Contents

Chapter 1

Statistical Theory

1.1 Overview of Statistical Data Analysis

1.1.1 Principles of Statistical Modeling

Overview of data analysis: the goal is to infer the underlying structure in the data and make predictions.

- Prediction problem: we consider the problem of predicting y from x. The challenges and ideas can be similarly applied in any data analysis problems. Examples:
  - Image analysis: image $\rightarrow$ semantics/category.
  - Text analysis: sentence/text $\rightarrow$ semantics/category.
  - Genetics: genotype $\rightarrow$ phenotype.
- Model multiplicity/overfitting: when we do not know clearly the underlying process generating the data, we fit a "statistical model" to the data, and this leads to the problem of overfitting or model multiplicity: there are many ways of fitting the data and it is not clear whether the model fits the true signals or simply noises (also called Roshoman Effect, [Breiman, Stat Sci, 2001]). Specific sources of Roshoman Effect may include:
  - Number of parameters is much larger than the number of observations: curse of dimenionality. The solution of a system of linear equation is underdetermined.
  - The predictors are correlated: singular design matrix.
  - Nonlinearity/Interactions among predictors: this could increase the number of variables exponentially.
- Heterogeneity: related to model multiplicity is the heterogeneity: data are generated from different processes. In the context of prediction, the same y may be caused by many different combinations of x, with complex rules. Examples:
  - Text analysis: the same meaning can be encoded by various syntactic structures, and the same entity may be represented by many different noun phrases.
  - Genetics: the same disease can be caused by changes in different genes, in different combinations.

Signal and noise: the difficulty of extracting signals from noises:

- Mistaking noises for signals: for example, consider a problem of predicting y from many explanatory variables x_j's. Even if none of the x_j is relavant to y, there may be correlations between some x_j and y in the data, due to chance. This is basically the overfitting problem with relatively complex models, or type I error (in classical statistics).

- Mistaking signals for noises: for example, suppose we are trying to identify the relation between x and y (a true predictor). Our model may include irrelevant covariates z that are correlated with x: then we may explain away the correlation between x and y using the relationship between z and y. Thus incorporating irrelevant covariates may lose power.
  - Example: a LRT with high df. loses power, because a lot of signals are simply explained by the model as noises (a high df. model implies more noises).

The challenges of overfitting: we understand this through typical examples:

- Regression problem: suppose we predict Y from features $X_j, 1 \leq j \leq D$, where D is large. Then many X_j's may correlate with Y by chance (noise), and it is not clear which one is the true signal.
- Generative model for classification: suppose we have a generative model $Y \to X_1, \cdots, X_D$, where each $X_j|Y$ follows Bernoulli distribution, with parameter depends on y. For most j's, the parameters should be the same in different classes, but because of sampling noises, they may appear different.
- Clustering: suppose data are generated from a small number of true clusters, but by chance, some points may appear close to each other, forming extra clusters.
- Density estimation: suppose our true density is $f(x) = U(x)$, the uniform distribution. But the histogram of data generated may not appear very smooth. So if we use kernel density estimation with a small λ, we obtain irregular density function.

Paradigms of statistical data analysis:

- Probabilistic modeling paradigm: the data is generated from a probabilistic process, thus infer the underlying structure/process through statistical inference.
  - The model should respect the underlying process. Often a multi-step process, and may involve latent variables in some stage (which help to expose the structure/pattern of the data/problem).
  - The characteristics / desired properties of solutions, should have high probability to occur in the model.
  - The statistical model should capture the structure, dependency, relation in the data. For example, in a spatial model, the correlation of variables in adjacent regions. The dependency can be very informative: a variable would carry information of another correlation variable.

  The heterogeneity challenge can be addressed best through Bayesian statistics: model averaging, latent variables (structure), priors and random-effects (allow variations across related processes), etc.
- Predictive modeling paradigm: find the underlying associations between x and y, for future predictions. The heterogeneity and nonlinearity challenges can be addressed through: model shrinkage, model averaging, partition-based methods, non-parametric/prototype methods, etc.
- Informative patterns: recognize the patterns informative of the underlying process and/or desired properties of the solutions being searched form. These can be expressed as statistic (in classical statistics) or features (in machine learning), and used for testing hypothesis or make predictions. Ex. in ANOVA, the informative pattern is the intra- vs. inter-group variations.
- Model analysis: for a prediction problem, need to address the multiplicity and heterogeneity challenges. The main idea for the former is: a simple model is prefered - regularization, and additional structure in the data to limit the space of possible models; for the latter, important to analyze the source of heterogeneity and model it properly. The general solution to a statistical learning problem from Nature is: hierarchical representation of data/object that at each step, extract key/stable features that normalize the data/input. Examples:

- Image processing: image → geometrical features → object → image semantics.
- Natural language processing: sentence → syntatic strutucre and entities → semantics.
- Genetics: genotype (mutations at nucleotide level) → changes of genes → effects on cells → phenotype.

Normalization need not be unidirectional: e.g. for a sentence, may try different parsing to find out which one leads to meaningful semantics.

Principles of statistical data modeling:

- Goal: translate the basic understanding/intuition of data into formal models. Need to capture the characteristics of data (or data generating process), or speaking in different terms, explain the data (variations in the data). It is important to control model complexity.
  - The analysis of problem: the underlying structure/process, the source of heterogeneity in classification problem, the prior knowledge/extra data that can be used, an appropriate measure of similarity/distance, whether the developed model and inferred results could reflect the properties, etc.
  - Model flexibility: a model should be able to explain various possible scenarios. Ex. in genetics, a model be able to account for the situation where a small number of loci each having strong effect or the situation where many loci each having small effect. In practice, consider several special cases and see if the model can explain all cases.
  - Model identifiability/complexity: a model cannot be too complex (over-parameterized) or it may not be identifiable, or very diffusive posterior. This is particularly important when the model contains missing variables.
- Modeling associations: some property of an object may suggest another property; or if two objects are similar in some aspect, then they may be similar in another. This is the basic idea of statistical learning: infer some property Y from features $\mathbf{X}$. Alternatively, the principle can be applied to group similar objects, as in cluster analysis.
  - Examples: cancer prediction from gene expression patterns; prediction of document quality from hyperlink structures; etc.
  - Strategies of modeling associations/dependency: conditional distribution (regression as a special case), samples from the same distribution (to capture similarity), MRF-like model (undirectional dependency), kernel method (local smoothing).
- Features/patterns/properties: from the raw data, define features, as functions of the data, that are better predictive of the property of interest. Patterns (particular arrangements of basic elements) or functional properties can be powerful features.
  - Ex. to predict function of a sequence, use its physical properties (stability, etc.) as features.
  - Ex. SVM method, by using more features (e.g. functions of the original features), the positive and negative classes are more likely to be separatable in the hyperplane.
  - Ex. a complex geometric object may be represented as a function/curve with a small number of parameters, then these parameters are treated as features.
- Dealing with overparameterizatio/overfitting: Bayesian random effects or sparse models. Under Bayesian framework: the parameters come from a common (or similar) distribution, thus the effective number of parameters is much smaller. Under the sparsity assumption: e.g. most parameters are zero, and the free parameters is also small.

- Dealing with uncertainty by integration/averaging: in general, if there are uncertain parameters/variables, integrate over the unknowns, or average over multiple observations/models, etc. This is a fundamental idea of Bayesian approach, where the parameters are integrated over. A special case is: combining models - aggregating over a large set of competing models can reduce the nonuniqueness while improving accuracy.

Probabilistic modeling:

- Generative models: for some data (such as experimental measurements), some physical/natural process can be used; for many other data types, this is not the case (e.g. image data). The idea is to model the data as generated from a sampling process. Ex. in genetics, even though genotype determines phenotype, we could assume we sample genotypes from persons with a given phenotype.
- Explaining the data: the goal of a statistical modeling can be stated as finding a model that "explains" the data. Most importantly, this can be said to explain the variations in the data. Two general sources of variations: (1) other related variables; (2) errors/probabilistic processes.
- Informative statistic(s): inference relies on the statistic (summary of data) that contains information of the parameters of interest, where "information" means the different values of the parameters would lead to different distributions of the statistic. In particular, we could obtain $E(T)$ of the test statistic T, as a function of the parameter θ, then the value of T would suggest the value of θ (similar to how physical parameters are estimated, treating T and θ just as two physical quantities). Examples: testing parameters in a linear model:
  - Estimator of the parameter: this is essentially the correlation between random variables, and is informative of the regression coefficient (larger coefficient would imply a larger correlation).
  - ANOVA: the informative statistics are between-group variance, which contains information of whether the level means are equal, and the within-group variance, which contains information of the intrinsic errors.
- Likelihood function: this is a special informative statistic that is applicable whenever a full probabilistic model is available. The information that this statistic captures is the Fisher information.

Sparse modeling/regularization: this can be expressed as $p >> n$ problem.

- Most parameters are 0/Lasso: Take an example of generative model for classification, let $p_1(x) = f(x|y = 1)$, and $p_0(x) = f(x|y = 0)$. p_1 shold be mostly close to p_2 except in a few features, e.g. suppose $p_c(x_j) = \theta_j^{(c)}$ where $c = 0, 1$, then we should have $\theta_j^{(1)} = \theta_j^{(0)}$ for most j's.
- Group and spatial structure: the parameters within a group should be similar. In spatial model, the adjacent parameters should be similar.
- Leveraging other hidden structure: e.g. tree structure, which implies that some features are more correlated than with other features.

Classical vs. Bayesian statistics:

- Classical statistics: if the data model is valid, a strong framework for inference. It aims to solve:
  - Parameter estimation: the estimators may be developed from MLE, MOM, error minimization, etc., and the assessment is usually carried out by the Mean Squared Error, which is the sum of the square of bias and variance.
  - Test of hypothesis concerning parameters: determine the distribution of the estimators (under H_0), and assess the power of test.

However, classical statistics has limited tools to establish validity of the data model, most imporrantly, goodness-of-fit test, residual analysis. Very limited power with more than four to five independent variables [Breiman, Stat Sci, 2001].

- Model selection in classical statistics: one creates a general model, and make the problem of model selection a problem of inference on parameters. Examples:
  - Variable selection in regression: inference of whether $\beta_j = 0$.
  - Mixture model: selecting the number of components. Suppose the number is bounded by $K_{\max}$, then a model of $K_{\max}$ components, inference of whether $\theta_k = 0$, where θ_k is the mixture weight of the k-th component.
  - Bayesian networks: to select among many possible structure G, define a multivariate normal distribution (suppose all variables are normally distributed), then infer a certain structure exists in the covariance matrix Σ.
- Difficulty with model selection: classical statistics is not designed to select models from many alternatives, with possibly different complexities.
  - Maximum likelihood parameter estimation: cannot compare models with different complexities, thus a poor strategy for selecting models.
  - Hypothesis testing: classical statistics is mainly concerned with H_0 vs. H_A, and this could be very inadequte. Example: for variable selection problem, testing individual $\beta_j = 0$; however, because variables tend to be correlated, it is likely that few of them will pass the threshold (especially consider multiple testing), even for true variables.
  - Non-nested model: even when we limited to two model comparison, when the two are not nested, they may pose a problem for classical statistics.
- Bayesian statistics: a model M is assessed by the posterior probability $P(M|D)$, where D is the data. Bayesian statistics provides a number of strategies to deal with the model multiplicity issue:
  - Averaging: over possible values of parameters or types of models.
  - Random effects: the data need not be generated by a single process with a single set of parameters. Instead, the parameters may be variables (related in some way), and this may allow more explanation of data, and a better normalization of data (by incorporating structure of parameters, and by averaging over parameters).
  - Latent variables: this could achieve the effect of normalization, or capturing variations (a lot of variations may be explained by a few unobserved variables).
  - Prior: this would generally favor simpler models, and also could guide the model search using prior knowledge.
- Remark:
  - Despite the limitations, sometimes classial statistics can be very useful for some model selection problem. Example, when comparing a small set of alternative models for a Bayesian network.
  - Classical statistics may address the problem of model complexity by introducing contraints in the parameters. Ex. for variable selection in regression, constraints on the L_1 norm of the parameters (Lasso).

Evaluating model fit:

- Principles: (1) agreement of model predictions and observation. (2) how much variation in the data is explained by the model.

- Example: linear model. (1) Residual plot. (2) R^2 measure of goodness-of-fit.
- Example: fitting parameteric distribution. (1) Histogram vs. PDF of the fitted model. (2) Similar R^2 measure?

Predictive modeling: the essence is to impose constraints on the model explaining the data.

- General strategy: search for a model that minimizes the generalization error, defined as $E[L(Y, \phi(X)]$, where $L()$ is the loss function, and ϕ is the model. The generalization error is often estimated by cross-validation in practice. Since generalization error is often hard to obtain, often search for a model that minimize some loss/error function or maximize the explanation of variations in the training data, with appropriate regularization. Statistical perspective is generally important: data is often modeled as generated from some stochastic processes.
- Geometric perspective: a model can be viewed as a geometric surface approximating the data points (x_i, y_i) (in regression), or as providing a decision boundary (in classification).
- Regularization: (e.g. in the case of classification) geometrically, a naive/complex model has a complex/rugged decision boundary. A good model should be simpler, have a smooth decision boundary (hence, the term, "regularization"). Shrinkage methods directly impose constraints/penalty on parameters such as simpler models are favrored. Ex. Lasso.
- Margin methods: the intution is for any classifier of the training data, the one with the low margin of error is likely to be wrong in the future (small perturbations may cross the decision boundary), and this class of models is more complex (many possible modesl to have low margin of error). Thus favor models with high margins.
- Partition-based methods: partition the data into regions, where in each region, there may be simpler (linear) model. Ex. decision tree. The general difficulty is that the correct partitioning is unknown. Some form of soft partitioning, and combining partition with prediction may help (form a partition such that within each region, there is a simple model).
- Prototype/non-parametric methods: the idea is that locally the model may be simple/linear. Use the training data to define the local regions. This is similar to the partition-based methods in that essentially each sample point defines a local region. The difficulty is that how to define locality is not clear, and in high-dimensional space, there may not be an instance in training data that is close to an instance to be predicted, thus the model may be not very generalizable.
- Semi-supervised learning and the use of external data: first, it could help uncover additional structure on features s.t. a better partition can be formed where local models can be learned; second, it may allow better estimation of parameters relevant to the data.
- Model averaging/ensemble learning: if not a single good is available. Similar to the idea of partitioning, though does not do that explicitly.
- Connection between parametric and non-parametric methods: Ex. linear regression (model-based): the solution can be written as a linear function of $\mathbf{y}$ (response), thus may be interpreted as weighted contribution of the training examples; KNN (instance-based): may be understood as a model using certain centroids, but the position of centroids must be learned from the data.

Comparison and relation between generative and predictive modeling paradigms:

- Interpretation: both could have appropriate interpretations. E.g. X = gene expression pattern, and Y = phenotype (e.g. growth rate), (1) phenotype is a function of expression pattern; (2) phenotype indicates the internal status or environmental condition of the cell (e.g. growth rate reflects the nutrient availability), then expression pattern is determined by this status/condition.

- Comparison of two paradigms:
  - Feature expansion: it is much easier to incorporate additional features in a regression framework than probabilistic models.
  - Density modeling: this may be difficult, especially as the process of sampling X may be biased. Thus the generative approach may be more sensitive to outliers, e.g. in LDA, the estimation of class centroids (mean of normal distribution is sensitive to outliers).
  - Latent variables: easier in the generative models.
  - Model averaging: easier in the generative models.
  - Nonlinearity: if no structure is known, then regression models may be more flexible by using prototype/non-parametric methods, or by adding more features (kernel trick).
- Connection between regression and generative modeling: the optimal method may need to model both $P(x)$ and $P(y|x)$. In the regression approach, $P(x)$ is ignored, however, it may be informative: e.g. there is cluster structure in the space of X, and the same cluster tends to have the same class label. In the classification problem, this is semi-supervised learning that takes advantage of the unlabeled data.
- Combine generative and regression models: suppose we want to classify using BFs - a generative model approach. But if we can partition the information into multiple parts, each part captured by a BF. Then we can use each BF as a feature and train a classifier. The idea is not limited to BFs: we can use LRT, even p-values.
  - Example: classify cancer genes. Information from mutation rates and spatial clustering. BF from two features, and classify using the linear combination of two features.

Strategies of parametric probabilistic models:

- Simple linear model: suppose we condiser the problem of predicting Y from $X_1, \cdots, X_p$. The simplest model would be a linear model of Y from X. This model suffers from a number of problems, including: noisy features, nonlinearilty, dependence between features (e.g. one feature affects only with certain values of another features).
- Group and locality structure: often exists despite heterogeneity, this can be at the sample level or the variable level. Possible structures: (1) The same group of samples has the same model; or related/close samples have similar models; (2) Variables in the same group tend to have non-zero effects at the same time; or within each group, only one variable should be chosen.
  - Benefit: heterogeneity in the data is captured, while at the same time regularization and variance stabilization (without overparameterization).
  - Models: a number of ways of modeling these structures, including structure models (e.g. introducing group variables), hierarchical models (modeling group parameters), kernel smoothing (e.g. varying coefficient model).
  - Example: in genomics, X is gene features (such as regulator binding), Y is expression, then genes in the same group should share a model (using the same features); or related genes should have similar regulators.
  - Example: decision tree is effectively partitioning the samples and learn models in each sample. Ex. partition the samples by the discrete variables using decision tree, and then at each leaf node, learn a linear model with the remaining contiuous variables.
  - Remark: sample and variable structures are related: e.g. one could introduce group membership variables for samples, then the structure is stated in terms of variables.
- Feature interaction and expansion: introduce additional variables as functions of features. They may be non-linear functions of individual features or interaction terms of multiple features.

  - Benefit: non-linearity and feature dependence (the effect of one feature depends on another features). Feature dependence is one way of modeling heterogeneity: e.g. the model parameters may depend on some other variable (such as time).
  - Example: define features $\sigma(X_j)\sigma(X_k)$ where $\sigma(\cdot)$ is the sigmoid function, to approximate logic AND, and OR.

- Structure models: the conditional distribution or the MRF-like models of the variables, in particular, graphical models.
  - Benefit: greatly limit the possible models (version space), i.e. reduce model complexity.
  - Example: model joint distribution of X_j's and Y's using a MRF model, e.g. predicting the spin state of a grid in the Ising model.

- Hierarchical model: a probabilistic model (prior distribution) of parameters.
  - Benefit: regularization of parameters or variance stablization.
  - Example: model parameters as functions of additional variables or features. Could be used in variable selection, e.g. $\beta_j \sim \text{Mixture}(0, N(\tau, \sigma^2)$.

- Latent variable models: the actual explanatory variables for Y may be latent, or missing. Latent space model: all true explanatory variables are latent.
  - Benefit: a smaller number of explanatory variables (assuming a good model relating observed variables and latent variables), thus reducing variance.
  - Example: document class is a function of latent topics; phenotype if a function of latent gene activities.

- Variable selection/sparsity: in general, when p is large, there may be only a subset of variables that influence Y. The correlation of one variable X_j to Y may actually be due to a correlated variable X_k to Y.
  - Benefit: a simpler model reduces the variance of the estimators/prediction, or reduces model complexity in general.
  - Example: learn the structure of X_j's from unlabeled data (e.g. a Bayesian network model), this may limit the possible explanatory variables or variable interactions for Y.

- Nonlinear functions: directly capture the non-linear aspect in the data, e.g. GLM, splines, decision tree, cyclic structure, semi-parametric models (e.g. varying coefficient model), etc.
  - Benefit: capture the specific aspect/characteristic of data.
  - Example: a continuous version of decision tree, Y is a sum of product terms of features.

- Independence assumption: one often makes independence assumption in the model. Sometimes this may be violated. Ex. in Naive Bayes model, if features are correlated, the OR (or BF) will be inflated. Suppose we have 10 highly correlated features, each feature contributes to OR by 2; the total contribution would be $2^10 = 1,024$, while the true contribution is much smaller.

- Prior knowledge and external data: when available, use them to constrain the model. In Bayesian, this can be added as priors; in sparse model learning, can be added as regularization terms of the objective function.
  - Benefit: limit the model search space, thus reducing complexity.
  - Example: in regression problem, one may know the importance of variables before hand, e.g. $\beta_1 > \beta_2$, or $\beta_1 > 0$ if and only $\beta_2 > 0$, this can be incorporated in the model.

- Example: application in document/sentence classification:
  - Group and locality structure: using document meta-data, e.g. authorship (one author has certain topic preference), date, link structure, etc.
  - Feature interaction and expansion: N-gram features.
  - Structure models: HMM or CRF models at sentence level.
  - Hierarchical models & variable grouping: words in the same category (or synonymous words) have similar weights.
  - Latent variable models: supervised LDA.
  - Variable selection: IDF weighting or removing of stop words.
- Example: application to genotype to phenotype mapping:
  - Group and locality structure: family relations among samples.
  - Feature interaction and expansion: SNP interactions.
  - Structure models: Markov model of sequential SNPs.
  - Hierarchical models & variable grouping: genes in the same pathway tend to have similar regression coefficients.
  - Latent variable models: gene activity as the latent variables.
  - Variable selection: weighting SNPs by prior evidence; causal variants in LD blocks.
  - Nonlinear function: non-additive genetic models (dominant and recessive models).
- Example: the same ideas can be applied to model joint probability distributions (no explicit response variables), e.g. model DNA sequence evolution. The structure in the data, i.e. the variation of evolutionary rates, can be modeled in a number of different ways:
  - Structure models & Latent variable model: a hidden variable of functional class of a position (fast or slow), and allow switching of classes across different positions with HMM.
  - Hierarchical model: the rate itself (across all positions) follow a random distribution.
  - Kernel smoothing: the adjcant positions should have similar rates, model the rate as a function of position, and apply the varying coefficient model.
- Remark: many ideas can be combined and further improve inference. Examples:
  - Variable grouping and variable selection: choose one variable per group. Ex. genetic association, where features are SNPs in LD.
  - Structure models with latent variables: e.x. HMM.
  - Feature expansion and latent variables: the expanded features (interactions) are related to latent variables. Ex. the interaction between two genes correspond to the state of a pathway (latent).
  - Sample grouping and variable grouping: one sample group may use one variable group.

Understanding probabilistic models:

- Importance: while in an inference problem, one can often follow the generic procedure (e.g. MLE, MCMC), it is often important to understand the properties of the probability models/distributions, such as expectations, covariance, etc. This would help one to understand the consequences/implications of a prob. model: what patterns are informative of model parameters (and thus help parameter estimation and hypothesis testing).
- General steps:

  - Model identification: the first step of understanding a model, for a complex model, it may not be identifiable.
  - Patterns/consequences of the model: how informative patterns are related to the model parameters.
  - Estimation procedure: an intuitive understanding of the estimator/test statistic, the estimation algorithm (e.g. many algorithms are itereative procedures).

- Examples:
  - Multivariate normal distribution: the covariance matrix, marginal and conditional distributions.
  - Bayesian networks: the conditional independence structure.
  - Ising model (MRF): the covariance structure of the spin states of the lattice sites (not necessarily adjacent) - this would help understand the equilibrium of the Ising model.

- Another way of "understanding" a model is: analyzing the statistic that one uses to solve the inference problem, e.g. Bayea factor for model selection problem - how does this statistic depend on the properties of data? What is its behavior (for different types of data/input)? Does it make intuitive sense?

- Remark: related to Method of Moments parameter estimation, though the idea of understanding the consequences (patterns) of a prob. model is very general.

Optimization perspective:

- Data analysis with optimization: one can formulate certain desired properties of the solutions of a problem, and usually these can be expressed as an objective function to be optimized.

- Regression: the parameters of such a problem should maximize the fitting of data, or minimize the prediction error. Ex. least square fitting of linear models.

- Clustering: the positions of clusters should minimize the total intra-cluster distance.

- Missing data: the values should minimize some kind of errors (with respective to known data). Ex. in alignment problem: maximize the similarity of two sequences.

- Statistical perspective: while the optimization perspective sometimes is enough to solve a problem, a statistical perspective often brings benefits, including: uncertainty of estimation; parameter estimation when training data is available; etc.

Common statistical considerations:

- Model identifiability: whether data is sufficient to estimate parameters/models (if not, choose simpler models, etc.)

- Information: whether all the information is used. Ex. when a procedure involves discretization of contiuous values, information may be lost (two different objects may be treated equal).

- Alternative models/data explanations: whether there are alternative models not considered in the model. Ex. the alternative hypothesis may not include all possibilities, thus rejection of null may not guarantee the acceptance of the current alternative hypothesis. (Bias is one special case.)

- Distribution assumption/outliers and independence assumption: whether data follows the assumed distribution: e.g. normality assumption; whether data points can be considered as i.i.d. If these assumptions not held, how sensitive the method is to outliers, or to dependence of data points.

- Mathematical functions: whether it is appropriate to add terms (often used in models such as regression, SVM, may need transform variables so that they may be added), etc.

- Simplification of problems by using data summary: we may not need to model the complete/original data. If we can use statistics to capture all the information in the data, we can work on the summary statistics, which is often much easier.
  - Ex. meta-analysis of linear regression.
  - Ex. hierarchical linear model, where β follows some distribution. We can make inference of β on each group, and then model the estimated and standard error of β using normal distribution.

1.1.2 Statistical Theory and Methods

Reference: [Breiman, Stat Sci, 2001]

Hypothesis testing: a key consideration is to increase power in testing hypothesis.

- Degree of freedom: H_A needs to fit significantly better than H_0 for one to accept H_A, thus if one has a complex H_0, it will be difficult for H_A to do significantly better. Therefore, a complex H_0 could result in power loss.
- Multiple hypothesis testing correction: reduce the number of hypothesis tested will increase the power of testing.

Regression modeling:

- Variable selection: when independent variables are correlated, the additivity assumpion in the linear model may not hold (only independent effects are additive), thus variable selection is crucial. From the hypothesis testing perspective, having more independent variables than necessary (a complex model) generally reduces power.
  - Shrinkage method: Lasso, etc. that penalizes more parameters.
  - Bayesian variable selection: select a subset that provides a balance of data fitting and model complexity.

Classification through hypothesis testing:

- Class density modeling and classical hypothesis testing: the class density approach, where $P(x|y)$ is modeled, can be treated as testing two hypothesis: $y = 1$ and $y = 0$. The classical hypothesis testing can then be applied, where some test statistic T is used to reject or accept H_0. If use LRT, then the test score is similar to the Bayesian posterior probability, $P(y|x)$, usually used for classification.
- Determine the threshold under the supervised setting: the threshold t can be determined by minimizing some appropriately defined error function.
- Determine the threshold under the unspuervised setting: when the training data is not available, the threshold is chosen to meet certain level of type I error, or FDR (usually need multiple hypothesis testing correction). Alternatively, choose a certain sensitivity level (e.g. threshold as the test statistic of the top K prediction), and evaluate the FDR. The latter is particularly useful when comparing methods.

Unsupervised learning: two basic perspectives for analysis of unsupervised data:

- Hidden patterns: search for objects with certain properties or certain types of relations in the data. Examples:
  - Objects: (1) human behavior patterns: infer terrorist suspect; (2) genes with certain expression profiles.
  - Relations among objects: (1) human data, infer social network; (2) functional interactions among genes.

  - Relations among features: (1) items that are bought togeter from transaction data; (2) grouping tissues from expression data.
  - Relations among both objects and features: e.g. identify a set of people that work for the same company (share features such as income, city, favorite restaurant, etc.).

  Note that the types of objects and relations considered depend on the problem, and can be very complex. Ex. for association rule mining, the rules can be any logic/algebric functions the variables satisfy.

- Latent structure: identify latent variables or structure/grouping (how objects are organized) in the data. Example:
  - Object grouping: e.g. social network (how people are related to each other).
  - Latent variables: e.g. gene expression pattern of cells, the cellular condition is not directly measured, but important latent variable.

Dealing with heterogeneity/overparameterization:

- Problem: in a problem of many objects, each of them may have some unique properties (heterogeneity). Using the same parameter for all would be too simple and may lead to false conclusions, while using different parameters for each object lead to a model with too many parameters (overparameterization).
- Strategies:
  - The general idea is to introduce structure into the model. Important cases include: certain objects share certain properties (random effect), hierarchical model, a smaller number of hidden variables (principle component analysis).
  - Random effects: each parameter is a random sample from a common effect. Then one could test/estimate the shared distribution. Its advantage is that the evidence of multiple objects can thus be combined to make inference.
  - Mixture model/grouping: an important special case of random effects is the mixture model approach. The idea is that: each object belongs to one of multiple classes (each class: the same parameter), and the class assignment is a hidden variable. Effectively, this is to group similar objects together.
- Examples:
  - Molecular evolution of proteins: different sites may evolve at different rates. Random effects model: the rate of each site is from a probability distribution, and test the parameter of this distribution.
  - Functional properties of pathways: different genes may be related, but still have individual differences. Random effects model: each gene has unique contribution, but the parameter is from a common distribution, and test this distribution. [Gene group association with clinical outcome, Goeman & van Houwelingen, Bioinfo, 2004]

Latent variable models:

- Strategy: in many problems, it is natural/advantageous to introduce additional latent variables: one can build a better model/explanation of data in terms of these latent variables.
- Applications:
  - Dimensionality reduction: e.g. PCA, a small set of latent variables explain most of the variations of observations.

  - Causal model with latent variables: e.g. SEM.
  - Prediction with latent variables (instead of the observables): e.g. factor regression (or factor analysis).

- Benefits of latent variable models:
  - Imposing additional structure in the model, reducing model complexity. This is similar to hierarchical models, where instead of having one model per group, the models of all groups are related in some way.
  - Reducing dimensions and better interpretability: these latent variables may represent concepts/themes (text analysis), objects/patterns (vision), pathway activity (genomics), etc.
  - Better predictive model: a response may better be predicted with (fewer) latent variables, in particular, all the relevant observations are used to learn the effect of latent variables, and this achieves the effect of pooling, and improve inference.

- Relation to multi-level modeling: in some applications of multi-level models, the group-level parameters (that are subject to modeling) can be viewed as latent variables (e.g. group means), and so they are special cases of latent variable models.

- Latent variable in regression models: suppose we are predicting Y (response) from X (observations), and Z are latent variables that are better predictors of Y (e.g. more direct relations with Y). There are a number of ways of modeling the relation among these variables:
  - Generative model: $Y \to Z \to X$. Ex. in text analysis, we have Document class $\to$ Topic $\to$ Words.
  - Regression model: $X \to Z \to Y$. Ex. in genetics, we have SNP $\to$ Gene $\to$ Phenotype.
  - Joint model of predictors and responses: $Y \leftarrow Z \to X$. Ex. in genomics, we have Class $\leftarrow$ Module/Pathway activity $\to$ Gene expression.

  In a particular problem, we may use any of the three possible models, e.g. in text analysis, joint modeling may be the used, where topics (latent) determine both document class and words (supervised LDA).

Feature development:

- Goals of feature development:
  - Feature representation: for complex objects, need a relatively simple representation that allows e.g. comparison of similar objects. Ex. image analysis: to recognize similarity between images, represent images by vectors, where features correspond to spatial regions; then image similarity can be simply defined as the correlation.
  - An important part of the learning procedure is to develop/recongize features that may distinguish different types of objects, e.g. "fingerprints".

- Developing features:
  - Elements: of the objects. Ex. text classification problem, the words are natural elements.
  - Patterns: elements are often insufficiently discriminative (e.g. single words for text, or single AA for sequences) or not recognizable (e.g. image data), then the recurrent patterns (some particular arrangements of elements) may serve as good features.
  - Properties: functions defined on the elements/patterns/objects that reflect certain properties of the objects. Ex. DNA sequence classification: the physicochemical properties inferred from DNA sequences; CpG islands as markers of genes; evolutionary footprints.

- Features as compact representations of complex objects: it may be possible that a few simple features explain variations of complex objects. Ex. variation of beak shapes in Darwins's finches can be explained by three geometric parameters (scaling, etc.).
- Examples of features for different types of objects:
  - Sequences: presence of motifs/k-mers; co-occurrence of motif pairs; property of sequences (e.g. conservation, DNA bending, stability, TF-binding, etc.)
  - Sentences/text: words and phrases; the synatic structure of sentences (e.g. Entity-Verb structure, where Verb is one of a list of key verbs).
  - Gene expression profiles: the expression of pathways (e.g. an entire is up-regulated), the co-expressed genes (modules) - the module strcuture may characterize one type of profiles vs the other.
  - Images/structures: spatial patterns of geometric objects or atoms (values at spatial units within a reference framework).

Feature learning: to learn features important for a class of objects is often part of the learning problem/goal.

- Classical statistics: by testing statistical significance of the parameters, e.g. testing the hypothesis that $\theta > 0$ vs. $\theta = 0$ for some parameter θ.
- Nonparametric tests: e.g. testing overrepresentation: the features important for discrimination may be overrepresented in one class only relative to the other. Ex. motif finding in sequences.
- Regression: by testing the effect on the performance when the feature is removed or permuted [Breiman, Stat Sci, 2001].
- Difficulty of learning important features: features are often correlated, then removing one feature (as in both classical statistics and ML) may have small effect, as the correlated feature may make up for the lost feature.

Data normalization:

- Why do we need normalization? Often we need to compare some variables, but the measurement (data) is influenced not only by the quanity of interest, but also other sources. So we will need to remove the influence of other sources/noises.
- Examples:
  - Compare expression of a gene in multiple conditions. The expression in a condition is influenced by batch effects, biological variations, sample quantity, etc.
  - Identify CNVs from arrayCGH or sequencing data. Variation of measurement is large across genome even in background (no CNV) regions due to differences in capture efficiency (for WES), PCR, sequencing, etc.
  - Calling peaks from ChIP-seq data. Read depth is influenced by GC content, mappability, etc.
- Strategies of normalization: in general, analyze what factors could cause problems (data points not directly comparable), and develop strategies accordingly.
  - Paired treatment-control design: e.g. for arrayCGH or ChIP-seq, use paired controls to obtain background signal.
  - Controlling known confounders: e.g. for aCGH data, control for GC content, read depth. For expression data, control batch effect (covariate), etc.

  - Controlling unknown confounders: find out if some hidden variables explain the variation of variables, then these variables can be controlled. Typically through PCA.
  - Using "local background" as controls: the idea is to mimic paired treatment-control, using similar data points. Ex. in ChIP-seq, use local genomic regions as control. Quantile normalization.
  - Data transformation: we could transform the data in such a way that it becomes comparable across data points. Ex. expression of different genes: obtain values such as RPKM that removes dependency on gene length and library size.

Confounding variables and bias:

- Association not equal to causation & confounding factors: these are essential considerations for drawing a valid conclusion, as there may be factors not considered or encoded in the model (which contribute to the observed patterns). To deal with this issue: defining an appropriate controls; incorporating the confounding factors/variables in the models; etc.
- Bias: what are the possible biases/whether data are comparable, e.g. to compare statistic of objects of different sizes. This often involves the analysis of what other factors (other than the main effect we are studying) may contribute to observed data.
- Subtle distinction between confounding and bias: in confounding, we are concerned with relationship between two RVs. In bias, we are talking about a general issue of some estimate quantity or some test. Ex. we compare genes in GWAS, and define gene p-value as min-p. There is no confounding here (as we are not explicitly studying two variables), but there is a gene-size bias.
- Testing associations with genomic features: often we are interested in whether some genomic features are associated with some properties, for example:
  - GWAS: test if GWAS hits are more likely to be localized within enhancers.
  - De novo mutations: test if de novo mutations are more likely to be in enhancer regions.
  - TFBS distribution: test if TFBSs tend to be in evolutionarily conserved regions.

  In all these cases, there are possible confounding variables that may create association, such as: distance to TSS, GC content of genomic regions, mutation/recombination rates, mappability.
- Inferring genomic properties: we may want to infer some underlying properties of genomic regions, e.g. TFBS, their interaction, chromatin states, etc. Examples:
  - Chromatin interaction from Hi-C: infer the strength of interaction. Confounding variables: random DNA looping and genomic context (e.g. GC content).
  - ChIP-seq: infer the location of peaks. Confounding variables: mappability.
- Strategies of dealing with confounding variables/bias:
  - Linear model and matching confounding variables: test the hypothesis at matching values of confouding variables. Under the linear model, assume the coefficent is the same with different values of Z.
  - Permutation that controls for confounding variables: permutate data in such a way that preserves confounding of x and z (confouding variable) and y and z.

Strategies of quality control:

- Data filtering: often the first step is remove dubious data points, outliers. Example:
  - GWAS: filter SNPs by HWE, by AF.

- Proper normalization of data: controlling for confounders, data transformation, etc.
- Negative control: generally needed to obtain the "background" - how would data look like if there is no signal.
- Positive control: can we find the signals that we are supposed to find?
- Metrics for QC: what do we expect if the data is good and data processing procedure is working? What are the expected properties of things that we are trying to find. Examples:
  - GWAS: few causal SNPs, thus the QQ plot should be roughly linear. If a study finds SNPs, are they enriched in functional regions?
  - ChIP-seq: the peaks should be close to TCC, evolutionarily conserved, etc.

Physical/biological process-motivated models:

- Statistical mechanics: Markov random field.
- Random walk/diffusion: Markov chains.
- Network models: network flow, etc.

1.1.3 Statistical Problems

Statistical theory:

- LRT for non-nested hypothesis: what is the asymptotic distribution? A simple case is: $H_0 : \theta = \theta_0$ vs. $H_1 : \theta = \theta_1$. The proof of χ^2 distribution depends on Taylor expansion around MLE, however, for non-nested hypothesis, the parameters (under two hypothesis) are not necessarily close.
- Model identifiability: in a complex model, e.g. hierarchical model with missing data, the identifiability is not obvious. How to formally analyze the model identifiability? Bayesian approach using the posterior distribution/sample of $P(\theta|D)$ may be the solution.
- Information theoretical approach to inference: it can be shown that minimizing KL divergence is equivalent to MLE. Similarly, we could use information theory for hypothesis testing, e.g. to test $H_0 : \theta = \theta_0$ vs. $H_1 : \theta = \theta_1$, we may want to compare $KL(\hat{f}||f_{\theta_0})$ and $KL(\hat{f}||f_{\theta_1})$, where $\hat{f}$ is the empirical distribution.
- Evaluating estimators: Ex. for parameter estimation problems, the unbiased estimator with minimum variance can be thought of as the optimal one. In general, what is the theoretical framework of assessing estimators, say statistical decision theory? Can we prove that our common estimator (e.g. MLE) are optimal under reasonable loss functions?
- Performance analysis of statistical learning: a general problem in statistical learning (similar to estimator evaluation) how to evaluate the performance of a method and comparison of different methods, and can we prove optimality in some cases.
  - Ex. the error of the KDE method; the MSE of linear classifiers such as SVM; etc.
  - Ex. Lasso regression, why Lasso penalty is good (how the performance may depend on the correlation structure of independent variables)? Can we prove it is optimal under some assumptions (e.g. a small number of true variables and they are generally uncorrelated with each other)? This may be analyzed using bias-variance decomposition, Equation 7.19.
- Statistical analysis of algorithms: in machine learning, we often have an algorithm of doing things, say regression or classification (e.g. Lasso). How do we analyze the statistical properties of these algorithms: what is the rate of false positive findings (e.g. for Lasso, what is the FDR of the selected variables)?

- Learning important features: when features are correlated, testing individual coefficients or even groups of coefficients by F test is insufficient. Ex. X_1 and X_2 are correlated, then removing X_1 would have a small impact on SSE. What is the best way of learning feature importance, taking into account the feature correlations?
- Bayesian statistics: introduce prior distributions would reduce the variance of the learned model (informative prior vs. uniform/noninformative prior)?

Bayesian statistics:

- Types of priors: Jeferreys, objective, etc. And how they may influence the inference if the priors are not "fully anchored in past experience" [Efron, A Two-Hundred-and-Fifty-Year Argument]?
- Frequentist behavior of Bayesian inference/posterior: a genearl question is that as we increase sample size, how would the posterior converge/behave?
  - Example: suppose we have a linear regression model, and we have two correlated variables X_1 and X_2. Intuition: if the two are highly correlated, we need a big sample to learn the separate effects - posterior converge to one peak for each of the coefficients. How do we formalize such intuition?
- Parametric bootstrap as approximation of Bayesian [Efron, A Two-Hundred-and-Fifty-Year Argument]?
- Derivation of Bayesian information criterion?
- Related to Variational inference: if a distribution can be factorized, what does it say geometrically in terms of the contour plot of the PDF?
  - Remark: for MVN, independence means orthogonality (of eigenvectors). In general, do we have something similar (orthogonality)?
- Convergence of the variational inference algorithm? The relationship to convexity [Bishop, Chapter 10]

Statistical models:

- Two sample comparison: e.g. differential expression between two samples. How do we control the hidden confounders? More generally, use a linear model to study the effect of X on Y, what's the impact of missing confounders? (False associations: the effect can be explained by the hidden confounders, but we falsely attribute the effect to X).
- Linear regression: suppose our goal is to test if an independent variable is associated with a response. Does including additional variables (covariates) increase the power of test or reduce the false positives? Intuitively, stratification on the covariates, and testing the variable.
- Linear regression: do we gain power (of testing β) if we model $P(x, y)$, instead of $P(y|x)$?
- Linear regression with multiple testing: suppose we have a complex testing/model selection problem, e.g. whether a subset of parameters are 0, vs. only one parameter is 0. And among many such tests, we want to estimate the fraction of each scenario. An example: the effect of something (treatment, SNP) on gene expression could be: tissue-specific, or all-tissue, or 0 for all-tissues.
- ANOVA: can the idea be generalized to, e.g. non-linear models, for selecting variables or dimensionality reduction, etc.?

- Lasso: does group Lasso helps aggregate statistics across multiple members of a group to increase power? If formulating in hypothesis testing terms, what is the criteria by Lasso to select features? Guess: group Lasso, to choose a group (suppose there are no other groups), is effectively comparing two hypothesis: $H_0 : \beta_j = 0, \forall j$, and H_A: for some j, $\beta_j \neq 0$. Thus this is similar to Hotelling's T^2 test.
- Logistic regression: how standard errors of coefficients are computed?
- PCA: can we have a model where we have both latent variables and known covariates? In the typical gene expression data analysis, PCA first, then correct for PCs and other covariates. Can we correct covariates and do PCA at the same time?
- Hierarchical models: as a tool to deal with heterogeneity in the data. How would this compare with mixture model? Ex. among K groups, instead of modeling some parameter β_k for each group as a sample from some common distribution, we could also have a mixture model: some groups have parameter β_1 and the others with β_2.
- Hierarchical models: how to model the network (as opposed to group) structure of samples? How to model the overlapping groups?
- Markov random fields: only certain conditional independence conditions are satisfied, a probability distribution can be called a MRF. Why?
- Structural preference in regression and Bayesian networks: in a regression setting, it may be helpful if put constraints on X_j's, e.g. certain pairs of features may have similar coefficients. There is a similar problem in learning Bayesian networks, where one may prefer how the nodes are linked. In general, the constraints may take the form: certain variables should be grouped (similar explanatory or outcome variables), similarity between pairs of variables, etc.
- Soft partitioning: in partition-based methods, how to perform soft partitioning? Regularization on partitions: so that the models in related groups are also closer? Combining partitioning with prediction: otherwise, partitioning may not be relevant to the prediction task.
- Sample imbalance in nearest neighbor methods: when there is an imbalance in the samples, these methods such as KNN easily give biased results. What is the source of this problem and how to address it?
- Justification of local likelihood methods?
- Physical analogy of Markov random field: what is a continuous version of Markov random field? Can we view function $f(x)$ as the state at any point over the space, e.g. the particle density, or potential at the point x. Can this perspective be applied to other problems, e.g. curve fitting?
- Causal inference: how to translate correlations to causality? In two group comparison, if two groups are randomized in every other aspect except the test factor, then then difference between the two groups must be due to the test factor. In general, if $X \to Y$, then when X varies, Y should also varies; for a potential confounder, its variation is generally random wrt. X.

1.2 Probability Theory

Expectation and variance:

- Total expectation and variance: Suppose we are interested in $\mathrm{E}(X)$ and $\mathrm{Var}(X)$ for some random variable X. The conditional expectation and variance of X under given Y are easier to find, so we can express $\mathrm{E}(X)$ and $\mathrm{Var}(X)$ in terms of the conditional expectation and variance.

$$\mathrm{E}(X) = \mathrm{E}_Y(\mathrm{E}(X|Y))) \tag{1.1}$$

$$\text{Var}(X) = \text{E}_Y(\text{Var}(X|Y)) + \text{Var}_Y(\text{E}(X|Y)) \tag{1.2}$$

Approximating a distribution:

- Motivation: how do we approximate a probabilty distribution (or summariz a set of numbers) using a single number?
- Theorem: let $x_1, \cdots, x_n$ be any numbers, for any number c, we have:

$$\sum_{i=1}^{n}(x_i - c)^2 = \sum_{i=1}^{n}(x_i - \bar{x})^2 + n(\bar{x} - c)^2 \tag{1.3}$$

  The number c that minimizes the sum of squared deviation (SSD) is $\bar{x}$.

- Theorem (continuous RV with L_2 loss): let X be a RV, for any constant c, we have:

$$\text{E}(X - c)^2 = [\text{E}(X) - c]^2 + \text{Var}(X) \tag{1.4}$$

  A corollary of this theorem is: the constant c that minimize $\text{E}(X - c)^2$ is $\text{E}(X)$.

- Theorem: let $x_1, \cdots, x_n$ be any numbers, for any number c, we define the sum of absolute deviation (SAD) of x_i to c as:

$$\text{SAD}(c) = \sum_i |x_i - c| \tag{1.5}$$

  The number c that minimizes SAD is the median (if n is even, then any point between the two middle elements is fine).
  Proof: consider all possible interval where c may fall into (thus removing the absolute sign).

- Theorem (continuous RV with L_1 loss): let X be a RV, then the median of X minimizes $\text{E}(|X - c|)$.
- Theorem (discrete RV): Let X be a discrete RV with K possible values $g_1, \cdots, g_K$, define a 0/1 loss function for any categorical value c: $L(g_k, c) = 0$ if $c = g_k$ and 1 otherwise, and the deviation from c is defined as:

$$D(c) = \sum_{k=1}^{K} L(g_k, c)p_k \tag{1.6}$$

  It is easy to show that c that minizes $D(c)$ is: $\hat{c} = \text{argmax}_k p_k$.

- **Remark**:
  - Decomposition of deviation: it has two components: (1) departure of the mean to the number and (2) the variance. This idea is generally applicable for analyzing the errors/variance.
  - Approximation: the general idea of using something simpler distribution to approxiate a more complex distribution, e.g. normal distribution to approximate any RV that is bell-shaped. To define an approximation problem, need: a loss function and averaging.
  - Approximation perspective in different contexts: (1) point estimate of a probability distribution: since the functional form is known, only parameter value matters, thus use MSE as loss function; (2) prediction problem: loss function defined on the joint distribution of (X, Y); (3) approximation probability distribution: KL divergence.

Markov's Inequality: [Wiki]

- Theorem: if X is an non-negative RV and $a > 0$, then

$$P(X \geq a) \leq \frac{E(X)}{a} \tag{1.7}$$

- Proof: let $f(x)$ be the PDF of X,

$$a \cdot \int_a^{+\infty} af(x)dx \leq \int_a^{+\infty} xf(x)dx \leq \int_0^{+\infty} xf(x)dx = E(X) \tag{1.8}$$

- Remark: the intuition is that for any non-negative RV, it cannot be too large, and this upper bound depends on its expectation (of course, the higher the expectation is, the more likely X is large).

Chebyshev's inequality:

- Intuition: suppose X has a finite variance, the departure of X from its mean should depend on the variance: if the variance is small, small departure. We could define a bound of the departure using the variance.

- Theorem: X is a random variable with mean μ and variance σ^2, then we have:

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2} \tag{1.9}$$

- Proof 1: we consider only the case of $\mu = 0$. The idea is with large departure, i.e. large x, $x^2 f(x)$ integration can be large, but this integral is bounded by σ^2. We have:

$$\sigma^2 = \int_{-\infty}^{+\infty} x^2 f(x)dx \geq \int_{-\infty}^{-k\sigma} k^2\sigma^2 f(x)dx + \int_{-k\sigma}^{+k\sigma} x^2 f(x)dx + \int_{+k\sigma}^{+\infty} k^2\sigma^2 f(x)dx \tag{1.10}$$

  The RHS is larger than or equal to $k^2\sigma^2 P(|x| \geq k\sigma)$.

- Proof 2: we can also apply Markov's Inequality to the random variable, $Y = |X - \mu|/\sigma$.

Functions of random variables:

- Application of Change of Variable Theorem: suppose we have n-dim. random variable X, and $Y = \phi(X)$ be a function of X also in n-dim. We consider the probability mass $f_Y(y)dy$ near y, where f_Y is the pdf. of Y. Under the mapping ϕ^{-1}, the volume of the region dy becomes $\det D\phi^{-1}(y)dx$ where D is the derivative (Jacobian) of ϕ^{-1}. The probability mass should be equal, thus we have (eliminating dx):

$$f_Y(y) = f_X(\phi^{-1}(y))|\det D\phi^{-1}(y)| \tag{1.11}$$

- Unequal dimensions: when the dimensions of X and Y are not equal, e.g. we know the joint distribution of (X, Y), and want to find the distribution of $g(X, Y)$, we could add additional auxilary variables s.t. the dimensions are equal. In this example, we could define:

$$U = g(X, Y) \qquad V = Y \tag{1.12}$$

  And apply the Theorem on the mapping $(X, Y) \to (U, V)$.

Moment generating functions (MGF):

- MGF: a function may be characterized (or even defined) via all of its moments, thus we could define a generating function of the moments, and the original function can be studied using this MGF. Defintion: MGF of a random variable X:

$$M_X(t) = \sum_{k=0}^{\infty} \frac{\mathrm{E}\,(X^k)}{k!} t^k = \mathrm{E}\,[e^{tX}] \tag{1.13}$$

Note that the last step comes from the Taylor expansion and we add the coefficients $1/k!$ so that the MGF is an expectation. For continuous RVs, we have:

$$M_X(t) = \int_{-\infty}^{+\infty} e^{tx} f_X(x) dx \tag{1.14}$$

If X is discrete:

$$M_X(t) = \sum_k P(X = k) e^{kt} \tag{1.15}$$

From the Taylor expansion above, it is easy to see that the k-th moment is the k-th derivative of the MGF at 0.

$$\mathrm{E}\left(X^k\right) = M_X^{(k)}(0) \tag{1.16}$$

Note that the MGF of a RV may not exist if the series does not converge.

- Uniqueness: it is possible to have two differnet distributions with exactly the same sets of moments. The following conditions uniquely define a distribution:
  - If X and Y have bounded support, then $F_X(u) = F_Y(u)$ for all u if and only if $\mathrm{E}\left(X^r\right) = \mathrm{E}\left(Y^r\right)$ for all integers $r = 0, 1, \cdots$.
  - If the MGF exist and $M_X(t) = M_Y(t)$ for all t in a neighborhood of 0, then $F_X(u) = F_Y(u)$ for all u.
- Convergence: under some conditions, the convergence of MGF implies the convergence of CDF. Suppose

$$\lim_{n\to\infty} M_{X_n}(t) = M_X(t) \qquad \text{for all } t \text{ iin a neighborhood of } 0 \tag{1.17}$$

then there exists a unique CDF F_X whose moments are determined by $M_X(t)$ and for all x, we have:

$$\lim_{n\to\infty} F_{X_n}(t) = F_X(t) \tag{1.18}$$

- Remark: the proof of uniqueness and convergence Theorems rely on the theory of Laplace transform.

Properties of MGF:

- Linear function of RVs: the MGF of the random variable $aX + b$ is given by:

$$M_{aX+b}(t) = e^{bt} M_X(at) \tag{1.19}$$

- Sum of independent RVs: X and Y are independent RVs, then

$$M_{X+Y}(t) = M_X(t) M_Y(t) \tag{1.20}$$

1.2.1 Convergence of Random Variables

Reference: [Casella, Statistical Inference, Chapter 5]

Conceptual foundation of frequentist statistics:

- Estimator behavior: the basic idea is that estimator is a random variable indexed by n (the sample size), and under frequentist statistics, we are interested in whether the estimator, W_n (1) converges to the true parameter θ, and (2) how fast/efficient the convergence is, e.g. the variance of $\sqrt{n}(W_n - \theta)$.
- Simple estimators: we start from the simplest case, that $\bar{X}_n$ is an estimator of μ, and the behavior of this estimator. These are provided by WLLN and CLT. The most sophisticated estimators then can be built from sample mean, sample variance, etc.

- Idea of approximation: a distribution is characterized by its mean, variance, and other moments, and intuitively, the higher-order moments are less important. Ex. if $Z_n \to 0$, then the distribution of Z_n is mostly determined by its variance, so we can study the convergence behavior of Z_n through the behavior of its variance.
  - Technically, this can be addressed using MGF or characteristic function.
- Remark: in real analysis, we define or approximate a function through an infinite sequence or series of functions. Similarly, we could study/approximate a random variable by a sequence of random variables, or conversely, study the convergence behavior of a sequence of random variables (e.g. large-sample behavior of estimators).

Convergence in probability and Weak Law of Large Numbers (WLLN):

- Convergence in probability: a sequence of RVs $\{X_n\}$ converges in probability to a RV X if $\forall \epsilon > 0$, $P(|X_n - X| < \epsilon) \to 1$ as $n \to \infty$.
  - Consistency: a squence of the "same" sample quantity approaches a constant as $n \to \infty$.
  - Remark: convergence in probability means as $n \to \infty$, most of the probability mass is concentrated around X. It is a strong form of convergence, often used for convergence to a constant. For instance, if $X_n = X$, then clearly the sequence converges in distribution, but not in probability in general.
- Theorem (10.1.3): if $\{X_n\}$ satisfies: (1) $\mathrm{E}(X_n) \to \mu$, and (2) $\mathrm{Var}(X_n) \to 0$, then $X_n \to \mu$ in probability as $n \to \infty$.
  Proof: by Chebshev's inequality,
$$P(|X_n - \mu| \geq \epsilon) \leq \frac{\mathrm{Var}(X_n)}{\epsilon^2} \tag{1.21}$$
- Theorem: if $\{X_n\}$ convergence in probability to X, and h is a continuous function, then the sequence $\{h(X_n)\}$ converges in probability to $h(X)$.
- Consistency of sample mean (WLLN): let $X_1, X_2, \cdots$ be i.i.d random variables with $\mathrm{E}(X_i) = \mu$ and $\mathrm{Var}(X_i) = \sigma^2 < \infty$. Then $\bar{X}_n$ converges in probability to μ.
  Proof: check the condition of the previous theorem, in particular, the variance of $\bar{X}_n$ is equal to σ^2/n, which converges to 0.
- Consistency of sample variance: let $X_1, X_2, \cdots$ be i.i.d random variables with $\mathrm{E}(X_i) = \mu$ and $\mathrm{Var}(X_i) = \sigma^2 < \infty$. Define:
$$S_n^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X}_n)^2 \tag{1.22}$$
  Then S_n^2 converges in probability to σ^2.

Convergence in distribution and Central Limit Theorem (CLT):

- Convergence in distribution: a sequence of RVs $\{X_n\}$ converges in distribution to X if
$$\lim_{n \to \infty} F_{X_n}(x) = F_X(x) \tag{1.23}$$
  at all points x where $F_X(x)$ is contiuous, i.e. the CDF of X_n converges pointwise to the CDF of X.
- Relationship to convergence in probability: if a sequence $\{X_n\}$ converges to probability in X, then it must converge to X in distribution. The two are equivalent if X is a constant.

- CLT: suppose $\{X_n\}$ is a sequence of i.i.d. random variables with $\mathrm{E}(X_i) = \mu$ and $\mathrm{Var}(X_i) = \sigma^2 < \infty$. Then

$$\sqrt{n}(\bar{X}_n - \mu)/\sigma \to N(0,1) \text{ in distribution} \tag{1.24}$$

Proof: we consider the case where the MGF of X_n exists in a neighborhood of 0. Let $Y_i = (X_i - \mu)/\sigma$, and $M_Y(t)$ be the common MGF of Y_i, then the MGF of our target RV can be expressed as:

$$M_{\sqrt{n}(\bar{X}_n - \mu)/\sigma}(t) = \left[M_Y\left(\frac{t}{\sqrt{n}}\right)\right]^n \tag{1.25}$$

Note that as $n \to \infty$, $t/\sqrt{n} \to 0$, thus near 0, the above MGF can be approximated by Taylor expansion, noting that $M_Y(0) = 1, M_Y'(0) = 1, M_Y''(0) = 1$:

$$\left[M_Y\left(\frac{t}{\sqrt{n}}\right)\right]^n \approx \left(1 + \frac{t^2}{2n}\right)^n \to e^{t^2/2} \tag{1.26}$$

1.2.2 Random Vectors

Expectation and variance of random vectors:

- Ref: `http://www.statpower.net/Content/313/Lecture%20Notes/MatrixExpectedValue.pdf`
- Expectation: let x and y be n-dim. random vectors. Suppose A is a given matrix, then:

$$\mathrm{E}(Ax) = A\mathrm{E}(x) \qquad \mathrm{E}(x+y) = \mathrm{E}(x) + \mathrm{E}(y) \qquad \mathrm{E}(x^T) = (\mathrm{E}(x))^T \tag{1.27}$$

The implication is that we can do expected value algebra for matrices, e.g.

$$\mathrm{E}(ABxyC) = AB \cdot \mathrm{E}(xy) \cdot C \tag{1.28}$$

- Covariance matrix: suppose we have a n-dim. random vector (column) x, with $\mathrm{E}(x) = \mu$. The covariance matrix of x is given by:

$$\mathrm{Cov}(x) = [\mathrm{Cov}(x_i, x_j)]_{n\times n} = \mathrm{E}[(x_i - \mu_i)(x_j - \mu_j)] = \mathrm{E}[(x-\mu)(x-\mu)^T] = \mathrm{E}(xx^T) - \mu\mu^T \tag{1.29}$$

- Covariance of two random vectors: let x and y be two random vectors of $m \times 1$ and $n \times 1$ respectively. The covariance of x and y is $m \times n$ matrix defined by:

$$\mathrm{Cov}(x, y) = \mathrm{E}(xy^T) - \mathrm{E}(x)\mathrm{E}(y)^T \tag{1.30}$$

Suppose A and B are two matrices that can multiply with x and y, then we have:

$$\mathrm{Cov}(Ax, By) = A\mathrm{Cov}(x, y)B^T \tag{1.31}$$

- Linear transformation of random vector: suppose x is a $n \times 1$ random vector with mean μ and variance Σ, and A a matrix $m \times n$, then we have:

$$\mathrm{E}(Ax) = A\mu \qquad \mathrm{Var}(Ax) = A\Sigma A^T \tag{1.32}$$

Note: this result is true even if x is not normally distributed.
Proof: we use the fact that $\Sigma = \mathrm{E}(xx^T) - \mu\mu^T$:

$$\mathrm{Var}\, Ax = \mathrm{E}((Ax)(Ax)^T) - (A\mu)(A\mu)^T = A\mathrm{E}(xx^T)A^T - A\mu\mu^T A^T = A\Sigma A^T. \tag{1.33}$$

Similarly, we have:

$$\mathrm{E}(x^T A) = \mathrm{E}(x^T)A \qquad \mathrm{Var}(x^T A) = A^T \cdot \mathrm{Var}(x^T) \cdot A \tag{1.34}$$

- Covariance of random variables that are linear functions of a random vector: let G be a random vector, and u, v are vectors (constants). Let $X = G^T u$ and $Y = G^T v$ (both are scalar), then the sample covariance between the two is:

$$\mathrm{Cov}\,(X, Y) = \mathrm{Cov}\,(\sum_i G_i u_i, \sum_i G_j v_j) = \sum_{i,j} u_i v_j \mathrm{Cov}\,(G_i, G_j) = u^T \mathrm{Cov}\,(G) v \tag{1.35}$$

  where $\mathrm{Cov}\,(G)$ is the covariance matrix of the random vector G. We also have: the variance $\mathrm{Var}\,(X) = u^T \mathrm{Cov}\,(G) u$.
  Remark: this is useful in the case of MR and TWAS, where X and Y are exposure and outcome, respectively.

Quadratic form of random vectors:

- Ref: `quadratic-form-random-vector.pdf`.
- Theorem: x is n-dim. random vector with mean μ and variance Σ, and A is a symmetric matrix, we have:

$$\mathrm{E}\,(x^T A x) = \mathrm{tr}\,(A \mathrm{E}\,(x x^T)) = \mathrm{tr}\,(A\Sigma) + \mu^T A \mu \tag{1.36}$$

- Proof: we first use the fact that scalar of trace is just the scalar, and the fact that scalar and expectation could commute:

$$\mathrm{E}\,(x^T A x) = \mathrm{tr}\,(\mathrm{E}\,(x^T A x)) = \mathrm{E}\,(\mathrm{tr}\,(x^T A x)) \tag{1.37}$$

  Now we use $\mathrm{tr}\,(AB) = \mathrm{tr}\,(BA)$ for any two matrices:

$$\mathrm{E}\,(x^T A x) = \mathrm{E}\,(\mathrm{tr}\,(A x x^T)) = \mathrm{tr}\,(\mathrm{E}\,(A x x^T)) = \mathrm{tr}\,(A \mathrm{E}\,(x x^T)) \tag{1.38}$$

  This completes the first part of the theorem. For the second part, we use the covariance matrix of x: $\Sigma = \mathrm{E}\,(x x^T) - \mu \mu^T$:

$$\mathrm{E}\,(x^T A x) = \mathrm{tr}\,(A \cdot (\Sigma + \mu \mu^T)) = \mathrm{tr}\,(A\Sigma) + \mathrm{tr}\,(A \mu \mu^T) = \mathrm{tr}\,(A\Sigma) + \mu^T A \mu \tag{1.39}$$

  where we use: $\mathrm{tr}\,(A \mu \mu^T) = \mathrm{tr}\,((A\mu) \cdot \mu^T) = \mathrm{tr}\,(\mu^T A \mu) = \mu^T A \mu$.

1.3 Probability Distributions

Reference: [Casella, Statistical Inference, 5.2]
Negative Binomial distribution:

- Ref: Wiki and `https://probabilityandstats.wordpress.com/tag/poisson-gamma-mixture/`
- Waiting time in Bernoulli process (Pascal distribution): when r is an integer, NB is the number of successes before the r-th failure in a Bernoulli process, with probability p of successes on each trial. When $r = 1$, this is geometric distribution. The PMF of NB:

$$P(X = k | r, p) = \binom{k + r - 1}{k} p^k (1 - p)^r \tag{1.40}$$

  The expectation and variance of X:

$$\mathrm{E}\,(X) = \frac{pr}{1 - p} \qquad \mathrm{Var}\,(X) = \frac{pr}{(1 - p)^2} \qquad \frac{\mathrm{Var}\,(X)}{\mathrm{E}\,(X)} = \frac{1}{1 - p} \tag{1.41}$$

  The last term is the "index of overdispersion", and it depends only on p. When p is small, the index is close to 1, and the distribution is close to Poisson.

- Poisson-Gamma mixture: if $X|\theta \sim \text{Pois}(\theta)$ and $\theta \sim \text{Gamma}(\alpha, \beta)$, where α, β are shape and rate parameter, respectively, then $X \sim NB(r = \beta, p = \frac{\alpha}{\alpha+1})$.
- Parameterization in NB regression: in regression problem, we parameterize with mean, which depends on covariates. Typically, we use $X \sim NB(\mu, \theta)$, where $\mu = \text{E}(X)$, and θ is the overdispersion parameter defined by $\text{Var}(X) = \mu + \theta\mu^2$. Some authors parameterize using overdispersion parameter as $1/\theta$.

Gaussian integral:

- Gaussian integral:
$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi} \tag{1.42}$$
- Proof: compute the following integral by double integration:
$$\int_{-\infty}^{\infty} e^{-(x^2+y^2)} dxdy = \left(\int_{-\infty}^{\infty} e^{-x^2} dx \right)^2 \tag{1.43}$$
The LHS can also be computed using the polar coordinate transformation. Equating the two gives the equation.

Sample mean and sample variance: parameter estimation for $N(\mu, \sigma^2)$. Suppose $X_1, \cdots, X_n$ iid $N(\mu, \sigma^2)$:

- Sample mean: the estimator of μ: $\bar{X} \sim N(\mu, \sigma^2/n)$.
- Sample variance: the estimator of σ^2 is the mean squared error:
$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2 \tag{1.44}$$
Intuition: S^2 is the estimator of the variability in the sample, if μ is known, then S^2 should be the mean variance (divided by n); since $\bar{X}$ is used, the n terms are subject to one constraint (sum to 0), thus the total variability is slightly less (thus divided by $n-1$).
- Theorem: given $x_1, \cdots, x_n$, we have
$$(n-1)s^2 = \sum_{i=1}^{n} (x_i - \bar{x})^2 = \sum_{i=1}^{n} x_i^2 - n\bar{x}^2 \tag{1.45}$$
Proof: take $c = 0$ in the Equation 1.3.
- Theorem: $E(\bar{X}) = \mu$, $\text{Var}(\bar{X}) = \sigma^2/n$, $E(S^2) = \sigma^2$.
Proof: the first two can be easily proven with the iid. property of the random sample. For the last, using the Theorem above:
$$\sum_{i=1}^{n} (X_i - \bar{X})^2 = \sum_{i=1}^{n} (X_i - \mu + \mu - \bar{X})^2 = \sum_{i=1}^{n} (X_i - \mu)^2 - n(\bar{X} - \mu)^2 \tag{1.46}$$
- Theorem: $\bar{X}$ and S^2 are independent RVs, $\bar{X} \sim N(\mu, \sigma^2/n)$ and $(n-1)S^2/\sigma^2 \sim \chi^2_{n-1}$.
  - Independence of $\bar{X}$ and S^2: define $Y_1 = \bar{X}$, $Y_2 = X_2 - \bar{X}, \cdots, Y_n = X_n - \bar{X}$, and S^2 can be expressed as a function of $Y_2, \cdots, Y_n$. It can be shown that $Y_1 = \bar{X}$ is indepenent of $Y_2, \cdots, Y_n$: (1) by apply the linear transformation to the joint pdf. of $X_1, \cdots, X_n$); (2) using the fact that both $\bar{X}$ and $X_j - \bar{X}$ are linear functions of $X_1, \cdots, X_n$ (calculate the covarance of the two RVs).

- Distribution of sample variance: proof by induction. Wlos., assume that $\mu = 0$ and $\sigma = 1$. First, at $n = 2$, $S_2^2 = (X_2 - X_1)^2/2$, and it is easy to show that S_2^2 follows χ_1^2. Then at $n = k + 1$, kS_{k+1}^2 can be decomposed as a sum of two χ^2 distributions: $(k-1)S_k^2$ and $(X_{k+1} - \bar{X}_k)^2$ (up to a constant).
- Remark: when μ is known, $(n-1)S^2/\sigma^2 = \sum_i (\frac{X_i - \mu}{\sigma})^2$ is a sum of n i.i.d standard normal RVs, thus follows χ_n^2 distribution. When μ is unknown, needs some correction, but still χ^2 distribution.

- Z-score: when σ^2 is unknown, the distribution of $\bar{X}$ is unknown. Thus using S instead of σ:

$$\sqrt{n}(\bar{X} - \mu)/S \sim t_{n-1} \tag{1.47}$$

  This would allow one to determine the confidence interval of μ.

- Evaluating the estimators: from the previous theorems, $\bar{X}$ and S^2 are unbiased estimators of μ and σ^2. The variance of $\bar{X}$ is σ^2/n, and the variance of S^2 can be obtained from the χ^2 distribution:

$$\text{Var}\left((n-1)S^2/\sigma^2\right) = \frac{(n-1)^2}{\sigma^4}\text{Var}\left(S^2\right) = 2(n-1) \Rightarrow \text{Var}\left(S^2\right) = \frac{2\sigma^4}{n-1} \tag{1.48}$$

  Thus, both $\bar{X}$ and S^2 converge to the true values of the parameters at the rate of $1/n$.

Properties of normal distribution:

- Sum of normal random variables: suppose $X_i \sim N(\mu_i, \sigma_i^2)$, then

$$\sum_i w_i X_i \sim N(\sum_i w_i \mu_i, \sum_i w_i^2 \sigma_i^2) \tag{1.49}$$

- Product of normal density functions: suppose we have $N(x|\mu_1, \sigma_1^2)$ and $N(x|\mu_2, \sigma_2^2)$, the product of the density functions:

$$N(x|\mu_1, \sigma_1^2)N(x|\mu_2, \sigma_2^2) = N(x|\mu, \sigma^2)N(\mu_1|\mu_2, \sigma_1^2 + \sigma_2^2) \tag{1.50}$$

  where:

$$\mu = \frac{1/\sigma_1^2}{1/\sigma_1^2 + 1/\sigma_2^2}\mu_1 + \frac{1/\sigma_2^2}{1/\sigma_1^2 + 1/\sigma_2^2}\mu_2 \tag{1.51}$$

  and

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \tag{1.52}$$

  The proof follows from the quadratic form of x. Take the integral of x:

$$\int N(x|\mu_1, \sigma_1^2)N(x|\mu_2, \sigma_2^2)dx = N(\mu_1|\mu_2, \sigma_1^2 + \sigma_2^2) \tag{1.53}$$

- Mixture of normal random variables: suppose we have $X|\theta \sim N(\theta, \sigma^2)$ and $\theta \sim N(\mu, \tau^2)$, then the marginal distribution:

$$X \sim N(\mu, \sigma^2 + \tau^2) \tag{1.54}$$

  To see this, we write the marginal as the integral of $N(\theta|x, \sigma^2)N(\theta|\mu, \tau^2)$ over θ, and apply the Equation 1.53 above.

Pooled variance:

- Problem: suppose there are K groups, each group is from a normal distribution, with different mean but the same variance (e.g. additional dependent variable that changes mean, but not variance), and our goal is to estimate the variance by pooling all groups.

- Pooled variance [Wiki]: suppose the sample variance of the i-th group is S_i^2, and the sample size of the i-th group is n_i, then the estimated variance S_p^2 is given by:

$$S_p^2 = \frac{\sum_i (n_i - 1) S_i^2}{\sum_i (n_i - 1)} \tag{1.55}$$

Multinomial distribution:

- Suppose $X_1, \cdots, X_k \sim \text{Mul}(n; p_1, \cdots, p_k)$, where $\sum_{i=1}^k p_i = 1$. The pmf is given by:

$$P(x_1, \cdots, x_k) = \frac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k} \tag{1.56}$$

- Properties: $E(X_i) = np_i$. The covariance matrix is given by:

$$\text{Var}(X_i) = np_i(1 - p_i) \qquad \text{Cov}(X_i, X_j) = -np_i p_j \tag{1.57}$$

  Proof: the variance is easy to prove using binomial distribution. For the covariance, use proof by induction: simple at $k = 2$; at larger k, reduce by letting $X_k + X_{k+1}$ as a single RV, and apply the induction hypothesis.

Gamma and inverse gamma distributions:

- Gamma distribution: defined on $x \geq 0$

$$f(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha - 1} e^{-\beta x} \tag{1.58}$$

  where α and β are called the shape and rate(scale) parameter, respectively.

- Inverse gamma distribution: if X is gamma RV, then $1/X$ follows inverse Gamma distribution. Its density:

$$f(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha - 1} \exp\left(-\frac{\beta}{x}\right) \tag{1.59}$$

- Remark: a number of other distributions have this form of density function and special cases of gamma and inveser-gamma distributions.

χ^2 distribution:

- Definition: the density of χ^2 distribution with dof. equal to ν:

$$f(x; \nu) \propto x^{\nu/2 - 1} \exp(-x/2) \tag{1.60}$$

- Theorem: if $Z \sim N(0, 1)$, then $Z^2 \sim \chi_1^2$.
- Theorem: if $X_1, \cdots, X_n$ are independent, and $X_i \sim \chi_{p_i}^2$, then $X_1 + \cdots + X_n \sim \chi_{p_1 + \cdots + p_n}^2$.
- Mean and variance: if $X \sim \chi_k^2$, then $\text{E}(X) = k$ and $\text{Var}(X) = 2k$.

Inverse-χ^2 distribution:

- Inverse χ^2 distribution: if X has the χ^2 distribution with ν degrees of freedom, then $1/X$ has the inverse-χ^2 distribution with ν degrees of freedom.

$$f(x; \nu) \propto x^{-(\nu/2 + 1)} \exp\left(-\frac{1}{2x}\right) \tag{1.61}$$

  Intuitively, with large degree, chi-square RV would have large mean, thus its inverse (inverse-χ^2 RV) would have a large peak near 0.

- Scaled-inverse χ^2 distribution: if $X \sim$ Scaled-Inv-$\chi^2(\nu, \sigma^2)$, then $\frac{X}{\sigma^2 \nu} \sim$ Inv-$\chi^2(\nu)$. Thus it is basically an inverse χ^2 distribution with σ^2 as the unit. Its shape is determined by ν, and its scale determined by σ^2 (large σ means that the distribution is broader).

$$f(x; \nu, \sigma^2) \propto x^{-(\nu/2+1)} \exp\left(-\frac{\nu\sigma^2}{2x}\right) \tag{1.62}$$

  - Mean: when $\nu > 2$, the mean is $\frac{\nu}{\nu-2}\sigma^2$.
  - Mode: $\frac{\nu}{\nu+2}\sigma^2$.

- Remark: these are all special cases of inverse gamma distribution.

Student's t and F distribution:

- Student's t distribution: if U is a standard nomral distribution, V is χ^2_p, then $U/\sqrt{V/P}$ follows t distribution with dof $p-1$.

- Theorem: given a random normal sample,

$$\frac{\bar{X} - \mu}{S/\sqrt{n}} \sim t_{n-1} \tag{1.63}$$

  Proof: divide by $\sigma/\sqrt{n}$ in both the numerator and denomiator.

- F distribution: if $U \sim \chi^2_p$ and $V \sim \chi^2_q$ and U, V are independent, then $(U/p)/(V/q)$ follows F distribution with dof $(p-1, q-1)$.

- Theorem: let $X_1, \cdots, X_n$ iid. $N(\mu_X, \sigma_X^2)$, and $Y_1, \cdots, Y_m$ iid. $N(\mu_Y, \sigma_Y^2)$ (independent of X), then:

$$\frac{S_X^2/\sigma_X^2}{S_Y^2/\sigma_Y^2} \sim F_{n-1,m-1} \tag{1.64}$$

  Proof: use the χ^2 distributions of S_X^2 and S_Y^2.

Laplace distribution [Murphy, Section 2.4]

- PDF: $\text{Lap}(x|\mu, b) = \frac{1}{2b}\exp\left(|(x-\mu)|/b\right)$, where μ is the mean and b scale parameter. The probability mass is more concentrated near 0, and has a large tail (similar to spike-and-slab).

- Application in linear regression with outliers: normal error function is sensitive to outliers, which have large effect on the regression estimates. Using Laplace error function, the estimates are less sensitive.

Log-normal distribution:

- Defintion: a random variable X is log-normally distributed, if $Y = \log X$ is normally distributed. We write:

$$X \sim LN(\mu, \sigma^2) \Leftrightarrow Y = \log X \sim N(\mu, \sigma^2) \tag{1.65}$$

- Moments of log-normal distribution: if $X \sim LN(\mu, \sigma^2)$, we have:

$$\mathrm{E}(X) = e^{\mu + \frac{1}{2}\sigma^2} \tag{1.66}$$

  The s-th moment, where s is a real or complex number is given by:

$$\mathrm{E}(X^s) = e^{s\mu + \frac{1}{2}s^2\sigma^2} \tag{1.67}$$

- Multivariate log-normal distribution: if X is MNV $N(\mu, \Sigma)$, then $Y = \exp(X)$ follows multivariate log-normal, and the mean of Y has a simple closed form:

$$\mathrm{E}(Y_i) = e^{\mu_i + \frac{1}{1}\Sigma_{ii}} \tag{1.68}$$

- For computing the higher-order central moments of multivariate log-normal distribution: see Reference: A Recursive Formula for Computing Central Moments of a Multivariate Lognormal Distribution.

1.4 Parameter Estimation

1. Methods of point estimation

   Reference: [Casella, Statistical Inference, 7.2, 7.3]

   Overview of parameter estimation:

   - Intuition: suppose W is an estimator of θ, then W should contain information of θ. Intuitively, different values of θ should lead to different (mean) values of W. Example, in normal distribution, $\bar{X}$ is informative of μ but not σ^2, as its expectation is equal to μ but independent of σ.
   - The perspective of matching histogram: from the frequentist perspective, as the sample size approaches infinity, the empirical distribution should match the true distribution. Thus the parameter θ that leads to a match between f_θ and the empirical distribution $\hat{f}$ is a good estimator. Based on this perpsective, we have:
     - MOM estimator: match the moments of f_θ and the sample moments.
     - MLE estimator: minimizing the KL divergence, $KL(\hat{f}||f_\theta)$, gives the ML estimator (see the section on information theoretical perspective on statistical inference).
   - Evaluating estimators: we need a way to quantify how much information W contains on θ. The important measures of an estimator are: whether it is unbiased, the variance of W, and how fast it converges to the true value θ. In particular, $\mathrm{Var}\,(W)$ can be used to derive the confidence interval of θ, thus important.
   - Method of moment (MOM) estimation: the general idea is suppose we define an informative pattern W, i.e. $\mathrm{E}\,(W)$ is some function of θ, $h(\theta)$, then we could use $h^{-1}(W)$ as an estimator of θ. Often the patterns are mean, variance, covariance, but any other patterns follow the same idea.
   - MLE: follows from the likelihood principle, that all information of θ is contained in the likelihood function $L(\theta)$.
   - Error minimization for prediction problems: the idea is to minimize the difference between the model predictions and observations. We could define the error as the objective function to be minimized. For the least square method, let $f(\cdot)$ be our function, the parameters are estimated by:
     $$\min F(\theta) = \sum_i [y_i - f(x_i; \theta)]^2 \tag{1.69}$$
   - Partial likelihood and conditional distributions: sometimes we do not have to model the complete likelihood of the data, especially if we are interested in only part of the parameters. We could then use part of the likelihood that contain the interested parameter; or use conditional distributions that is independent of nuisance parameters.
     - Two-sample Poisson test: suppose we only know the ratio of t_1/t_2, then use the conditional distribution $P(x_1|x_1 + x_2)$ has the advantage that it does not depend on the absolute values of t_1 and t_2.
     - TDT in genetics: using $P(G|G_p, Y)$ where G, G_p are genotypes of child and parent, and Y the phenotype of child, we can avoid parameters of genotype frequency in the population.

   Properties of point estimator: these are desirable properties of $\hat{\theta}_n$:

   - Unbiasedness: $E(\hat{\theta}_n) = \theta$.
   - Sufficiency: $f(x_1, \cdots, x_n|\hat{\theta}_n)$ does not depend on θ.
   - Minimum MSE: $E[(\hat{\theta}_n - \theta)^2]$ is minimum.
   - Minmum variance unbiased: $\mathrm{Var}(\hat{\theta}_n) \leq \mathrm{Var}(\hat{\theta}_n^*)$ for any other estimator $\hat{\theta}_n^*$.

   Procedure of point estimation:

- Estimator: for a given parameter estimation problem, find the estimator (MOM, MLE, etc.) of θ, called W.
- Evaluating the estimator: using the Theorem of Point Approximation, the mean squared error (MSE) of an estimator W is given by:

$$\mathrm{MSE}_\theta(W) = E_\theta(W-\theta)^2 = [E_\theta(W)-\theta]^2 + \mathrm{Var}_\theta(W) \tag{1.70}$$

  The subscript means MSE, bias and variance all depend on the true value of θ. Since this is usually unknown, the evaluation of W in a practical problem is performed on the estimated value of θ, e.g. MLE. For unbiased estimator, the MSE is given by the variance of the estimator.
- Obtaining variance of estimator: and similarly for condifence interval. For difficult problem, one has some options:
  - Asymptotic results: e.g. asymptotic normality of MLE.
  - Parametric bootstrap: suppose $\hat{\theta}$ is the estimator (often MLE) of θ from the data. We could simulate data many times assuming $\hat{\theta}$ is the true value of θ, and compute the estimator W for each data set (note: W is the estimator of the paramter of interest). This allows one to obtain the distribution of W.
  - Sampling with replacement (nonparameteric bootstrapping): the benefit is that it is independent of the assumption on the parametric form of the distribution.
- What determines the variance of an estimator? Sample size is a major determinant. In many other cases, the variance may be thought of some variation or effective sample size.
  - Normal distribution: $X_i \sim N(\mu, \sigma^2)$, the variance of $\hat{\mu}$ is σ^2/n.
  - Poisson distribution: $X \sim \mathrm{Pois}(\lambda t)$, then $\hat{\lambda} = X/t$. It follows Poisson distribution, and $\mathrm{Var}\,\hat{\lambda} = X/t$, where t is similar to the sample size.
  - Simple linear regression: $\mathrm{Var}\,(\hat{\beta}_1) = \sigma^2/\sum_i (X_i - \bar{X})^2$, where the variance of X can be viewed as some effective sample size. Effectively, when $X_i = \bar{X}$, then sample i does not contribute (no information of β).
- Remark:
  - The decompoision of the MSE of the estimator is the result of Theorem of Point Approximation, the perspective here is opposite: we are not approximating a RV with a constant, but rather, estimate the error when we are infering an (unknown) constant using some distribution.
  - The general idea of MSE decomposition is: partition the error into two parts, one part depends on the truth (bias), and the other only depends on the property of the estimator itself (variance).

Method of moment (MOM) estimation:

- General strategy: suppose T is some statistic (pattern) from data, if we have $\mathrm{E}\,(T) = h(\theta)$, then we could define $W = h^{-1}(T)$ as an estimator of θ. If h is linear, we have:

$$\mathrm{E}\,(W) = \mathrm{E}\,[h^{-1}(T)] = h^{-1}(\mathrm{E}\,(T)) = \theta \tag{1.71}$$

  Thus W is an unbiased estimator of θ. In general, if h is not linear (e.g. convex), then the above inequality does not hold, but W may still be an estimator.
- MOM: a special type of patterns in the data can be expressed as the tendency (mean) of variables and the relationship (covariance) between variables. To formulate this idea, we consider the joint distribution of random variables involved in the model, and analyze the population mean, variance and covariance of the random variables: (generally)

$$\mathrm{E}\,(X_i) = f_i(\theta) \qquad \mathrm{Cov}\,(X_i, X_j) = g_{ij}(\theta) \tag{1.72}$$

where θ is model parameters. If we can derive the sample mean, variance, covariance (that estimate the population quantities above), then we could equate these statistics with the functions f_i and g_{ij} above.

- Other examples of MOM strategy:
  - Regression: the conditional covariance of Y on X_j when other variables are fixed, carries information of β_j.
  - SEM: a special case of MVN, the sample covariance matrix is equal to covariance matrix $\Sigma(\theta)$, which is a function of θ.
  - Markov chain: suppose we have a k-state Markov chain, the statistics such as: frequencey of each state, the mean length of sequential runs of each state, the frequency of transitions, are informative of the parameters of the Markov chain.
- Remark: the difficulties of MOM:
  - Informative estimators: may not be obvious. Ex, for a HMM where the states are not observed, the informative statistics are not obvious, unlike the Markov chain case.
  - Non-linearity: suppose $\mathrm{E}(T) = h(\theta)$, if h is nonlinear, then $W = h^{-1}(T)$ is not an unbiased estimator of θ.

Model identifiability:

- Intuition/motivation: even if a statistical model is well-defined, the parameters may not be identifiable (see examples below). Intuitively, it is identifiable only when different model parameters lead to different distributions/data characteristics. The definition: a statistical model with parameters θ is identifiable if:
$$P_{\theta_1} = P_{\theta_2} \Rightarrow \theta_1 = \theta_2 \qquad \forall \theta_1, \theta_2 \in \Theta \tag{1.73}$$
where Θ is the parameter space.
- Examples:
  - Linear regression: if there is linear dependence among features, then the matrix $(X^T X)$ is singular, and the coefficients not identifiable. Intuitively, if X_1 and X_2 are correlated, then having large β_1, small β_2 would be indistinguishable from small β_1, large β_2.
  - Mixture model: two classes of sites with the mixing θ unknown, some evolve faster α, some slower α_0. The model may not be identifiable, as having large θ and small α may be similar to having a smaller θ, but larger α.
- Model identification via the number of parameters: intuitively, if there are more parameters than the number of constraints/degree of freedom in the data, then the model is not identified.
  - Example: linear regression, $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$, X_1 is perfected correlated to X_2. Then the joint distribution of all variables has only one true parameter related to the covariance between X_1 (or X_2) and Y, but we have two model parameters β_1 and β_2, thus not identifiable.
  - Example: Error-in-variable (EIV) model.
- Model identification via symmetry: if the parameters are exchangable, then they may not be identifed. Ex. mixture model, the indices of components are clearly exchangable.
- Fisher information matrix: the parameter θ is identified at $\hat{\theta}$ if and only if the inverse of the information matrix, $I(\theta)$, exists at $\hat{\theta}$.
  - Intuition: at $\hat{\theta}$, the log-likelihood function has derivative 0. If the second derivative is also 0 (or singular information matrix), the the function is locally flat (hyperplane), thus the parameters are not identified.
  - Alternatively, a small value of $I(\theta)$ implies a large value of $\mathrm{Var}(\hat{\theta})$, at the extreme case of $I(\theta)$ is singular, this suggests that the variance is infinitely larger, i.e. the model is not identified.

2. Fisher information and Cramer-Rao lower bound

   Log-likelihood function and entropy:

   - Motivation: while we are generally inferring an unknown parameter θ from the data, we need to consider the fact that, different θ might generate data with simliar characteristics. So to infer the unknown parameter, we need to consider a family of distributions parameterized by θ, and study how the data characteristics depend on θ - technically this is the likelihood function.
   - Log-likelihood function and parameter identification: suppose we have data x, and we form the log-likelihood function $l(\theta) = \log f(x|\theta)$. In the region where $\log f(x|\theta)$ is flat, different values of θ could lead to the same data, so it is difficult to infer the true value of θ. To characterize the intrisic difficulty of inference (instead of basing on a particular dataset), we should consider the "flatness"/curvature averaged over X, or the curvature of the log-likelihood function when $n \to \infty$. Note: this is the average over the distribution parameterized by the same θ.
   - Uncertainty (information): if $f(X|\theta)$ is always 1, then there is no uncertainty, and in general, we can use the expectation of $f(X|\theta)$ as a measure of uncertainty:

   $$H(\theta) = \mathrm{E}\left(-\log f(X|\theta)\right) = -\int f(x|\theta) \log f(x|\theta) dx \tag{1.74}$$

   Fisher information:

   - Efficient score (or just score): describes how fast the likelihood function changes with the parameter values. Let $f(\theta; X)$ be the dentify function, it is defined as:

   $$V(\theta) = \frac{\partial \log f(\theta; X)}{\partial \theta} \tag{1.75}$$

     V is random variable defined on X. However, we cannot use expectation of V as a measure of how flat the likelihood surface is, as the mean of V is 0. The proof follows from (rewriting the likelihood in terms of the PDF):

   $$V(\theta) = \frac{1}{f(X;\theta)} \frac{\partial f(X;\theta)}{\partial \theta} \tag{1.76}$$

     The integral over the RV $X|\theta$:

   $$\mathrm{E}\left(V|\theta\right) = \int \frac{1}{f(X;\theta)} \frac{\partial f(X;\theta)}{\partial \theta} f(X;\theta) dX = \int \frac{\partial f(X;\theta)}{\partial \theta} dX = \frac{\partial}{\partial \theta} \int f(X;\theta) dX = 0 \tag{1.77}$$

     A better measure is to use the expectation of V^2 (see below).

   - **Remark**: It is important to understand what the property says. When evaluating $\mathrm{E}\left(V|\theta\right)$, we are averaging over all X, assuming X are generated from the same θ. In other words, suppose θ is the true value, and we generate x_i (data) from θ, then we compute the score (E_i) at θ from x_i. Repeat the experment n times, the average of score E_i would be close to 0. Had we evaluate score at a different value of θ, the expectation is not necessarily 0.
     - This property says that the expectation of score should be 0. Intuitively, this should be easy to understand, on average, the log-likelihood function is maximized at the true value θ, so its derivative at θ, on average, should be 0.
   - Fisher information: measures how much information, the data points X carries on the unknown parameter θ, averaging over possible X. It is defined as the square of V, averaging over all possible data:

   $$I(\theta) = E\left[\left(\frac{\partial \log f(\theta; X)}{\partial \theta}\right)^2\right] \tag{1.78}$$

Since the mean of V is 0, it is also the variance of the V. It can be shown that:

$$I(\theta) = -E\left(\frac{\partial^2}{\partial^2\theta}\log f(\theta; X)\right) \tag{1.79}$$

The proof follows from applying the product rule to the second partial derivative:

$$\frac{\partial^2}{\partial^2\theta}\log f(x|\theta) = -\frac{1}{[f(x|\theta)]^2}\left[\frac{\partial}{\partial\theta}f(x|\theta)\right]^2 + \frac{1}{f(x|\theta)}\frac{\partial^2}{\partial^2\theta}f(x|\theta) \tag{1.80}$$

The expectation of the second term above is 0. Therefore, Fisher inforamtion at any θ may be seen as a measure of the "curvature" of the log-likelihood curve, averaging over data points. It measures the intrinsic difficulty of making inference at any θ, thus independen of data (x) and independent of the estimator function used to infer θ.

- Fisher informtion in the iid. case: if we expand the likelihood function using iid data points, we have:

$$I_n(\theta) = nI_1(\theta) \tag{1.81}$$

where $I_n(\theta)$ is Fisher information with n data points, and $I_1(\theta)$ is information at $n = 1$.

- Example: a Bernoulli process with x 1's and $n - x$ 0's. Suppose the probability of success per trial is θ, the score is given by:

$$V = \frac{\partial l(\theta)}{\partial\theta} = \frac{x}{\theta} - \frac{n-x}{1-\theta} \tag{1.82}$$

It is easy to check that $\mathrm{E}(V) = 0$:

$$\mathrm{E}(V|\theta) = \frac{\mathrm{E}(x)}{\theta} - \frac{\mathrm{E}(n-x)}{1-\theta} = \frac{n\theta}{\theta} - \frac{n(1-\theta)}{1-\theta} = 0 \tag{1.83}$$

And the variance is given by:

$$I_n(\theta) = \mathrm{Var}(V|\theta) = \mathrm{Var}\left(\frac{x-n\theta}{\theta(1-\theta)}\right) = \frac{\mathrm{Var}(x)}{\theta^2(1-\theta)^2} = \frac{n}{\theta(1-\theta)} \tag{1.84}$$

We see that for Bernoulli distribution, it is easier to infer θ when θ is close to 0, and more difficult when θ is close to $1/2$ (intuitively, large number of 1's and 0's - large noise).

- Fisher information matrix [Wiki]: in the multivariate case:

$$I(\theta)_{ij} = -\mathrm{E}\left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log f(\theta; X)\right] \tag{1.85}$$

Cramer-Rao lower bound:

- A single parameter: let $\hat{\theta}$ be an unbiased estimator of θ, and it satisfies some regularity condition, then we have:

$$\mathrm{Var}(\hat{\theta}) \geq \frac{1}{I(\theta)} \tag{1.86}$$

Since $\hat{\theta}$ is unbiased, its variance is also its MSE. The intuition of this inequality: the LHS is the error of the estimator. It depends on how difficult it is to estimate θ from data (or how discriminate θ is), or the information observation carries on θ.

- Function of parameter: suppose T is an unbiased estimator of $\tau(\theta)$, we have:

$$\mathrm{Var}(T) \geq \frac{[\tau'(\theta)]^2}{I(\theta)} \tag{1.87}$$

The dependence on $\tau'(\theta)$ can be explained by: when it is large, smaller error of θ (from estimation) means larger error of $\tau(\theta)$.

3. Common parameter estimation problems

   Comparing the means of two samples:

   - Problem: given two independent samples of sizes n_1 and n_2 respectively, where the first sample is from $N(\mu_1, \sigma_1^2)$, and the second from $N(\mu_2, \sigma_2^2)$. We want to estimate the effect size $\Delta = \mu_1 - \mu_2$.
   - The estimator is given by:
   $$D = \bar{X}_1 - \bar{X}_2 \tag{1.88}$$
   The variance of D is given by:
   $$\mathrm{Var}(D) = \mathrm{Var}(\bar{X}_1) + \mathrm{Var}(\bar{X}_2) = \frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2} \approx \frac{S_1^2}{n_1} + \frac{S_2^2}{n_2} \tag{1.89}$$
   where S_1^2 and S_2^2 are the sample variances. The approximation is applied as the true variances of two samples are unknown.

   Estimating odds-ratio in a 2 by 2 table: [Odds ratio, Wiki]

   - Problem: given a 2 by 2 table, let p_{ij} be the probability of the (i, j) cell, and n_{ij} be the observed counts of the (i, j) cell. The odds ratio is defined by:
   $$\mathrm{OR} = \frac{(p_{11}/(p_{11} + p_{10}))\,/\,(p_{10}/(p_{11} + p_{10}))}{(p_{01}/(p_{01} + p_{00}))\,/\,(p_{00}/(p_{01} + p_{00}))} = \frac{p_{11}p_{00}}{p_{10}p_{01}} \tag{1.90}$$
   Our problem is to estimate OR from the observed counts.
   - The problem of estimating odds: to simplify, we first consider the case where we have a binomial sample, and estimate the odds (the prob. of success over the prob. of failure). Suppose our sample has x successes in n trials, with $X \sim \mathrm{Bin}(n, p)$, the log-odds is thus: $\log \frac{p}{1-p}$. The estimator is simply:
   $$\hat{\mathrm{Odds}} = \log \frac{\hat{p}}{1 - \hat{p}} \tag{1.91}$$
   where $\hat{p} = x/n$. The variance of the estimator (MLE) can be calculated using the asymptotic results of MLE. The likelihood function is: $L(p) = p^x(1-p)^{n-x}$, and thus the Fisher information can be computed as the negative second derivative of $\log L(p)$:
   $$I(p) = \frac{x}{p^2} + \frac{n - x}{(1-p)^2} \tag{1.92}$$
   The derivative of the $\log(p/(1-p)) = 1/p + 1/(1-p)$, and plug the asymptotic variance of MLE:
   $$\mathrm{Var}\left(\log \frac{\hat{p}}{1 - \hat{p}}\right) = \left.\frac{\left(\frac{1}{p} + \frac{1}{1-p}\right)^2}{\frac{x}{p^2} + \frac{n-x}{(1-p)^2}}\right|_{p=\hat{p}} = \frac{1}{x} + \frac{1}{n - x} \tag{1.93}$$
   - Estimator of log odds-ratio: it can be shown easily that the estimator of log-odds ratio is:
   $$L = \log \mathrm{OR} = \log \frac{n_{11}n_{00}}{n_{10}n_{01}} \tag{1.94}$$
   The variance of L is simply the sum of the variance of log-odds in the two groups (given above):
   $$\mathrm{Var}(L) = \frac{1}{n_{11}} + \frac{1}{n_{10}} + \frac{1}{n_{01}} + \frac{1}{n_{00}} \tag{1.95}$$

4. Missing data problem

   EM algorithm:

- Intuition of EM algorithm: suppose we have observed data x, parameters θ, and missing data y. Our goal is to maximize the likelihood:

$$l(\theta) = \log P(x|\theta) = \log \int P(x, y|\theta) dy \tag{1.96}$$

  Our intuition is this: if we know θ, then we could estimate the missing data y; using the estimated y in the computation of log-likelihood would lead to a better estimate of θ. However, since the missing y cannot be completely determined, we need to consider the log-likelihood averaging over all possible values of y.
- EM algorithm: at the E-step, we compute this function:

$$Q(\theta|\theta^t) = \mathrm{E}_{y|x,\theta^t} [\log P(x, y|\theta)] \tag{1.97}$$

  assuming that the current estimate is θ^t. At the M-step, we maximize $Q(\theta|\theta^t)$ as a function of θ. For the generalized EM (GEM) algorithm, we only need to find θ that increases $Q(\theta|\theta^t)$.
- Proof of convergence: we show that the EM algorithm always increases the Q function. We note that the log-likelihood of x is related to the complete log-likelihood:

$$\log P(x|\theta) = \log P(x, y|\theta) - \log P(y|x, \theta) \tag{1.98}$$

  Since this is true for any value of y, we could average over $y|x, \theta^t$:

$$\log P(x|\theta) = \mathrm{E}_{y|x,\theta^t} [\log P(x, y|\theta)] - \mathrm{E}_{y|x,\theta^t} [\log P(y|x, \theta)] \tag{1.99}$$

  The first term is exactly the Q function. We have:

$$\log P(x|\theta) - \log P(x|\theta^t) = Q(\theta|\theta^t) - Q(\theta^t|\theta^t) + \mathrm{E}_{y|x,\theta^t} \left[\frac{\log P(y|x, \theta^t)}{P(y|x, \theta)} \right] \tag{1.100}$$

  The last term is always nonnegative according to the KL divergence.
- EM algorithm for independent samples: the Q function can be written as:

$$Q(\theta|\theta^t) = \sum_i \mathrm{E}_{y_i|x_i,\theta^t} [\log P(x_i, y_i|\theta)] \tag{1.101}$$

Extensions of EM:

- Missing information principle: according to Equation 1.98, we take second derivative wrt. θ, and state in terms of Fisher information:

$$I_o(\hat{\theta}|x) = I_{oc} - I_{om} \tag{1.102}$$

  where I_o is the Fisher information of the observed data:

$$I_o(\theta|x) = -\frac{\partial^2 \log p(x|\theta)}{\partial \theta^2} \tag{1.103}$$

  and I_{oc} is the Fisher information of the complete data (averaging over $y|x, \theta$):

$$I_{oc} = \mathrm{E}_{y|x,\theta}[I_o(\theta|x, y)]|_{\theta=\hat{\theta}} \tag{1.104}$$

  and I_{om} is the information from the missing data:

$$I_{om} = \mathrm{E}_{y|x,\theta} \left[-\frac{\partial^2 \log p(y|x, \theta)}{\partial \theta^2} \right] |_{\theta=\hat{\theta}} \tag{1.105}$$

- Finding the covariance matrix of θ (SEM algorithm): [Using EM to obtain asymptotic variance-covariance matrices: The SEM algorithm] we often need to determine the covariance matrix at $\hat{\theta}$, e.g. for the confidence interval of θ. The covariance matrix at $\hat{\theta}$ is the inverse of the I_o matrix. See the paper for the details of the SEM algorithm.

1.5 Hypothesis Testing

Neyman-Pearson paradigm: the goal is to choose a decision rule for accepting or rejecting a hypothesis.

- Intuition of hypothesis testing: to distinguish two hypothesis H_0 and H_A, suppose we choose some informative statistic/pattern T to be test statistic if it is expected to be different under H_0 or H_A. The two distributions $T|H_0$ and $T|H_A$ should be different, and the more different they are, the easier to distinguish the two hypothesis (Figure: hypothesis-testing.gif).
- Assessing a decision rule: Suppose the rule takes the form: H_0 is rejected iff $T \in C$, where T is the test statistic and C is the critical region. Then the rule can be assessed by the two types of errors:
  - Type I error: $\alpha = P(T \in C|H_0)$.
  - Type II error: $\beta = P(T \notin C|H_A)$.

  Alternatively, a test can be assess by these two measures:
  - Significance level: probability of incorrectly rejecting the null hypothesis (α).
  - Power: probability of correctly rejecting the null hypothesis ($1 - \beta$).
- Remark: for a given pair of hypothesis, to choose one of them is similar to assign the distribution of which a data point comes from under a mixture model. While this task is generally performed by Bayesian or likelihood ratio test, the hypothesis testing approach is different.
- Procedure: suppose we choose T as the test statistic, and we need to determine the critical region C. Typically, we choose C s.t. Type I error is bounded by a pre-specified level α:

$$P(T \in C|H_0) \leq \alpha \tag{1.106}$$

  To design a test: choose among all tests that meet certain significance level (e.g. $\alpha < 0.05$), the one that maximizes the power.
- Power of a test: generally depends on the significance level, the sample size, and the alternative hypothesis. When the parameter of the alternative hypothesis is not specified (θ), then we define power function as $\text{Pw}(\theta) = 1 - \beta(\theta)$.
  - Most powerful test: of a specified H_A, if no other test of the same sample size has greater power.
  - Uniformly most powerful test: of a class of alternative hypothesis, if it is the most powerful test for each specified alternative in this class.
- Comparison of tests: generally compare power at a specified significance level. Note that the relative power may depend on alternative hypothesis. Also, robustness to violations of parametric assumptions is another important consideration in practice (see Nonparameteric tests).
- Relation to classification: the metrics of classification performance measure:
  - Type I error or false positive rate (FPR): specificity or true negative rate ($1 - \alpha$).
  - Type II error or false negative rate (FNR): sensitivity, recall or true positive rate, TPR ($1 - \beta$).
  - The fraction of correct predictions among all that are predicted in the positive class: precision or false dicovery rate: $= P(H_A|H_A \text{ is accepted})$
- Trade-off between specificity-sensitivity or type I/II errors: a more accurate method (high specificity) may not generalize well (low sensitivity). A common situation is the preference of simpler models, which tend to be less specific.

Limitations of Neyman-Pearson paradigm:

- Limitations of the number of hypothesis: choose among a large set of models, e.g. structural learning in Bayesian networks.
- Decision theory: a more general framework for hypothesis testing and model selection is probably decision theory. Ex. for choosing among multiple models, we could define the cost function that may take into account the similarity of different models (thus even if we choose a wrong model, but if it is close to the true model, our penalty would be smaller).

p value:

- Motivation: it is often not sufficient to just have a decision rule. One may need to quantify the confidence of a decision, e.g. if T is far from the critical region, then one may know that it is safe to accept H_0 (as opposed to the case where T lies in the boundary).
- Idea: since in general, the value of T is not comparable across different tests, we want to map T to some function $\phi(T)$ s.t. $\phi(T)$ is comparable, and then effectively we could have a simple decision rule for all tests, e.g. $\phi(T) < 0.05$.
- Percentile transformation: this is similar to the situation where we want to say whether a number randomly sampled from a distribution is large or not. Clearly, it has to be normalized against the underlying distribution. This can be achieved by transform any value x to its percentile or CDF $CDF(x)$. It is easy to show that the new R.V., $CDF(X)$ follows uniform distribution (if not, some ranges must have more probability mass than expected, violating our definition of percentile at the first place).
- p value: we apply the percentile transformation to test statistic T, and this gives the p value of an observed statistic (thus p value can be simply understood as the normalized test statistic). Assume we reject H_0 when T is large, then p value of the observed statstic $t$is given by:

$$p = P(T > t|H_0) \tag{1.107}$$

  Alternatively, we can understand p value as: if we choose the observed statistic as the decision boundary, what is the significance level.
- Uniform distribution of P-value: the intuition is that percentile score should be unbiased, e.g. we would only see 1% of samples with p value 0.01. Proof: we want to show that $P(p < u) = u$. Suppose x_u is the value of X s.t. $P(X > x_u) = u$, then whenever $x > x_u$, we have $p < u$ and vice versa, thus $P(p < u) = P(x > x_u) = u$.
- Remark: p-value is simpler than the Neymen-Pearson framework of testing, because it does not require one to specify the alternative distribution, and it has the advantage providing confidence of the results, instead of a yes/no answer. Neyman-Pearson however has the advantage of computing the power of the test.

Power analysis:

- Assessing a test: suppose we have test statistic T for a hypothesis, to evaluate its performance, we compute the power of the test at a certain significance level α. The analysis consists of two steps:
  - Determine the threshold of T according to α: from the distribution of T under H_0, choose the threshold t (suppose $T < t$ will reject H_0) s.t. $P(T < t|H_0) \leq \alpha$.
  - Power calculation at the threshold: suppose t is the chosen threshold, the power of the test is: $P(T < t|H_A)$. The crucial step is thus to calculate the distribution of T under H_A.
- Example: binomial distribution. Suppose we want to test the parameter p of a distribution, Bernoulli(p): $H_0 : p = p_0$ and $H_A : p = p_1$ (suppose $p_1 < p_0$). The goal here is here to choose sample size n s.t. the test reaches power $1 - \beta$, at significant level α. We choose the test statistic $\hat{p}$ as the fraction of successes in the sample.

– Threshold determination: the expectation and variance of the statistic under H_0:

$$E(\hat{p}) = p_0 \qquad \text{Var}(\hat{p}) = \frac{p_0(1-p_0)}{n} \tag{1.108}$$

Thus the distribution of $\hat{p}$ can be approximated by a normal distribution $N(p_0, p_0(1-p_0)/n)$, and at the significance level α, we have the test:

$$\hat{p} < p_0 - z_\alpha \sqrt{\frac{p_0(1-p_0)}{n}} \tag{1.109}$$

– Power calculation: under H_A, the distribution of $\hat{p}$ can be similarly derived: $\hat{p} \sim N(p_1, p_1(1-p_1)/n)$. To have the power above a threshold $1-\beta$ (or type II error below β), we should have:

$$P(\hat{p} > p_0 - z_\alpha \sqrt{\frac{p_0(1-p_0)}{n}} | H_A) \leq \beta \tag{1.110}$$

This is equivalent to:

$$p_0 - z_\alpha \sqrt{\frac{p_0(1-p_0)}{n}} - p_1 \geq z_\beta \sqrt{\frac{p_1(1-p_1)}{n}} \tag{1.111}$$

And the sample size should satisfy:

$$n \geq \left[\frac{z_\alpha \sqrt{p_0(1-p_0)} + z_\beta \sqrt{p_1(1-p_1)}}{p_0 - p_1} \right]^2 \tag{1.112}$$

Why p value is not a good measure?

- Multiple testing issue: under this situation, p value does not tell the false positive rate. Thus need correction or Bayesian methods.
- Power is not considered: suppose we are doing hypothesis testing for two tasks, if the power of two tests are different, then the p values under two tests are not really comparable (thus fails to reflect our confidence of accepting the alternative hypothesis). Ex. testing mean of normal distribution: $H_0 : \mu = 0$, vs two alternative hypothesis, A) $H_1 : \mu = 0.1$; B) $H_1 : \mu = 3.0$. Clearly, the test of A) is much harder than that of B). Both A) and B) would use the same Z-test, thus for the same $\bar{\mu} = 3$, even though the p value is the same under A) and B), our confidenence of accepting A) is much smaller than accepting B).
- Why Bayesian approach is a promising solution:
  - First, the multiple testing problem is addressed by introducing prior, the confidence of an hypothesis is evalauted by posterior ratio, which is a product of prior ratio and Bayes factor. The prior ratio could encode the information: the fraction of true signals.
  - Second, the power problem is addressed by Bayes factor: BF $= \frac{P(D|H_1)}{P(D|H_0)}$. In the above example: A) $P(D|H_1)$ is also small, thus BF is small; B) $P(D|H_1)$ is large, thus BF is large. Therefore, for the same p value, we have different BF, reflecting the difference confidence resulting from different plausibility under alternative hypothesis.

Confidence interval [Rice, Mathematical Statistics and Data Analysis, Section 8.5.3]

- Definition: a confidence interval for a population parameter θ is a random interval, calculated from the sample, that contains θ with some specified probability.

- Example: normal distribution $N(\mu, \sigma^2)$, the sample mean and sample variance are:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{1.113}$$

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 \tag{1.114}$$

  The MLE of μ is $\bar{X}$. The confidence interval of μ is based on the fact that $\frac{\sqrt{n}(\bar{X}-\mu)}{S} \sim t_{n-1}$. Let $t_{n-1}(\alpha/2)$ denote the point beyond which the t distribution with $n-1$ degrees of freedom has probability $\alpha/2$ (t distribution is symmetric). Then the confidence interval is given by $\bar{X} \pm St_{n-1}(\alpha/2)/\sqrt{n}$.

- Duality of confidence intervals and hypothesis tests [Rice, 9.4]: A $100(1-\alpha)\%$ confidence interval for θ consists of all those values of θ_0 for which the hypothesis θ equals θ_0 will not be rejected at level α.

- The hypothesis θ equals θ_0 is accepted if θ_0 lies in the confidence interval.

Techniques for finding confidence interval:

- Exact method: if the distribution of the estimator (or related functions) can be found, then it can used for finding the confidence inteval. The idea is to transform the distribution (which usually contains unknown parameters) s.t. the random interval no longer contains unknown parmaters. Ex. confidence interval for μ of the normal distribution (see above).

- Asympotic method for confidence interval of MLE: normal distribution.

- Bootstrapping method: sample from the estimated values of the population parameter and obtain the distribution of the estimator.

1.5.1 Strategies of Developing Tests

How to construct a test?

- General idea: look for patterns in the data that would be different under H_A and H_0. This is expressed as a statistic T, that is informative of the relevant hypothesis. Technically, it would have different distributions under H_A and H_0. Typically, one may have: $\mathrm{E}(T|H_0) = 0$ and $\mathrm{E}(T|H_A) > 0$ (one could use expectation as a surrogate of whether the distributions under two hypothesis are equal). See below for some common ideas.

  - Ex. for linear model, when $\beta = 0$, X and Y are independent - so we test this independence.

- Estimator-based test: to test hypothesis about a parameter θ, suppose W_n is an estimator of θ, then we obtain the sampling distribution of W_n (dependent on the parameter θ). In general, we could then construct a test statistic using this distribution through CLT, Wald test, etc.

  - Often, we will need to determine the variance or standard deviation of the estimator, called standard error. The distinction is that the standard deviation often depends on some unknown parameter, while the standard error needs to be estimated from the data (so we replace the unknown parameter with its estimate).
  - The idea can be extended to functions of parameters: if we have a statitsic that is an estimator of a function of parameter (e.g. $E(T_n)$ is a function $\tau(\theta)$), then T_n has information of θ. Suppose we can obtain the sampling distribution of $T_n|H_0$, then we could form a test of θ. Since the form of function $\tau(\cdot)$ is very general, this is a very broad strategy to test hypothesis.

- Likelihood-basesd test: LRT or score test, this allows one to construct a simpler test statistic, or one could use asymptotic χ^2 distribution. Under some conditions (e.g. testing simple hypothesis), Neyman-Pearson Lemma guaranttes that LRT is optimal.

Patterns in the data that carry information of the parameters being tested:

- Variance and variance partitioning: the hypothesis regarding parameters may be manifested as how big the variance is, and how variance is partitioned. The best example is ANOVA: if the treatment has no effect, then the variance between the group (treatment and no-treatment) would be the same as the variance within the group (up to some constant).
- Covariance based tests: covariances between variables are important patterns encoded in the data. Example: $Y = \beta X$, the larger β, the bigger the covariance $\text{Cov}(X, Y)$.
- Errors or model fit: for prediction problems, the parameter values are related to the predictor error or other measures of model fitting (e.g. log-likelihood). Ex. $Y = \beta X$ with error term $\epsilon \sim N(0, \sigma^2)$, to test σ^2, the smaller σ^2, the smaller the predictor error (i.e. most of Y can be explained by X).
- Matching histogram: to test if θ is a certain value, we could compare the distribution f_θ and the empirical distribution $\hat{f}$, if the two are very close, then we know that θ is a good value. Example, χ^2 test or goodness-of-fit test.
- Combining multiple statistics: a test statistic may be constructed from multiple components, e.g. χ^2 test statistic from the sum of variables for all cells. The random variables to be combined (added, subtracted, etc.) need to be comparable.
- Nonparametric test in terms of ranks, e.g. Mann-Whitney test.
- Remark: the ideas of parameter estimation: MOM, MLE, LS, etc. can all be applied to develop hypothesis testing strategies.

Lessons for constructing tests:

- Consideration of sampling variance: if we use T as the test statistics, in general, lower variance of T (under a hypothesis) would be preferred. Intuitively, if T does not vary much under H_0, then it is easy to tell if any departure of T (from expectation) is significant or not.
  - Example: suppose we have ChIP-seq type of experiment, and we want to test if a peak is statistically significant. If the count (or normalized counts, or some statistic) does not vary much across the genome, then it's easier to determine the significant peaks.
  - Example: linear model, where y is response variable and x the independent variable of interest. Suppoze we have another variable z, even if we are not interested in z per se, we should include z if it correlates with y. The intuition is that conditioned on z, we can remove some of the variance of y, thus our test statistic of x will have lower variance (which depends on the variance of y).

  The general idea is that if we can explain away some variance/noise in the data, then we should always do that.
- Borrowing information if possible: the idea is that if we additional information of the same parameter of interest, we could borrow information. This is the basic idea of hierarchical Bayes.
  - Example: we are interested if a SNP is associated with a trait y_1. Now it is reasonable to speculate that if it is associated, then it is likely associated with another trait y_2 as well. So we could test the association simultaneously.

Preference of simple models:

- Occam's Razor: It is harder to accept complex hypothesis: under hypothesis testing, if H_A is complex, then a higher LRT statistic would be needed to reject H_0 (it would be easier to get a good LRT by chance, thus LRT should be discounted). This is reflected by higher d.f. of the test under complex H_A. When the actual H_A is simpler than the specified H_A: the model fitting statistics would be hard to reach the LRT threshold (because the actual model generating data is not that complex), resulting a lower power.

- Example: Fisher's method vs. combined likelihood ratio test. Suppose we have two independent datasets D_1 and D_2 testing the same null hypothesis. The true model of D_1 and D_2 are related by a single parameter θ. Suppose with single dataset, we use χ^2 test and get p-value $p_1 = 0.01$ and $p_2 = 0.01$. We consider two tests:

  - Fisher's method: combining p_1 and p_2, and use the χ^2 distribution with dof. 4, we have $p = 10^{-3}$.
  - Combined LRT: the χ^2 statistic for each dataset is 6.6. Since the same θ should maximize (approximately) the likelihood in both datasets, the combined LRT is $6.6 + 6.6 = 13.2$, giving p-value $2 \cdot 10^{-4}$ under χ^2 with d.o.f. 1.

  In this example, Fisher's method ignores the common θ in both datasets; effectively, the alternative model under Fisher's method can fit θ independently in D_1 and D_2, increasing the model complexity.

Dealing with nuisance parameters:

- Standarization and pivotal quantity: it is desirable to have T whose distribution does not depend on nuisance paramaters, e.g. $T|H_0$ follows some standard distribution, such as $N(0,1)$ or χ^2 with certain dof. Intuitively, we want T whose distribution (our decision rule) is not influenced by things other than what we are testing.

  - Identifying factors that may affect the proposed statistic (T): if there are additional factors (beyond the one being tested) that may influence the distribution of T, then it is important to control for them. These factors may be: the degree of freedom, sample size.
  - Ex. for univariate linear regression, the parameter β_1 is related to the fraction of total variance explained by the regression, i.e. $R^2 = SSR/SST$. However, R^2 does not control for sample size, so cannot be directly used for testing β_1.
  - Ex. for ANOVA, need to standardize the bewteen-group variation using within-group variation.

- Replacing nuisance parameters with their estimators: develop the test statistic T that is independent of the nusiance parameter(s) τ. E.g. if T is a test statistic of θ, whose distribution involves τ, we could develop a new test statistic T' by replacing τ with its estimator $\hat{\tau}$ s.t. the distribution of T' is no longer dependent on τ.

- Using partial likelihood or derived statistic from data: suppose the likeihood is $f(D|\theta, \eta)$ where η is the nuisance parameter. Our idea is to derive some new RV T from data, that does not depend on η; in other words, the likelihood in terms of T, $g(T|\theta)$ depends only on θ.

  - Two-sample Poisson test: event 1 follows Poisson distribution of rate λ and event 2 λR, and we are interested in if $R > 1$. Our statistic is: given that event 1 or 2 occurs, how often event 2 occurs, or
  $$P(x_2|x_1 + x_2 = n) = \text{Binom}\left(n, \frac{R}{R+1}\right) \tag{1.115}$$
  - Example: Parent Assymetry Test (PAT) in testing imprinting and matenral effect, the event $M > P$ (mother has more minor alleles than father) conditioned on the mating type follows binomial distribution whose parameter does not depend on the genotype frequency of parents.

- This represents a genearl case where the nuisance parameter is a scale parameter (the base time of Poisson distribution). In this case, one can use conditional distribution of some test statistic - this means one consider fractions, effectively canceling out the scale parameter (similar to Z-score, which normalize a test statistic, e.g. difference of mean).
- More generally, when some test statistic Z is a sufficient statistic of the nuisance parameter η, then the conditional distribution of data given Z is independent of η.

Composite hypothesis and nuisance parameters:

- Problem: the key step of constructing a test is the distribution of T under H_0. However, H_0 may not completely identify a parameter (for composite hypothesis), or T distribution depends on additional (unknown) parameters.
- Obtaining T distribution under given parameters: under Neyman-Pearson paradigm, we need to control type I error, i.e. find t s.t. (assuming $T \geq t$ rejects H_0):

$$\sup_{\theta \in \Theta_0} P(T \geq t|\theta) \leq \alpha \tag{1.116}$$

  Thus as long as we know $T|\theta$ distribution, we could construct a test.

- Problems with maximization: the technical problem with a test that depends on nuisance parameters is that the maximization in Equation 1.116 may not well-behave. Ex. to test mean in the distribution $N(\mu, \sigma^2)$ where σ^2 is unknown, if our test statistic is simply $T = \bar{X}$, without normalization, maximization in Equation 1.116 is unbounded or equal to 1.

Obtaining null distribution by sampling: for complex test statistic, its null distribution may be obtained by simulation. We sample data under the null hypothesis $T|H_0$.

- Parametric bootstrap: normally, the model has other parameters than the one(s) specified by H_0, so we need to estimate those parameters (e.g. MLE), and then sample data from H_0 and the fitted nuisance parameters.
- Permutation test: for hypothesis involving relationship between variables, permutating the data s.t. the sample can be considered to be generated from H_0 (no relation).

Design considerations: to avoid the cases where departure from H_0 is caused by some reasons other than H_A.

- Comparablility: if some variables used in T are not comparable, the test may be biased (size bias is one common problem). Ex. find pathways that are differentially expressed in samples by the number of DE genes: the size of pathways are different. In statistical terms, it means the relevant quantities should have the same distributions.
- Avoid information loss: this often occurs, for instance, when some form of cut-off is considered. Ex. test enrichment of gene groups: if only the most significant genes are tested, the information in the marginally signicant genes is lost.
- Correlation/dependency in the data: this may make tests invalid. Ex. to find genes DE across many samples: if samples are correlated (e.g. some samples are from the same patients), then the test statistic may be inflated.
- Other considerations: outliers, heterogeneity, etc. Any other implicit assumptions that may be violated.

Strategies of dealing with possible biases:

- Modifying test statistic: e.g. to make some variables comparable, use P value, or normalization/calibration (the general idea is to use relative values as test statistics); to maximize use of information, use some weighting scheme s.t. weak evidence can still be utilized.
- Probabilistic modeling: a full model with ML or Bayesian methods can avoid many possible biases.
- Null distribution: the null distribution should take the possible biases/dependence/etc. into account.
- Examples: (1) gene set enrichment analysis, pathway association with diseases - the pathway sizes are different [Tian & Park, PNAS, 2005]. (2) Gene association with diseases in GWAS studies: gene size, the SNP density and LD pattern in the gene region, etc. are different, thus need normalization.

A general form of hypothesis testing is comparison: H_0: foreground (FG) = background (BG). Important considerations:

- The positive and negative (or FG and BG) sets should be identical in all aspects except the one that is being investigated. The test statistic should reflect the difference of FG and BG and the null distribution in general should be sampled from the BG distribution
- Dealing with confounding variables: the general strategy is stratification of the confounding variables, which could be implemented in a regression framework. Or seeking better controls that match the test objects in the confounding variables (which can be used to obtain the null distribution of test statistic).

Some exampels:

- Comparison of sequence groups: the two groups may have some systematic difference, e.g. GC content, or level of conservation. For instance, to find motifs enriched in the positive group, if the positive group is AT rich, then any AT-rich motif may be found to be enriched (not specific).
- Finding differentially expressed genes in two conditions: the expression profiles of the two conditions must be comparable, otherwise, many genes will be differentially expressed. Solution: use a reference set for each condition.
- Association mapping: if the case and control groups have different population structure, then may give false SNPs.

1.5.2 Common Statistical Tests

Bernoulli distribution:

- Suppose $X_i, 1 \leq i \leq n$ iid from Bernoulli(p), we want to test the hypothesis: $H_0 : p = p_0$, vs. some alternative, $H_1 : p \neq p_0$, or $H_1 : p = p_1$.
- Estimator-derived test: we have the estimator of p as $\hat{p} = X/n$. The sampling distribution of $\hat{p}$ is given by CLT, as $n \to \infty$:
$$\sqrt{n}(\hat{p} - p) \to N(0, p(1-p)) \tag{1.117}$$
Under $H_0 : p = p_0$, thus we have the test statistic:
$$Z = \frac{X - np_0}{\sqrt{np_0(1-p_0)}} \tag{1.118}$$
which follows a standard normal distribution.
- LRT: suppose we are testing p_0 vs. p_1. The LRT statistic would have the form:
$$-2(\log L(p_0|X) - \log L(p_1|X)) = -2X \log \frac{p_0}{p_1} - 2(n - X) \log \frac{1-p_0}{1-p_1} \tag{1.119}$$
where X follows Bin(n, p) distribution. The distribution of the LRT statistic is easily derived as a linear function of binomial random variable.

- Pattern-based test: let $X = \sum_i X_i$, the number of 1's in the data, then the extreme value of X would reject H_0. We have the sampling distribution $X|H_0 \sim \text{Binom}(n, p_0)$, and this allows one to compute p-value for any observed X.

Poisson distribution:

- Two sample test: comparing the rates of two samples [Krishnamoorthy & Thomson, A more powerful test for comparing two Poisson means]. Suppose we have x_1 events in interval t_1, and x_2 events in t_2, we are testing if the two rates are equal: $\lambda_1 = \lambda_2$. The idea is that given a total of $x_1 + x_2$ events, the expected number of events in t_1 follows Binomial distribution:

$$X_1 \sim \text{Binom}(x_1 + x_2, p) \tag{1.120}$$

  where $p = (\lambda_1 t_1)/(\lambda_1 t_1 + \lambda_2 t_2)$. Under H_0, we have $p = t_1/(t_1 + t_2)$. This test can be implemented in R using `binom.test()` or `poisson.test()` (two sample version) and the results are equivalent.

Normal distribution:

- Test μ with known σ^2:
  - Estimator-based test: the estimator of μ is $\bar{X}$, and its distribution:

$$\frac{\sqrt{n}(\bar{X} - \mu)}{\sigma} \sim N(0, 1) \tag{1.121}$$

    When σ is known, we could have test statistic $T = \sqrt{n}(\bar{X} - \mu_0)/\sigma$ for $H_0 : \mu = \mu_0$.
  - LRT: suppose we are testing $H_0 : \mu = \mu_0$ vs. $H_1 : \mu \neq \mu_0$. The LRT statistic is reduced to:

$$\sum_i \left[(x_i - \mu)^2 - (x_i - \bar{x})^2\right] = n(\bar{x} - \mu)^2 \tag{1.122}$$

    The test statistic is thus similar to the estimator based test.
- Test σ^2 with known μ: the estimator of σ^2 is the sample variance, S_n^2. We know that the sampling distribution of S_n^2/σ^2 follows χ^2 distribution of dof. n.
- Test μ with unknown σ^2:
  - Estimator-based test: similar to the case above where σ^2 is known, the difference being that we replace σ with its estimator S_n.
  - LRT: see [Casella, Example 8.2.2].

Simple linear regression:

- Problem: a simple univariate regression $Y = \beta_1 X + \beta_0 + \epsilon$, and we want to test $H_0 : \beta_1 = 0$.
- Estimator-based test: the estimator of β_1 is $b_1 = \hat{\text{Cov}}(X, Y)/\hat{\text{Var}}(X)$. The sampling distribution of b_1 is normal, with mean β_1 and variance $\sigma^2/\sum_i (x_i - \bar{x})^2$. We can thus form a test of β_1 by b_1 divided by its standard deviation (t-test, replacing σ^2 by its estimator, MSE).
- LRT: when $\beta_1 = 0$, we fit the data of Y using the mean of Y; when β_1 is free, we fit the data of Y using X and thus have larger likelihood or smaller squared error. We could form the LRT (equivalent to F-test using RSS).

- Pattern-based test/variance partitioning: the variance of Y is partitioned by: those explained by X, and those not (within group variance). If $b_1 = 0$, the variance explained by X should be 0. We thus look at MSR of the data:
$$MSR = \hat{\beta}_1^2 \sum_i (x_i - \bar{x})^2 \tag{1.123}$$
MSR is the estimator of the function of parameter:
$$\mathrm{E}(MSR) = \sigma^2 + \beta_1^2 \sum_i (x_i - \bar{x})^2 \tag{1.124}$$
thus clearly carrying information of β_1. We already know $\hat{\beta}_1$ follows normal distribution, thus it can be shown that:
$$\frac{MSR}{\sigma^2} = \frac{\sum_i (x_i - \bar{x})^2 \hat{\beta}_1{}^2}{\sigma^2} \sim \chi_1^2 \tag{1.125}$$
With σ^2 unknown, we replace it by its estimator MSE, thus we have the F-test MSR/MSE.

Two sample t-test:

- Suppose we have two samples: $X_i, 1 \leq im$, and $Y_j, 1 \leq j \leq n$, $X_i \sim N(\mu_1, \sigma_1^2)$, $Y_j \sim N(\mu_2, \sigma_2^2)$. We want to test the hypothesis: $H_0 : \mu_1 = \mu_2$ vs. $H_A : \mu_1 \neq \mu_2$.
- Test: the idea is the difference of the means is represented by $\hat{\delta} = \bar{X}_m - \bar{Y}_n$. This needs to be normalized by the standard error of $\hat{\delta}$. If σ_1 and σ_2 are known, we have:
$$\mathrm{se}(\hat{\delta}) = \sqrt{\frac{\sigma_1^2}{m} + \frac{\sigma_2^2}{n}} \tag{1.126}$$
Since they are not known, we use the sample variance s_1^2 and s_2^2 instead. Thus we have the test statistic:
$$T = \frac{\bar{X}_m - \bar{Y}_n}{\sqrt{\frac{s_1^2}{m} + \frac{s_2^2}{n}}} \tag{1.127}$$

Pearson's χ^2 test:

- The general idea is to compare an observed distribution/counts with the expected distribution counts. The strategy is to discretize, obtain counts in each category, then compare the expected vs. observed counts.
- Given data from a multinomial distribution $p_1, \cdots, p_K$, we want to test $H_0 : p_i = p_{i0}, 1 \leq i \leq K$ vs. $H_A : \exists i, p_i \neq p_{i0}$.
- The idea is: for each cell i, the difference between observed and expected, $O_i - E_i$, is a indicator of the departure of H_0. Since different cells have different E_i's, this statistic needs to be normalized s.t. they are comparable across cells. To estimate the variance, we assume that the cell count follows a binomial distribution, $\mathrm{Bin}(N, p_i)$, and the variance is: $Np_i(1 - p_i) = E_i(1 - p_i)$. But since the cell counts are not independent, we obtain the variance as E_i (to be proved). The test statistic is:
$$X^2 = \sum_{i=1}^{K} \frac{(O_i - E_i)^2}{E_i} \tag{1.128}$$

Mann-Whitney test:

- Problem: compare the mean of two groups, where no parameteric distribution can be assumed.

- The idea is if the means are equal, then if we rank all numbers from two groups, the ranks should appear random. Specifically, consider the total rank of one group, and the total rank of the other, and the difference between the two total ranks is a good indicator of how different the means are.

Fisher's method of combining tests:

- Method: suppose we are testing the same hypothesis using different independent datasets. Let p_i be the p-value of the test using the i-th dataset, then we have:

$$T = -2\sum_{i=1}^{k} \log p_i \tag{1.129}$$

  It's easy to show that T follows the exact χ^2 distribution with dof. $2k$: negative log of uniform distribution is exponential, and its sum χ^2.

- Remark: limitations
  - Fisher's method does not assume any relationship between the tests, if actually the datasets share parameters, Fisher's method loses power.
  - The sample sizes of the multiple datasets are not taken into account. If the sample sizes are very unbalanced, the p-value from low-powered tests may actually hurt the performance.

Testing non-nested models by J-test [Google search: "A Specification Test for Non-Nested Regression Models"]:

- Problem: suppose we want to test two models:

$$y = f(\beta_0; X) + u_0 \tag{1.130}$$

$$y = g(\beta_1; Z) + u_1 \tag{1.131}$$

- J-test: the idea is to estimate the comprehensive model:

$$y = (1-\lambda)f(\beta_0; X) + \lambda g(\beta_1; Z) + u \tag{1.132}$$

  When no a priori information is available, the mixing parameter is not identifiable in the comprehensive model. The J-test works around this by replacing $g(\cdot)$ with the fitted values from a regression of y on Z and testing the mixing parameter, λ for statistical significance.

- Remark: testing non-nested models is an active area of research, especially in econometrics. Probably no established, widely-used method available.

1.6 Large Sample Theory

Reference: [Casella, Statistical Inference, Chapter 10]

Convergence in distribution of combination of random variables:

- Motivation: suppose we have multiple random variables with known convergence properties (in probability or in distribution), how do we know the sum/product/etc. of these random variables?

- Slutsky's Theorem: if $X_n \to X$ in distribution and $Y_n \to a$ in probability, then $Y_n X_n \to aX$ in distribution; and $X_n + Y_n \to X + a$ in distribution.

- Example: suppose

$$\frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma} \to N(0,1) \tag{1.133}$$

but σ is unknown. Suppose we have $S_n^2 \to \sigma^2$ in probability, then $\sigma/S_n \to 1$ in probability, we have:

$$\frac{\sqrt{n}(\bar{X}_n - \mu)}{S_n} = \frac{\sigma}{S_n}\frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma} \to N(0,1) \tag{1.134}$$

This result is very useful for normal approximation when the variance is unknown, but an estimator of variance is available.

Delta Method: convergence in distribution of functions of random variables:

- Motivation: similar to before, suppose we know the convergence properties of some random variable (e.g. sample mean from CLT), how do we know the functions of these RVs?
- Theorem: suppose Y_n satisfies $\sqrt{n}(Y_n - \theta) \to N(0, \sigma^2)$ in distribution, for a given function g and a specific value of θ, suppose $g'(\theta)$ exists and is not equal to 0. Then

$$\sqrt{n}[g(Y_n) - g(\theta)] \to N(0, \sigma^2[g'(\theta)]^2) \text{ in distribution} \tag{1.135}$$

Proof: Taylor expansion of $g(Y_n)$ around $Y_n = \theta$:

$$g(Y_n) = g(\theta) + g'(\theta)(Y_n - \theta) + \text{Remainder} \tag{1.136}$$

The remainder term $\to 0$ in probabiliyty. Applying Slutsky's Theorem to

$$\sqrt{n}[g(Y_n) - g(\theta)] = g'(\theta)\sqrt{n}(Y_n - \theta) \tag{1.137}$$

the result follows.

1.6.1 Asymptotic theory of point estimation

Reference: [Casella, Statistical Inference, Chapter 10]

Tools for studying asymptotic behavior of estimators or statistics:

- Sample moments: the WLLN and CLT gives the asymptotic behavior of sample mean, and similar results exist for other sample moments. If an estimator or statistic (including likelihood function) can be expressed as a function of sample moments, then its behavior may be obtained.
- Convergence of empirical distribution: at large sample size, the histogram of data points converges to the true distribution. Consider the case of log-likelihood function, $l(\theta, x)$, where θ_0 is the true parameter. As $n \to \infty$: the data points $\{x_i\} \to f(x|\theta_0)$, aka. the true distribution. This property can be combined with the fact that $\hat{\theta}_n$ maximizes likelihood to understand the asymptotic behavior. The same can be said for e.g. derivatives of $l(\theta, x)$.
- Approximating log-likelihood function and its derivatives: around the true value θ_0 (since we are interested in the convergence to some constants dependent on θ_0). This allows us to express log-likelihood function and its derivatives in terms of θ_0, entropy at θ_0 and Fisher information at θ_0, etc.

Consistency and efficiency of estimators:

- Consistency: if the estimator W_n converges in probability to the true value θ_0 as $n \to \infty$, then W is a consistent estimator.

- Theorem: if W_n is a sequence of estimators of θ satisfying $\mathrm{E}_\theta(W_n) - \theta \to 0$ and $\mathrm{Var}_\theta(W_n) \to 0$ as $n \to \infty$ for all values of θ, then W_n is a consistent estimator of θ.
  Proof: follows from the property of convergence in probability (using Chebshev's inequality).

- Motivation for efficiency: consistency does not say anything about the variance of the estimator. Ex. different estimators may all be consistent estimators of θ, but their rate of convergence (or variance) is different.

  - Remark: the concept of efficiency is similar to the "rate of convergence" defined in the context of numerical analysis. Ex. for two sequences:

    $$\lim_{n\to\infty} \frac{1}{n} = 0 \qquad \lim_{n\to\infty} \frac{1}{n^2} = 0 \tag{1.138}$$

    Apparently, both converge to 0, but the rate is different. So, we use the value of k s.t. $n^k a_n$ converges to a positive number to measure the rate of convergence.

  Our idea is thus to study the rate of convergence of the variance of the estimator $\mathrm{Var}(W_n)$ (it converges 0 if $W_n \to \theta$ in probability). For a good estimator, its variance should approach the Cramer-Rao lower bound.

- Limiting variance and asymptotic variance: for technical reasons (see Example 10.1.8), we do not directly study the rate of convergence of the sequence $\mathrm{Var}(W_n)$. Instead, we define: for estimator T_n, if $k_n(T_n - \theta) \to N(0, \sigma^2)$ in disribution, where k_n is a sequence of constants, then σ^2 is called the asymptotic variance of T_n.

- Efficient estimator: estimator W_n is asymptotically efficient for a parameter $\tau(\theta)$, if

  $$\sqrt{n}\,[W_n - \tau(\theta)] \to N(0, v(\theta)) \text{ in distribution} \tag{1.139}$$

  where $v(\theta)$ is the Cramer-Rao Lower Bound:

  $$v(\theta) = \frac{[\tau'(\theta)]^2}{I(\theta)} \tag{1.140}$$

Convergence of log-likelihood function and its derivatives:

- Motivation: we often need to study the convergence of log-likelihood function and its derivatives (these are all sample statistics). These are related to the entropy, score and Fisher information of the distribution.

- Log-likelihood function: we evaluate the log-likelihood function at θ:

  $$\frac{1}{n}\log L(\theta|X) = \frac{1}{n}\sum_i \log f(x_i|\theta) \to \mathrm{E}\left[\log f(X|\theta)\right] \tag{1.141}$$

  where the expectation is taken over the true distribution. We approximate this at $\hat{\theta}$:

  $$\frac{1}{n}\log L(\hat{\theta}|X) \approx H(\theta) \tag{1.142}$$

  Thus the average log-likelihood function at MLE approximates the entropy of the distribution.

- Derivatives of log-likelihood function: similarly, we have

  $$\frac{1}{n}\frac{\partial}{\partial\theta}\log L(\theta|X) = \frac{1}{n}\sum_i \frac{\partial}{\partial\theta}\log f(x_i|\theta) \to \mathrm{E}\left[\frac{\partial}{\partial\theta}\log f(X|\theta)\right] \tag{1.143}$$

where the expectation is taken over the true distribution. Approximation at $\hat{\theta}$:

$$\frac{1}{n}\frac{\partial}{\partial\theta}\log L(\hat{\theta}|X) \approx 0 \tag{1.144}$$

Similary for the second derivatitive:

$$\frac{1}{n}\frac{\partial^2}{\partial^2\theta}\log L(\theta|X) = \frac{1}{n}\sum_i \frac{\partial^2}{\partial^2\theta}\log f(x_i|\theta) \to \mathrm{E}\left[\frac{\partial^2}{\partial^2\theta}\log f(X|\theta)\right] \tag{1.145}$$

Approximation at $\hat{\theta}$, we have the observed information as an estimator of Fisher information:

$$\hat{I}_n(\hat{\theta}) = -\frac{\partial^2}{\partial\theta^2}\log L(\hat{\theta}|X) \approx I_n(\theta) \tag{1.146}$$

- Remark: the log-likelihood and its derivatives are sample means of entropy, average score (0) and Fisher information, so they follow asymptotic normality according to CLT.

Consistency and efficiency of MLE:

- Intuition: the true value θ_0 should generally lead to large likelihood. Speaking in other words, as the sample size gets very large, the MLE and the true value θ_0 should be very close.
- MLE consistency: let $\hat{\theta}_n$ be the MLE of θ, and $\tau(\theta)$ be a continuous function of θ, we have $\tau(\hat{\theta}_n) \to \tau(\theta_0)$ in probability, as $n \to \infty$.
- Proof of MLE consistency: only consider the case $\tau(\theta) = \theta$. We first see that the log-likelihood function:

$$l_n(\theta) = \sum_{i=1}^{n}\log f(x_i|\theta) \tag{1.147}$$

where $f(\cdot)$ is the pdf. As $n \to \infty$, the distribution of x_i approaches the true PDF, $f(x|\theta_0)$, thus we have:

$$\frac{1}{n}l_n(\theta) \to \int f(x|\theta_0)\log f(x|\theta)dx = \mathrm{E}\left[\log f(X|\theta)\right] \tag{1.148}$$

From the KL divergence, we know that the above integral is maximized at $\theta = \theta_0$ (i.e. the log. of the density equal to the true density). Therefore, as $n \to \infty$, θ_0 maximizes $l_n(\theta)$, i.e. $\hat{\theta}_n \to \theta_0$.

- Efficiency of MLE - single parameter: let n be the sample size, $\hat{\theta}_n$ be MLE, and let $\tau(\theta)$ be a continuous function of θ, then under certain regularity conditions, as $n \to \infty$, we have:

$$\sqrt{n}(\tau(\hat{\theta}_n) - \tau(\theta_0)) \to N\left(0, \frac{[\tau'(\theta_0)]^2}{I_1(\theta_0)}\right) \tag{1.149}$$

where θ_0 is the true value of θ. The variance term is the Cramer-Rao lower bound, thus, $\tau(\hat{\theta}_n)$ is asymptotically efficient estimator of $\tau(\theta)$. Note: in the RHS, to have a constant, instead of variance that depends on n, we have $I_1(\theta)$, thus $\sqrt{n}$ term in the LHS.

- Proof idea: we consider only the case where $\tau(\theta) = \theta$. To show $\hat{\theta}_n$ is close to θ_0, the idea is, if for some function, the values at $\hat{\theta}_n$ and θ_0 and very close, then the two must be close by the Taylor expansion. Another idea is that: the convergence of log-likelihood function (and its derivatives) can be approximated by CLT since it can be written as an average. We choose this function to be the derivative of log-likelihood function, taking its expansion near θ_0:

$$l'(\theta|x) = l'(\theta_0|x) + (\theta - \theta_0)l''(\theta|x) \tag{1.150}$$

At $\theta = \hat{\theta}_n$, the LHS is 0, so we have:

$$\sqrt{n}(\hat{\theta}_n - \theta_0) = \frac{-\frac{1}{\sqrt{n}} l'(\theta_0|x)}{\frac{1}{n} l''(\theta_0|x)} \tag{1.151}$$

For the numerator, we apply the CLT:

$$-\frac{1}{\sqrt{n}} l'(\theta_0|x) \to N(0, I_1(\theta_0)) \tag{1.152}$$

where the Fisher information is the variance of the score (first derivative of the log-likelihood) For the denomator, we apply the WLLN:

$$\frac{1}{n} l''(\theta_0|x) \to I_1(\theta_0) \tag{1.153}$$

where the Fisher information is the second derivative of the log-likelihood.

- Efficiency of MLE - multiple parameters: [Wiki] for the case of simple function $\tau(\theta) = \theta$:

$$\sqrt{n}(\hat{\theta} - \theta_0) \to N(0, I^{-1}(\theta_0)) \tag{1.154}$$

where I is the Fisher information matrix (single point, or PDF) evaluated at θ_0 (in practice, replace with $\hat{\theta}$). This theorem can be used to derive the confidence interval of MLE.

Studying asymptotic efficiency/variance of estimators:

- General strategy: for a parameter estimation problem,
  - Consistent estimator: the first step is generally to find a consistent estimator $W_n \to \theta$ in probability. For functions of parameters, we could use the Theorem: $h(W_n) \to h(\theta)$ in probability if h is a continuous function.
  - Standard error: the next question is often the variance (or standard error) of the estimator. The first step:

$$\mathrm{Var}_\theta(W_n) = \sigma_n^2(\theta) \tag{1.155}$$

    Note that the variance is a function of the true value θ. Since θ is unknown, we have the second step, which replaces θ with $\hat{\theta}$:

$$\hat{\mathrm{Var}}_\theta(W_n) = \mathrm{Var}_\theta(W_n)|_{\theta=\hat{\theta}} \tag{1.156}$$

    In some other cases (e.g. Wald test for parameters), we need a consistent estimator of the variance, i.e. $S_n/\sigma_n \to 1$ in probability.
  - Asymptotic normality: for many cases, e.g. for testing parameters, we want to estabilish normality of the estimator. Typically, we have:

$$\frac{W_n - \theta}{\sigma_n} \to N(0, 1) \tag{1.157}$$

    Or we may replace σ_n with S_n (may not be necessary if we are working on the distribution under H_0).
  - Basic tools: for studying asymptotic distribution include CLT and the Delta Method (if the estimator is a function of sample mean), and the asymptotic normality of MLE.
- Approximation of variance of MLE: we apply the two-step procedure for the variance of MLE. First,

$$\mathrm{Var}(\tau(\hat{\theta})) \approx \frac{[\tau'(\theta)]^2}{I_n(\theta)} \tag{1.158}$$

Next, from our discussion of covergence of derivative of log-likelihood function, we know that observed information is a consistent estimator of $I_n(\theta)$.

$$\text{Var}(\tau(\hat{\theta})) \approx \frac{[\tau'(\theta)]^2|_{\theta=\hat{\theta}}}{\hat{I}_n(\hat{\theta})} \tag{1.159}$$

- Example: suppose $X_1, X_2, \cdots X_n$ iid. Bernoulli(p) distribution. The estimator of p is $\hat{p} = X/n$. We have two ways of obtaining its variance. First, direct calculation:

$$\text{Var}_p(\hat{p}) = \frac{p(1-p)}{n} \tag{1.160}$$

We approximate this at $p = \hat{p}$:

$$\hat{\text{Var}}_p(\hat{p}) = \frac{\hat{p}(1-\hat{p})}{n} \tag{1.161}$$

And the distribution is normal. The second way is to apply the MLE approximation above:

$$\hat{I}_n(\hat{p}) = -\frac{\partial^2}{\partial^2 p} \log L(\hat{p}|X) = \frac{n}{\hat{p}(1-\hat{p})} \tag{1.162}$$

From both methods, we have:

$$\sqrt{n}\frac{\hat{p}-p}{\sqrt{\hat{p}(1-\hat{p})}} \to N(0,1) \tag{1.163}$$

1.6.2 Large Sample Tests

Reference: [Casella, Statistical Inference, Chapter 10]

Likelihood ratio test:

- LRT: suppose we are testing $H_0 : \theta \in \Theta_0$ vs. $H_1 : \theta \in \Theta_1$, the test statistic

$$\lambda(x) = \frac{\sup_{\Theta_0} L(\theta|x)}{\sup_{\Theta_1} L(\theta|x)} \tag{1.164}$$

- Theorem (asymptotic distribution of LRT - simple H_0): for test $H_0 : \theta = \theta_0$ vs. $H_1 : \theta \neq \theta_0$, under H_0, as $n \to \infty$:

$$-2\log\lambda(X) \to \chi^2_1 \text{ in distribution} \tag{1.165}$$

Proof: suppose $\hat{\theta}$ is the MLE under H_1, Taylor expansion of $\log L(\theta|X)$ near $\hat{\theta}$:

$$l(\theta|x) = l(\hat{\theta}|x) + l'(\hat{\theta}|x)(\theta-\hat{\theta}) + l''(\hat{\theta}|x)\frac{(\theta-\hat{\theta})^2}{2} + \cdots \tag{1.166}$$

Thus:

$$-2\log\lambda(x) = -2l(\theta_0|x) + 2l(\hat{\theta}|x) \approx -l''(\hat{\theta}|x)(\theta-\hat{\theta})^2 \tag{1.167}$$

Using the Theorem of MLE efficiency, we have that this converges in distribution to χ^2_1.

- Theorem (asymptotic distribution of LRT - composite H_0): for composite H_0, we have that $-2\log\lambda(x)$ converges in distribution to χ^2_k, where k is the difference between the number of free parameters in Θ_0 and Θ_1.

Wald test:

- Test: suppose we have an estimator $W_n \in \theta$, if W_n has asymptotic normal distribution then we can construct a Z score. Suppose S_n is an estimator of σ_n, the standard error of W_n, with $\sigma_n/S_n \to 1$. Then to test $H_0 : \theta = \theta_0$, we define:

$$Z_n = \frac{W_n - \theta_0}{S_n} \tag{1.168}$$

  which converges in distribution to $N(0,1)$ under H_0. When W_n is the MLE, we have:

$$S_n = \frac{1}{\sqrt{\hat{I}_n(W_n)}} \tag{1.169}$$

- Example: suppose $X_1, X_2, \cdots X_n$ iid. Bernoulli(p) distribution. The estimator of p is $\hat{p} = X/n$. From CLT,

$$\frac{\hat{p} - p}{\sigma_n} \to N(0,1) \text{ in distribution} \tag{1.170}$$

  where $\sigma_n = \sqrt{p(1-p)/n}$. An estimate of σ_n is $S_n = \sqrt{\hat{p}(1-p)/n}$, so we have:

$$\sqrt{n}\frac{\hat{p} - p_0}{\sqrt{\hat{p}(1-\hat{p})}} \to N(0,1) \tag{1.171}$$

- Remark: note that under H_0, the value of θ may be specified, so σ_n may be a given function of θ, and in this case, we could use σ_n instead of S_n in the test. For the Bernoulli example, to test if $p = p_0$, this is:

$$\sqrt{n}\frac{\hat{p} - p_0}{\sqrt{p_0(1-p_0)}} \to N(0,1) \tag{1.172}$$

  If H_0 is not simple, e.g. to test if $p \le p_0$, then σ_n is not known, and we would need S_n instead of σ_n in the test.

Score test:

- Intuition: our goal is to test $H_0 : \theta = \theta_0$. If θ_0 is the true value that generates the data, then the score $S(\theta_0) \approx 0$ because $\mathrm{E}(S(\theta_0)) = 0$. So we use $S(\theta_0)$ as our test statistic, whose null distribution has mean 0. Larger values would reject H_0. When H_0 is not true, the data is not generated from θ_0, but we evaluate the score at θ_0, so the score is not necessarily close to 0.
- Relation between score and MLE: suppose our data is generated by θ, then the score at θ, $S(\theta)$ is close to 0 as sample size gets large, because the expectation of $S(\theta)$ is 0. On the other hand, at MLE $\hat{\theta}$, the derivative $\partial l(\theta)/\partial\theta = 0$, thus the MLE must be close to the true value θ.
- Test: the derivative of the log-likelihood function is score:

$$S(\theta) = \frac{\partial}{\partial\theta} \log L(\theta|x) \tag{1.173}$$

  This is a function of RV x. Its mean over x is 0 (see the sectoin on "Fisher information"), and variance is simply the Fisher information $I(\theta)$. To test $H_0 : \theta = \theta_0$, we define:

$$Z_S = \frac{S(\theta_0)}{\sqrt{I(\theta_0)}} \tag{1.174}$$

  Under H_0, Z_S has mean 0 and variance 1, and $Z_S|H_0$ follows $N(0,1)$ (see Convergence on the log-likelihood function and its derivatives). To test composite H_0, we replace θ_0 with the maximum of θ under H_0.

- Alternative (multiple variable) forms of the score test: let U be the score and I be its variance (Fisher information matrix), then $U^2/I \sim \chi_1^2$. In multi-dim. case, we have

$$S = U^T I^{-1} U|_{\theta_0} \sim \chi_k^2 \tag{1.175}$$

where k is the rank of I.

- Binomial score test. To test $H_0 : p = p_0$, we have score and Fisher information:

$$S(p) = \frac{x}{p} - \frac{n-x}{1-p} = \frac{\bar{p} - p}{p(1-p)/n} \qquad I(p) = \frac{n}{p(1-p)} \tag{1.176}$$

The score test statistic is thus:

$$Z_S = \frac{S(p_0)}{\sqrt{I(p_0)}} = \frac{\bar{p} - p_0}{\sqrt{p_0(1-p_0)/n}} \tag{1.177}$$

which is the same as Equation 1.118.

- Pearson's chi-square test as a score test. We are testing if a table follows multinomial distribution with paramater μ_i for the i-th cell (or if rare, we use Poisson distributions). Our data is $\{O_1, \cdots, O_n\}$. The likelihood is then:

$$L(\mu) = \prod_i P(O_i|\mu_i) = \prod_i \frac{\mu_i^{O_i} e^{-\mu_i}}{O_i!} \tag{1.178}$$

The log-likelihood is:

$$l(\mu) = \sum_i (O_i \log \mu_i - \mu_i - \log O_i!) \tag{1.179}$$

This allows to compute the derivative and second derivative and we have:

$$\frac{\partial l}{\partial \mu_i} = \frac{O_i}{\mu_i} - 1 \qquad \frac{\partial l^2}{\partial \mu_i \partial \mu_j} = -\delta_{ij} \frac{O_i}{\mu_i^2} \tag{1.180}$$

The Fisher information matrix is:

$$I = -\mathrm{E}\left[\frac{\partial l^2}{\partial \mu_i \partial \mu_j}\right] = \left[\delta_{ij} \frac{\mathrm{E}(O_i)}{\mu_i^2}\right] = \mathrm{Diag}\left(\frac{1}{\mu_1}, \cdots, \frac{1}{\mu_n}\right) \tag{1.181}$$

And its inverse is $\mathrm{Diag}(\mu_1, \cdots, \mu_n)$. The score test is:

$$S = U^T I^{-1} U = \sum_i \mu_i \left(\frac{O_i}{\mu_i} - 1\right)^2 = \sum_i \frac{(O_i - \mu_i)^2}{\mu_i} \tag{1.182}$$

where U is the score vector. S follows chi-square distribution, as discussed earlier.

- Linear model score test: suppose we have a simple regression model, $y = \beta_0 + x\beta + \epsilon$, and we test $H_0 : \beta = 0$. For simplicity, assume β_0 is known. The log-likelihood function:

$$l(\beta) = -\frac{1}{2\sigma^2} \sum_i (y_i - \beta_0 - x_i\beta)^2 + \text{const} \tag{1.183}$$

The score is then:

$$S(\beta) = \frac{\partial l(\beta)}{\partial \beta} = \sigma \sum_i (y_i - \beta_0 - x_i\beta) x_i \tag{1.184}$$

At $\beta = 0$, we have

$$S(\beta) = \sigma \sum_i (y_i - \beta_0) x_i = \sigma(Y - \beta_0) \cdot X \tag{1.185}$$

So it is proportional to the inner product of Y (if it is standardized) and X. The Fisher information:

$$I(\beta) = -\frac{\partial S(\beta)}{\partial \beta} = \sigma \sum_i x_i^2 \tag{1.186}$$

The test statistic is thus:

$$Z = \left.\frac{S(\beta)}{\sqrt{I(\beta)}}\right|_{\beta=0} = \frac{(y-\beta_0)\cdot x}{\|x\|} \tag{1.187}$$

- Remark: it is also called Langrange Multiplier test (in econometrics literature), because the test involves maximization of likelihood of a restricted model (H_0), which may often be obtained by Langrange Multiplier.

Comparison of LRT, Score test and Wald test: suppose we have $H_0 : \theta = \theta_0$, and $\hat{\theta}$ denotes the MLE of θ under H_1 (more general model).

- In the log-likelihood function surface: LRT is based on the difference in the y-axis, $l(\hat{\theta}) - l(\theta_0)$; Wald test is based on the difference in the x-axis, $\hat{\theta} - \theta_0$; and Score test is based on the derivative $\frac{\partial l(\theta)}{\partial \theta}|_{\theta_0}$. See Figure in: `http://www.ats.ucla.edu/stat/mult_pkg/faq/general/nested_tests.htm`
- LRT needs to evaluate MLE under both H_0 and H_1; Wald test to evaluate MLE under H_1; and Score test to evaluate MLE under H_0. Since H_0 is often simple, Score test may avoid the difficult optimization problem of calculating MLE under H_1.

1.7 Information Theory

Concepts:

- Entropy: for a RV with pdf. $f(x)$, its entropy is defined as the negative of the expected information content:

$$H(X) = -\int f(x) \ln f(x) dx \tag{1.188}$$

- KL divergence. Given two probability distributions P and Q, the "distance" of P from Q is defined by the KL divergence:

$$H(P||Q) = \int \log \frac{P(x)}{Q(x)} P(x) dx \tag{1.189}$$

  So it is the expected log ratio of pdf of P and Q, averaged over P. The KL divergence has the important property: for any P and Q

$$H(P||Q) \geq 0 \tag{1.190}$$

  - Proof: let $Y = \frac{Q(X)}{P(X)}$ be a random variable (function of X). Y represents the difference of density between the two distributions and $\mathrm{E}_P(Y) = 1$. Thus we have $H(P||Q) = -E_P(\log Y)$. Using Jensen's inequlity:

$$\mathrm{E}_P(\log Y) \leq \log \mathrm{E}_P(Y) = 0 \tag{1.191}$$

- Remark: in KL divergence, we can think of Q as true distribution, and P as the empirical distribution (data). Then we should integrate over P, as we consider the log-likelihood over all data.

Maximum entropy method:

- Background: calculus of variation. Example, for a physical system with certian state function $f(x)$ (e.g. pressue), where x represents the spatial coordinate. Suppose the entropy (density) at the point x is related to the state at x by: $\phi(f(x))$, where ϕ is given, then the total entropy of the system is:

$$S[f] = \int \phi(f(x))dx \tag{1.192}$$

  We see that S is a "functional" of the state function f, and to apply the Second Law of Thermodynamics, we need to maximize S wrt. f, typically subject to certain constraint (e.g. mass conservation):

$$\int f(x)dx = C \tag{1.193}$$

- Maximum entropy: given certain constraints of a probability distribution, typically given in the form of moments, the unknown distribution should be the one that maximizes the entropy. Ex. the distribution that maximizes the entropy subject to:

$$\mathrm{E}(X) = \mu \qquad \mathrm{Var}(X) = \sigma^2 \tag{1.194}$$

  is the normal distribution $N(\mu, \sigma^2)$.

- Remark: a generalization of the Method of Moments of parameter estimation. Instead of assuming a parameteric form, we find the distribution (with maximum entropy) that matches the moments of unknown distribution and the empirical distribution (data).

Parameter estimation from information theory perspective:

- Minimizing KL divergence: the idea is that θ should minimize f_θ and the empirical distribution $\hat{f}$. We have:

$$KL(\hat{f}||f_\theta) = \int \hat{f}(x)\log\hat{f}(x)dx - \int \hat{f}(x)\log f_\theta(x)dx \tag{1.195}$$

  Clearly, minimizing KL divergence is equivalent to maximizing the second term, which is the log-likelihood function (divided by n). Thus minimizing KL divergence leads to ML estimator (for parameteric distributions).

- Implications to nonparameteric methods: the above equation applies to all cases, even if we do not have a parameteric form of f. In this case, we could choose f (subject to certain constraints) that has the lowest $KL(\hat{f}||f)$.

- Sample entropy: the term

$$S = \int \hat{f}(x)\log\hat{f}(x)dx \tag{1.196}$$

  is the entropy in the sample. It represents all "uncertain" in the data, and in the perfect case, equal to the maximum log-likelihood funciton.

Hypothesis testing from information theory perspective:

- KL divergence: to test if $\theta = \theta_0$, the idea is that the KL divergence $KL(\hat{f}||f_{\theta_0})$ should be really close to 0. This could be used for a goodness-of-fit test, e.g. the normality of a distribution [A test of normality based on sample entropy].

1.8 Model Selection

Overview of model comparison and selection:

- Purposes: compare multiple models: choose the best one or testing if one is significantly better than another. Often, this is needed to compare models with different complexities or select models with the appropriate complexity.
- Scenarios: we have basically two scenarios:
  - Probability distributions/processes: which distribution/process better explains the data, $\{x_1, \dots x_n\}$.
  - Predictive models: which model better predicts responses y_i from predictors, x_i, where $1 \leq i \leq n$.

  The two different scenarios are equivalent, in that an approach for one case could be adapted to the other. Specifically, assuming some error distribution of y_i from $f(x_i)$, where $f(\cdot)$ is the true function, then the latter problem can be reduced to the former. Similarly, the former problem, is the latter problem when the predictors are empty.
- Goodness-of-fit: another scenarior is to test if a model is good enough to explain the data, without involving comparison of multiple models.

Methods for model selection:

- Hypothesis testing: if models are nested, then reduced to the problem of testing parameter values: Neyman-Pearson paradigm, Likelihood-ratio test (or asympotic results in general), etc. If non-nested, could still use LRT, but the distribution will have to be obtained from e.g. bootstrapping [Goldman, JME, 1993]; or use other tests, e.g. J-test.
- Bayesian model selection: computing the Bayes factor, which involves integrating over all parameter values.
- Analytic methods: BIC as an approximation of Bayes factor, MDL, etc.
- Cross validation: divide the data into training and validation sets, and define a measure of error, typically expected prediction error for predictive models (with certain loss function). But it could be other reasonable measure of errors.

Model assessment problem [Hastie, Section 7.2]:

- Prediction error: our problem is to learn a function $f : X \to Y$. Suppose the function learned from training data is $\hat{f}$, the loss function is denoted as $L(Y, \hat{f}(X))$, then the model is assessed by the test error or generalization error, defined as:

$$\text{Err} = E[L(Y, \hat{f}(X))] \tag{1.197}$$

- Training error: the training error is the average loss over the training samples:

$$\bar{\text{err}} = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}(x_i)) \tag{1.198}$$

  However, the training error is not a good prediction of the test error.

Typical loss functions:

- Squared error loss: $L(Y, f(X)) = (Y - f(X))^2$.
- Zero-one loss: $L(Y, f(X)) = I(Y \neq f(X))$.

- Log likelihood loss: if instead of a function f, we want to estimate the parameter θ of some density function, then $L(Y, \theta(X)) = -2 \log P(Y|\theta(X))$.

Structural risk minimization [Murphy, Section 6.5]

- Let λ be a parameter for penalizing model complexity, we should choose a model, denoted as δ, to minimize:

$$\delta_\lambda = \text{argmin}_\delta \left[R_{emp}(D, \delta) + \lambda C(\delta) \right] \tag{1.199}$$

  where $R_{emp}(D, \delta)$ is the empirical risk (assess the fitting of data by δ), and $C(\delta)$ controls complexity. The issue is: how to choose λ?

- Choosing λ by cross-validation: suppose we partition the data into K fold, and let D_k be the test data of fold k, and D_{-k} be the training data. Under a given λ, the CV estimate of the risk is: fit D_{-k} and let the best function be $f_\lambda^k(\cdot)$. We then apply it to every data point $i \in D_k$, and let the error/loss be $L(y_i, f_\lambda^k(x_i))$. Summing over all data points in D_k gives the risk in the k-th fold. We then sum over all folds, and obtain the average risk for each data point. See Equation (6.60) in Murphy:

$$R(\lambda, D, K) = \frac{1}{N} \sum_{k=1}^{K} \sum_{i \in D_k} L(y_i, f_\lambda^k(x_i)) \tag{1.200}$$

  So λ should be chosen to minimize this function.

- Example: using CV to pick λ for ridge regression. The loss function is Negative log-likelihood (NLL), or equivalently, squared error. The parameter estimation for a given λ is the MAP estimate with normal prior.

- 1 standard error (1SE) rule: the difference of loss may not be significant under different λ, so we choose the largest λ s.t. it is still within 1 SE of the value of λ that minimizes the risk.

- Choosing λ by Empirical Bayes: suppose λ is a hyperparameter of β prior, we choose λ by:

$$\hat{\lambda} = \text{argmax}_\lambda \int P(y|\beta) P(\beta|\lambda) d\beta \tag{1.201}$$

  Note that this involves integration over β, which can be computationally expensive.

Chapter 2

Bayesian Inference

2.1 Bayesian Statistics Background

Bayesian paradigm:

- Setting up the model: this involves two parts, the likelihood function and the prior. The prior distribution should reflect prior belief, or noninformative. An important issue is to ensure the posterior distribution is proper (finite integral) when the prior is improper.
- Posterior inference: the basic equation:

$$P(\theta|y) \propto P(\theta)P(y|\theta) \tag{2.1}$$

  In some simple cases, the posterior distribution can be analytically determined. In most cases, however, we will need to sample θ from $p(\theta|y)$. From the posterior samples, it is easy to obtain posterior summary (mean, median, quantile and posterior interval, etc.) and any quantity/function of parameters of interest.

- Posterior predictive distribution: defined as

$$P(\tilde{y}|y) = \int P(\tilde{y}|\theta)P(\theta|y)d\theta \tag{2.2}$$

  When we already have the posterior sample, $\theta^l, l = 1, \cdots, L$, we could sample $\tilde{y}$ from $p(\tilde{y}|\theta)$ using the sampled values of θ.

- Model checking: one strategy is to simulate replicate data using the posterior predictive distribution, and compare with the observed data to see if there is discrepancy.
  - Example: in the ETS example (Section 5.5), simulate the data of 8 schools, and check the maximum of the simulated data. The observed maximum can be compared with simulation to estimate the probability that the maximum could reach the observed maximum.
- Remarks:
  - Problem of posterior mode: this may not reflect the uncertainty in the inference, and insufficient to capture the inference results. Ex. in the ETS example [Section 5.5] of hierarchical normal model, the posterior mode of τ (the variance of group means) is 0, and if we accept it, all groups have the same mean (complete pooling), and there is no benefit of Bayesian.

Advantages of Bayesian over frequentist statistics (especially on hypothesis testing): [personal notes]

- Reference: [Servin & Stephens, Imputation-Based Analysis of Association Studies: Candidate Regions and Quantitative Traits, PLG, 2007].
- Advantage of Bayes factor over p-values: p-values do not reflect the power of the test, thus a small p-value may not mean much if the test is also very unlikely under the H_1 for a test with very low power. Thus when the problem effectively involves combining multiple tests, p-values from low-powered ones (less informative ones) create noises that hide signals from the most informative ones. Examples:
  - GWAS: test association of multiple SNPs in a region with the trait. If a number of SNPs in the region are not informative (irrelevant or in low LD with the causal variant), p-value of the region loses power.
  - Meta-analysis of multple studies: when the power (sample size, MAF for SNPs, etc.) of the studies are different.
  - Combining multiple evidence of a gene: de novo data, case/control data. The power may be very different.

Lessons for Bayesian modeling in practice [personal notes]:

- Analyzing influence of prior: the use of prior may create bias in the inference problem (overriding data), so the results sometimes may not be desired. This has to be analyzed carefully. Ex. TADA case-control model, if the prior of q is strong, then it is possible to have weird cases, e.g $B(1,2) > 1$. So one should analyze the nature of this bias and see if it is acceptable.
- Check distribution of Bayes factors: when Bayesian statistics is used for decision: model selection/hypothesis testing, it is important to check if the model leads to false decision. If the inference is based on Bayes factor (BF), then it's important to check the distribution of BF under the null model (when we should not make a certain decision).
- Sensitivity analysis: how robust the results are to the parameters, specifically prior parameters. In the hypothesis testing problem, this means analyzing how power and false positive rate depend on parameters.

Issues of Bayesian inference:

- Exchangability: the data $\{y_i, 1 \leq i \leq n\}$ is exchangable, if the joint density $p(y_1, \cdots, y_n)$ is invariant to permutations of the indices. Exchangability reflects our ignore of the difference between data points (other than those reflected in the explanatory variables).
  - Exchangability vs iid: often we model exchangable data as iid. conditioned on unknown parameters. However, the two concepts are not the same, e.g. (X_1, X_2) follows bivarirate normal distribution, thus they are exchangable, but certainly not iid.
- Inference of models with nusiance parameters: suppose we have a model with parameters θ_1 and θ_2, and we are interested in only θ_1. The joint posterior distribution of θ_1, θ_2 is:

$$p(\theta_1, \theta_2|y) \propto p(\theta_1, \theta_2)p(y|\theta_1, \theta_2) \tag{2.3}$$

  There are two ways of obtaining $p(\theta_1|y)$. First, we average the joint posterior over θ_2:

$$p(\theta_1|y) = \int p(\theta_1, \theta_2|y)d\theta_2 \tag{2.4}$$

  Second, suppose the conditional posterior distribution of θ_1 when θ_2 is given is easy to obtain (e.g. for normal distribution with known variance), then we have:

$$p(\theta_1|y) = \int p(\theta_1|\theta_2, y)p(\theta_2|y)d\theta_2 \tag{2.5}$$

Note that this would need to determine $p(\theta_2|y)$. Yet another way of obtaining $p(\theta_1|y)$ through conditional distribution is:

$$p(\theta_1|y) = \frac{p(\theta_1, \theta_2|y)}{p(\theta_2|\theta_1, y)} = \frac{p(\theta_2|y)p(\theta_1|\theta_2, y)}{p(\theta_2|\theta_1, y)} \tag{2.6}$$

Note that the equation is valid for any value of θ_2, thus one may plug-in a special value of θ_2. However when applying this equation, the constant term in the denominator depends on θ_1, thus effectively, we need to determine the normalizing constant of $p(\theta_2|\theta_1, y)$, requiring integration over θ_2.

Prior distributions: [GCSR, Section 2.9]

- Conjugate prior: choose the form of prior s.t. the posterior distribution would have the same form of distributions.
- Proper and improper prior: prior may often be improper, e.g.

$$p(\theta) \propto 1 \tag{2.7}$$

  However, given improper prior, the posterior distribution may be proper. Ex. for normal distribution $N(\mu, \sigma^2)$ where σ^2 is know, under the uniform prior of μ, the posterior is proper.
- Checking posterior by simulation: to check if posterior is proper (when prior is improper), sample the posterior distribution, and test if it is proper (e.g. the probability should be close to 0 when the parameters approach infinity).
- Common noninformative prior distributions: in general, find the right scale (transformation) s.t. the prior is uniform. Examples:
  - $\theta > 0$: uniform at the log. scale, i.e. $\log(\theta) \propto 1$ or $\theta \propto 1/\theta$.
  - $\theta \in [0, 1]$: uniform at the logit scale, i.e. $\text{logit}(\theta) = \log \frac{\theta}{1-\theta} \propto 1$.
  - Location and scale parameters: normally, noninformative prior for location parameter is uniform, and the noninformative prior for the scale parameter is uniform in the log. scale.
- Dealing with improper posterior: generally, if some improper prior makes the posterior improper, change the prior distribution. The idea is to reduce the probability mass at $\theta \to \infty$ s.t. the posterior will be less dispersed.
  - Ex. rat tumor example [Section 5.3]: if $p(\theta) \propto \theta^{-1}$ does not lead to proper posterior, change it to $p(\theta) \propto \theta^{-3/2}$.

Normal approximation of posterior distribution: [GCSR, Chapter 4]

- Normal approximation: the Taylor expansion of $\log p(\theta|y)$ near the posterior mode:

$$\log p(\theta|y) = \log p(\hat{\theta}|y) + \frac{1}{2}(\theta - \hat{\theta})^T \left[\frac{d^2}{d\theta^2} \log p(\theta|y)\right]_{\theta=\hat{\theta}} (\theta - \hat{\theta}) + \cdots \tag{2.8}$$

  Ignoring the higher order term, this is a quadratic function of θ, thus it follows normal distribution:

$$\theta|y \sim N(\hat{\theta}, [I(\hat{\theta})]^{-1}) \tag{2.9}$$

  where $I(\theta)$ is Fisher information:

$$I(\theta) = -\frac{d^2}{d\theta^2} \log p(\theta|y) \tag{2.10}$$

- Convergence to normal distribution at large sample size: [Appendix B]

- Theorem: given a parameter θ, we define the KL divergence between the distribution $p(\cdot|\theta)$ and $f(y)$, defined as:

$$H(\theta) = \int \log\left(\frac{f(y_i)}{p(y_i|\theta)}\right) f(y_i) dy_i \tag{2.11}$$

  Thus $H(\theta)$ measures how good θ is or how close θ is to the true value. Suppose the true parameter is θ_0, i.e. $f(y) = p(y|\theta_0)$, then θ_0 minimze the KL divergence.

- Theorem: under some regularity conditions, as $n \to \infty$, the posterior distribution of θ approaches normal distribution with mean θ_0 and variance $(nJ(\theta_0))^{-1}$, where J is the Fisher information, and θ_0 minimizes the KL divergence.

- Using normal approximation to estimate the posterior interval: suppose we want to find the posterior interval covering 95% of posterior probability mass (in terms of log. density relative to the density at the mode), we note that if X is a RV of d-dimensional MVN, then $\log p(X)$ is a quadratic function of X, and thus

$$Y = -2 \log p(X) \tag{2.12}$$

  is a RV with χ^2_d distribution. The interval of X thus can be found via the interval of Y. Example, $d = 2$, the 95% interval of χ^2_2 is 5.99, this corresponds to the log. density above $\exp(-5.99/2) = 0.05$ relative to the density at the mode. This is useful in posterior sampling to find the region around the mode that covers most probability mass.

Relation of Bayesian posterior distribution and estimator distrubtion under classical statistics [personal notes]:

- Motivation: suppose we are interested in the posterior of $\theta|D$. It often depends on the estimator $\hat{\beta}$. How would this posterior relates to the estimator distribution under classical statistics? Intuitively, if the estimator has large standard error under classical statistics, the posterior of θ would also have large posterior interval.

- Let $\hat{\theta}$ be the estimator of θ. Suppose it is a sufficient statistic, then we have $P(\theta|D) = P(\theta|\hat{\theta})$ since $\hat{\theta}$ captures all the information in the data. By Bayes Theorem:

$$p(\theta|\hat{\theta}) \propto p(\hat{\theta}|\theta)p(\theta) \tag{2.13}$$

  The form of the distribution $p(\hat{\theta}|\theta)$ would be given by the results of classical statistics. In particular, when $p(\theta)$ is uniform, the two distributions would be identical if we view $p(\hat{\theta}|\theta)$ from a Bayesian perspective. The key of applying these results are: (1) show that $\hat{\theta}$ is a sufficient statistic; (2) from the distribution of estimator $p(\hat{\theta}|\theta)$, determine the distribution of θ given $\hat{\theta}$ (view $\hat{\theta}$ as given).

- Example: normal distribution with known variance. Let x_i iid $N(\mu, \sigma^2)$, and $|hat\mu = \frac{1}{n}\sum_i x_i$. From classical statistics, we know that $\hat{\mu} \sim N(\mu, \sigma^2/n)$. Our posterior:

$$p(\mu|x) = p(\mu|\hat{\mu}) \propto p(\hat{\mu}|\mu)p(\mu) \propto N(\hat{\mu}|\mu, \sigma^2/n) \tag{2.14}$$

  assuming uniform prior of μ. Suppose $\hat{\mu}$ is given, it is easy to see that μ follows normal $N(\hat{\mu}, \sigma^2/n)$.

2.1.1 Bayesian Model Selection

Reference: [Mackay, Chapters 27, 28], [Bishop, Section 3.4, 4.4]

Issues/thoughts about Bayesian model selection:

- Advice from Gelman: avoid model selection, rather, what matters in practice is the posterior predictive distribution. BFs are often very sensitive to prior (model specification).

- Expected log-pointwise predictive density (ELPD): defined as

$$\text{Elpd} = \text{E}\left[\log \int p(x^*|\theta)p(\theta|x)d\theta\right] \tag{2.15}$$

  where x is the given data and x^* is what is to be predicted. The expectation is over x^*. Remarks:

  - Averaging over many data points, so it should be normally distributed.
  - It is similar to prediction error (generalized) in Machine Learning.

- Using posterior predictive density for model comparison: this could be problematic. Suppose we compare two models M_1 and M_2 with very different priors for θ. Now the parameter θ in our data comes from M_2, say, and we have very large data. Then the posterior under M_1 and M_2 are very similar, and thus lose the ability to distinguish the two models.

  - Remark: perhaps we can use how fast the posterior converges, as a measure of how good a prior model is? Similar to PAC learning.

Model selection:

- Model evidence: a model is assessed by its posterior probability:

$$P(M|D) = P(D|M)P(M)/P(D) \tag{2.16}$$

  where $P(D|M)$ is the model evidence. It is given by:

$$P(D|M) = \int P(D|w, M)P(w|M)dw \tag{2.17}$$

  where w is the model parameters.

- Bayes factor: the comparison of two models is determined by the ratio:

$$\frac{P(M_1|D)}{P(M_2|D)} = \frac{P(D|M_1)}{P(D|M_2)} \cdot \frac{P(M_1)}{P(M_2)} \tag{2.18}$$

  In practice, we often use log-BF as the evidence, and the posterior odds ratio is given by:

$$\text{log-posterior odds} = \text{log-BF} + \text{log-prior odds} \tag{2.19}$$

Interpretation of model evidence:

- Parameter constraint: for a model M, assumming the posterior distribution $P(D|w)P(w)$ is approximated by its peak at w_{MAP}, with width $\Delta w_{\text{posterior}}$ and the prior $P(w|M)$ is flat with width Δw_{prior}, so that $P(w) = 1/\Delta w_{\text{prior}}$, we have:

$$\ln P(D|M) \approx \ln P(D|w_{MAP}) + \ln \frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}} \tag{2.20}$$

  Thus for complex models: the prior distribution is more diffused, thus Δw_{prior} is larger; or $\Delta w_{\text{posterior}}$ is smaller. Therefore the second term penalizes the complex models. If there are multiple parameters, we would also have more penalization terms.

- Data generation: the model evidence $P(D|M)$ is the probability of generating a specific dataset D from M. While simpler models can only generate datasets that are fairly similar to each other, complex models can generate a great variety of different datasets. Thus the simpler model cannot fit the data well, whereas the more complex model spreads its predictive probability over too broad arange of data sets and so assigns relatively small probability to any one of them.

The Laplace approximation [MacKay, Chapter 27]:

- Problem: suppose we want to compute the integral:

$$Z = \int f(z)dz \tag{2.21}$$

  where z is K-dimensional variable and $f(z)$ is the (unnormalized) density function.

- The log-density function can be approximated with normal distribution, or equivalently, we apply the Taylor expansion of $\ln f(z)$ at its maximum z_0:

$$\ln f(z) \approx \ln f(z_0) - \frac{1}{2}(z - z_0)^T A(z - z_0) \tag{2.22}$$

  where

$$A_{ij} = -\frac{\partial^2}{\partial z_i \partial z_j} \ln f(z)\bigg|_{z=z_0} \tag{2.23}$$

  Plug in the approximation of $\ln f(z)$ to the integral, and using Gaussian integral, we have:

$$Z \approx f(z_0)\sqrt{\frac{(2\pi)^K}{\det A}} \tag{2.24}$$

Approximating model evidence by Laplace approximation:

- Apply Laplace approximation to $P(D|M)$ (dropping M):

$$\ln P(D) = \ln P(D|\theta_{\text{MAP}}) + \ln P(\theta_{\text{MAP}}) + \frac{K}{2}\ln(2\pi) - \frac{1}{2}\ln\det A \tag{2.25}$$

  where A is the Hessian matrix of second derivatives of the negative log posterior:

$$A = -\nabla\nabla \ln P(D|\theta_{\text{MAP}})P(\theta_{\text{MAP}}) \tag{2.26}$$

- Bayesian information criterion (BIC): If we assume that the Gaussian prior distribution over parameters is broad, and that the Hessian has full rank, then we can approximate very roughly using Equation 2.25:

$$\ln P(D) = \ln P(D|\theta_{\text{MAP}}) - \frac{1}{2}K \ln N \tag{2.27}$$

  where K is the number of parameters and N is the number of data points (the additive constants are omitted).

Bayesian model comparison [BDA ed3, Chapter 7]

- Example: linear regression with noninformative prior. $P(\sigma^2|y)$ follows inverse-chi square and $P(a, b|\sigma^2, y)$ follows normal.

- Measure of predictive accuracy: **log predictive density**, $\log p(y|\theta)$. Connection with KL divergence. However, we cannot use it directly to assess a model in future observations. The out-of-sample predictive accuracy for a single observation $\hat{y}_i$ can be measured by **out-of-sample log predictive density**:

$$\log p_{\text{post}}(\hat{y}_i) = \log \int p(\hat{y}_i|\theta)p(\theta|y)d\theta \tag{2.28}$$

  But since the future data is not given, we should average over them, and this leads to **expected out-of-sample log predictive density**, or ELPD:

$$\text{ELPD} = \text{E}_f(\log p_{\text{post}}(\hat{y}_i)) = \int \log p_{\text{post}}(\hat{y}_i) f(\hat{y}_i) d\hat{y}_i \tag{2.29}$$

- Evaluating a model fit in practice: we generally cannot compute expectation of LPD, since we do not know f in general. So we use the **log pointwise predictive density** (LPPD):

$$\text{llpd} = \sum_{i=1}^{n} \log p_{\text{post}}(y_i) = \sum_{i=1}^{n} \log \int p(y_i|\theta)p(\theta|y)d\theta \tag{2.30}$$

  The posterior integration can be achieved by sampling: let θ^s be the s-th draw from $\theta|y$, we have:

$$\text{llpd} \approx \sum_{i=1}^{n} \log \left(\frac{1}{S} \sum_{s=1}^{S} p(y_i|\theta^s) \right) \tag{2.31}$$

  To estimate predictive accuracy in future data, we have to adjust by: (1) Adjust for the model complexity using within-sample predictive accuracy. (2) Cross-validation.

- Background: if $x \sim N(\mu, \Sigma)$ is n-dim random vector, then $(x-\mu)^T\Sigma^{-1}(x-\mu)$ follows χ^2_n.

- Background: Deviance measures the model fit. It is -2 times the likelihood ratio of two nested models. Based on LRT, it follows χ^2 distribution with dof. k, where k is the difference of number of parameters. Note that the deviance has the scale of k, as the expectation of χ^2_k is k.

- AIC: log predictive density evaluated at the MLE. To account for model complexity, subtract number of model parameters k to penalize complex models.

$$\hat{\text{elpd}}_{\text{AIC}} = \log p(y|\hat{\theta}_{\text{MLE}}) - k \tag{2.32}$$

  The intuition is that the complex model would have higher log-likelihood, with the difference of LL follows χ^2_k.

  - The asymptotic distribution of log predictive density $\log p(y|\theta)$: note that y here refers to in-sample data (technically likelihood not predictive density). Here θ is a random variable follows posterior distribution (normal), and the predictive density is a function of θ. Using Taylor expansion near θ_0 (the posterior mean of θ), one can show that the log predictive density follows χ^2_k distribution with k being the dim of θ (in Taylor expansion, we have the second derivative of log predictive density, which is the covariance matrix of the posterior of θ).

- Deviance information criteria (DIC): when θ has informative prior distribution, the effect nubmer of parameters is not k, so AIC is not appropriate. Similar to AIC, but evaluate $\log p(y|\theta)$ at the mean posterior of θ, and penalize with the effective number of parameters:

$$\hat{\text{elpd}}_{\text{DIC}} = \log p(y|\hat{\theta}_{\text{MP}}) - p_{\text{DIC}} \tag{2.33}$$

  where p_{DIC} is the effective number of parameters, defined as:

$$p_{\text{DIC}} = 2\left[\log p(y|\hat{\theta}_{\text{MP}}) - \text{E}_{\,\theta|y} \log p(y|\theta)\right] \tag{2.34}$$

  The first term is the LPD at Bayesian point estimate, and the second is the LPD averaged over posterior of θ. Clearly, the difference gets larger when we have a complex model, so the difference of the two gives the effective number of parameters.

- WAIC: similar to DIC, but slight variation in computing the effective number of parameters, averaging instead of using a single mean posterior.

- LOO-CV: We first consider the general case: suppose we have training data y and testing data $\hat{y}$, we can evaluate the model by its LPD at testing data:

$$\log p_{\text{post}}(\hat{y}) = \sum_{i} \log \int p(\hat{y}_i|\theta)p(\theta|y)d\theta \tag{2.35}$$

In the leave-one-out cross-validation setting, we treat each data point as a testing data, and all the other data as training data, this leads to the lppd. with LOO-CV:

$$\text{lppd}_{\text{loo-cv}} = \sum_i \log p_{\text{post}(-i)}(y_i) = \sum_i \log \int p(y_i|\theta)p(\theta|y_{-i})d\theta \tag{2.36}$$

To calculate this, we make inference on each y_{-i}, summarized as posterior draws θ^{is} for the s-th draw of i-th partition. LOO-CV can be compared with other ICs: the difference is the penalty (effective number of parameters).

- Estimation of LOO-CV ELPD: [Vehtari, Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC, 2017]. To compute/sample $p(\theta|y_{-i})$ for each i is expensive. One can importance sampling, however not stable. Using Pareto smoothed importance sampling.
- Scenarios of model comparison: (1) Nest model: e.g. model expansion, see adding more parameters lead to better fit; (2) Non-nested models: e.g. compare two sets of predictors in regression.
- Evaluating predictive error comparison: if two models differ (in AIC, DIC, WAIC, or LOO-IC) by 5, say, does the difference matter? (1) Statistical significance: asymptotic theory say if order of 1, then due to chance. (2) Practical significance: e.g. AUC for prediction.
- Problems of all approaches: AIC, DIC not Bayesian. WAIC and LOO-IC: may not apply with hierarchical model. LOO-IC can be hard to compute.
- Remark: WAIC or LOO-IC, not incorporate prior, or not adjust for multiple testing. Could convert ICs to p-values, then standard multiple testing correction.

2.1.2 Bayesian Decision Theory

Motivation:

- Decision theory: an unified framework for inference and prediction problems. The goal of decision theory is to decide what is the best action to take under the uncertain circumstance. If we view the estimation or label prediction as an action to take, then all these problems can be formulated in a decision theory framework.
- Bayesian decision theory: in Bayesian statistics, we are generally interested in the posterior distribution. But if we need to take an action, e.g. an estimator or selecting a model, decision theory may provide the theoretical background.
- Inference problem: the action to take is the value of an unknown parameter (estimator), or the label of a new instance.
- Learning problem: the action is defined on a set of putative instances, thus a procedure/function.

Inference problem: parameter estimation and prediction on response variables/labels:

- Loss function: suppose we are predicting some variable, whose true value is y, and our action is a, the loss function is written as, $L(y, a)$. For example, we may have 0-1 loss if y is binary; or L_2 loss for continuous y:
$$L(y, a) = (y - a)^2 \tag{2.37}$$
- Posterior expected loss: given an instance x, and we want to predict y, the optimal action should minimize the expected loss:
$$\delta(x) = \text{argmin}_a \text{E}\left[L(y, a)\right] \tag{2.38}$$

In the Bayesian approach, the expected loss is averaged over the posterior distribution $p(y|x)$. Thus the expected loss when the action is a is defined by:

$$\rho(a|x) = \mathrm{E}\left[L(y,a)\right] = \int L(y,a)p(y|x) \tag{2.39}$$

The Bayes estimator is thus given by:

$$\delta(x) = \mathrm{argmin}_a \rho(a|x) \tag{2.40}$$

- Binary classification and MAP estimator: when y is binary, and we take 0-1 loss (symmetric), we have the rule:
$$y^*(x) = \mathrm{argmax} p(y|x) \tag{2.41}$$
Thus the optimal y is the one that maximizes the posterior.

- Predicting continuous variables and posterior mean: the posterior expected loss is:
$$\rho(a|x) = \mathrm{E}\left[(y-a)^2|x\right] = \mathrm{E}\left(y^2|x\right) - 2a\mathrm{E}\left(y|x\right) + a^2 \tag{2.42}$$
Minimize this as a function of a, we have:
$$y^*(x) = \mathrm{E}\left(y|x\right) = \int yp(y|x)dy \tag{2.43}$$
Thus the optimal estimator/predictor is the posterior mean.

Supervised learning problem:

- Goal: when we are solving a learning problem, we are not just minizming the loss over any single instance x, instead we will need to minimize the loss over a distribution of x. Furthermore, we will need to explicitly represent the unknown parameter θ.

- Generalization error: suppose our action is δ (a prediction procedure, as we would not to predict for any value of x), the loss of δ when the true parameters are θ is called the generalization error:
$$L(\theta,\delta) = \mathrm{E}_{(x,y)\sim p(x,y|\theta)}[L(y,\delta(x))] = \int\int L(y,\delta(x))p(x,y|\theta)dxdy \tag{2.44}$$

- Posterior expected loss: since θ is unknown, we need to take the expectation over the posterior distribution of θ:
$$\rho(\delta|D) = \int p(\theta|D)L(\theta,\delta)d\theta \tag{2.45}$$

2.2 Bayesian Modeling in Practice

Using Bayesian framework to borrow information [personal notes]

- Motivation: a key advantage of Bayesian inference is the incorporation of prior, which allows one to use information elsewhere (prior). This is important, for example, when the data is sparse relative to the parameters to be estimated. This could be some knowledge one has before analyzing data, but also the information from other parts (e.g. other data samples) of the same dataset. Some general strategies how this could be done.

- Hierachical model: this is the most common strategy of borrowing information from other data points. The idea is that some parameters (objects of the same group) share a common prior distribution.

- Similar parameters in similar objects: discrete version. Hierarhical model requires some discrete grouping and the assumption of a common prior, this may be too stringent. A general idea is that the parameters of similar objects should be similar, but not necessarily of common distribution, and the similarity can be defined across a contiuous spectrum (thus less similar objects would have less similar parameter). Ideas capturing this intuition:
  - MVN: model the covariance of parameters, which is coupled to the similarity of objects. Ex. G-prior for regression.
  - Ising model: encourge similar β for connected objects.
- Spatial model: suppose we have consecutive $\beta_1, \cdots, \beta_n$, we could imagine a stochastic process (e.g. HMM, random walk) that relates these parameters, e.g. $\beta_t \sim N(\beta_{t-1}, \sigma^2)$.

Choosing a good prior:

- Importance of prior: often very important for the final results. Ex. for testing associations (using BF) in genetics: the prior of effect size has a large impact on the final BF.
- Examine the prior distribution and see if it is consistent with domain knowledge. This often means examine the mean, variance, the probability of very rare events (tail distribution), and so. For example, the prior of the relative risk of a genetic variant, one could consider several criteria, including: the mean effect, the fraction of risk vs. protective variants, the percent of variants with very large effects.
- Model selection vs. parameter estimation: in Bayesian inference, there is no clear-cut between the two. Model selection may be understood as a particular kind of prior: the mixture prior, i.e. the prior has multiple components. For example, in genetic association analysis, the effect size is generally close to 0 (for non-causal genes/variants), but could be large for causal genes/variants (e.g. LoF). In this case, it may be difficult to fit a single parametric distribution for prior, and a mixture prior is more appropriate (e.g. 0 and a normal prior).

Modeling prior information: two general strategies of utilizing prior information [Personal notes]

- Using prior information to set the prior distributions: effectively, our inference is conditioned on the prior data, which appears in the distribution $P(\theta|\phi, R)$ where ϕ is the hyperparameters and R represents the prior data. The parameters of the prior, ϕ, can be determined by empirical Bayes or posterior inference.
- Modeling prior information as additional data: there may be advantages of modeling the prior information R as additional data. The prior data R may contain information of some hyperparameters ϕ, and one can use independent data to bette estimate ϕ. In other words, we model:

$$P(\theta, R|\phi) = P(R|\phi)P(\theta|\phi, R) \tag{2.46}$$

  where the distribution $P(R|\phi)$ encodes information of ϕ, and may be estimated from independent data. Example, in rare variant association analysis, V_j, the variant information can be treated as data, and $P(V_j|Z_j = 1)$ can be estimated from independent data.

Sensitivity analysis:

- Goal: how the results depend on prior parameters. For model selection/hypothesis testing problems, we calculate how the BF of a test (or BF distribution of multiple tests in the genome-wide seting) depends on the prior parameters.
- Example: [Bayesian statistical methods for genetic association studies, NRG, 2009] Sensitivity tends to be greatest in situations with less information, such as small studies, or low MAF. For a SNP with $p = 4.1e - 9$, The $\log_{10}(BF)$ depends on σ (prior standard deviation of effect size): e.g. when $\sigma = 0.2$ (low), it is only 2.2, and when $\sigma = 0.8$, it is 5.2.

- Example: [A Bayesian Measure of the Probability of False Discovery in Genetic Epidemiology Studies, Wakefield, AJHG, 2007]. In a lung cancer study of 131 SNPs, the number of predictions change dramatically with π_0 (about 20-30 positive discoveries vs. 2-3 discoveries). However, the predictions (number, and ranking of SNPs) vary little with the other parameters (Figure 8).

Bayesian diagnistic/goodness of fit:

- Idea: suppose our data is x, we could compute the marginal distribution using the estimated model, and compare that with the observed distribution (histogram) of the data.
- Example: [Detecting differential gene expression with asemiparametric hierarchical mixture method, Newton et al, Biostatistics, 2004] Figure 3: QQ plot of the marginal distribution of the model and the empirical distribution. Figure 5: histogram of the fitted and observed expression measurements.

Performance evaluation:

- Individual test: type I error and power.
- Multiple tests: under a target FDR, run the Bayesian methods on simulated data, and estimate the realized FDR and power (number of true discoveries). The results could be represented in a ROC curve.

Examples of Bayesian inference in genetics:

- A hidden Markov random field model for genome-wide association studies (PMID:19822692).
  - Data: NB is a common and lethal pediatric malignancy. GWAS of 1000 cases and 2000 controls, and anlaysis on 31K SNPs in Chr. 6. Single SNP analysis identified three SNPs.
  - Using PPA from HMRF model identified two additional SNPs in LD with these SNPs (PPA close to 1). In addition, one SNP in chr. 6 has PPA 0.74. Including all these 6 SNPs give FDR 0.046.
  - Analysis on 2 predicted regions: 100 permutations, only in 1 simulation, find a SNP with PPA greater than 0.5.
- Detecting differential gene expression with asemiparametric hierarchical mixture method, Newton et al, Biostatistics, 2004:
  - Data: n genes, with expression in the first set of conditions $x_{g,i}$, and expression in the second set of conditions $y_{g,j}$. The goal is to compare if the means of the two conditions are the same.
  - Model: (1) Individual gene: expression depends on the mean, modeled as Gamma distribution (constant CV, so not normal distribution). (2) Hierarchical model: mixture of three cases: equal mean in two conditions; one of the condition has higher mean.
  - Simulation: three secenarios (corresponding to different π's), each method (Bayesian and t-test plus FDR control) targets FDR at 0.05, and compare the performance: the sensitivity and realized FDR.
- Reporting and interpretation in genome-wide association studies [Wakefield, Int. J Epiderm, 2008].
  - Approximate BF (ABF): in terms of the estimate of the effect size, its confidence interval, and the standard deviation of the prior effect size.
  - Dependence of BF on MAF: at small MAF, the same small p-value has lower BF because to achieve such p-value with low MAF, the effect size must be big, which is quite unlikely under the prior.

2.3 Bayesian Inference of Common Probability Distributions

Reference: Gelman *et al.* [2003]

Overview: for Bayesian inference of common probability distributions, we are interested in:

- Posterior distribution of parameters: this is often the goal of Bayesian inference.
- Posterior predictive distribution: important for the purpose of making predictions (e.g. in classification or regression).
- Marginal likelihood: the dependence between data and the hyperparameter(s). This is important in hierarchical models, where we are often interested in the higher-level parameters. Also important in model selection problems. This can be obtained in two ways: (1) by integrating out the parameter:

$$p(y) = \int p(y|\theta)p(\theta)d\theta \tag{2.47}$$

  Or (2) by using the posterior distribution:

$$p(y) = \frac{p(y|\theta)p(\theta)}{p(\theta|y)} \tag{2.48}$$

  The latter may be easier when the posterior has already been solved.

Bernoulli and binomial distribution:

- Model: our data follows the distribution $y \sim \text{Bin}(n;\theta)$, and the prior distribution $\theta \sim \text{Beta}(\alpha, \beta)$.
- Posterior: this is given by:

$$\theta|y \sim \text{Beta}(\alpha + y, \beta + n - y) \tag{2.49}$$

- Posterior predictive distribution: suppose we want to predict a new data point $\hat{y}$ (single observation), which is 0 or 1, thus Bernoulli distribution. The mean of the distribution is:

$$\text{E}(\hat{y}|y) = \int \text{E}(\hat{y}|\theta)p(\theta|y)d\theta = \int \theta p(\theta|y)d\theta = \text{E}(\theta|y) = \frac{\alpha + y}{\alpha + \beta + n} \tag{2.50}$$

  Thus $\hat{y}|y \sim \text{Ber}(\hat{\theta})$, where $\hat{\theta}$ is the posterior mean of θ.

- Marginal likelihood: this is obtained by integrating θ in the likelihood function:

$$p(y|\alpha, \beta) = \int p(y|\theta)p(\theta|\alpha, \beta)d\theta = \frac{\binom{n}{y}}{B(\alpha, \beta)} \int \theta^y (1-\theta)^{n-y} \theta^{\alpha-1}(1-\theta)^{\beta-1} d\theta \tag{2.51}$$

  Apply the definition of Beta function:

$$p(y|\alpha, \beta) = \binom{n}{y} \frac{B(\alpha + y, \beta + n - y)}{B(\alpha, \beta)} \tag{2.52}$$

Poisson distribution:

- Basic model: suppose we have y_i i.i.d. with $y_i \sim \text{Pois}(\theta), 1 \leq i \leq n$, and the prior distribution $\theta \sim \text{Gamma}(\alpha, \beta)$. The posterior distribution is:

$$\theta|y \sim \text{Gamma}\left(\alpha + \sum_i y_i, \beta + n\right) \tag{2.53}$$

Thus the prior can be viewed as α events in β observations. The marginal likelihood is given by:

$$p(y|\alpha,\beta) = \frac{\prod_i \text{Pois}(y_i|\theta) \cdot \text{Gamma}(\theta|\alpha,\beta)}{\text{Gamma}(\theta|\alpha+\sum_i y_i, \beta+n)} \tag{2.54}$$

Plug-in the relevant terms, we have:

$$p(y|\alpha,\beta) = \frac{\Gamma(\alpha+\sum_i y_i)}{\Gamma(\alpha)\prod_i y_i!} \frac{\beta^\alpha}{(\beta+n)^{\alpha+\sum_i y_i}} \tag{2.55}$$

- Model with rate and exposure: suppose for the i-th observation, the count depends on the exposure x_i, we have:

$$y_i \sim \text{Pois}(x_i\theta) \tag{2.56}$$

And we have the same prior $\theta \sim \text{Gamma}(\alpha,\beta)$. The posterior distribution is given by:

$$\theta|y \sim \text{Gamma}\left(\alpha+\sum_i y_i, \beta+\sum_i x_i\right) \tag{2.57}$$

The marginal likelihood is:

$$p(y|\alpha,\beta) = \prod_i \frac{x_i^{y_i}}{y_i!} \cdot \frac{\Gamma(\alpha+\sum_i y_i)}{\Gamma(\alpha)} \frac{\beta^\alpha}{(\beta+\sum_i x_i)^{\alpha+\sum_i y_i}} \tag{2.58}$$

- Relation with negative binomial distribution: when there is a single observation y with exposure x, the marginal likelihood is:

$$p(y|\alpha,\beta) = \frac{\Gamma(\alpha+y)}{\Gamma(\alpha)y!} \cdot \left(\frac{\beta}{\beta+x}\right)^\alpha \left(\frac{x}{\beta+x}\right)^y = \text{NegBin}\left(y|\alpha, \frac{x}{\beta+x}\right) \tag{2.59}$$

The expectation of y is:

$$\text{E}\left(y|\alpha,\beta\right) = \frac{\alpha \cdot \frac{x}{\beta+x}}{1-\frac{x}{\beta+x}} = x\frac{\alpha}{\beta} \tag{2.60}$$

which is the product of exposure and the average prior rate. Thus the negative binomial can be understood as a model of discrete distribution, similar to Poisson, but with variance possibly different from the rate parameter.

Univariate normal distribution with known variance:

- Likelihood function: given observations $y_1, \cdots, y_n$ iid. $N(\mu,\sigma^2)$. The likelihood is:

$$p(y|\mu,\sigma^2) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^n (y_i-\mu)^2\right] \tag{2.61}$$

- Conjugate prior: the likelihood function is the exponential of a quadratic function of μ, thus choose the prior of the same form (normal distribution):

$$\mu \sim N(\mu_0,\tau^2) \tag{2.62}$$

- Posterior distribution: express the posterior distribution in the form of exponential of a quadratic, and we find that the posterior is also normal:

$$\mu|y \sim N(\mu_n,\tau_n^2) \tag{2.63}$$

where:

$$\mu_n = \left(\frac{1}{\tau_0^2} + \frac{n}{\sigma^2}\right)^{-1}\left(\frac{1}{\tau_0^2}\mu_0 + \frac{n}{\sigma^2}\bar{y}\right) \qquad \frac{1}{\tau_n^2} = \frac{1}{\tau_0^2} + \frac{n}{\sigma^2} \tag{2.64}$$

Thus the posterior is a combination of prior and data. Its expectation is the weighted average of μ_0 and $\bar{y}$, with weights equal to the precision (inverse of variance) of prior and data. Its precision is the sum of precision of prior and data.

- Posterior predictive distribution: is the average over $\theta|y$. It is normal distribution with:

$$\mathrm{E}\left(\tilde{y}|y\right) = \mathrm{E}_{\theta|y}[\mathrm{E}\left(\tilde{y}|\theta, y\right)] = \mathrm{E}\left(\theta|y\right) = \mu_n \tag{2.65}$$

$$\mathrm{Var}\left(\tilde{y}|y\right) = \mathrm{E}_{\theta|y}[\mathrm{Var}\left(\tilde{y}|\theta, y\right)] + \mathrm{Var}_{\theta|y}[\mathrm{E}\left(\tilde{y}|\theta, y\right)] = \mathrm{E}\left(\sigma^2|y\right) + \mathrm{Var}\left(\theta|y\right) = \sigma^2 + \tau_n^2 \tag{2.66}$$

Thus the variance of $\tilde{y}|y$ has two components: one from the variance of $\theta|y$, and the other from the inherent error (σ^2).

Univariate normal distribution with known mean but unknown variance:

- Conjugate prior: from the likelihood function, the conjugate prior should have the form:

$$p(\sigma^2) \propto (\sigma^2)^{-(\alpha+1)} e^{-\beta/\sigma^2} \tag{2.67}$$

We thus choose the scaled inverse χ^2 distribution: $\sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2)$ where ν_0 is dof. and σ_0^2 is (roughly) the mean. The density function:

$$p(\sigma^2) \propto (\sigma^2)^{-(\nu_0/2+1)} \exp\left(-\frac{\nu_0\sigma_0^2}{2\sigma^2}\right) \tag{2.68}$$

- Posterior: the mean is θ, the sample variance is:

$$v = \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta)^2 \tag{2.69}$$

The posterior of σ^2 is given by:

$$\sigma^2|y \sim \text{Inv-}\chi^2\left(\nu_0 + n, \frac{\nu_0\sigma_0^2 + nv}{\nu_0 + n}\right) \tag{2.70}$$

Thus the posterior of σ^2 has dof. equal to $\nu_0 + n$ (larger n, sharper distribution), and the scale parameter is the weighted average of prior mean (roughly) and sample variance, where weights are given by the dof.

- Alternative proof using distribution of sample variance: consider a problem y_i iid $N(0, \sigma^2)$ and we want to infer σ. Let $S^2 = \frac{1}{n}\sum_i y_i^2$ be the sample variance. It is easy to show that S^2 is a sufficient statistic. We have this result from classical statistics:

$$\frac{(n)S^2}{\sigma^2} \sim \chi_n^2 \tag{2.71}$$

Note that we have n instead of $n-1$ here because mean is given. From this we have (using inverse of χ^2 distribution):

$$\frac{\sigma^2}{nS^2} \sim \text{Inv} - \chi_n^2 \Rightarrow \sigma^2|S^2 \sim \text{Scaled-Inv} - \chi^2(n, S^2) \tag{2.72}$$

Thus the posterior of σ^2 is inverse chi-square with dof n (large number of samples, sharper peak), and the scale determined by S^2.

Univariate normal distribution with unknown mean and variance: conjugate prior

- Conjugate prior:

$$\begin{array}{rcl} \sigma^2 & \sim & \text{Inv-}\chi^2(\nu_0, \sigma_0^2) \\ \mu|\sigma^2 & \sim & N(\mu_0, \sigma^2/\kappa_0) \end{array} \tag{2.73}$$

  Thus ν_0 is the dof. of the σ^2 (higher dof., more accurate), σ_0 is the scale of σ^2; μ_0 is the location of μ, and κ_0 is the number of measurements.

  – Remark: To see why we want prior of μ to depend on σ, we note that if μ has a different variance, then in the posterior, the exponential terms for the prior and the likelihood cannot be combined.

- Posterior distribution:

$$p(\mu, \sigma^2|y) \propto (\sigma^2)^{-(\frac{\nu_0+n}{2}+\frac{3}{2})} \exp\left(-\frac{1}{2\sigma^2}[\nu_0\sigma_0^2 + \kappa_0(\mu-\mu_0)^2]\right) \exp\left(-\frac{1}{2\sigma^2}[(n-1)s^2 + n(\bar{y}-\mu)^2]\right) \tag{2.74}$$

  where s^2 is the sample variance:

$$s^2 = \frac{1}{n-1}\sum_i (y_i - \bar{y})^2 \tag{2.75}$$

- Conditional posterior distribution $p(\mu|\sigma^2, y)$: this is similar to the case of known variance, we have $\mu|\sigma^2, y \sim N(\mu_n, \sigma^2/\kappa_n)$, where

$$\mu_n = \frac{\kappa_0}{\kappa_0+n}\mu_0 + \frac{n}{\kappa_0+n}\bar{y} \qquad \kappa_n = \kappa_0 + n \tag{2.76}$$

- The marginal posterior distribution $p(\sigma^2|y)$: we integrate out μ in the joint posterior density:

$$p(\sigma^2|y) \propto (\sigma^2)^{-(\frac{\nu_0+n}{2}+\frac{3}{2})} \exp\left(-\frac{1}{2\sigma^2}[\nu_0\sigma_0^2 + (n-1)s^2]\right) \cdot I \tag{2.77}$$

  where

$$I = \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2\sigma^2}[\kappa_0(\mu-\mu_0)^2 + n(\bar{y}-\mu)^2]\right) d\mu \tag{2.78}$$

  We apply the complete-the-square trick:

$$\kappa_0(\mu-\mu_0)^2 + n(\bar{y}-\mu)^2 = (\kappa_0+n)\left(\mu - \frac{\kappa_0\mu_0 + n\bar{y}}{\kappa_0+n}\right)^2 + \frac{\kappa_0 n}{\kappa_0+n}(\bar{y}-\mu_0)^2 \tag{2.79}$$

  Plug in this to the integral I:

$$I = \sigma\sqrt{\frac{2\pi}{\kappa_0+n}} \exp\left(-\frac{1}{2\sigma^2}\frac{\kappa_0 n}{\kappa_0+n}(\bar{y}-\mu_0)^2\right) \tag{2.80}$$

  The posterior distribution $\sigma^2|y$ thus follows scaled inverse-χ^2 distribution: Inv-$\chi^2(\nu_n, \sigma_n^2)$, where

$$\nu_n = \nu_0 + n \qquad \nu_n\sigma_n^2 = \nu_0\sigma_0^2 + (n-1)s^2 + \frac{\kappa_0 n}{\kappa_0+n}(\bar{y}-\mu_0)^2 \tag{2.81}$$

- The marginal posterior distribution $p(\mu|y)$: by integrating out σ^2 in the joint posterior density, we have:

$$\mu|y \sim t_{\nu_n}(\mu_n, \sigma_n^2/\kappa_n) \tag{2.82}$$

- Sampling: first sample σ^2 from $p(\sigma^2|y)$, then sample μ from $p(\mu|\sigma^2, y)$. For posterior predictive distribution: after sampling μ and σ^2, sample $\tilde{y}$ from $p(\tilde{y}|\mu, \sigma^2)$.

- Noninformative prior: as a special case of the conjugate prior, we have:

$$p(\mu, \sigma^2) \propto (\sigma^2)^{-1} \tag{2.83}$$

Note that the prior is improper, i.e. the integrate is infinite; however, the posterior is proper.

- Alternative form of conjugate prior [Bishop]: it is sometimes easier to work with precision (inverse of covariance), $\tau = 1/\sigma^2$. The conjugate prior of τ is the Gamma distribution:

$$p(\tau) = \text{Gamma}(\tau|a_0, b_0) = \frac{b_0^{a_0}}{\Gamma(a_0)}\tau^{a_0-1}e^{-b_0\tau} \tag{2.84}$$

The conjugate prior of the mean:

$$p(\mu|\tau) = N(\mu|\mu_0, (\lambda_0\tau)^{-1}) \tag{2.85}$$

The posterior distribution of τ and μ follow the Gamma and normal distributions, respectively (see [Bishop, 2.3.6]).

- Alternative derivation: we use the properties of MVN, given the distribution of μ and $y|\mu$ (multivariate), we derived the distribution $\mu|y$ and y (σ^2 is treated as constant):

$$\mu|\sigma^2 \sim N(\mu_0, \sigma^2/\kappa_0) \tag{2.86}$$

$$y|\mu, \sigma^2 \sim N(\mu\mathbf{1}, \sigma^2 I) \tag{2.87}$$

where $\mathbf{1}$ is the vector consisting of 1's. Then we obtain $\mu|y, \sigma^2$ as before, and:

$$y|\sigma^2 \sim N(\mu_0\mathbf{1}, S_n) \tag{2.88}$$

where

$$S_n = \sigma^2\left(I + \frac{1}{\kappa_0}\mathbf{1}\cdot\mathbf{1}^T\right) \tag{2.89}$$

Univariate normal distribution with unknown mean and variance: semi-conjugate prior

- Semi-conjugate prior: in some cases, we don't want the prior of μ to be dependent on the variance parameter. So we have the prior:

$$\mu \sim N(\mu_0, \tau_0^2) \qquad \sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2) \tag{2.90}$$

- Conditional posterior distribution $p(\mu|\sigma^2, y)$:

$$\mu|\sigma^2, y \sim N(\mu_n, \tau_n^2) \tag{2.91}$$

where

$$\mu_n = \frac{\frac{1}{\tau_0^2}\mu_0 + \frac{n}{\sigma^2}\bar{y}}{\frac{1}{\tau_0^2} + \frac{n}{\sigma^2}} \qquad \frac{1}{\tau_n^2} = \frac{1}{\tau_0^2} + \frac{n}{\sigma^2} \tag{2.92}$$

- Posterior distribution $p(\sigma^2|y)$: since we already know $p(\mu|\sigma^2, y)$, we could solve it by:

$$p(\sigma^2|y) = \frac{p(\mu, \sigma^2|y)}{p(\mu|\sigma^2, y)} \propto \frac{N(\mu|\mu_0, \tau_0^2)\text{Inv-}\chi^2(\sigma^2|\nu_0, \sigma_0^2)\prod_i N(y_i|\mu, \sigma^2)}{N(\mu|\mu_n, \tau_n^2)} \tag{2.93}$$

This is true for any value of μ, so we choose $\mu = \mu_n$ s.t. the denominator is simple: $(\sqrt{2\pi}\tau_n)^{-1}$. So we have:

$$p(\sigma^2|y) \propto \tau_n N(\mu_n|\mu_0, \tau_0^2)\text{Inv-}\chi^2(\sigma^2|\nu_0, \sigma_0^2)\prod_i N(y_i|\mu_n, \sigma^2) \tag{2.94}$$

Even though this does not have a simple conjugate form, this can be easily computed for any value of σ^2.

- Posterior sampling: first sample from $\sigma|y$ using the numerical method; and then sample $\mu|\sigma^2, y \sim N(\mu_n, \tau_n^2)$.

Multivariate normal distribution with known variance:

- Likelihood function: given $y_1, \cdots, y_n$ i.i.d. $N(\mu, \Sigma)$,

$$p(y_1, \cdots, y_n|\mu, \Sigma) \propto |\Sigma|^{-n/2} \exp\left[-\frac{1}{2}\text{tr}\,(\Sigma^{-1}S)\right] \tag{2.95}$$

  where S is the matrix of sum of squares:

$$S = \sum_{i=1}^{n}(y_i - \mu)(y_i - \mu)^T \tag{2.96}$$

- Conjugate prior: $\mu \sim N(\mu_0, \Lambda_0)$.
- Posterior distribution: this is similar to the univariate case. $\mu|y \sim N(\mu_n, \Lambda_n)$, where

$$\mu_n = (\Lambda_0^{-1} + n\Sigma^{-1})^{-1}(\Lambda_0^{-1}\mu_0 + n\Sigma^{-1}\bar{y}) \qquad \Lambda_n^{-1} = \Lambda_0^{-1} + n\Sigma^{-1} \tag{2.97}$$

- Posterior predictive distribution: we have

$$\text{E}\,(\tilde{y}|y) = \mu_n \qquad \text{Var}\,(\tilde{y}|y) = \Sigma + \Lambda_n \tag{2.98}$$

Multivariate normal distribution with unknown variance:

- Conjugate prior: similar to the univariate case, but replace inverse-χ^2 with inverse Wishart distribution:

$$\begin{array}{rcl} \Sigma & \sim & \text{Inv-Wishart}_{\nu_0}(\Lambda_0^{-1}) \\ \mu|\Sigma & \sim & N(\mu_0, \Sigma/\kappa_0) \end{array} \tag{2.99}$$

- Posterior distribution: similar to the univariate case, the conditional posterior distribution

$$\mu|\Sigma, y \sim N(\mu_n, \Sigma/\kappa_n) \tag{2.100}$$

  where

$$\mu_n = \frac{\kappa_0}{\kappa_0 + n}\mu_0 + \frac{n}{\kappa_0 + n}\bar{y} \qquad \kappa_n = \kappa_0 + n \tag{2.101}$$

  And the marginal posterior of $\Sigma|y \sim \text{Inv-Wishart}_{\nu_n}(\Lambda_n^{-1})$, where

$$\nu_n = \nu_0 + n \qquad \nu_n\Lambda_n = \Lambda_0 + S + \frac{\kappa_0 n}{\kappa_0 + n}(\bar{y} - \mu_0)(\bar{y} - \mu_0)^T \tag{2.102}$$

  The marginal posterior $\mu|y$ is multivariate t-distribution (see the book).

2.4 Bayesian Computation: MCMC

Lessons in Bayesian computation [personal notes]:

- Adaptive MCMC [Stephens]: in general, the adaptive MCMC is not theoretically correct: not Markov Chain (the distribution changes). So it is hard to prove convergence. In practice, often stop at a fixed step size after some point, then a standard MCMC.
- Computing Bayes factors [Stephens, Gelman]: generally difficult with more than a few parameters.

- How to study the convergence of MCMC: study the geometric rate of convergence [Stephens].

MCMC review [personal notes]

- Initialization: warm start, but need to have multiple chains started at different places in the parameter space.
- Strategy: mixing MH and Gibbs.
- Visualization of posterior draws: could help monitor chain behavior.
- Testing convergence: multiple chains, and test both mixing and stationarity.

Sampling and expectation [MacKay, Chapter 29]:

- Two computational problems: sample from a distribution $P(x)$ and computing the expectation of some function under $P(x)$:

$$\Phi = \langle \phi(x) \rangle = \int \phi(x) P(x) dx \tag{2.103}$$

- If we can solve the first problem, we can solve the second one, by sampling $x^{(r)}, 1 \leq r \leq R$, and compute the estimator

$$\hat{\Phi} = \frac{1}{R} \sum_r \phi(x^{(r)}) \tag{2.104}$$

  It is easy to check that the estimator is unbiased, $\mathrm{E}(\hat{\Phi}) = \Phi$ and its variance:

$$\mathrm{Var}(\hat{\Phi}) = \sigma^2 / R \tag{2.105}$$

  where $\sigma^2 = \mathrm{Var}(\phi(x))$.

- **Remark**: we can use the frequentist approach to evaluate the estimator of the integral to be evaluated: its mean and variance (and how it changes with R). In particular, most estimators are unbiased, so we need to analyze the variance.
- Why sampling is difficult? Often we can evaluate $P^*(x) = P(x)/Z$, but cannot evaluate $P(x)$ as Z is unknown. Ex. we want to sample $p(\theta|y) = p(\theta)p(y|\theta)/p(y)$: we can evaluate the numerator, but do not know the denomiator. Even if we know the true $P(x)$, we may not be able to sample it: we do not know whether a $P(x)$ is large unless we evaluate $P(x)$ for all x.

Importance sampling [MacKay, Chapter 29]

- Suppose we want to evaluate $\Phi = \mathrm{E}(\phi(x))$, we assume that we can sample from a distribution $Q(x)$. The idea is that, if $P(x) < Q(x)$, we over-sampled x, and if $P(x) > Q(x)$, we under-sampled x, so we need to adjust for this. Suppose we can evaluate $Q^*(x)$, where $Q(x) = Q^*(x)/Z_Q$. We then define:

$$w_r = \frac{P^*(x^{(r)})}{Q^*(x^{(r)})} \tag{2.106}$$

  Our estimator is:

$$\hat{\Phi} = \frac{\sum_r w_r \phi(x^{(r)})}{\sum_r w_r} \tag{2.107}$$

- Proof: intuitively, the numerator is proportional to Φ, and one can show that the constant is given by the denominator.
- Behavior of importance sampling: in general, if Q is substantially smaller than P at the places where P has prob. mass (typical set), then we will have very large w_r in those points. This would lead to large variance of the estimator. So in general, the distribution Q should be long-tailed.

Practical issues in sampling posterior distribution (general issues): [GCSR, Section 3.7, 5.3, Chapter 10]

- Estimating the range of parameters: the first step is often to obtain a crude estimate of the parameters.
  - Ex. for hierarchical models, one can obtain the rough estimate through complete pooling (population mean) and no-pooling (group mean).
  - Posterior mode approximation: normal distribution centered on the posterior mode can serve as an approximation of the posterior.
- Sampling strategy: if appropriate analytic forms exist, to sample (θ, ϕ), first sample marginal posterior distribution, e.g. $p(\phi|y)$, then sample conditional posterior distribution $p(\theta|\phi, y)$. If no analytic forms exist, MCMC is the general strategy.
  - Discretization: if involved (sampling in a lattice with grids), the range of grids can be roughly estimated via normal approximation [Section 4.1, page 103]. Also, since we can only sample at the resolution of grids, could add random jitter within each gride for sampled points.
- Visualization and inspection: 1-D we could use histogram; for 2-D, use scatter plot or contour plot for density.
- Debugging via fake data: to test if the sampling procedure works properly, simulate the data using the some known parameter value, and apply the sampling procedure, and test if the correct value can be recovered (within posterior interval).
- Example: bioassay experiment [Section 3.7]. Two parameters α, β, describe the dose-response relation of a drug. The posterior distribution $p(\alpha, \beta|x)$ can be calculated for given value of α, β (but no analytic form). Perform simulation with 2D contour plot.

MCMC: [GCSR, chapter 11]

- Idea of MCMC: to sample from a distribution $P(x)$, we design a Markov chain whose equilibrium distribution is $P(x)$. This is done through implementing the detailed balance.
- Metropolis algorithm: suppose we want to sample from $P(x)$, the Metropolis algorithm proposes a move from x^t to x' using the jumping distribution $Q(x'|x^t)$. Note that Q is symmetric in Metropolis. x' is accepted with probability:
$$r = \frac{P(x')}{P(x^t)} \tag{2.108}$$
If accepted, we have $x^{t+1} = x'$. Intuitively, this is similar to stochastic hill climbing for optimization: if $P(x')$ increases, then we should accept it; otherwise accept with only a probability.
- M-H algorithm: relax the assumption that $Q(\cdot)$ must be symmetric: if not symmetric, we accept with probability:
$$r = \frac{P(x')Q(x^t|x')}{P(x^t)Q(x'|x^t)} \tag{2.109}$$
This is important: e.g. to have Gibbs sampler as a special case. The most common proposal function Q is the normal jumping kernel, $X'|X \sim N(X, \Sigma)$.
- Gibbs sampler as a special case of M-H algorithm: in each cycle, Gibbs sampler performs d steps (d variables). We only need to show that the true distribution P is an equilibrium distribution of the Markov chain. To see that:
  - First, the chain is irreducible, i.e. one can move from any state to any other state of the chain (accessible). So there is a unique equilibrium distribution.

– Suppose the chain is at the distribution P, it is easy to see that in each step, the distribution at $t+1$, $P^{t+1} = P^t = P$, since detailed balance is satisfied at each of the d steps. Therefore, P is an equlibrium distribution.

Remark: for a MC, as long as each step the detailed balance is satisfied, the distribution will converge to the target distribution.

- Combining Gibbs sampler and M-H algorithm: in a practical problem, some conditional distributions are conjugate (easy to sample) while the others maybe not, so mixing MCMC and Gibbs may be necessary. The convergence to the target distribution follows from the fact that each step satisfies the detailed balance (thus not changing the equilibrium distribution).
  - M-H embedded in a Gibbs sample structure: to sample (X, Y), we repeatedly sample $X|Y$ and $Y|X$, each with M-H algorithm.
  - Mixing M-H and Gibbs sampling: to sample (X, Y), we repeatedly (1) sample $X|Y$ using Gibbs, then (2) sample from $Y|X$ with M-H updating.
  - Block-level MCMC: mix M-H and Gibbs at the level of blocks of variables/parameters.

Practical issues of MCMC: [GCSR, Chapter 11]

- Unnormalized probability: needs to checked. In most cases, working with unormalized density would not affect the sampling algorithm.
- Inference from iterative simulation: challenges include (1) the simulations may be unrepresentative of the target distribution, if the chains have not converged. (2) Within sequence correlation reduces the number of effective draws.
- Discard early runs: called warm-up. Generally discard half. Ex. we run 200 iterations, if not converge, then run another 200, and discard the original 200.
- Thinning: we could improve the efficiency of simulation runs by use every k-th simulation draw.
- Assessing convergence: general strategies
  - Run with different sequences/chains with starting points dispersed throughout parameter space. If converged, the different sequences should have the same distributions. See example of drawing from 2D-MVN (BDA V3: Figure 11.1).
  - Monitor convergence by some scalar estimands: parameters or some functions on parameters. Often good to check log-posterior density. It is better to transform the estimands s.t. they are normally distributed.
  - Importance of both mixing and stationarity (BDA V3: Figure 11.3): (1) two chains each stationary, but not mixing; (2) two chains mix well, but neither reach stationary distribution.
- Multi-chain strategy for testing convergence: after burn-in period, split each chain in half and check that all the resulting half sequences have mixed. This checks both mixing and stationarity (comparing first and second half). A test statistic for convergence (Rhat): let W be within chain variance and B between chain variance of the scalar estimand. We compute the variance of posterior samples of the estimand ψ as:

$$\hat{\text{Var}}(\psi|y) = \frac{n-1}{n}W + \frac{1}{n}B \tag{2.110}$$

as the weighted average of W and B. We note that W always under-estimates with finite samples, but converge to truth as $n \to \infty$. So we compute:

$$\hat{R} = \sqrt{\frac{\hat{\text{Var}}(\psi|y)}{W}} \tag{2.111}$$

This ratio should decline to 1 as $n \to \infty$.

MCMC convergence analysis [Statistical Rethinking, Chapter 8]

- Defining Stan model: define the distribution of each parameter, and the distribution of data (likelihood). Define/initialize data.
- Summarizing posterior samples: use extract.samples() function, then plot pairwise correlation and histogram with pair() function.
- Diagnosis of chain: use plot() function to show traces of parameters. Check for: stationarity and mixing (no strong auto-correlation, zigzag behavior).
- Number of samples needed: no simple answer. If the goal is to find mean, generally fewer. If interested in the distribution at extreme values, need more samples. Generally discard the earlier samples (warm-up). Note: different from burn-in, the warm-up are used to adapt sampling, not from target distribution.
- Number of chains needed: use multiple chains (often 4) to check. And if confirm the correct behavior, use a single long chain is more efficient - no multiple warm-up.
- Dealing with un-identifiable models: using flat prior can lead to wildly large parameters, e.g. in regression with strong colinearity. Use weakly informative prior can help a lot.

Efficient MCMC samplers [BDA, Chapter 12]

- Parameterization: Gibbs sampler is most efficient when the parameters are independent, so if possible, reparameterize s.t. the posterior distribution is independent (if normal, then perform linear transformation).
- Jumping rule: for M-H algorithm, suppose we could approximate the target distribution by a normal distribution with variance matrix Σ. Then the normal jumping kernel is:

$$Q(X'|X^t) \sim N(X^t, c^2\Sigma) \tag{2.112}$$

  The most efficient has scale $c \approx 2.4\sqrt{d}$, where d is the number of variables.
- Adaptive algorithm: the jumping rule (the scale) can be tuned. Intuitively, when close to the optimum, we reduce the step size s.t. the optimum is not missed. However, we need to fix the step size in the end to draw samples (otherwise, the chain may not converge).
- Acceptance rate: one could monitor the MCMC runs by checking the acceptance rates. For Metropolis jumps, tune the step size s.t. the acceptance rates are near 20% (when altering a vector of parameters) or 40% (when altering a single parameter a time).

Obtaining an approximate sample through posterior mode:

- Crude estimate: the first step of sampling is often to find a crude estimate of the parameter values. Ex. for hierarchical normal model, this could be done through estimation of the mean of each group, then estimate the population mean and variance from group means.
- Finding posterior modes: any optimization algorithm can be used if the (normalized) density can be evaluated for any parameter values.
  - For Bayesian problems, the conditional maximization (CM) algorithm and EM (for marginal posterior, see below) are often useful.
  - Finding multiple local modes: run the algorithm with different starting points to obtain all the local modes.

- Normal or normal-mixture approximations: suppose we find the posterior mode at $\hat{\theta}$, and the covariance matrix V_θ can be found (e.g. through numerical derivative), then we could approximate the posterior by $N(\hat{\theta}, V_\theta)$. If we find multiple modes, then the posterior is approximated by normal mixture:

$$p_{\text{approx}}(\theta) \propto \sum_k w_k N(\theta|\hat{\theta}_k, V_{\theta k}) \tag{2.113}$$

  where w_k is the weight of the k-th mode. The weights can be determined by equating the actual density at each mode to the approximate density from the equation above. For instance, if the modes are well separated, we can solve w_k:

$$w_k = q(\hat{\theta}_k|y)|V_{\theta k}|^{1/2} \tag{2.114}$$

  where $q(\cdot)$ is the unnormalized density.

- Marginal posterior: this is imporant for two reasons:
  - Nuisance parameters and missing data: we may have a number of parameters/variables in the model that are of no interest.
  - Conditional sampling: when the number of parameters in the model is large, sampling/approximation may be difficult (especially with techniques based on posterior mode). Using marginal posterior may greatly reduce the number of parameters/variables, and samping in a low dimensional space is much safer.

- Finding marginal posterior via EM algorithm: suppose we have a model of $\theta = (\gamma, \phi)$ and want to find the posterior mode of $p(\phi|y)$. The EM algorithm for missing data can be applied. At the E-step, we find the expectation of the log posterior density:

$$Q(\phi|\phi^t) = \mathrm{E}_{\gamma|\phi^t,y}[\log p(\gamma, \phi|y)] \tag{2.115}$$

  And at the M-step, the Q function is maximized.
  - ECM algorithm: maximiation of the Q function can be achieved through conditional maximazation (CM).
  - Multiple modes: the EM algorithm can be run with different starting points to obtain multiple local modes.
  - Covariance matrix: normal approximation requires the covarance matrix of $\phi|y$ at $\hat{\phi}$. This could be done via the SEM algorithm.

Example: hierarchical normal model:

- Model: within the j-th group ($1 \le j \le J$), we have:

$$y_{ij} \sim N(\theta_j, \sigma^2) \tag{2.116}$$

  And the mean of each group:

$$\theta_j \sim N(\mu, \tau^2) \tag{2.117}$$

  The unknown parameters are θ, μ, τ and σ^2. The prior is: $p(\mu, \log\sigma, \tau) \propto 1$. The posterior:

$$p(\theta, \mu, \tau, \log\sigma) \propto \prod_{j=1}^{J} N(\theta_j|\mu, \tau^2) \prod_{j=1}^{J} \prod_{i=1}^{n_j} N(y_{ij}|\theta_j, \sigma^2) \tag{2.118}$$

- Gibbs sampling: the conditional posteriors can be easily determined:
  - Conditional posterior of θ: $\theta_j|\mu, \tau, \sigma, y$ follows normal distribution with conjugate prior $N(\mu, \tau^2)$.

  - Conditional posterior of μ: only depends on θ_j and τ, $\mu|\theta,\tau,\sigma,y \sim N(\hat{\mu},\tau^2/J)$ where $\hat{\mu}$ is the mean of θ_j.
  - Conditional posterior of σ^2: $\sigma^2|\theta,\mu,\tau,y$ only depends on θ and y, inverse-χ^2 distribution.
  - Conditional posterior of τ^2: $\tau^2|\theta,\mu,\sigma,y$ only depends on θ and μ. Also inverse χ^2 distribution.

- M-H and Gibbs sampling: notice that at the population level, only three parameters μ,τ,σ. So we could perform Metroplis jumping of the three parameters in low-dimensional space, and once the three parameters are sampled, we sample θ_j conditioned on these values.

- Normal approximation: the marginal posterior of $\mu,\tau,\sigma|y$ can be approximated by normal distribution (low-dimensional space), so we need to determine the mode of the posterior. The marginal posterior is obtained by integrating over θ in the joint posterior (similar to the posterior predictive of normal distribution). It can be maxizied by EM, the log. of joint posterior density:

$$\log p(\theta,\mu,\tau,\log\sigma|y) = -n\log\sigma-(J-1)\log\tau-\frac{1}{2\tau^2}\sum_j(\theta_j-\mu)^2-\frac{1}{2\sigma^2}\sum_j\sum_{i=1}^{n_j}(y_{ij}-\theta_j)^2+\text{Const} \quad (2.119)$$

  At the E-step, only the last two terms depend on θ (this missing parameters), and the expectation can be easily determined (both are quadratic of θ_j).

Computing marginal likelihood and Bayes factors:

- Laplacian approximation.

- Sampling from prior: suppose we want to compute $P(D) = \int p(D|\theta)p(\theta)d\theta$, we sample θ_i from $p(\theta)$, then

$$P(D) \approx \frac{1}{n}\sum_i P(D|\theta_i) \quad (2.120)$$

  The problem of this is that it has a large variance. The posterior of θ is often a narrow peak, so $P(D|\theta)$ is close to 0 most of times (when it is outside that peak), and occasionally get big values.

- Harmonic mean estimator: the problem of using prior is that we often sample from the region with little support. So instead, we sample θ from the posterior, $\theta^{(i)}, 1 \le i \le m$. Now we can compute the integral with importance sampling where the weights are:

$$w_i = \frac{P(\theta^{(i)})}{P(\theta^{(i)}|D)} = \frac{P(D)}{P(D|\theta^{(i)})} \quad (2.121)$$

  Plug in this to the equation of importance sampling, we have the estimator:

$$\hat{P}(D) = \left[\frac{1}{m}\sum_i P(D|\theta^{(i)})^{-1}\right]^{-1} \quad (2.122)$$

  The problem of the Harmonic mean estimator is that: it depends only the sample from the posterior, which is somewhat insensitive to the prior. When we have large D, then two different models would have the same posterior, and thus same $\hat{P(D)}$. The true marginal likelihood, on the other hand, depend strongly on the prior. See `https://radfordneal.wordpress.com/2008/08/17/the-harmonic-mean-of-the-likelihood-worst-monte-carlo-method-ever/`. So the Harmonic mean estimator works well only when the prior has a large influence on the posterior.

- Using long-tailed distribution: the problem with Harmonic mean is that the posterior distribution is often 0, outside the peak; so it has a vary large variance, and not a good proposal distribution. To fix it, the general intuition of choosing the proposal distribution is to have a long tail distribution, e.g. mixture of prior and posterior distribution [Weighted Average Importance Sampling and Defensive Mixture Distributions, Hesterberg]. We can also sample from posterior, but combining it with KDE to obtain a smooth distribution.

Improving marginal likelihood estimation for Bayesian phylogenetic model selection [Xie and Chen, Syst Biol, 2011]

- Problem: let y be the data, θ the model parameters. We define $f(\theta)$ be the prior of θ, $f(y|\theta)$ the likelihood. Our goal is to evaluate $f(y) = \int f(y|\theta)f(\theta)d\theta$.
- Harmonical mean estimator is biased: intuitively, we would mostly sample from high posterior regions, and as a result, we will have overrepresentation of high likelihood.
- Idea of Stepping stone sampling: we use importance sampling, but we use a progression of importance distributions that vary from prior to posterior. During progression, the distributions change incrementally so that at every step, the importance distribution is a good approximation for computing the marginal likelihood.
- Power posterior density: we define density function

$$q_\beta(\theta) = f(y|\theta)^\beta f(\theta) \tag{2.123}$$

  This function is prior when $\beta = 0$ and posterior (unnormalized) when $\beta = 1$. The normalization constant is:

$$C_\beta = \int q_\beta(\theta)d\theta \tag{2.124}$$

  And the normalized PDF as $p_\beta(\theta) = q_\beta(\theta)/C_\beta$. It is easy to see that the marginal likelihood is c_1, and we can write it as a product of $c_{\beta_k}/c_{\beta_{k-1}}$ as we vary β from 0 to 1.
- Computing $c_{\beta_k}/c_{\beta_{k-1}}$ by importance sampling: when evaluating both numerator and denomator, we use $p_{\beta_{k-1}(\theta)}$ as the importance distribution. The weights is given by:

$$w(\theta) = \frac{f(\theta)}{p_{\beta_{k-1}(\theta)}} = \frac{c_{\beta_{k-1}}}{f(y|\theta)^{\beta_{k-1}}} \tag{2.125}$$

  Apply the importance sampling equation:

$$c_{\beta_k} = \frac{1}{n}\sum_i f(y|\theta_i)^{\beta_k} w(\theta_i) / \frac{1}{n}\sum_i w(\theta_i) = c_{\beta_{k-1}} \sum_i \frac{f(y|\theta_i)^{\beta_k}}{f(y|\theta_i)^{\beta_{k-1}}} / \sum_i w(\theta_i) \tag{2.126}$$

$$c_{\beta_{k-1}} = \frac{1}{n}\sum_i f(y|\theta_i)^{\beta_{k-1}} w(\theta_i) / \frac{1}{n}\sum_i w(\theta_i) = c_{\beta_{k-1}} \sum_i \frac{f(y|\theta_i)^{\beta_{k-1}}}{f(y|\theta_i)^{\beta_{k-1}}} / \sum_i w(\theta_i) = nc_{\beta_{k-1}} / \sum_i w(\theta_i) \tag{2.127}$$

  where θ_i are samples of θ from $p_{\beta_{k-1}(\theta)}$. Now divide the two, we have:

$$\frac{c_{\beta_k}}{c_{\beta_{k-1}}} = \frac{1}{n}\sum_i \frac{f(y|\theta_i)^{\beta_k}}{f(y|\theta_i)^{\beta_{k-1}}} \tag{2.128}$$

2.4.1 Advanced MCMC methods

Background: Hamiltonian mechanics

- Reference: Chapter 5 of MCMC handbook. `http://www.mcmchandbook.net/HandbookChapter5.pdf`. See Reference/Stat-ML/Bayesian/.
- Hamiltonian mechanics: a dynamic system (e.g. a particle) that can be characterized by vectors (q, p) where q is position vector and p momentum vector. Let $H(q, p)$ be the Hamiltonian of the system

(energy): $H(q,p) = U(q) + K(p)$ where $U(q)$ is the potential energey and $K(p)$ the kinetic energy. One can thus obtain the description of the system in terms of H:

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i} \quad \frac{dp_i}{dt} = \frac{-\partial H}{\partial q_i} \tag{2.129}$$

Consider a single particle. The first equation says: the velocity equals the derivative of its kinetic energy with respect to its momentum. The second equation is effectively Newton's second law, where the force equals the negative gradient of potential energy. Geometrically, we can think of a system in the generalized coordinates (q_i) and generalized velocities (p_i).

- Common assumption about K function: usually kinetic energy, $K = p^T M^{-1} p/2$, where M is the diagonal matrix of mass. Or $K = \sum_i p_i^2/m_i$.
- Example: simple harmonic oscillator, $U(q) = q^2/2, K(p) = p^2/2$. The system can be described by a *phase portrait*: how (q,p) changes in the phase diagram. The solution/behavior of the system: circle in the phase diagram, where p and q both are periodic functions.
- Properties of Hamiltonian dynamics: Reversibility, Conservation of the Hamiltonian, Volume Preservation in the phase space (Liouvilles theorem).
- Numerical algorithms to solve Hamiltonian system: Euler's method, we update p and q at each step of size ϵ:

$$p_i(t+\epsilon) = p_i(t) - \epsilon \frac{\partial U}{\partial q_i}(q(t)) \qquad q_i(t+\epsilon) = q_i(t) + \epsilon \frac{p_i(t)}{m_i} \tag{2.130}$$

  where the first equation describes the change of moment because force (from potential energy change), and the second is the update of position from momentum. However, Euler's method does not preserve volume and may diverge. Modified Euler method does better. Leapfrog is even better: do update of q for a half-step, then update p for a full step, and then update q for another half-step.

Hamiltonian MC (HMC) [MCMC handbook, Chapter 5]

- Basic idea of HMC: suppose we are minimizing a function, we model it as a physical system to minimize the potential energy. The idea is that given a physical/dynamic system, let it evolve over time, it will reach the point of lowest potential energy. So we create a dynamic system whose potential energy is the objective function and take advantage of the Hamiltonian dynamics. This is generally faster than random walk. In HMC case (as opposed to minimization), we are not minimizing a function, but simulation under Hamiltonian dynamics would give the distribution one wishes to sample from.
- Intuitions of HMC: consider the landscape of the potential energy, which may have many local minimum. Imagine we have a ball that travels this landscape, to reach the mimimum, we will "release" it and follow its motion. Naturally it will reach some local minimum. To escape from the trap, we give the ball some random momentum, then it will have a chance to escape from the current minimum, and by natural dynamics reach a second minimum. If we repeat this many steps, we will reach the global minimum.
- HMC procedure: to reach a posterior defined on parameters q, we define a system with

$$U(q) = -\log[\pi(q)P(D|q)], \quad K(p) = \sum_i \frac{p_i^2}{2m_i} \tag{2.131}$$

  where m_i are parameters of the procedure (similar to step size in MCMC). To create a MC, we define

$$P(q,p) = \frac{1}{Z}\exp(-H(p,q)/T) \tag{2.132}$$

  The procedure has two steps: (1) it updates p by sampling $p_i \sim N(0, m_i)$. (2) Run the leapfrog method to update q: following the Hamiltonian dynamics. The leapfrog method is effectively numerical method of solving the PDE of Hamiltonian equations.

- Why HMC converges to the target distribution? Ex. normal distribution. Hamiltonian dynamics: given by simple Harmonic oscillator with $U(q) = q^2/2$. Given an initial p: the distribution of q is oscillation around $q = 0$. Initial value of p itself follows normal distribution. The overall results samples from the entire normal distribution: intuitively, p is usually close to 0, and q revolves around the circle defined by p; overall the entire distribution of p, q would follow normal distribution.
- Why is HMC more efficient than standard MCMC? Hamiltonian dynamics is a more efficient way of exploring parameters using gradient information. Consider one parameters case, the posterior can be thought of as surface of a bowl. At the MAP, when the posterior changes quick or large gradient (steep surface), Hamiltonian dynamics will quickly take the particle to the MAP; when the gradient is small (flat surface), the particle will move more freely and could be far from the MAP.
- More general case: given an initial p, HMC samples the landscape determined by p: usually, HMC explores local neighborhood. With small probability p is large, this allows HMC to explore distal regions.

2.5 Variational Inference

Reference: [Bishop, Chapter 10], [Murphy, Chapter 21]
Background: Calculus of Variations [Wiki]

- Problem: given a functional, a mapping from a function to $\mathbf{R}$, our goal is to optimize it, i.e. finding the function that optimizes the real value. Ex. given two points, find the curve with the minimum length.
- Euler-Lagrange Equation: consider the functional (e.g. length of curve) defined on function y and its derivative y':

$$J[y] = \int_{x_1}^{x_2} L(x, y(x), y'(x))dx \tag{2.133}$$

  Our goal is to minimize $J[y]$. We can derive conditions similar to the result in calculus (vanishing gradient). Consider a function around f, denoted as $f + \epsilon\eta$ for some arbitrary function η and small number ϵ, where $\epsilon\eta$ is called *variation*. The key idea is that at the minimum, we should have $\frac{dL}{d\epsilon} = 0$. Using the basic rule of total derivative, we have:

$$\int_{x_1}^{x_2} \eta(x) \left(\frac{\partial L}{\partial f} - \frac{d}{dx} \frac{\partial L}{\partial f'} \right) dx = 0 \tag{2.134}$$

  Since this must be held for any function $\eta(x)$, we must have:

$$\frac{\partial L}{\partial f} - \frac{d}{dx} \frac{\partial L}{\partial f'} = 0 \tag{2.135}$$

  This is called the Euler-Lagrange equation, and the left hand is called the *functional derivative* of $J[f]$. In general this gives a second-order ordinary differential equation which can be solved to obtain the extremal function $f(x)$.
- Example: given two points (x_1, y_1) and (x_2, y_2), show that the straight line minimizes the curve length connecting the two points.

Variational inference:

- Motivation: approximate a complex joint distribution. Suppose we have a distribution $p(z)$ to sample from, or to calculate the integral. Example: Bayesian posterior distribution, Ising model, etc. We want to approximate $p(z)$ by some simpler distribution: certain parameteric form, or factorizable into multiple distributions. Relation to calculus of variations: find a function q that minimizes the functional $L(q)$, the distance of q to the target distribution p.

- Minimizing KL divergence: we want to find the distribution $q(z)$ s.t. the KL divergence $D(q||p)$ is minimized:

$$D(q||p) = \int q(z) \ln \frac{q(z)}{p(z)} dz \tag{2.136}$$

Note: we minimize this instead of $D(p||q)$ because it would involve computing expectation over true distribution p, which is unknow. We define $L(q)$ as:

$$L(q) = \int q(z) \ln \frac{p(z)}{q(z)} dz \tag{2.137}$$

It is easy to see that $D(q||p) + L(q) = 0$, thus minimizing $D(q||p)$ is equivalent to maximizing $L(q)$. We write it as:

$$L(q) = \int q(z) \ln p(z) dz - \int q(z) \ln q(z) dz = \mathrm{E}_q[\ln p(Z)] + H(q) \tag{2.138}$$

The first term is called "energy" (the target distribution p fits q, imagining data is generated from q; in other words, q should generate a good sample according to p), and the second term is the entropy of q. We thus call $L(q)$ the negative free energy. Thus q should balance (1) approximate the target distribution and (2) maximizing the entropy.

- Intuition of KL divergence: $KL(q||p)$ avoids regions where $q(x)$ is large/moderate, but $p(x)$ small. $L(q) = E_q[\ln p(z)] + H(q)$, so intuitively, we want q to cover "critical regions" of $p(z)$, at the cost of sacrificing regions where $p(z)$ is small. Generally this means the approximation will be too compact comparing with the true distribution. See Bishop Figure 10.2.

- Multi-mode distributions: another example of Variational approximation. p is a mixture of two normal. Numerical algorithm usually leads to approximation of a single mode. Bishop Figure 10.3.

Mean-field approximation:

- Factorization: a common assumption is that q can be factorized:

$$q(z) = \prod_{i=1}^{M} q_i(z_i) \tag{2.139}$$

- Maximizing $L(q)$: we infer how q_j is related to other components $q_{i \neq j}$.

$$L(q) = \int q_j(z_j) \left(\ln p(z) \prod_{i \neq j} q_i(z_i) dz_{i \neq j} \right) dz_j + H(q_j) + \sum_{i \neq j} H(q_i) \tag{2.140}$$

We used the fact that the entropy of the product of distributions is the sum of entropy of each component. We define:

$$\ln \tilde{p}(z_j) = \mathrm{E}_{q_i, i \neq j}[\ln p(z)] = \int \ln p(z) \prod_{i \neq j} q_i(z_i) dz_{i \neq j} \tag{2.141}$$

Then:

$$L(q) = \int q_j(z_j) \ln \tilde{p}(z_j) dz_j - \int q_j \ln q_j(z_j) dz_j + \sum_{i \neq j} H(q_i) = - \int q_j \ln \frac{q_j(z_j)}{\tilde{p}(z_j)} dz_j + \sum_{i \neq j} H(q_i) \tag{2.142}$$

When $q_i(z_i)$ is fixed for all $i \neq j$, and only q_j is free to change, we note that the above is the negative of KL divergence between q_j and $\tilde{p}(z_j)$ and is maximized at $q_j^*(z_j) = \tilde{p}(z_j)$. Thus we have the general expression for the optimal solution:

$$\ln q_j^*(z_j) = \mathrm{E}_{q_i, i \neq j}[\ln p(z)] \tag{2.143}$$

- Intuition of the mean-field equation: wlos, we assume we only have two dimensions z_i and z_j (imagine 2D normal distribution, with j horizontal dim.). Apparently, the best approximation of z_j should be $p_j(z_j)$, marginalizing over z_i. Why don't we do that? This is often difficult, e.g. imaging Bayesian Variable Selection for regression, the marginal of $p(\beta_j)$ is the PIP, which is difficult to find. So the mean-field VB approximation equation is computing the "pseudo-marginal" over other variables, by assuming they follow the approximate distribution. Specifically, to obtain $q_j(z_j)$ at a strip near z_j: we need to marginalize the density $p(z_i, z_j)$ over all values of z_i. The density near z_i can be approximated by $q_i(z_i)dz_i$. In VB, we actually consider log-pdf, so we have:

$$p_j(z_j) = \int p(z_i, z_j)dz_i \Rightarrow \ln q_j(z_j) = \int \ln p(z_i, z_j) \cdot q_i(z_i)dz_i \tag{2.144}$$

  Eventually, after VB, $q_j(z_j)$ should converge to the true marginal of z_j.

- Procedure of variational inference: We cycle through the components $q_j(z_j)$: at each time, assume the distribution of all other components are known, and solve the optimal q_j^* according to Equation 2.143. Remark: the idea is a generalization of EM or Gibbs sampling: instead of sampling or maximization given the other variables, variational inference find the optimal distribution of one component given the distribution of the rest. See [Bishop, Figure 10.4] for an example of (μ, τ) in Gaussian distribution - very intuitive.

- Why the procedure is more tractable than sampling the joint distribution? Each time, we are dealing with distribution of 1D, and the expectation of $\ln p(z)$ over other variables can be simpler.

- Properties of factorized approximation: the main assumption of variational method is the independence assumption. Example: we could approximate a bivariate normal distribution, but the independence assumption is clearly invalid when the two components are correlated. See [Bishop, Figure 10.2].
  - Minimizing $\mathrm{KL}(q||p)$: from the integral, the integrand is large at small $p(z)$, so the solution (which tries to minimize KL) will try to avoid the regions where $p(x)$ is small, but $q(x)$ large. Thereforem, the solution will cover the modes of the true distribution, but do not expand beyond those.
  - Minimizing $\mathrm{KL}(p||q)$: the opposite situation. The solution will try to avoid regions where $p(x)$ is substantial.

Variational Bayesian: approximate the posterior distribution

- Maximizing $L(q)$: suppose we want to solve the posterior distribution $p(z|x)$, where Z represents all the unknowns (parameters and missing variables), and X are data. We first relate to the joint distribution $p(x, z)$ by:

$$\ln p(x) = \ln p(x, z) - \ln p(z|x) \tag{2.145}$$

  The above is true for any Z. Let $q(z)$ be the approximation of $p(z|x)$. We take the expectation over $q(z)$ in the equation above, and this leads to:

$$\ln p(x) = L(q) + \mathrm{KL}(q||p) \tag{2.146}$$

  where we have defined:

$$L(q) = \int q(z) \ln \frac{p(x, z)}{q(z)} dz \tag{2.147}$$

$$\mathrm{KL}(q||p) = \int q(z) \ln \frac{q(z)}{p(z|x)} dz \tag{2.148}$$

  Since $p(x)$ is independent of q, thus minimizing KL divergence is equivalent to maximizing $L(q)$. We note that $L(q)$ provides a lower bound of the marginal log likelihood $\ln p(x)$:

$$L(q) \leq \ln p(x) \tag{2.149}$$

  It is called ELBO. It can be used in several ways: (1) In VB, we should see monotonic increase of $L(q)$. (2) It provides estimate of marginal likelihood, which is useful for Bayesian selection.

- Procedure of Variational Bayesian: similar to the general case, we have the iterative equation:

$$\ln q_j^*(z_j) = \mathrm{E}_{q_i, i\neq j}[\ln p(x,z)] + \text{const} \tag{2.150}$$

where the expectation is defined as:

$$\ln \tilde{p}(x,z_j) = \mathrm{E}_{q_i, i\neq j}[\ln p(x,z)] = \int \ln p(x,z) \prod_{i\neq j} q_i(z_i) dz_{i\neq j} \tag{2.151}$$

- Direction optimization of ELBO $L(q)$ [SuSiE paper]: we can write $L(q)$ as functions of q:

$$L(q) = \mathrm{E}_q[\ln p(z)] + \mathrm{E}_q[\ln p(x|z)] - \mathrm{E}_q[\ln q(z)] \tag{2.152}$$

where the three terms correspond to expectation of prior, of likelihood, and the entropy of q. It is possible to directly optimize this function by coordinate descent: find the best q_j, supposing other $q_i, i \neq j$ are given.

Thoughts about VB:

- How does VB equation relate to conditional distribution? When VB converges, can we say that the expectation of an unknown is correct?

Example: Variational Inference of Univariate Gaussian distribution: [Bishop, 10.1.3]

- Problem: suppose $X \sim N(\mu, \tau^{-1})$, where the parameters follow conjugate prior:

$$p(\tau) = \text{Gamma}(\tau|a_0, b_0) \tag{2.153}$$

$$p(\mu|\tau) = N(\mu|\mu_0, (\lambda_0\tau)^{-1}) \tag{2.154}$$

Given the data $X_1, \cdots, X_n$, the posterior of $\tau, \mu|X$:

$$\ln p(\tau,\mu|X) = \ln p(\tau) + \ln p(\mu|\tau) + n/2 \cdot \ln\left(\frac{\tau}{2\pi}\right) - \left[\frac{\tau}{2}\sum_i (x_i-\mu)^2\right] + \text{const} \tag{2.155}$$

- Variational inference: we start with inference of μ given τ. In this case, any term in $\ln p(\tau,\mu|X)$ that does not depend on μ would not matter. We have:

$$\ln q_\mu^*(\mu) = -\frac{\mathrm{E}_\tau[\tau]}{2}\left[\lambda_0(\mu-\mu_0)^2 + \sum_i (x_i-\mu)^2\right] + \text{const} \tag{2.156}$$

We see that $q_\mu(\mu)$ has normal density with mean and variance dependent on E_τ. And

$$\ln q_\tau^*(\tau) = (a_0-1)\ln\tau - b_0\tau + \frac{n+1}{2}\ln\tau - \frac{\tau}{2}\mathrm{E}_\mu\left[\lambda_0(\mu-\mu_0)^2 + \sum_i (x_i-\mu)^2\right] + \text{const} \tag{2.157}$$

We see that $q_\tau(\tau)$ is Gamma with parameter dependent on $\mathrm{E}[\mu]$ and $\mathrm{E}[\mu^2]$. This leads to VB iteration: we determine the parameters of $q_\mu(\mu)$ (normal) and $q_\tau(\tau)$ (Gamma) using $\mathrm{E}[\tau], \mathrm{E}[\mu]$ and $\mathrm{E}[\mu^2]$ from the previous iteration. Note: Equations (10.28) and (10.29) should have $(N+1)/2$ instead of $N/2$.

- Remark: the challenge of variational inference is that when computing $q_j(z_j)$, we do not know the distribution (form) of other components. To address this, we take expectation over other parameters, then the unknown distributions of $q_i(z_i)$ $(i \neq j)$ enter into the equation in the form of some expectation, e.g. E_μ over a quadratic function of μ in the example above. As a result, we may obtain the form of $q_j(z_j)$ with some unknown constants (expectations over other parameters) - these constants can be obtained from previous iterations. It makes problem easier if we can recognize the form of $q_j(z_j)$ from its log-likelihood function, and how its parameters depend on expectation over other dimensions.

Variational Bayesian inference for linear and logistic regression [Drugowitsch, arxiv, 2013]

- Model: linear regression

$$P(y|x,w,\tau) = N(y|w^T x, \tau^{-1}) = \left(\frac{\tau}{2\pi}\right)^{1/2} \exp\left(-\frac{\tau}{2}(y-w^T x)^T (y-w^T x)\right) \tag{2.158}$$

  The prior of w and τ is conjugate normal inverse-gamma:

$$p(w,\tau|\alpha) = N(w|0,(\tau\alpha)^{-1}I) \cdot \text{Gamma}(\tau|a_0,b_0) \tag{2.159}$$

- VB update: assuming α is given, we update w and τ. We first write $\log p(w,\tau,\alpha,D)$ as (ignoring the term constant wrt. w):

$$\log p(w,\tau,\alpha,D) = \log N(w|0,(\tau\alpha)^{-1}I) - \frac{\tau}{2}(y-w^T x)^T (y-w^T x) + \text{const} \tag{2.160}$$

  This is a quadratic function of w: the first term has $\tau \mathrm{E}(\alpha) w^T w$. We can thus solve mean and variance of w in terms of $\mathrm{E}(\alpha)$ and τ. Similarly, we can show that τ would follow Gamma distribution, with parameters depend on $\mathrm{E}(\alpha)$ and mean and variance of w.

Variational Bayes EM (VBEM) [Murphy, 21.6]

- Idea of VBEM: (1) M-step: account for uncertainty of parameters, instead of MAP. (2) E-step: posterior of z_i, averaging the posterior of parameters, instead of at MAP. In practice, posterior at posterior mean of parameters.

- VBEM for GMM: let z_i be cluster label, with $z_{ik} = 1$ if sample i belongs to cluster k and 0 o/w. The probability of $z_{ik} = 1$ is π_k.

$$x_i | z_{ik} = 1 \sim N(\mu_k, \Lambda_k^{-1}) \tag{2.161}$$

  The prior probabilities of parameters $\theta = (\pi, \mu, \Lambda)$ are given by:

$$p(\theta) = \text{Dir}(\pi|\alpha_0) \prod_k N(\mu_k|m_0,(\beta_0\Lambda_k)^{-1}) \text{Wi}(\Lambda_k|L_0,\nu_0) \tag{2.162}$$

  We define $q(\theta, z) = q(\theta)\prod_i q(z_i)$ as approximate posterior of θ and z_i. Inference has two steps:

- Variational M-step: determine forms of $q(\theta)$. We first write the log-likelihood as:

$$\log p(x,z,\theta) = \log p(\pi) + \sum_k \log p(\mu_k,\Lambda_k) + \sum_i \log p(z_i|\pi) + \sum_i \sum_k z_{ik} \log N(x_i|\mu_k,\Lambda_k^{-1}) \tag{2.163}$$

  To determine $q(\pi)$ and $q(\mu_k, \Lambda_k)$, we take expectation over z_i with density $q(z_i)$. Suppose we have r_{ik} as the expectation of z_{ik} (probability of i assigned to cluster k) in the previous iteration. We can show that expectation over z_i can be expressed in terms of r_{ik}. So we have:

$$q(\pi) = \text{Dir}(\pi|\alpha) \qquad \alpha_k = \alpha_0 + N_k \qquad N_k = \sum_i r_{ik} \tag{2.164}$$

  So this is similar to standard EM (M-step). Similarly, we can show that $q(\mu_k, \Lambda_k)$ is also normal with parameters similar to what we have under EM. Ex. the mean of cluster k is weighted average of all data points assigned to k, where weights are given by r_{ik}.

- Variational E-step: determine $q(z_i)$. Problem becomes computing expectation of functions defined on θ. From the equation of $\log p(x,z,\theta)$, it is easy to see that only two terms depend on π are: $\log p(\pi) + \sum_i \log p(z_i|\pi)$. Expectation over $q(\theta)$ are given by Equation (21.128). The problem reduces to: (1) finding $\mathrm{E}(\log \pi_k)$, when we know π_k follows a given Dirichlet distribution; (2) find expectation of quadratic form $(x_i - \mu_k)^T \Lambda_k (x_i - \mu_k)$, over $q(\mu_k, \Lambda_k)$.

- Model selection: using ELBO as approximation of marginal likelihood.
- Large number of clusters with small α_0: shrink mixing weight to 0. However, the difference between mixture model and variable selection is: the total weight is fixed at 1, so small clusters get shrinked, but big clusters get bigger.

2.6 Hierarchical Model and Empirical Bayes

Modeling strategy:

- Intuition: suppose we have a baseline model, where a number of parameters are used. If there are additional structure in the data, e.g. certain objects are similar, then we could incorporate the structure by modeling the distributions of parameters. So we have an additional layer of model of parameters.
- Modeling group structure: suppose the samples can be grouped, and within each group, we have a parametric model. The parameters of the different groups may be related, e.g. the group parameters may depend on certain group-level variables. Ex. hierarchical normal model; hierarchical regression model.
- Modeling structure of parameters: parameters may have specific interpretations, and can be modeled. Ex. a parameter may represent the influence of one variable on response variable, and the influence of multiple varaibles may be similar (thus parameters should be similar).
- Fixed vs random effects: under fixed effect model, certain variables/parameters are constant across groups(complete pooling) ; under the random effect model, the group-level variable/parameter are still random, even though they share certain things in common (partial pooling). Ex. hierarchical normal model: (1) fixed-effect model, the mean of each group is constant; (2) random-effect model, the group mean is a sample from a normal distribution.
- Analysis of multi-level modeling: analyze how the individual variation can be partitioned, and if the model appropriately captures all variations. Ex. under hierarchical regression, the individual variation consists of: individual variation within the group and variation of group, and the latter consists of the group mean plus some random variation from the mean.

Example of multi-level modeling: the disease risk of individuals.

- Group-level modeling: suppose the risk depends on genetics (individual), and environmental factors such as diet (individual), pollution (region) and water quality (region). Under the multi-level modeling, one would write the risk as a function of genetics, diet and a variable for regions; then the region variable can be modeled as a function of pollution, water quality, etc.
- Parameter-level modeling: the genetics can be partitioned into many genes, however, the effect of a gene can be modeled. Suppose β_j is the effect of the i-th gene, and the gene belongs to K groups, then β_j is a function of the sum of effect of all groups the gene belongs to, plus some random variation.

Applications of multi-level modeling: [Ji & Liu, NBT, 2010]

- Motivation: the data contains additional structures in the form of groups (which could be nested), or similarity between objects; or additional determinants that may influence the group-level properties. These additional structure or factors can be treated by modeling the relevant parameters/effects. Specifically, grouping or similarity can be expressed as: the relevant parameters viewed as samples from a common distribution; the group-level effect may be determined from other group-level factors; etc.

- Benefits: data can be aggregated for inference of certain parameters: all the groups contain information of top-level parameters, which in turn change of our inference on group-specific parameters (acting like prior distributions). This benefit is stronger when the heterogeneity is small. When hierarchical modeling is applied to estimate variance, it is called "variance stabilization".
- Variance partition: similar to ANOVA (or ANOVA can be viewed as a special form of multi-levle models), the variation can be parititioned into the top-level variation (group variation), and the intrinsic variations within groups.
- Examples: (1) Differentially expressed (DE) genes: assume the variance of each gene is a sample from a top-level distribution. Important when the sample size is small, thus variance of individual genes is not robust. (2) SNPs: of the same genes/pathways may follow the same top-level distribution; or the effect of a SNP can be regressed on the properties of the SNP such as its position, cross-species conservation, etc.

2.6.1 Bayesian Hierarchical Models

Reference: [Gelman04, chapter 5]

General procedure for Bayesian hierarchical models:

- Overview: in general, we are interested in two types of problems in a hierarchical model setting:
  - Population-level parameters: average over all groups, while taking the heterogeneity across groups into account. It is thus not a simple average over all data points, e.g. two groups, but one has a much larger group variance than the other (thus should be discounted).
  - Group-level parameters: borrow information from other groups (i.e. population level parameter) to better infer group-level parameters. Ex. a group with a very small number of instances.

  The key problem is to infer the posterior distribution of the hyperparameter, $p(\tau|y)$, as: (1) it may be the objective of study; (2) it helps the inference of group parameters (θ), as once τ is known (sampled), the inference of θ is often a standard Bayesian problem.
- Model: suppose there are J groups, for the j-th group, the data $y_j, 1 \leq j \leq J$ (vector) is generated from the parameters θ_j. The parameters come from a common population distribution, parameterized by the hyperparameter(s) ϕ. We write the model as:

$$\phi \rightarrow [\theta_j \rightarrow y_j] \tag{2.165}$$

  The prior distribution:

$$p(\phi, \theta) = p(\phi)p(\theta|\phi) \tag{2.166}$$

  and the posterior distribution:

$$p(\phi, \theta|y) \propto p(\phi, \theta)p(y|\theta, \phi) = p(\phi, \theta)p(y|\theta) \tag{2.167}$$

  Note: in hierarchical model, it is important to note that the constant terms (in non-hierarchical model) can depend on the parameters (see the rat tumor example below).
- Posterior sampling: some similarlity with the case of nuisance parameters. The difference is that we are interested in both ϕ and θ; while we may want to integrate out the nuisance parameters. Typically, sampling may consists of the three steps (if marginal posterior is possible to sample, Step 3):
  - The posterior distribution: $p(\phi, \theta|y)$.
  - The conditional posterior distribution: $p(\theta|\phi, y)$, this may be reduced to a familiar case.

– The marginal posterior distribution: $p(\phi|y)$. This is often the key step. Several strategies:

$$p(\phi|y) = \int p(\phi, \theta|y) d\theta \tag{2.168}$$

Or apply the following equation to any value of θ:

$$p(\phi|y) = \frac{p(\phi, \theta|y)}{p(\theta|\phi, y)} \tag{2.169}$$

Often if the marginal likelihood is available, the inference of $p(\phi|y)$ is made easier:

$$p(\phi|y) \propto p(\phi)p(y|\phi) = p(\phi) \int p(y|\theta)p(\theta|\phi) d\theta \tag{2.170}$$

The sampling step thus involves: first sampling from $p(\phi|y)$, then $p(\theta|\phi, y)$. One could use alternate Gibbs sampling:

$$\theta \sim p(\theta|\phi, y) \tag{2.171}$$

$$\phi \sim p(\phi|\theta, y) \tag{2.172}$$

We use hierarchical linear model as an example. The first equation: within group regression, sample each group-specific effect parameter, suppose ϕ is given. This is a Bayesian regression problem with prior parameterized by ϕ. The second equation: once each group-specific effect is estimated/sampled, the between group variation and population parameters can be estimated using group-level regression. With both steps, additional Gibbs sampling may need to be performed.

- Remark: in general, the inference of hierarchical model consists of (1) inference of population level parameters $\phi|y$; (2) inference of group-level parameters: $\theta_j|\phi, y$.

Hierarchical binomial model: rat tumor experiment [Section 5.3]

- Model: the data consists of J groups, with each group of size n_j, and y_j is the number of rats survived.

$$y_j \sim \text{Bin}(n_j, \theta_j) \tag{2.173}$$

where θ_j is the mean of the j-th group, and is assumed to follow a prior of Beta distribution:

$$\theta \sim \text{Beta}(\alpha, \beta) \tag{2.174}$$

We are interested in learning the general trend of the population (over all groups).

- Inference: we first infer α and β. Using the marginal likelihood of binomial distribution:

$$p(y|\alpha, \beta) = \prod_j p(y_j|\alpha, \beta) \propto \prod_j \frac{B(\alpha + y_j, \beta + n_j - y_j)}{B(\alpha, \beta)} \tag{2.175}$$

And the posterior $p(\alpha, \beta|y) \propto p(\alpha, \beta)p(y|\alpha, \beta)$. We would use numerical method to compute this function for any value of α, β. Once we have α, β, the parameter of the j-th group:

$$\theta_j|y_j, \alpha, \beta \sim \text{Beta}(\alpha + y_j, \beta + n_j - y_j) \tag{2.176}$$

- Prior of α, β: uniform prior in the scale of $(\alpha/(\alpha + \beta), (\alpha + \beta)^{-1/2})$.

Hierarchical normal model:

- Model: consider J groups, the goal is to estimate the mean of each group. Within each group, the data points are from iid sample:

$$y_{ij}|\theta_j \sim N(\theta_j, \sigma^2) \qquad i = 1, 2, \cdots, n_j \tag{2.177}$$

  The parameters θ_j are from a common distribution:

$$\theta_j|\mu, \tau \sim N(\mu, \tau^2) \tag{2.178}$$

  where θ_j are independent. Note that the parameter τ reflects the between-group variation, if it is close to 0, then complete-pooling; if it is approaching infinity, then no pooling. By exploring the distribution of τ, one can assess the heterogeneity of groups.

- Classical approach: either complete pooling - use all population to estimate a single mean; or no pooling - estimate one mean for each group. To choose between the two, ANOVA would test if the means of groups (from no pooling) are significantly different (heterogeneity test). If significant, no pooling; otherwise, complete pooling.

- Bayesian approach:

  - The prior distribution of hyperparameters: choose noninformative prior:

$$p(\mu, \tau) \propto p(\tau) \propto 1 \tag{2.179}$$

    Note that the prior $p(\tau) \propto 1/\tau$ leads to improper posterior (so in fact, the prior is determined after some experimentation).

  - Joint posterior distribution: let $\bar{y_j}$ be the mean of group j, and $\sigma_j^2 = \sigma^2/n_j$ be the sample variance of group j, then:

$$p(\theta, \mu, \tau|y) \propto p(\mu, \tau) \prod_j N(\theta_j|\mu, \tau^2) \prod_j N(\bar{y_j}|\theta_j, \sigma_j^2) \tag{2.180}$$

  - Conditional posterior distribution given the hyperparameters $p(\theta|\mu, \tau, y)$: this is from the standard results of posterior distribution of normal means - normal distribution, with mean equal to the weighted average of $\bar{y_j}$ and μ, and precision equal to the sum of precision (of prior and of the data).

  - Marginal posterior distribution $p(\mu, \tau|y)$:

$$p(\mu, \tau|y) \propto p(\mu, \tau)p(y|\mu, \tau) \tag{2.181}$$

    This can be achieved by marginalization of θ (similar to finding the posterior predictive distribution in normal distribution).

  - Conditonal posterior distribution $p(\mu|\tau, y)$: from the equation above, $p(\mu|\tau)$ is normal (or uniform), and log of $p(y|\mu, \tau)$ is quadratic of μ, thus this is normal distribution. u

  - Marginal posterior distribution $p(\tau|y)$:

$$p(\mu, \tau|y) = p(\mu|\tau, y)p(\tau|y) \tag{2.182}$$

    The first term is a normal distribution, and the second from integrating out μ (the integrand is exponential to the quadratic of μ), or from $p(\mu, \tau|y)/p(\mu|\tau, y)$.

  - Sampling: first sample $p(\tau|y)$ (e.g. 1-D grid sampling), then sample from $p(\mu|\tau, y)$ (normal), and then $p(\theta_j|\mu, \tau, y)$ (also normal).

- Example: the ETS data, J schools, for each school, the improvement of student scores from a policy (one student per score). The goal is to estimate the treatment effect, i.e. whether the score improvement is significantly different from 0.

- $p(\tau|y)$: maximum at 0 and is very small at $\tau > 15$ points, meaning the within-group variation is small. However, still signficiant probability mass from 0 to 15, thus signficant uncertainty.
- $p(\theta_j|\tau, y)$: draw the distribution at different values of τ. At $\tau = 0$: all groups have the same mean score θ_j. The difference increases at larger τ. Obtain standard deviation or posterior quantile of θ_j, this allows one to analyze each school, e.g. for school A, the probability that the score effect is great than 28 is very small, even at large $\tau = 10$.

- Example: meta-analysis of drug effects. J trials, at each trial, the death rate of the control group, and the treatment group. Estimate the treatment effect. Let p_{1j} and p_{0j} be the death probability in group j of treatment and control respectively. For a single group, the hypothesis is to test $p_{1j} = p_{0j}$. The Bayesian approach models p_{1j} and p_{0j} as from common distributions.

- Comparison with non-Bayesian model: e.g. the ETS example [Section 5.5]. The problems with the non-Bayesian approach:
  - No pooling: estimate θ_j for each group separately, and then obtain μ by averaging over θ_j. First, θ_j estimation is not accurate (because of small sample size per group); second, the average over θ_j does not weigh groups properly.
  - Complete pooling: ignores variation across groups. Ex. suppose there is a very large group in the data, then the average (after complete pooling) is dominated by the average of this group, but this average is simply one random sapmle from the population mean.

2.6.2 Empirical Bayes

Reference: [Efron, Large-Scale Inference: Empirical Bayes methods for estimation, testing and prediction, 2010], [Casella, An Introduction to Empirical Bayes Data Analysis, Am Stat, 1985]

Bayesian point estimation:

- Strategy overview: instead of full posterior inference, we could performs point estimation, i.e. find a best value that "summarizes" the posterior distribution. This may make inference much simpler.

- Mean posterior estimator: one can find a number that best represent the posterior distribution of the unknown parameter. This could often be mean posterior (MP) estimator, or maximum a posterior (MAP) estimator. Example: normal distribution, the posterior distribution of the mean is also normal, thus the MP estimator is simply the mean of the normal distribution.

- Marginal distribution: the second main strategy for estimation. Suppose we want to infer ϕ, with extra parameter θ in the model. We could derive the marginal likelihood function of ϕ by integrating out θ:

$$P(y|\phi) = \int P(y|\theta, \phi)d\theta \tag{2.183}$$

  This is important for cases like: nuisance parameters (e.g. variance parameter in normal distribution), hierarchical model (infer the population parameters by marginaling group parameters).

- Assessing estimators: suppose we have $\hat{\theta}(X)$ as our estimator of θ, we could assess it by the mean squared error (MSE):

$$\text{MSE} = \text{E}\left(\theta - \hat{\theta}(X)\right)^2 \tag{2.184}$$

  where the expectation is taken over X over given θ. The interpretation is thus the same as MSE for classical point estimation (with fixed θ). Note that we could also define the overall Bayes risk of an estimator by averaing over the prior distribution of θ.

- Remark: the two strategies of point estimation: MP estimator and marginal distribution can be used in the same problem for different parameters. In particular, one may need to estimate one set of parameters first, and then estimate the second set of parameters conditioned on the first set of parameters.

Empirical Bayes overview:

- Learning from the experience of others: the general idea is that in large-scale inference problems (simultaneous inference of multiple unknowns), there may be "mysterious" evidence lurking among the objects, and hierarchical Bayes modeling strategy could take advantage of that. For Empirical Bayes, the prior "may exist only as a motivational device".
- Example: baseball play, Table 1.1. in [Efron10]. We are estimating the rate of hits of players. Even though the relationship between all players is far from clear, the James-Stein (JS) estimator gives better estimate of the rates than MLE.
- EB strategy: the prior distribution is estimated from the data in contrast to the standard Bayesian approach. It may be viewed as an approximation to a fully Bayesian treatment of a hierarchical model wherein the parameters at the highest level of the hierarchy are set to their most likely values, instead of being integrated out. Also known as Maximum Marginal Likelihood.

Normal hierarchical model and JS estimator:

- Model: suppose we have n observations z_i with mean μ_i, and μ_i follows a prior normal distribution:

$$\mu_i \sim N(0, A) \qquad z_i|\mu_i \sim N(\mu_i, 1) \tag{2.185}$$

  where A is a constant. Our goal is to infer μ_i, trying to leverage all the data. The posterior distribution can be found easily:

$$\mu_i|z_i \sim N(Bz_i, B) \tag{2.186}$$

  where $B = A/(A+1)$. The Bayes point estimator is thus:

$$\hat{\mu}^{\text{Bayes}} = Bz = \left(1 - \frac{1}{A+1}\right) z \tag{2.187}$$

- Evaluating an estimator by overall risk: to evaluate the estimator, we could use the mean squared error (MSE) loss function over all parameters, as in frequentist statistics. First, the loss function:

$$L(\hat{\mu}, \mu) = \sum_i (\hat{\mu}_i - \mu_i)^2 = \|\hat{\mu} - \mu\|^2 \tag{2.188}$$

  Next, write $\hat{\mu} = t(z)$ as a function of data, then the expected squared error, over all possible data (z) is:

$$R(\mu) = \mathrm{E}\left[L(\hat{\mu}, \mu)\right] = \mathrm{E}\left[\|t(z) - \mu\|^2\right] \tag{2.189}$$

  For the MLE, $\hat{\mu}^{\text{MLE}} = z$, it's easy to show that

$$R^{\text{MLE}}(\mu) = N \tag{2.190}$$

  Plug in the Bayes estimator, we can show that:

$$R^{\text{Bayes}}(\mu) = (1-B)^2\|\mu\|^2 + NB^2 \tag{2.191}$$

  The MSE of an estimator depends on the true values of parameters. We could average MSE over the prior distribution of parameters, and this gives the overall Bayes riks. It's easy to see that:

$$R^{\text{MLE}} = N \qquad R^{\text{Bayes}} = N\frac{A}{A+1} \tag{2.192}$$

  Thus the Bayes estimator is a better estimator than MLE when all parameters are estimated simultaneously.

- JS estimator: since A is generally unknown, we want to estimate A and replace it with the estimator in the Bayes estimator of μ. We note that we only need to estimate $1/(A+1)$. This could be obtained by the marginal distribution of z:

$$z_i \sim N(0, (A+1)) \tag{2.193}$$

The problem is thus estimating the variance (acutally its inverse) from normal distribution. Let S be the sample variance $\|z\|^2$, and we have:

$$\mathrm{E}\left(\frac{N-1}{S}\right) = \frac{1}{A+1} \tag{2.194}$$

JS estimator is defined as:

$$\hat{\mu}^{\mathrm{JS}} = \left(1 - \frac{N-2}{S}\right) z \tag{2.195}$$

Its overall Bayes risk is:

$$R^{\mathrm{JS}} = N\frac{A}{A+1} + \frac{2}{A+1} \tag{2.196}$$

which is slightly higher than the risk of Bayes estimator (given that the true A is known), but still below the MLE risk.

- Theorem (The superiorty of JS estimator): for $N \geq 3$, the JS estimator everywhere dominates the MLE in terms of expected total squared error for every choice of μ:

$$\mathrm{E}_{\mu}\left[\|\hat{\mu}^{\mathrm{JS}} - \mu\|^2\right] < \mathrm{E}_{\mu}\left[\|\hat{\mu}^{\mathrm{MLE}} - \mu\|^2\right] \tag{2.197}$$

Note that this result does not require any prior belief about μ, and is completely frequestist (not overall Bayes risk). To understand this theorem:

  - Note that in comparing Bayes estimator and MLE, A can be any arbitrarily large number, so the fact that we have a prior distribution of μ_i does not really matter, i.e. the superiorty should be true regardless of our prior belief.
  - When three or more unrelated parameters are measured, their total MSE can be reduced by using a combined estimator such as the JS estimator.

Problems and extensions of EB estimators:

- Tyranny of the majority: with EB approach, the estimation of an outlier, may be influenced largely by the mean of multiple related objects. This could be a problem if the object is truely an outlier (e.g. a trully distinguished baseball player). One idea to deal with this problem is to use JS estimation subject to the restriction of not deviating too far from the MLE (limited translation estimator).
- Application to the regression setting: the prior distribution of μ_i may not be simple, but depends on other parameters. Example:

$$\mu_i \sim N(M_0 + M_1 \cdot \mathrm{age}_i, A) \qquad z_i|\mu_i \sim N(\mu_i, \sigma_0^2) \tag{2.198}$$

So our estimation of μ_i is based not only on z_i, but also other players with the age effect corrected for.

False Discovery Rates, A New Deal (ASH) [Stephens, Biostatistics, 2016]

- Motivation: two main limitations of existing approaches to FDR:
  - Zero assumption: most would assume that near $p = 1$ or $Z = 0$, the observed test statistics are from H_0. This leads to overestimation of π_0. This is necessary so that the model is identifiable, e.g. in Efron's approach.

  - Different power of different tests: e.g. different MAF in GWAS or different coverage in RNA-seq. Thus p-values mean different things for different tests. Mixing them reduce the power (for high-signal tests).

- Idea: model the underlying effects as a mixture (unimodal), and including the standard error (thus taking power into account - higher powered test will have smaller standard error).

- Model: we observed $\hat{\beta}_j$, assuming it follows distribution: $N(\beta_j, s_j^2)$ where s_j is the standard error (known). So our likelihood is $L(\beta_j) \propto \exp((\hat{\beta}_j - \beta_j)^2/s_j^2)$. The prior of β_j is:

$$\beta_j \sim \pi_0 \delta_0(\cdot) + (1-\pi_0) g(\cdot) \tag{2.199}$$

  where δ_0 is Dirac's delta function and $g(\cdot)$ needs to be infered. A convenient way to model g is a mixture of normal:

$$g(\beta; \pi) = \sum_k \pi_k N(\beta; 0, \sigma_k^2) \tag{2.200}$$

  Another option is to use uniform prior:

$$g(\beta; \pi) = \sum_k \pi_k U(\beta; a_k, b_k) \tag{2.201}$$

  where $U(a_k, b_k)$ is the uniform distribution on $[a_k, b_k]$. Or more generally, $g(\beta, \pi) = \sum_k \pi_k f_k(\beta)$. In both cases, we choose a large number of components in the mixture - very dense grid. We estimate π_k and σ_k through Empirical Bayes (EM algorithm). Once we estimate these parameters, we can infer the posterior of β_j: which will be shrinked towards 0. This method is thus called "adaptive shrinkage" (ASH).

- Penalty of π_0: let $l(\pi)$ be the log-likelihood function. We add a penalty term: $h(\pi; \lambda) = \prod_{k=0}^K \pi_k^{\lambda_k - 1}$. Default option: $\lambda_k = 1$ for all $k > 0$, and $\lambda_0 = 10$. This would introduce no penalty for π_k's, but a penalty for π_0 to encourage a large π_0. This is motivated by Dirichlet density, though we do not explicitly use prior of π.

- Inference: we maximize the penalized log-likelihood:

$$l(\pi) = \log P(\hat{\beta}|s, \pi) + \log \pi_k^{\lambda_k - 1} \tag{2.202}$$

  Next we describe the marginal likelihood:

$$p(\hat{\beta}_j|s, \pi) = \sum_{k=0}^K \pi_k \tilde{f}_k(\hat{\beta}_j) \tag{2.203}$$

  where

$$\tilde{f}_k(\hat{\beta}_j) = \int_{\beta_j} f_k(\beta_j) N(\hat{\beta}_j|\beta_j, s_j^2) d\beta_j \tag{2.204}$$

  When we use normal prior, we have:

$$\tilde{f}_k(\hat{\beta}_j) = N(\hat{\beta}_j|0, s_j^2 + \sigma_k^2) \tag{2.205}$$

  When we use uniform prior, we can show that the marginal is CDF of normal (integration of normal density over an interval):

$$\tilde{f}_k(\hat{\beta}_j) = \frac{1}{b_k - a_k} \left[\Psi\left(\frac{\hat{\beta}_j - a_k}{s_j}\right) - \Psi\left(\frac{\hat{\beta}_j - b_k}{s_j}\right) \right] \tag{2.206}$$

  Maximization is simple: the function is convex, so we can use convex optimization method (e.g. Interior Point method) or EM. The EM alternates between two steps: let ϕ_{jk} be the posterior probability that data point j belongs to the k-th component, and L_{jk} be the probability data point j is from component k, then:

  - E-step: $\phi_{jk} = \pi_k L_{jk} / \sum_l \pi_l L_{jl}$.
  - M-step: $\pi_k = \sum_j \phi_{jk}/p$, where p is the number of tests.

- Application to the estimation problem: when we only estimate β_j, we may not need the special component $\pi_0\delta_0$. Or more generally, we can view this as a special case of normal or uniform prior.

- Model behavior: intuition of the shrinkage effect. Suppose we have a large $\hat{\beta}_j$, and population mean at 0. The estimate of β_j will be shrinked towards 0: how much it will be shrinked, i.e. the weights of data ($\hat{\beta}_j$) and prior, depends on the local density of $g(\cdot)$. If this is large near $\hat{\beta}_j$, then we will have small shrinkage. To apply this to the uniform prior case, if many intervals contain $\hat{\beta}_j$, then the prior density is high in the neighborhood, and we will small degree of shrinkage.

  - Remark: The problem of unstable estimate of π: when the number of components is large, it is probably hard to get a stable estimate of π, especially, π_k of adjacent components would be unidentifiable. This is likely not a main problem, however, as our goal is to obtain the shape of $g(\cdot)$ and the posterior of β_j, which should not be sensitive to the exact values of π.

- Local false sign rate (lfsr): when most of the tests are not null, FDR will be small, even for those observations with p-value ≈ 1. The reason is that FDR cannot distinguish $\beta = 0$ and β "very small". The solution is to use local FSR: the probability that we mis-claim the sign of the effect size. This is based on the posterior of β. For example, suppose $P(\beta > 0|\hat{\beta}) = 0.9$ is large, we would claim that it is positive, then our probability of being wrong is $P(\beta \leq 0|\hat{\beta}) = 0.1$. More generally, our rule is to choose:

$$\max\{P(\beta > 0|\hat{\beta}), P(\beta < 0|\hat{\beta})\} \tag{2.207}$$

  Our probability of error is thus the minimum of the two possible errors:

$$lfsr_r := \min\{P(\beta \leq 0|\hat{\beta}), P(\beta \geq 0|\hat{\beta})\} \tag{2.208}$$

  The case where we decide to choose $P(\beta = 0|\hat{\beta})$ is just LFDR. However, FDR can be underestimated (as explained previously, e.g. most alternative models have small, but non-zero effects). We have: $lfsr_j \geq lfdr_j$. FSR is more robust to model specificiations than FDR. We can also define aggregate measure of FSR at a certain thresold of lfsr, just as in the Bayesian FDR case.

- ASH has different behavior than current methods `qvalue` or `locfdr` (Figure 1): a simulation with only true effects (mean 0), FDR methods produce a hole in the H_1 component because it assumes Z-scores near 0 are from H_0. ASH does not have this behavior, because it has explicit alternative distribution with mean 0.

- Estimation of π_0 by ASH: simulation under different $g(\cdot)$, e.g. spiky, flat or bimodal (Figure 2A). π_0 estimates of ASH are generally conservative, except the case of bi-modal (Figure 2B). They are more accurate than FDR methods, which are too conservative.

- LFSR is less conservative than LFDR (Figure 2C): FDR methods are sensitive to π_0, since π_0 tends to be significantly overestimated, LFDR can be over-estimated as well. LFSR is a better metric: it is much closer to true LFSR.

- Estimation of $g(\cdot)$: "post-selection" interval estimate is extremely desirable, however, it is hard to achieve this with frequentist paradigm.

- Calibration of posterior interval: generally close to 0.95.

- Remark/Questions:

  - How the model is sensitive to the parameter λ_0? The value of 10 seems arbitrary. Comparing this way of estimating π_0 vs. William Wen's model, which uses the fact that BF under H_0 has expectation 1.

– Generalize to other likelihood model: e.g. in RVAT, we are testing if $\sigma = 0$. Or the data has some dependence, e.g. HMM or LD.

– Multi-parameter cases: e.g. suppose we analyze two GWAS traits simultaneously, we need a prior for $P(\beta_{1j}, \beta_{2j})$ for two traits.

Chapter 3

Basic Probabilistic Methods

3.1 Multivariate Normal Distribution

3.1.1 Properties of Multivariate Normal Distribution (MVN)

Reference: [Bishop, Pattern Recognition and Machine Learning, 2.3]

Definition of multivariate normal distribution:

- Background: Gaussian integral at 1D is given by:

$$\int_{-\infty}^{+\infty} e^{-x^2} dx = \sqrt{\pi} \tag{3.1}$$

  The proof follows variable substitution (using polar coordinates). Its general form in n-D is:

$$\int \exp\left(-\frac{1}{2} x^T A x\right) dx = \sqrt{\frac{(2\pi)^n}{\det A}} \tag{3.2}$$

- Definition: a random vector $X = (X_1, \cdots, X_D)$ follows $N(\mu, \Sigma)$ if:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{\det \Sigma^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)\right] \tag{3.3}$$

  where x is a column vector and Σ is a (symmetric) positive definite matrix, i.e. all eigenvalues of Σ are positive (otherwise, the PDF is not properly normalized). The exponent is the quadratic form. So it is equivalent to say any distribution in the above form, where the exponent can be written in quadratic form (with both quadratic and linear terms) is a Gaussian distribution.

- Interpretations: the Gaussian distribution follows from the average of multiple i.i.d. random variables according to the Central Limit Theorem. In addition, it is the distribution that maximizes the entropy among all continuous random variable with finite first and second moments (could be multivariate):

$$H(X) = -\int p(x) \ln p(x) \tag{3.4}$$

Representation and interpretation of multivariate Gaussian distribution:

- Diagonalization: the matrix Σ is real and symmetric, thus according to the Spectral Theorem, Σ has the Eigen Decomposition. Let $\mathbf{u_i}$ be the i-th eigenvector of Σ, and λ_i be the eigenvalue, then:

$$\Sigma = U^T \cdot \mathrm{diag}(\lambda_1, \cdots, \lambda_D) \cdot U \tag{3.5}$$

where U is a matrix whose rows are given by the vector $\mathbf{u_i}^T$. Plug in this equation into the quadratic form, and let $\mathbf{y} = U(\mathbf{x} - \mu)$, we have:

$$(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) = \sum_i y_i^2 / \lambda_i \tag{3.6}$$

- Interpretation: this equation implies that the random variable $\mathbf{y}$ follows D independent Gaussian distribution in each dimension, where the dimensions are defined by the orthogonal eigenvectors of Σ. The PDF of $\mathbf{y}$ is given by the theorem of variable transformation (Jacobian is equal to 1 since U is an orthogonal matrix):

$$p(\mathbf{y}) = p(\mathbf{x}) \det J = \prod_{j=1}^{D} \frac{1}{(2\pi\lambda_j)^{1/2}} \exp\left(-\frac{y_j^2}{2\lambda_j}\right) \tag{3.7}$$

  Thus λ_j is the covariance of Y_j. The contour plot of $\mathbf{x}$ is an ellipse centered on μ in 2D, with two axis given by y_1 and y_2. When Σ is diagonal, the axis are parallel to the axis of $\mathbf{x}$ (Figure 2.7).

- Alternative interpretation: suppose, suppose x follows MVN with covariance Σ. We can think view x as linear combination of independent RVs, i.e. the covariance between x is due to the linear combination. Ex. if z_1, z_2 are independent with variance σ_1^2 and σ_2^2, then $x_1 = z_1 + z_2$ and $x_2 = z_1 - z_2$ are generally not independent:

$$\text{Cov}\,(x_1, x_2) = \text{Cov}\,(z_1 + z_2, z_1 - z_2) = \sigma_1^2 - \sigma_2^2 \tag{3.8}$$

  Formally, we can write $x = U^T y$, where U is the matrix in the EVD of Σ and y independent RVs. A special case is the result used in sampling MVN: if Z_i iid $N(0, 1)$, and A is the Cholesky decomposition of Σ, then $X = AZ$ follows $N(0, \Sigma)$.

- Regression perspective: MVN can also be viewed from regression. We can treat one variable, say x_1 as response variable. The fact that x_1 corrleates with other variables means that we can view x_1 as a linear model of other variables.

- Relation to factor analysis: the idea of writing MVN as linear combination of independent RVs is similar to PCA and factor analysis in general.

Moments of multivarirate Gaussian distribution:

- Normalization constant: the integral of the distribution in the $\mathbf{y}$ coordinate system is 1 by multiplying the integral along each dimension (using Gaussian integral in each dimension).

- Expectation: replace $\mathbf{z} = \mathbf{x} - \mu$:

$$E[\mathbf{x}] = \frac{1}{(2\pi)^{D/2}} \frac{1}{\det \Sigma^{1/2}} \int \exp\left[-\frac{1}{2}\mathbf{z}^T \Sigma^{-1} \mathbf{z}\right] (\mathbf{z} + \mu) d\mathbf{z} \tag{3.9}$$

  The term $\mathbf{z}$ in $\mathbf{z} + \mu$ vanishes because of symmetry, and so: $E[\mathbf{x}] = \mu$.

- Covariance matrix: defined as the following matrix:

$$\text{Cov}(X) = E[(\mathbf{x} - E(\mathbf{x}))(\mathbf{x} - E(\mathbf{x}))^T] \tag{3.10}$$

  We need to compute $E(\mathbf{x}\mathbf{x}^T)$. Let $\mathbf{z} = \mathbf{x} - \mu$, and plug in to the equation of $E(\mathbf{x}\mathbf{x}^T)$, we have:

$$E(\mathbf{x}\mathbf{x}^T) = E(\mathbf{z}\mathbf{z}^T) + \mu\mu^T \tag{3.11}$$

  The first term can be computed using diagonalization of Σ: $\mathbf{y} = U\mathbf{z}$:

$$E(\mathbf{z}\mathbf{z}^T) = \frac{1}{(2\pi)^{D/2}} \frac{1}{\det \Sigma^{1/2}} \sum_{i,j} \mathbf{u_i}\mathbf{u_j}^T \int \exp\left(-\sum_k \frac{y_k^2}{2\lambda_k}\right) y_i y_j d\mathbf{y} = \sum_i \mathbf{u_i}\mathbf{u_i}^T \lambda_i = \Sigma \tag{3.12}$$

  where the $i \neq j$ terms vanish because of symmetry. So we have the covariance matrix as: $\text{Cov}(X) = \Sigma$.

Linear normal distributions: based on [Bishop] Eq 2.113-2.117, but with different notations.

- Property 1: if $y|x \sim N(x, \Sigma)$ and $x \sim N(\mu, \Lambda)$, we have: $y \sim N(\mu, \Lambda + \Sigma)$.
- Property 2: if $y|x \sim N(Ax + b, \Sigma)$ for a matrix A, and $x \sim N(\mu, \Lambda)$, the marginal distribution is: $y \sim N(Ax + b, A\Lambda A^T + \Sigma)$, and the conditional distribution is:

$$x|y \sim N(V(A^T\Sigma^{-1}(y - b) + \Lambda^{-1}\mu), V), \qquad V = (\Lambda^{-1} + A^T\Sigma^{-1}A)^{-1} \tag{3.13}$$

  An important special case is: if $x \sim N(\mu, \Sigma)$, then for a matrix A, we have $Ax \sim N(A\mu, A\Sigma A^T)$. Note: this can be easily proved using the fact that: $\mathrm{Var}\,(Ax) = A \cdot \mathrm{Var}\,(x) \cdot A^T$.

- Remark: the interpretation of the marginal distribution, the variance of y has two components: (1) Σ, which is given by y itself (when x is given), and (2) $A\Lambda A^T$, which is introduced by x: the variance of x, Λ is scaled by A.

Marginal and conditional distributions: a joint distribution $N(\mu, \Sigma)$ with precision matrix $\Lambda = \Sigma^{-1}$, and $\mathbf{x} = (\mathbf{x_a}, \mathbf{x_b})^T$ be some partition of all dimensions.

- "Completing the square" technique: we could write the exponent of a general Gaussian distribution as:

$$-\frac{1}{2}(\mathbf{x} - \mu)^T\Sigma^{-1}(\mathbf{x} - \mu) = -\frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x} + \mathbf{x}^T\Sigma^{-1}\mu + \text{const} \tag{3.14}$$

  Thus the second order term in $\mathbf{x}$ has coefficient Σ^{-1}, and the linear term in $\mathbf{x}$ has coefficient $\Sigma^{-1}\mu$. For a distribution of interest, write its first and second order terms, and this would allow one to determine mean and covariance.

- Relationship between covariance and precision matrix:

$$\Sigma_{aa} = (\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})^{-1} \tag{3.15}$$

- Joint $\to$ marginal distribution: $\mathbf{x_a}$ follows Gaussian distribution $N(\mu_a, \Sigma_{aa})$ where Σ_{aa} is the corresponding submatrix of Σ.
- Joint $\to$ conditional distributions: expand the exponent of the Gaussian distribution into $\mathbf{a}$ and $\mathbf{b}$ parts, and apply the "completing the square" technique. This leads to $\mathbf{x_a}|\mathbf{x_b}$ follows normal distribution with:

$$\mu_{a|b} = \mu_a - \Lambda_{aa}^{-1}\Lambda_{ab}(\mathbf{x_b} - \mu_b) \tag{3.16}$$

$$\Lambda_{a|b} = \Lambda_{aa}^{-1} \tag{3.17}$$

  Alternative, state in terms of covariance matrix:

$$\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(\mathbf{x_b} - \mu_b) \tag{3.18}$$

$$\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba} \tag{3.19}$$

  Note: $\mu_{a|b}$ is linear function of $\mathbf{x_b}$, and the covariance matrix is independent of the value of $\mathbf{x_b}$ (Figure 2.9).

- Conditional $\to$ joint distribution: suppose we have:

$$p(x) = N(x|\mu, \Lambda^{-1}) \tag{3.20}$$

$$p(y|x) = N(y|Ax + b, L^{-1}) \tag{3.21}$$

We have the joint distribution:

$$\begin{pmatrix} x \\ y \end{pmatrix} \sim N\left(\begin{pmatrix} \mu \\ A\mu + b \end{pmatrix}, \Sigma\right) \tag{3.22}$$

where

$$\Sigma = \begin{pmatrix} \Lambda^{-1} & \Lambda^{-1}A^T \\ A\Lambda^{-1} & L^{-1} + A\Lambda^{-1}A^T \end{pmatrix} \tag{3.23}$$

See [Gelman04] (Apppendix A) for Σ in terms of covariance matrix. The margininal distribution of $\mathbf{y}$ (also known as compound distribution) is given by:

$$p(y) = N(y|A\mu + \mathbf{b}, L^{-1} + A\Lambda^{-1}A^T) \tag{3.24}$$

The conditional distribution $x|y$ is given by:

$$p(x|y) = N(x|\Sigma_{x|y}\left[A^T L(y-b) + \Lambda\mu\right], \Sigma_{x|y}) \tag{3.25}$$

where

$$\Sigma_{x|y} = (\Lambda + A^T LA)^{-1} \tag{3.26}$$

- A special case of the linear model: if $\mathbf{x}$ is $N(\mu, \Sigma)$, and $\mathbf{y} = A\mathbf{x} + \mathbf{b}$, then $\mathbf{y} \sim N(A\mu + \mathbf{b}, A\Sigma A^T)$.

Bivariate normal distribution: [KNNL, chapter 2]

- Probability density function: consider the normal distribution with mean (μ_1, μ_2) and covariance matrix:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \tag{3.27}$$

Its pdf. is given by:

$$f(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left\{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2 - 2\rho\left(\frac{x_1-\mu_1}{\sigma_1}\right)\left(\frac{x_2-\mu_2}{\sigma_2}\right) + \left(\frac{x_2-\mu_2}{\sigma_2}\right)^2\right]\right\} \tag{3.28}$$

The parameter ρ is the correlation coefficient between X_1 and X_2: let $\sigma_{12} = \mathrm{Cov}(X_1, X_2)$, then

$$\rho = \frac{\sigma_{12}}{\sigma_1\sigma_2} \tag{3.29}$$

- Conditional probability distribution $X_2|X_1$: this is normal distribution with mean $\alpha_{2|1} + \beta_{2|1}X_1$ and standard deviation $\sigma_{2|1}$ with:

$$\begin{array}{ll} \alpha_{2|1} & = \mu_2 - \mu_1\rho\sigma_2/\sigma_1 \\ \beta_{2|1} & = \rho\sigma_2/\sigma_1 \\ \sigma_{2|1}^2 & = \sigma_2^2(1-\rho^2) \end{array} \tag{3.30}$$

- Parameter estimation: the MLE of ρ is:

$$r = \frac{\sum(X_{1i} - \bar{X}_1)(X_{2i} - \bar{X}_2)}{\sqrt{\sum(X_{1i} - \bar{X}_1)^2 \sum(X_{2i} - \bar{X}_2)^2}} \tag{3.31}$$

3.1.2 Inference of MVN

Sample variance, sample covariance and geometrical interpreations:

- Sample variance of one random variable: Suppose X is a random variable with sample $x_1, \cdots, x_n$, then:

$$\hat{\text{Var}}(X) = \frac{1}{n-1}\sum_i (x_i - \bar{x})^2 = \frac{1}{n-1}(\mathbf{x} - \bar{x})^T(\mathbf{x} - \bar{x}) \tag{3.32}$$

  It is the norm of the vector $\mathbf{x} - \bar{x}$ in the n-dim. space (up to the constant $1/(n-1)$). If X is centered, then we have the simple relation: the sample variance of X is simply the norm of the sample vector in the Eucledian space (up to the constant).

- Sample covariance between two RVs: Suppose X and Y are two random variables, with sample $x_1, \cdots, x_n$ and $y_1, \cdots, y_n$, respectively, then:

$$\hat{\text{Cov}}(X, Y) = \frac{1}{n-1}\sum_i (x_i - \bar{x})(y_i - \bar{y}) = \frac{1}{n-1}(\mathbf{x} - \bar{x})^T(\mathbf{y} - \bar{y}) \tag{3.33}$$

  It is the inner product of the $\mathbf{x} - \bar{x}$ and $\mathbf{y} - \bar{y}$ (up to constant). Similarly, if X and Y are centered, we have: the sample covariance is simply the dot product of the two sample vectors in Eucledian space. We could write the inner product as:

$$\hat{\text{Cov}}(X, Y) = \|\mathbf{x} - \bar{x}\|\|\mathbf{y} - \bar{y}\|\cos(\theta) = \sqrt{\hat{\text{Var}}(X)\hat{\text{Var}}(Y)}\cos(\theta) \tag{3.34}$$

  Thus the angle between the two sample vectors is simply the (sample) correlation coefficient of the two random variables.

- Sum of random variables: suppose $X = Y + Z$ where Y and Z are independent RVs, then $\text{Var}(X) = \text{Var}(Y) + \text{Var}(Z)$, in terms of sample variance, we have:

$$S_X = S_Y + S_Z \tag{3.35}$$

  where S_X, S_Y and S_Z are sample variance of X, Y and Z respectively. Geometrically, since sample variance is the norm of the vector representing the RV, this is essentially the Pythagorean Theorem, since Y and Z are independent (thus the two vectors are orthogonal).

- Sample covariance matrix (sample covariance of random vector): given a $N \times p$ data matrix, let $\mathbf{x_i}$ be the i-th data point (row vector) and X_j be the j-th random variable (j-th column). The covariance between X_j and X_k is given by:

$$\hat{\text{Cov}}(X_j, X_j) = \frac{1}{N-1}(X_j - \bar{X_j})^T(X_k - \bar{X_k}) \tag{3.36}$$

  where $\bar{X_j}$, $\bar{X_k}$ are the sample means. The sample covariance matrix is thus given by:

$$\hat{\text{Cov}}(X) = \frac{1}{N-1}\begin{pmatrix} X_1^T - \bar{X_1} \\ X_2^T - \bar{X_2} \\ \cdots \\ X_p^T - \bar{X_p} \end{pmatrix} \cdot (X_1 - \bar{X_1} \cdots X_p - \bar{X_p}) = \frac{1}{N-1}(X - \bar{x})^T(X - \bar{x}) \tag{3.37}$$

  where $\bar{x} = (\bar{X_1}, \cdots, \bar{X_p})$ is the mean vector. The proof simply follows the results for two RVs. When the matrix is standardized, we have sample mean is 0 for every variable, thus:

$$\hat{\text{Cov}}(X) = \frac{1}{N-1}X^T X \tag{3.38}$$

- Sample covariance matrix in terms of data points: using the equation above, but write the data matrix in terms of data points:

$$\hat{\mathrm{Cov}}(X) = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x_i} - \bar{\mathbf{x}})^{\mathbf{T}}(\mathbf{x_i} - \bar{\mathbf{x}}) \tag{3.39}$$

  The interpretation is simple, the basic equation of S applies to sample of any data points, suppose we have only 1 data point, we would have an estimate of Σ, now with n data points, we simply take the average.

MOM estimators:

- Simiilar to the UNV case, we have sample mean and sample covariance as unbiased estimators:

$$\mathrm{E}(\bar{X}) = \mu \qquad \mathrm{E}(S) = \Sigma \tag{3.40}$$

- Proof: show that the expectation of the jk-th element of the sample covariance matrix is equal to Σ_{jk}, using the results from univariate (for diagonal terms) and bivariate normal distributions (non-diagonal terms).

ML parameter estimation: [Matrix calculus and MLE for the multivariate Normal, Berkeley CS 281A notes]

- Log-likelihood function: Given a sample consisting of n independent observations $\mathbf{x_1}, \cdots, \mathbf{x_n}$ of d-dimensional MVN $N(\mu, \Sigma)$, the log-likelihood function is given by:

$$l(\mu, \Sigma) = \log P(\mathbf{x}|\mu, \Sigma) = -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log|\Sigma| - \frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^T\Sigma^{-1}(x_i - \mu) \tag{3.41}$$

  where $|.|$ is the determinant. The last term can be rewritten by using the invariance of cyclic permutations of matrix trace, we have:

$$l(\mu, \Sigma) = \log P(\mathbf{x}|\mu, \Sigma) = -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log|\Sigma| - \frac{n}{2}\mathrm{tr}(\Sigma^{-1}S) \tag{3.42}$$

  where S is the sample covariance matrix:

$$S = \frac{1}{N}\sum_{i}(x_i - \mu)(x_i - \mu)^T \tag{3.43}$$

- MLE of μ: take the derivative wrt. μ and apply the result of the derivative of the quadratic form:

$$\begin{array}{rcl} \frac{\partial l}{\partial \mu} & = & -\frac{1}{2}\sum_i \frac{\partial}{\partial \mu}\left[(x_i - \mu)^T\Sigma^{-1}(x_i - \mu)\right] \\ & = & -\frac{1}{2}\sum_i (\mu - x_i)^T 2\Sigma^{-1} \end{array} \tag{3.44}$$

  Solving the equation:

$$\sum_i (\mu - x_i)^T\Sigma^{-1} = 0 \tag{3.45}$$

  We have:

$$\hat{\mu} = \frac{1}{n}\sum_i x_i \tag{3.46}$$

  The unbiased estimator would replace n by $n-1$.

- MLE of Σ: take the derivative wrt. Σ^{-1} of the second form of log-likelihood above (the trace form), and use the results of matrix derivatives (the derivative of $\mathrm{tr}\,(AB)$ and of $\log|A|$):

$$\frac{\partial l}{\partial \Sigma^{-1}} = \frac{n}{2}\Sigma - \frac{nS}{2} \tag{3.47}$$

  Thus we have:

$$\hat{\Sigma} = S = \frac{1}{n}\sum_i (x_i - \hat{\mu})(x_i - \hat{\mu})^T \tag{3.48}$$

- Remark: in MLE, the scale constant is n^{-1} vs. the MOM estimators, the constants is $(n-1)^{-1}$.

Distribution of sample mean and sample covariance: [Manchester MT3732 class notes; Anderson, An Introduction to Multivariate Statistical Analysis, 3ed]

- Distribution of sample mean: $\bar{X} \sim N(\mu, \Sigma/n)$.
  Proof: to obtain the covariance matrix of μ_j, we show the general result: if X and Y are two MVN variables with iid. $N(\mu, \Sigma)$, then we have $\mathrm{Cov}\,(X+Y) = 2\Sigma$. Apply this result to the variance of $\bar{X}$.

- Wishart distribution: given $X_i, 1 \le i \le n$ iid. p-dim. MVN $N(0, \Sigma)$, let X be the $n \times p$ data matrix, and $M = X^T X$ be $p \times p$ matrix. Then M has Wishart distribution with scale matrix Σ and dof. n:

$$M \sim W_p(\Sigma, n) \tag{3.49}$$

  The Wishart distribution has the following properties:

$$\mathrm{E}\,(M) = n\Sigma \tag{3.50}$$

$$\mathrm{Var}\,(M_{ij}) = n(\Sigma_{ij}^2 + \Sigma_{ii}\Sigma_{jj}) \tag{3.51}$$

  When $p = 1$ and $\Sigma = 1$, this is the χ^2 distribution with dof. equal to 1.

- Distribution of sample covariance: for MVN $N(\mu, \Sigma)$, the scaled sample covariance matrix follows Wishart distribution:

$$(n-1)S \sim W_p(\Sigma, n-1) \tag{3.52}$$

  Furthermore, $\bar{X}$ and S are independent.

Testing parameters:

- Testing the mean of individual variable: suppose we want to test $H_0 : \mu_j = \mu_{j0}$. From the distribution of $\bar{X}$, we could obtain the marginal distribution of $\bar{X_j} \sim N(\mu_j, \sigma_j^2)$, where σ_j^2 is the variance of X_j. Replacing the variance with sample variance, we could construct the test statistic:

$$T_j = \sqrt{n}(\bar{X_j} - \mu_{j0})/\hat{\sigma_j} \tag{3.53}$$

  It follows t_{n-1} distribution under H_0.

- Testing the mean of MVN: suppose we want to test $H_0 : \mu = \mu_0$. The intuition is that we form a test statistic at each of the p dimensions, measuring the departure from μ_0, and add them together (squared from, since each individual statistic may be signed). This is easy when X_j's are orthogonal, so we use the EVD of the MVN distribution. Let $y = U(X - \mu)$, under $H_0 : \mu = \mu_0 \Rightarrow E(y) = 0$, thus we are testing if the mean of y is 0. We form this test statistic:

$$T^2 = n\sum_j \frac{\bar{y}_j^2}{\hat{\sigma}_j^2} \tag{3.54}$$

To express T in the orignal space, we use the Equation 3.6:

$$T^2 = n(\bar{x} - \mu_0)^T W^{-1} (\bar{x} - \mu_0) \tag{3.55}$$

where W is the sample covariance matrix (in place of Σ, which is unknown), and given by:

$$W = \frac{1}{n-1} \sum_i (x_i - \bar{x})(x_i - \bar{x})^T \tag{3.56}$$

T^2 under H_0 follows Hotelling's T^2 distribution, thus this test is called Hotelling's T^2 test.

3.1.3 Applications of MVN

Sampling from multivariate normal distribution (MVN):

- Theorem: d iid. random variables, $Z_i \sim N(0,1)$. Let $Z = (Z_1, \cdots, Z_d)^T$, and

$$X = \mu + AZ \tag{3.57}$$

where A is the Cholesky decomposition of the matrix Σ, $AA^T = \Sigma$. Then $X \sim N(\mu, \Sigma)$.

Projection of data matrix on a direction:

- Suppose v is a unit vector in p-dim space, the projection of X on v can be written as:

$$Xv = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} v = \begin{pmatrix} x_1 v \\ \vdots \\ x_n v \end{pmatrix} \tag{3.58}$$

Let Σ be the sample covariance matrix of X, then the sample variance of the vector Xv is:

$$\text{Var}\,(Xv) = \frac{1}{N}(Xv - \bar{x}v)^T (Xv - \bar{x}v) = \frac{1}{N}[(X - \bar{x})v]^T (X - \bar{x})v = \frac{1}{N} v^T \hat{\Sigma} v \tag{3.59}$$

3.2 Categorical and Count Data

Poisson ASH [Mengyin Lu thesis, 2018]

- Motivation: scRNA-seq data, estimate distribution of expression for each gene. Use ASH prior: mixture of uniform distributions can approximate any unimodal distribution.
- Model: let Y_{cg} be expression of gene g of cell c. Let α_c be the scaling factor of cell c,

$$Y_{cg} \sim \text{Pois}(\alpha_c \lambda_{cg}) \qquad \lambda_{cg} \sim G_g(\cdot) = \pi_g \delta_0 + (1 - \pi_g) H_g \tag{3.60}$$

where π_g captures zero-inflation, and H_g is the distribution. Use ASH for H_g: a mixture of uniform distributions. Note: do not use log-link function for λ_{cg}, not stable. Possible explanation: many cells with very low expression, log. expression would be very small (negative), thus need a large number of grids.

3.2.1 Contingency Tables

Exact tests of categorical data: [Hartl, Principles of Population Genetics, Section 2.3]

- Discrete test statistic: suppose the test statistic is a discrete RV, T, with probabilities p_i for the value a_i (un-ordered), what is the appropriate transformation that computes the p value of T? The solution: rank all i's by the value of p_i (ascending order), and the p value of T is the sum of all p_i's below T.
- Exact test of sample configuration: suppose we are testing a count table, we call the counts at each cell (possible values of combinations of variables) as a sample configurations. Ex. in HWE test, the counts of AA, Aa, aa are a sample configuration. The probability of obtaining any sample configuration under H_0 can be computed, and this allows one to calculate the p value of any observed sample configuration.
- Applications: in population genetics, test HWE of allele frequencies, or LD between two loci.

χ^2 test of multinomial distribution and tables: [Rice, Mathematical Statistics and Data Analysis]

- Testing multinomial distribution: suppose we have k categories (could be k cells in a table), with the count $(X_1, \cdots, X_m) \sim \text{MN}(p_1, \cdots, p_m)$. We are testing the hypothesis $H_0 : p_1(\theta), \cdots, p_m(\theta)$, where θ is k-dim. parameter, $k < m$, against $H_1 : p_1, \cdots, p_m$, i.e. m free parameters.
- Likelihood ratio χ^2 test: also called G-test. Let $p_1(\hat{\theta}), \cdots, p_m(\hat{\theta})$ be the MLE under H_0, and $\hat{p}_i = x_i/n$ be the MLE under H_1. We form the LRT:

$$-2\log\lambda = 2\sum_{i=1}^{m} x_i \log\frac{\hat{p}_i}{p_i(\hat{\theta})} = 2\sum_{i=1}^{m} O_i \log\frac{O_i}{E_i} \tag{3.61}$$

  where $O_i = n\hat{p}_i$ is the observed count at the i-th cell, and $E_i = np_i(\hat{\theta})$ is the expected count (under H_0). At large sample size (x_i generally greater than 5), the test follows χ^2 distribution with dof equal to $m - k$.
- Pearson's χ^2 test: defined as

$$\chi^2 = \sum_{i=1}^{m} \frac{(O_i - E_i)^2}{E_i} \tag{3.62}$$

  This similarly follows χ^2 distribution with dof. $m-k$. We can show the test is aympotically equivalent to LRT by expanding $-2\log\lambda$ near $p_i(\hat{\theta})$ (easier to calculate, but LRT is recommended).
- Multiple Binomial test: suppose we have m independent binomial distributions, with $X_i \sim \text{Bin}(N_i, p_i)$. We are testing $H_0 : p_i(\theta)$ against $H_1 : p_i, 1 \le i \le m$. We could similarly form the LRT:

$$-2\log\lambda = 2\sum_{i=1}^{m}\left[x_i \log\frac{\hat{p}_i}{p_i(\hat{\theta})} + (N_i - x_i)\frac{1-\hat{p}_i}{1-p_i(\hat{\theta})}\right] \tag{3.63}$$

  Using similar notations of O_i and E_i:

$$-2\log\lambda = 2\sum_{i=1}^{m}\left[O_i \log\frac{O_i}{E_i} + (N_i - O_i)\log\frac{N_i - O_i}{N_i - E_i}\right] \tag{3.64}$$

McNemar's test: [Sprent, Applied Nonparametric Statistical Methods]

- Example: (Section 5.2) climbing records of two rocks: number of successes and failures in two rocks respectively, and want to know if one is harder than the other. Since if a climber succeeds or fails in both rocks, no information is provided for the relative difficulty, only the diagoal cells provide information, and need to be considered.

- McNemar's test: give a 2-by-2 table, and we want to test if the diagonal cell counts are equal. Suppose the cell counts are a, b, c, d where b, c are diagonal counts. Our null hypothesis is $H_0 : p_b = p_c$, where p_b or p_c is the probability of cells. The McNemar's test is given by:

$$X^2 = \frac{(b-c)^2}{b+c} \tag{3.65}$$

  X^2 follows χ^2 distribution with dof 1 under the null hypothesis.

- Normal approximation to binomial test: we are testing the cell count $b \sim \text{Bin}(b+c, 1/2)$. Use the normal approximation, our test statistic should be:

$$Z = \frac{b-(b+c)/2}{0.5\sqrt{b+c}} \tag{3.66}$$

  Apply the χ^2 distribution (square of standard normal distribution) and we obtain the McNemar's test.

- Pearson's χ^2 test: we consider only the two diagonal cells and apply the Pearson's χ^2 test, where the expected count is $(b+c)/2$ for both cells.

- Difference of cell counts: we define the test statistic $T = b-c$, which should indicate the difference of two cells, i.e. $E(T|H_0) = 0$. We need to determine the variance of T under H_0. Use $b|H_0 \sim \text{Bin}(b+c, 1/2)$, we have $\text{Var}(b) = (b+c)/4$. Then

$$\text{Var}(T) = \text{Var}(2b-(b+c)) = 4\text{Var}(b) = b+c \tag{3.67}$$

  Assume T follows normal distribution, we have $T^2/\text{Var}(T)$ as our test statistic with χ^2_1 distribution.

3.3 Naive Bayes and Discriminant Analysis

1. Naive Bayes classifier

   Naive Bayes (NB) model and model fitting:

   - Naive Bayes classifier: suppose the data has D features, the likelihood of one data point is:

$$p(x|y=c,\theta) = \prod_{j=1}^{D} p(x_j|y=c,\theta_{jc}) \tag{3.68}$$

     where θ_{jc} is the model parameter for the j-th feature of class c. The commonly used models are: Gassussian distribution for continuous features, multivariate Bernoulli distribution for binary features.

   - Model fitting by MLE: clearly to fit the model, we fit the distribution for each class separately, and the problem is easily reduced to known problems for fitting Gaussian or Bernoulli distributions. Let π_c be the fraction of class c, and θ_{jc} be the Bernoulli parameter of the feature j of class c, then we have:

$$\hat{\pi}_c = \frac{N_c}{N} \qquad \hat{\theta}_{jc} = \frac{N_{jc}}{N_c} \tag{3.69}$$

     where N_c is the number of points in class c, and N_{jc} is the number of examples whose j-th feature is 1 in class c. The MLE suffers from overfitting, notably, the zero-count problem, where $\hat{\theta}_{jc} = 0$ if $N_{jc} = 0$.

   - Bayesian NB: assume a prior distribution $\pi \sim \text{Dir}(\alpha_1, \cdots, \alpha_C)$ and $\theta_{jc} \sim \text{Beta}(\beta_0, \beta_1)$, the posterior distribution can be easily determined:

$$p(\pi|D) = \text{Dir}(N_1+\alpha_1, \cdots, N_C+\alpha_C) \tag{3.70}$$

$$p(\theta_{jc}|D) = \text{Beta}((N_c - N_{jc})+\beta_0, N_{jc}+\beta_1) \tag{3.71}$$

Model prediction and analysis:

- Posterior of class: suppose we train the model using data D, and we need to predict the class label of a new instance x, the posterior of class is:

$$p(y = c|x, D) \propto p(y = c|D) \prod_{j=1}^{D} p(x_j|y = c, D) \tag{3.72}$$

  Since the posterior distribution of the parameters are known, we can compute the posterior predictive of y and x_j respectively. The result is:

$$p(y = c|x, D) \propto \bar{\pi}_c \prod_{j=1}^{D} (\bar{\theta}_{jc})^{I(x_j=1)} (1 - \bar{\theta}_{jc})^{I(x_j=0)} \tag{3.73}$$

  where $\bar{\pi}_c$ and $\bar{\theta}_{jc}$ are posterior mean. If we use the MLE estimator, the prediction is similar, except that we replace posterior mean by MLE.

- Decision boundary and relation to linear model: we can take the log. of the posterior and write it as:

$$\log p(y = c|x, \theta) = \log \pi_c + \sum_j [x_j \log \theta_{jc} + (1 - x_j) \log(1 - \theta_{jc})] + \text{const} \tag{3.74}$$

  To write it as a linear model:

$$\log p(y = c|x, \theta) = \log \pi_c + \sum_j \beta_{jc} x_j + \beta_{0c} + \text{const} \tag{3.75}$$

  where

$$\beta_{jc} = \log \frac{\theta_{jc}}{1 - \theta_{jc}} \qquad \beta_{0c} = \sum_j \log(1 - \theta_{jc}) \tag{3.76}$$

  In particular, when we have only two classes, the log. posterior ratio is given by:

$$\log \frac{p(y = 1|x, \theta)}{p(y = 0|x, \theta)} = \beta_0 + \sum_j x_j \beta_j = X\beta \tag{3.77}$$

  where

$$\beta_j = \log \frac{\theta_{j1}/(1 - \theta_{j1})}{\theta_{j0}/(1 - \theta_{j0})} \tag{3.78}$$

  Thus this is similar to logistic regression, the difference being how parameters are trained.

- Feature analysis/ranking: the weight of a feature for prediction is the log odds ratio between the two classes. For relative rare variables, this is roughly the log. of frequency ratio between the postive and negative classes. Consider document classification problem, in general, the scoring scheme favors rare words (desired); but for rare words, it is more likely that the log-OR may be large from random sampling, creating noises (undesired).

- Feature selection by mututal information: one way to reduce the noise to select only discriminative features. We could use the MI (which agrees with the analysis of the feature weights). The MI of the j-th feature in the case of multivariate Bernoulli model is:

$$I_j = \sum_c \left[\theta_{jc} \pi_c \log \frac{\theta_{jc}}{\theta_j} + (1 - \theta_{jc}) \pi_c \log \frac{1 - \theta_{jc}}{1 - \theta_j} \right] \tag{3.79}$$

  where $\theta_j = \sum_c \pi_c \theta_{jc}$.

Extending the basic NB model:

- Document classification by bag of words: we could use multinomial distribution to model word counts, which may be more informative than word presence. The model is easy to train and use. However, it does not work well for classification, one reason being the burstiness problem: the rare words often do not occur at all in a document, but once they do, they occur in bursts. This is not easily modeled by multinomial distribution. One idea is to use Dirichlet compound multinomial model for the density.
- Modeling dependency of features: (Exercise 3.20)
  - Intuition: why modeling dependency may help? In document classification problem, phrases are often more informative than single words. For example, "Windows operating system", each word alone may not be very discriminative between classes, but the phrase is. Modeling dependency allows one to capture phrases.
  - In general, feature dependency may help by capturing informative patterns (specific combination of individual features). Ex. we consider two classes in 2D space: all the points of the two classes belong to the same square, but the points of two classes are located in the two halfs of the square, separated by the diagonal line. Then neither x nor y dim. alone is very discriminative.
- L_1 regularization: [L_1-regularized naive Bayes classifiers] the idea is that θ_{jc} should be equal for any c for most features. Thus we define the penalized log-likelihood, in the fashion of group lasso. Adding the penalty term to negative log-likelihood:

$$\lambda J(\theta) = \sum_j \sqrt{\sum_c (\theta_{jc} - \theta_{j\cdot})^2} \tag{3.80}$$

  where $\theta_{j\cdot}$ is the mean of θ_{jc} over all c's. If the j-th feature can take multiple values (multinomial model instead of Bernoulli), we could extend the equation by summing over k as well (k is the index of the possible outcome of the j-th feature).
  - Remark: the penalty term does not take word frequency into account, the term tends to be dominated by common words (regularization of common words but not rare ones).

Lessons/questions:

- Decision boundary analysis: for a classification problem, to understand a classifier, the first question is what is the decision boundary of the classifier: is it linear or not, etc.
- Feature analysis: what features contribute most to the prediction function? Often helpful to intuitively understand it, e.g. for document classification, whether common or rare words are more important.
- Feature dependency and informative patterns: what kind of dependency between features may provide extra information? For document classification, whether phrases or higher structures are informative, etc.
- Q: under Bayesian NB, we would like a prior distribution that favors equal θ_{jc} for each c, for most j's. How would we define such prior?
- Q: how to deal with covariates in NB model: suppose we are mainly interested in finding relation between x and y, but we have covariates z that is associated with x or y. It is easy to model this in the regression framework; in the NB model, we need to model $y \to x \leftarrow z$, and y and z are associated.
  - A simple model is: define the distribution of x on each combination of (y, z). For model complexity purpose, we may add additional priors, e.g. the effect of y and z on x is independent (i.e. the effect size of y on x is the same across all values of z's).

– More generally, both regression and generative models are special cases of a graphical model involving variables x, y and z.

Reference: Chapter 3 of Murphy [2012].

3.4 Latent Variable Models

Expectation-Maximization (EM) algorithm [Murphy, 11.4]

- EM: let x be the data, z missing data and θ parameters. The complete data log likelihood is given by:

$$l_c(\theta) = \log p(x, z|\theta) \tag{3.81}$$

The E-step computes the expected complete log-likelihood:

$$Q(\theta|\theta^t) = \mathrm{E}_{z|\theta^t,x}[\log p(x, z|\theta)] \tag{3.82}$$

where the expectation is taken over the posterior of z given data and current parameters. The M-step maximizes the function, treating θ as parameter, but $z|\theta^t, x$ as given.

- Justification of EM: $Q(\cdot)$ is a lower bound of observed data log-likelihood $l(\theta) = \log p(x|\theta) = \log \sum_z p(x, z|\theta)$. One can show that in EM, $l(\theta)$ is always monotonically increasing, so this leads to local max. of $l(\theta)$.

3.4.1 Mixture Models and Missing Data Problem

Bayesian mixture model: [GCSR, Chapter 18]

- Latent variable model: suppose we have M groups with the m-th group defined by the model θ_m. Each sample belongs to one of the M groups, with the membership variable unobserved. Let ξ_i be the membership vector (unit vector) of the i-th sample, i.e. $\xi_{im} = 1$ if the i-th sample belongs to the m-th group and 0 otherwise. Our model for the latent variable is thus:

$$z_i \sim \text{Multinomial}(\lambda_1, \cdots, \lambda_K) \tag{3.83}$$

And the complete likelihood:

$$p(y, z|\theta, \lambda) = p(z|\lambda)p(y|z, \theta) = \prod_i \prod_{m=1}^{M} (\lambda_m f(y_i|\theta_m))^{z_{im}} \tag{3.84}$$

The prior of λ follows the Dirichlet distribution: $\lambda \sim \text{Dir}(\alpha_1, \cdots, \alpha_M)$.

- Equivalent model: we could elimiate ξ all together in the model. Instead, we have the likelihood as:

$$p(y_i|\theta, \lambda) = \sum_{m=1}^{M} \lambda_m f(y_i|\theta_m) \tag{3.85}$$

This is thus a single-level model with no latent variables. However, there is an advantage of the latent variable formulation (see below).

- Comparison with hierarchical model: the group membership is observed in the hierarchical model, but not in the mixture model. Thus for hierarchical normal model, we have: $y_i \sim N(\theta_{j[i]}, \sigma^2)$ where $j[i]$ is the group that the i-th sample belongs to; and for the normal mixture model, we have: $y_i|z_i, \theta \sim N(\theta_{z_i}, \sigma^2)$.

- Inference with EM algorithm: let $\phi = (\theta, \lambda)$ be the parameters. Suppose we are maximizing the posterior mode of θ, averaging latent variables z. In the E-step, we compute the objective function:

$$Q(\phi|\phi^{(t)}) = \mathrm{E}_{z|\phi^{(t)},y} [\log p(\phi, z|y)] \tag{3.86}$$

  Plug in $\phi = (\theta, \lambda)$, we have:

$$p(\theta, \lambda, z|y) \propto p(\theta)p(\lambda)p(z|\lambda)p(y|z, \theta) \tag{3.87}$$

  Note that the decomposition makes the E-step easier since some of the terms do not depend on the latent variables. In the M-step, sometimes CM can be used: effectively, we iteratively update λ assuming θ is given, and update θ assuming λ given.

- Inference with Gibbs sampling: alternatively sample from the conditional distributions: $p(\theta|\lambda, z, y) = p(\theta|z, y)$, $p(\lambda|\theta, z, y) = p(\lambda|z)$ and $p(z|\theta, \lambda, y) \propto p(z|\lambda)p(y|z, \theta)$.

- **Data Augmentation**: the general principle is that in many cases, it may actually facilitate inference if one introduces additional latent variables. Conceptually, this makes it possible to reduce a complex distribution (to be maximized or sampled) into multiple simpler ones.

Bayesian missing data problem: [GCSR, Chapter 21]

- Motivation: e.g. to infer a multivariate normal distribution, for some samples, some components are missing. Similarly, for regression problems, some explanatory variables of some samples may be missing.

- Model: note that we need to model the "missing data mechanism" (i.e. the random process by which some samples have missing data). Suppose $y = (y_{\text{obs}}, y_{\text{mis}})$ is the data, and I is the indicator variable (whether a sample is observed or not). The missing data mechanism is modeled by the distribution, $p(I|y_{\text{obs}}, \phi)$. Then the likelihood is given by:

$$p(y_{\text{obs}}, I|\theta, \phi) = p(I|y_{\text{obs}}, \phi) \int p(y_{\text{obs}}, y_{\text{mis}}|\theta) dy_{\text{mis}} \tag{3.88}$$

- Inference: an example of Data Augmentation, Gibbs sampling or EM algorithm. It involves iteration of (1) imputation of missing data, this is based on:

$$p(y_{\text{mis}}|y_{\text{obs}}, \theta) \propto p(y_{\text{obs}}, y_{\text{mis}}|\theta) \tag{3.89}$$

  and (2) inference of parameters, assuming there is no missing data.

Mixture model of multi-dimensional discrete variables: [Stephens' GTEx grant]

- Motivation: suppose we are trying to model variables $(Z_1, \cdots, Z_m)$ where Z_i is a binary variable. For example, Z may represent the expression of a gene in m tissues (discretized expression). The m variables are not independent, so we cannot use multi-Bernoulli to model it. There are 2^m different configurations, using a different prob. for each possible configuration is also unrealistics.

- Idea: we assume the samples form K clusters, where each cluster represents a different tendency of being 1 in different dimensions. For example, one cluster represents genes likely to be active in the second tissue, but not the first and third one, so the distribution of samples from this cluster can be represented as $(0.1, 0.9, 0.1)$. Another cluster may represent no expression in all tissues $(.01, .01, .01)$. Under this model, there may be ambiguity of assigning a configuration Z, which may belong to multiple clusters.

- Model: for the k-th cluster, we represent its "profile" as $q_k = (q_{k1}, \cdots, q_{km})$, so the probability of a configuration Z is:

$$p_k(Z|q_k) = \prod_{i=1}^{m} q_{ki}^{Z_i} (1 - q_{ki})^{1-Z_i} \tag{3.90}$$

  Furthermore, we define a prior/proportion of each cluster π_k.

- Remark: the model can be used in the context where Z_i are latent variables too.

Joint analysis of differential gene expression in multiple studies using correlation motifs (CorMotif) [Wei & Ji, Biostatistics, 2015]

- Biological motivation: suppose we want to test DE of a gene, and we have multiple conditions/samples. There is information shared between samples: e.g. a gene is activated in one tissue, then it's likely that it is also activated in a related tissue. The challenge is to model the relationship among conditions.
- Model intuition: we model the state of a gene in a condition as an indicator. To model the dependency of conditions, instead of directly modeling multivariate binomial RVs, we use a mixture model: each gene belongs to one motif, where each motif specifies a pattern of the indicator variable, e.g. it is likely to be 1 in all conditions; or likely 1 in the first three conditions and 0 otherwise. We assume that there are a small number of motifs.
- Model: let π_k be the probability of motif k, and for a motif k, q_{kd} is the probability of being 1 in the study d, i.e. Q specifies activity patterns of motifs. We have latent variables, B, the motif membership of genes, and $A = (a_{gd})$ be the activity (1 or 0) of gene g in study d. Given data T (test statistic from limma), our model can be specified by:

$$P(T, A, B|\pi, Q) = P(B|\pi)P(A|B, Q)P(T|A) \tag{3.91}$$

  Each component: (1) $B|\pi$: multinomial distribution. (2) $A|B, Q$: Bernoulli distribution, a_{gd} is given by the activity of the motif g belongs to. (3) $T|A$: we have $t_{gd}|a_{gd} = 1 \sim f_{d1}$ and $t_{gd}|a_{gd} = 0 \sim f_{d0}$. The paper uses t-distribution.
- Inference: to marginalize B and A, use EM algorithm. To find the number of motifs, using BIC.
- Interpretation of model: once we have the Q matrix, the correlation between two studies are measured by $\sum_k \pi_k q_{k1} q_{k2}$, summing over all motifs. This is the probability that a gene is active in both studies.
- Remark: alternative model using sparsity. Given D conditions, we have 2^D possible configurations. However, in truth, there are probably only a small number of configurations, so we can allow all configurations, and let π_k be the probability of configuration k (binary). But we assume π_k is sparse.
- Remark: we can also model the continuous relationship using covariance matrix. If we pre-specify the possible covariance, this leads to Matrix ASH (MASH). If not, this is similar to sparse Gaussian mixture model.

3.4.2 Principal Component Analysis (PCA)

Latent variable model and motivations of PCA:

- Motivation: in many problems/applications, there are certain unmeasured (latent) variables, which influence the observed variables. And variations of these latent variables explain the variations of all observed RVs, and the covariance among related RVs (those sharing the same latent variable(s)).
- Generic latent variable model: suppose we have two latent variables U and V, our observed variable X_j can be expressed as (ignoring the mean):

$$X_j = \beta_j U + \gamma_j V + \epsilon_j \tag{3.92}$$

  For the i-th data point, we could write x_i (D-dim. vector) in terms of u_i and v_i in vector form:

$$x_i = u_i \beta + v_i \gamma + \epsilon_i \tag{3.93}$$

  where $\beta = (\beta_1, \cdots, \beta_p)^T$ and $\gamma = (\gamma_1, \cdots, \gamma_p)^T$ are vectors representing the effect of U and V on $X_j, 1 \le j \le p$, respectively. Thus, a PCA model has two main components:

  - Latent variables $\mathbf{u} \perp \mathbf{v}$: that explain the variance and correlation of the observed variables. If the observed variables have some correlation structure, then most likely some lower-dimensional representation with latent variables will be possible.
  - Weights/coefficients: this can be understood in terms of how the observed variables depend on the latent variables. They could be seen as "contribution vectors" (of the latent variables).

  More generally, we have D observed variables with N samples, and we find L latent factors to explain the data.

- Examples:
  - Gene expression: expression profiles of D genes under N conditions. The hidden variables are nutrient availability, stress (such as temperature), etc. The eigenvectors of the two latent variables represent the contribution/effect of the two on gene expression.
  - Medicine: a person's risks of cancer, diabetes, heart disease, stroke depend on a few shared latent variables, e.g. the metabolic aspect/insuline resistence (U), and the inflammatory aspect (V). The eigenvectors represent the contribution of the two factors on the risks of various diseases.
  - Economics: there may be many variables to measure the economic activities of a country, e.g. manufacturing, inventory, GDP, corporate spending, employment rate, etc. They may all depend on a few latent variables, e.g. one for the level of consumer spending, one for inflation. The eigenvectors would then represent how strongly each economic index depends on these two factors.
  - Image processing: the pixel representation of an image really reflects the content of an image, e.g. in terms of what objects it has.

- Intuitive picture of PCA: identifiability issue, e.g. in 1D case ($D = 1$), any finite mixture model is identifiable, but infinite mixture, $L = 1$ is not. When $D > L$, the model is identifiable (with conditions, see below), and it tries to explain the correlation between variables in terms of some hidden variables. So for example, it first finds all strongly correlated variables, and define a latent variable to explain the correlation of these variables; it then repeats the process on the remaining variables or unexplained variances of the variables processed in the first step.

- The latent variable model in 2D: similar to least square regression, we could formulate the objective function as (2D case):
$$\min_{\beta,\gamma,u_i,v_i} \|x_i - u_i\beta - v_i\gamma\|^2 \tag{3.94}$$
  where we assume $\mathbf{u} \perp \mathbf{v}$. Note that this is to assume that all variables have the same variance (thus variables need to be standarized first to apply PCA).

What is principal component analysis? [NBT, 2008]

- Concept of PCA: reduces the dimensionality of the data while retaining most of the variation in the data set. It accomplishes this reduction by identifying directions, called principal components, along which the variation in the data is maximal.

- Example of two genes: Figure 1 A-C.

- Dimension reduction and visualization: Figure 1 D-E. Data of 10000 genes can be projected into 2D space.

Background: variance partition

- Theorem: if $X = Y + Z$ is a sum of two independent RVs Y and Z, then the sample variance of X is the sum of sample variance of Y and that of Z. The proof follows from the Pythagorean Theorem.

- Variance in multi-dimensional space: suppose we have $x_i \in \mathbb{R}^p$, let μ be the mean of all x_i's, we could define the total variance of the data points as:

$$V = \sum_i \|x_i - \mu\|^2 \tag{3.95}$$

- Variance partition: consider the j-th component of x_i's, let μ_j be the mean of the j-th component, and we could define the variance in the j-th component as:

$$V_j = \sum_i (x_{ij} - \mu_j)^2 \tag{3.96}$$

  Then the total variance can be expressed as: $V = \sum_j V_j$. In general, if we have an orthogonal basis of $\mathbb{R}^p$, $v_1, \cdots, v_p$, and project x_i's on v_j's, and let V_j be the variance in the projections on v_j, we have $V = \sum_j V_j$, simply from the Pythagorean Theorem.

- Error and variance: suppose we project x_i's (p-dimensional) on a lower-dimensional hyperplane of dim. q. Let V_p be the total variance, and V_q be the total variance of projections. We define the error as:

$$\text{Err} = \sum_i \|x_i - x_i'\|^2 \tag{3.97}$$

  where x_i' is the projection of x_i. Then according to the Pythagorean Theorem, we have

$$\text{Err} = V_p - V_q \tag{3.98}$$

  Thus the error is simply the unexplained variance in the data, which is the variance in the other dimensions (orthogonal to the hyperplane).

- Remark: this is similar to linear regression, where the total variance of the response variable can be partitioned: $SST = SSR + SSE$. Thus minimizing SSE is equivalent to finding the linear model that maximizes SSR. In both cases, we have one RV as a sum of two independent RVs (explanatory and error), so we have variance paritioning.

Geometric picture of PCA:

- A simple case of $D = 2, L = 1$: each x_i is a vector in a 2D space, and we find the direction w to project x_i (the projected x_i is $\hat{x_i}$). The objective function is the total distance from x_i to $\hat{x_i}$. The coordinates of x_i in w is z_i (scalar).

- Geometric mapping of the latent variable model: given N data points x_i in the D-dim. space, the latent variable model can be mapped geometrically:
  - The N points are close to a low-dim. hyperplane, P, (2D if there are two latent variables): the objective function (MSE) of the latent variable model corresponds to the total distance of the N points to the plane P.
  - In the low-dim. space, we have L orthogonal vectors $w_1, \cdots, w_L$, $w_k \in \mathbf{R}^p, 1 \leq k \leq L$, each of them representing a "principal component" (or principal direction).
  - The vector $z_i \in \mathbb{R}^L$ are the coordinates of x_i on principal components: the k-th coordinate is $z_{ik} = \langle w_k, x_i \rangle = w_k^T x_i$. In vector form: $z_i = W^T x_i$.
  - The projection of x_i on the plane (coordinates in the original D-dim. space), $\hat{x_i}$, can be written as: $\hat{x_i} = \sum_k z_{ik} w_k = [w_1 \cdots w_q] z_i = W z_i$.

- Alternative geometric view: We view X_j as a N-dim. vector, and the goal is to find an orthogonal set of N-dim. vector $Z_1, \cdots, Z_L$ s.t. linear combination of Z_k's explain all X_j's. Ex. suppose X_j's are all parallel to each other, then we can easily choose Z_1 that is parallel to all X_j's, which explains the data.

Statistical inference of latent variable model: PCA

- Reference: [Murphy, Chapter 12]
- Background: projection on orthogonal basis (see Linear Algebra notes, "Orthogonality"). Let U be an orthogonal basis (n-dim), consider the projection of a vector v (in the subspace defined by U) onto U. Let x be the coordinates of v on U, then v is a linear sum of basis vectors of U: $v = Ux$; and the coordinates $x = U^T v$.
- PCA model: we have N samples, D observed variables and L latent factors. We have $x_i \in \mathbb{R}^D$ as data, and $z_i \in \mathbb{R}^L$ as the latent variable for sample i. The k-th variable of sample i (assuming it is zero-centered) is given by:

$$x_{ik} = \sum_{j=1}^{L} w_{kj} z_{ij} + \epsilon_{ik}, \qquad \epsilon_{ik} \sim N(0, \sigma^2) \tag{3.99}$$

  Or in vector form: $x_i = W z_i + \epsilon_i$, where W is $D \times L$ matrix (factor loading matrix). In matrix form, we can write it as: $X = ZW^T$ where X is $N \times D$ data matrix, Z is $N \times L$ matrix, representing the projection of X on factors. The problem is:

$$\min_{W,z} \sum_i \|x_i - \hat{x}_i\|^2 \tag{3.100}$$

  where W is orthonormal matrix, with $w_j \in \mathbb{R}^D$ unit vector, and $\hat{x}_i = W z_i \in \mathbb{R}^D$.
- **Key notations of PCA**: let x_i be D-dim. data point, z_i its low-dim. representation (PC representation or projection), and $W = [w_1 \cdots w_L]$ be the L PCs (i.e. W is $D \times L$ matrix), we have:

$$\begin{cases} z_i = W^T x_i & \text{PC representation/projection of } x_i \\ \hat{x}_i = W z_i & \text{Reconstruction of } x_i \end{cases} \tag{3.101}$$

  This is similar to encoding and decoding, respectively. Write this in matrix form:

$$Z = XW \qquad \hat{X} = ZW^T \tag{3.102}$$

  where X is $N \times D$ matrix and Z is $N \times L$ matrix.
- **Theorem**: Minimizing reconstruction error by PCA. We are minimizing this function:

$$J(W, Z) = \frac{1}{N} \sum_{i=1}^{N} \|x_i - \hat{x_i}\|^2 \tag{3.103}$$

  where $\hat{x}_i = W z_i \in \mathbb{R}^D$, and $z_i \in \mathbb{R}^L$. The constraint is that the $D \times L$ matrix W is orthonormal, i.e. $w_j \in \mathbb{R}^D$ has unit norm. The optimal solution is given by $\hat{W} = V$ where V contains the L eigenvectors of the empirical covariance matrix $\hat{\Sigma} = X^T X$, and the coordinates $z_i = W^T x_i$.
- Analysis view of PCA (maximum variance): consider the case of $L = 1$, our objective is $J(w_1, z_1)$. Taking derivative wrt. z_{i1} and let it equal to 0, we can show that $z_{i1} = w_q^T x_i$, so z_{i1} is the orthogonal projection of x_i on w_1. And plug-in z_{i1}, we have:

$$J(w_1) = \text{const} - \frac{1}{N} \sum_{i=1}^{N} z_{i1}^2 \tag{3.104}$$

  This is effectively the variance of z_{i1}. So the problem is to find a direction where the projection has the largest variance.

- Solving the maximum variance problem: we plug-in z_{i1}, and the objective becomes:

$$\sum_{i=1}^{N} z_{i1}^2 = \sum_i w_1^T x_i x_i^T w_1 = w_1^T \left(\sum_i x_i x_i^T \right) w_1 = w_1^T \hat{\Sigma} w_1 \tag{3.105}$$

  where $\hat{\Sigma} = X^T X$ is the covariance matrix, assuming the data is centered. This is a quadratic form and the solution is given by the eigenvector of the maximum eigenvalue of $\hat{\Sigma}$, according to the Rayleigh quotient. The solution is the eigenvector, belonging to the largest eigenvalue, of the real symmetric matrix $X^T X$. Thus the maximum variance direction is the first PC, w_1, and the variance is λ_1, the first eigenvalue of $X^T X$. Similarly, we have: w_2 maximizes the variance of projections among all vectors orthogonal to w_1 (also the variance unexplained by w_1), and w_3 maximizes variance among all orthogonal to w_1 and w_2 (the variance unexplained by w_1 and w_2, and so on.

- Interpretation of the factor loading matrix W: $W = [w_1 \cdots w_L]$ is the matrix consisting of L PCs. W is $D \times L$ matrix, and $W_{jk}, 1 \leq j \leq D, 1 \leq k \leq L$ is the effect of the k-th latent variable Z_k on X_j. The row vector of W: W_j is the (regression) coefficients of all factors on X_j. The column vector of W (PC): W_k is the effect sizes of the factor k on all observed variables.

- Model identifiability and orthogonality of PCs: according to the geometric representation, suppose x_i's are close to a hyperplane, there are infinitely many ways of choosing the basis of the hyperplane, thus different coordinates of the projections (hence different values of the latent variables). To make the model identified, we could choose the PCs as the orthogonal basis of the hyperplane. Indeed, PCs are eigenvectors of $X^T X$, so they are orthogonal to each other.

- PCA and clustering: PCA only reflects the linearity in the data. If there is clustering structure in the data (e.g. all data points form clusters in low-dim. representation), this will not be captured by the matrix factorization.

Connection of PCA with Singular value decomposition (SVD):

- Motivation: suppose X lies in lower-dim. space (lower-rank), how do we understand geometrically why this leads to small eigen-values of $X^T X$, and the idea of SVD as an approximation of X?

- MVN perspective: suppose our data X is generated from MVN with covariance matrix $\Sigma = X^T X$. The contour of the PDF forms ellipse, with the axis defined by the eigenvectors of Σ. We can now use the diagonalization trick: let $\Sigma = VDV^T$, then we can view the PDF as ellipse defined by the axis v_i, and scaling d_i. Now it is clear that: X is low-dimensional means $d_i \approx 0$ for larger i (assuming d_i are sorted).

- How do we interpret SVD geometrically when X represents $n \times D$ data matrix? Let v_1 be the first singular vector, D-dim, of X. (1) In linear algebra, we view Xv_1 as the linear map of v_1 in $\mathbf{R}^n$. In statistics, we view Xv_1 as the projection of X on v_1: it has n coordinates along v_1. (2) In Linear algebra, we have the result: Xv_i and Xv_j are orthogonal. In statistics, it means that the projections along v_i and v_j are statistical independent. This means that the total variance of data can be partitioned along each v_i.

- Geometric intuition of why SVD leads to dimensionality reduction: consider a simple case where $D = 2$. Let v_1, v_2 be singular vectors with singular values d_1, d_2. Suppose $d_2 \approx 0$. Consider the projection of X on v_1 and v_2. We have $Xv_2 = d_2 u_2 \approx 0$. This means that projections of X along v_2 are mostly 0. This means that X lies largely in 1-D space of v_1.

- **Truncated SVD**: we consider only the top L singular vectors of X, then $X = U\Sigma V^T$, where U is $N \times L$ matrix, Σ is $L \times L$ matrix and V is $D \times L$ matrix. With this form, we have the PCs, W in PCA is just V: $\hat{W} = V$. The projection of X onto PCs $Z = XW$ is now:

$$Z = XW = U\Sigma V^T V = U\Sigma \tag{3.106}$$

and the reconstruction $\hat{X} = ZW^T = U\Sigma V^T = X$, which is just the truncated SVD of X!

- Variance explained by PCs: from the SVD, it is clear to answer how much variance of data is explained by PCs. Because PCs are orthogonal, the total variation of data is the sum of variation along the direction of each PC/eigenvector of X^TX. Let $\lambda_j = d_j^2$ be the eigenvalue of the j-th eigenvector of X^TX. For the j-th PC, its variance is explained is:

$$\text{Var}(Xv_j) = \text{Var}(u_j d_j) = d_j^2 \|u_j\| = d_j^2 = \lambda_j \tag{3.107}$$

  where we use the fact that u_j is a unit vector.

- Another connection between PCA and SVD: the outer-product form of SVD. If we want to approximate the matrix X, then we should use the largest L singular values. This is the PCA of X.

Application and interpretation of PCA [personal notes]

- Dimensionality reduction or "denoising": new representation of $x_i \in \mathbb{R}^D$ becomes $z_i \in \mathbb{R}^L$ in lower-dimensional space. Each PC $\in \mathbb{R}^D$ represents the effect of a latent variable on every dim. in the original space.
  - Ex. gene expression data: assuming expression of a gene is a result of multiple TFs. Then a PC could represent the effect of a TF on expression of every gene (mainly contributed by genes that are influenced by that TF); and all the samples can be represented now as the vector of all TF levels.
  - Ex. stock price: assuming stock prices are results of economics of multiple (lower-dim) sections. Then a PC may represent the activity of IT section (which is mainly contributed by stocks in the IT sector). And the stock price of one year can be represented as the vector of activities of all sectors.

- Recovering latent variables: sometimes we can recover the information of these latent variables. (1) Suppose we have additional information of each sample, we can correlate these additional variables with latent variables. In the stock price example, if we have actual measures of each sector, then we can correlate z_i's of every year with these measures. (2) We can also use information of variables: if a PC is mainly contributed by a subset of variables, then the common properties of this subset would be likely important for that PC. Stock price example: for each PC, see which stocks are weighted more.

- Adjusting for latent variables in comparison: suppose we have two transcriptome datasets, one treatment and one control. We would like to compare gene expression, but need to adjust for hidden covariates such as batch effects that could differ between treatment and controls. The idea is: we first do PCA to find out the latent factors, then for each sample, we have these z_i's, we could treat them as if they are observed, and adjust for them in gene expression test for every gene.

Statistical analysis of the genomic distribution and correlation of regulatory elements in the ENCODE regions [Zhang & Gerstein, GR, 2007]

- Representation of PCA results by biplots.

- Choosing eigenvectors: after PCA, we choose a few eigenvectors that capture most of the variance in X, typically two, and represent the results in this 2D space.

- Data point representation in the low-dim space: each observation is represented in the 2D space, by its projection in the plane. This data point representation may reveal the additional structure in the data, e.g. many observations may come from the same group, thus sharing the similar values of the PCs.

- Variable representation the low-dim space: a variable X_j is related to the latent variable u and v (2D case) by:
$$X_j = \beta_j u + \gamma_j v + \epsilon_j \tag{3.108}$$
This is a straight line in the (u, v) space (ignoring the error term), with β_j and γ_j reflects the effect of u and v on this X_j.

Extensions of PCA and relation with other methods:

- Nonlinear dimensionality reduction: the model of how observed variables are related to the latent variables does not have to be linear, and for nonlinear cases, we have x_i's close to a surface in the p-D space.
- PC-based clustering: one could perform clustering in the low-dim. space (PCs) to reveal the cluster structure.
- PC regression: regression of dependent variables on PCs (latent variables), instead of explanatory variables directly.
- Clustering vs PCA: clustering is based on the similar idea that a relatively small number of hidden variables would explain the data. The difference is: clustering - individual observation is a simple deviation from the hidden variables (cluster means) while PCA - individual variable is a linear combination of multiple hidden variables (principle components).
- Latent variable model perspective of PCA: e.g. the original model is not identifiable, but with more constraints, one may be able to learn the latent variables (e.g. latent variable must be some of the observed ones, with error - CFA model).

Biclustering with heterogeneous variance [Chen, PNAS, 2013]

- Problem: identify block structure in matrix.
- Approximation of matrix with SVD: $X \approx UDV^T$, where U and V are r-dimension. Write $X = \Psi + Z$, where Z is $MVN(0, \sigma^2 I)$, or $X = UDV^T + Z$
- Approximate Ψ using r-ranked matrix (lower than the dimension of X). To approximate the bi-cluster (checkboard structure): sparsity on the signal matrix, U and V are sparse.
- SVD algorithm: start with V (orthogonal matrix), then compute U and V in next iteration. To impose sparsity: only values in U and V larger than a threshold are chosen at each step.
- Heterogenesou sparse SVD: heterogenous variance. $X_{ij} = \mu_{ij} + \rho \cdot \Sigma \cdot \Phi$.
- Sigma: block structure for errors as well, e.g. case and control samples, genes in cases have similar errors.
- Application to methylation data in normal vs. cancer samples. Cancer samples: methylation more variable across all cancers.
- Lessons: express factor analysis as regression problem, where covariates are factors, then we can use machinery for regression, e.g. sparsity. Use SVD to approximate a matrix, and encode the structure.
- Q: why sparsity of U and V in SVD captures checkboard structure?

3.4.3 Factor Analysis

Basics of factor analysis [Murphy, 12.1]

- Model: our data is $x_i \in \mathbb{R}^p$. We assume they are generated from linear combination of a smaller number of latent variables $z_i \in \mathbb{R}^q$ with normal distribution. The model:

$$z_i \sim N(\mu_0, \Sigma_0) \qquad x_i|z_i \sim N(Wz_i + \mu, \Psi) \tag{3.109}$$

  where W is the factor **loading matrix** of $p \times q$ dimension. We assume Ψ is diagonal, and wlog, $\Sigma_0 = I$. Example: Figure 12.1.

- FA can be viewed as a way of representing MVN with fewer parameters: the marginal distribution of x_i is given by:

$$x_i \sim N(W\mu_0 + \mu, \Psi + W\Sigma_0 W^T) \tag{3.110}$$

  Note: covariance of x has $O(p^2)$ parameters, but this representation has $pq + p$ parameters.

- Posterior of latent variables z_i if W is given: it follows normal distribution with mean m_i and variance Σ. For simplicity, we assume $z_i \sim N(0, I)$.

$$\Sigma = (I + W^T\Psi^{-1}W)^{-1} \qquad m_i = \Sigma W^T\Psi^{-1}(x_i - \mu) \tag{3.111}$$

  Note that Σ is independent of data points (same for all i), and furthermore, it can be computed efficiently by Inversion Lemma. Also note that m_i is a linear operator of x_i.

  - Remark: if we assume $\Psi = I$, and somehow ignore the covariance of z_i, we have $m_i \approx W^T(x_i - \mu)$. This is basically the projection of x_i on PCs in PCA.

- Biplot: show z_i and show the representation of features X_j, both in terms of the latent variables (Figure 12.2).

- Identifiability of FA: we can rotate z_i, and have the same model (likelihood). Geometrically, suppose all data are close to a hyperplane, but the coordinates of the hyperplane can be chosen arbitrarily. Imagine a hyperplane, consider z_i and 2 PCs, w_1 and w_2. If we first rotate w_1 and w_2, and then z_i accordingly, we have the same likelihood.

- Addressing identifiability problem: (1) PCA: forcing W to be orthonomal, and rank the PCs by decreasing variance. Note: when $p = 2, q = 1$, maximizing variance leads to a unique solution. (2) Sparsity promoting priors on W.

EM algorithm for factor analysis: [Murphy, 12.1.5]

- Ref: The EM Algorithm for Mixtures of Factor Analyzers [Ghahramani and Hinton, 1996]. Also `Perera-FA-EM-slides.pdf`.

- Model: we assume $z_i \sim N(0, I)$, and $x_i|z_i, W, \mu, \Psi \sim N(Wz_i + \mu, \Psi)$. We denote $\theta = (W, \mu, \Psi)$.

- E-step: we first obtain the posterior distribution of z_i given x_i and current parameters $\theta^{(t)}$, given by Equation 3.111. To compute the Q function, we first obtain the log-likelihood of the complete data:

$$\log p(x, z|\theta) = \sum_i [\log p(z_i) + \log p(x_i|z_i, \theta)] \tag{3.112}$$

  We ignore the constant term and plug-in the normal likelihood:

$$\log p(x, z|\theta) = -\frac{1}{2}\sum_i [x_i - Wz_i - \mu]^T\Psi^{-1}[x_i - Wz_i - \mu] - \frac{n}{2}\log\det\Psi + \text{const} \tag{3.113}$$

We will now compute the expectation over z_i, given the current estimate of θ.

$$Q(\theta) = -\frac{n}{2}\log\det\Psi - \frac{1}{2}\sum_i \left[x_i^T\Psi^{-1}x_i - 2x_i^T\Psi^{-1}W\mathrm{E}\left(z_i|x_i\right) + \mathrm{E}\left(z_i^TW^T\Psi^{-1}Wz_i|x_i\right)\right] \tag{3.114}$$

We now use the result about the expectation of the quadratic form of random vector (in this case, z_i):

$$\mathrm{E}\left(z_i^TW^T\Psi^{-1}Wz_i|x_i\right) = \mathrm{tr}\left(W^T\Psi^{-1}W\mathrm{E}\left(z_iz_i^T|x_i\right)\right) \tag{3.115}$$

where $\mathrm{E}\left(z_iz_i^T|x_i\right)$ can be obtained from the posterior of z_i given x_i.

- M-step: See Appendix A of [Ghahramani and Hinton, 1996]. Use the fact that derivative and trace can commute to simplify the algebra.

Probabilistic PCA [Murphy 12.2.4]

- Marginal distribution and likelihood: wlog, we set $\mu_0 = 0$, $\Sigma_0 = I$, $\mu = 0$ (data is centered) and $\Psi = I$ (errors are independent), then the marginal distribution of X:

$$x_i|W \sim N(0, \sigma^2I + WW^T) \tag{3.116}$$

The covariance matrix $C = \sigma^2I + WW^T$, the LL function:

$$\log P(X|W,\sigma^2) = -\frac{N}{2}\log\|C\| + \frac{1}{2}\sum_i x_i^TC^{-1}x_i = -\frac{N}{2}\log\|C\| + \mathrm{tr}(C^{-1}S) \tag{3.117}$$

where $S = (1/N)X^TX$ is the sample covariance matrix.

- Interpretation of WW^T: we consider the row vectors of W, W_j is $1 \times q$ vector (the effects of PCs on X_j):

$$WW^T = \begin{bmatrix} W_1 \\ \cdots \\ W_p \end{bmatrix}\left[W_1^T \cdots W_q^T\right] = \left[W_jW_l^T\right]_{p\times p} \tag{3.118}$$

So the covariance of X_js (ignore σ^2I) is determined by the loading matrix. Intuitively, if two variables have similar weights in the loading matrix, then they are highly correlated.

- MLE: take derivative, dl/dW, and set it to 0. Analytic solution and $\hat{\sigma}^2$ is the average of the remaining eigenvalues.

- Connection with classical PCA: consider the case when $\sigma^2 \to 0$: then W should satisfy $\hat{W}^T\hat{W} = S$. Let the SVD of X be UDV^T then

$$X^TX = (UDV^T)^T(UDV^T) = (V^TD^T)(DV) \tag{3.119}$$

So we have: $\hat{W} \to DV$, where V is the $p \times q$ matrix where the columns are the first q eigenvectors of S. So the direction of $\hat{W}$ is given by V. If we require PCs to be unit vector, then we can ignore the scaling D, and the PCs are just V. The posterior mean of latent vectors is the projection of x_i on PCs. Difference: in classical PCA, we require the PCs to be unit vector; in PPCA, we require z to have unit variance.

- EM algorithm and interpretation: E-step: given the weight matrix, latent variables $\tilde{Z}$ can be determined by projection of X on W; M-step: given the latent variables, the weight matrix $\tilde{W}$ can be determined by multiple regression. Physical analogy (Figure 12.11): minimize the potential energy of spring.

Extensions of PCA [Murphy, 12.4, 12.5]

- Background: softmax, generalization of logistic regression on multi-class response variables, map K-dim. real values to K-dim. probability values that sum to 1. Let x be K-dim. values, and y is a multinomial random variable, then

$$P(y = j|x) = \frac{e^{x^T w_j}}{\sum_k e^{x^T w_k}} \tag{3.120}$$

  where w_k is K-dim. vector. Remark: relation to logistic normal distribution (similar to but more flexible than Dirichlet).

- Categorical PCA: y_{ij} are j-th response of sample i. It is a multi-class label (C classes). Ex. multiple binary traits measured in the same individuals. Each response y_{ij} depends on the latent variable z_i (q-dim) via softmax function using the weight loading matrix W_j (dim. $q \times C$), and there are R such matrices, $1 \le j \le R$. Consider each of the response j:

$$z_i \sim N(0, I) \qquad y_{ij}|z_i, \theta \sim S(W_j^T z_i + w_{0,j}) \tag{3.121}$$

  where $S(\cdot)$ is the multi-class logistic regression above. Discrete data can be similarly visualized using categorical PCA (Figure 12.18)

- Supervised PCA: both x_i and y_i depends on common latent variables z_i (Figure 12.19 (a)):

$$z_i \sim N(0, I) \qquad y_i|z_i \sim N(w_y^T z_i + \mu_y, \sigma_y^2) \qquad x_i|z_i \sim N(W_x z_i + \mu_x, \sigma_x^2 I) \tag{3.122}$$

  To make inference, we can marginalize z_i and infer the conditional distribution $p(y_i|x_i)$ in terms of w_x (matrix) and W_y (vector). This regression is called "information bottleneck". The idea is that: better to use information bottleneck to do regression on Y, instead of the original x.

- Extensions of supervised PCA: not all data have labels, but we can learn W_x via unlabeled data.

- Partial least square (PLS): a lot of covariance structure of x may have nothing to do with y, therefore, its better to do regression on y but using only relevant parts of x. So we assume x_i has two types of latent factors: x-specific factors, z_i^x, give covariance only specific to x, and shared factors, z_i^s, give common covariance between x and y. The model can be written as: Equation 12.83-12.85 and see Figure 12.19 (b).

- Canonical correlation analysis (CCA): similar to PLS, but make it more symmetric: y_i can also have specific latent factors. See Figure 12.19 (c). Note: in CCA, we do not have to distinguish explanatory variables and response, so the model can be generally applied to joint analysis of multiple related datasets.

Independent Component Analysis (ICA) [Murphy, 12.6]

- Why ICA? PCA solves only half the problem as the likelihood is invariant to rotations.

- ICA model ideas: use non-Gaussian distribution for latent variables. Assumptions: z_js are independent, and variance equal to 1. Example: Figure 12.21, 2D sources are uniformly distributed, PCA does not work well because of normality assumption.

- Background: whitening of input data. Make covariance matrix of data equal to I. Definition: suppose X is a random vector with mean 0 and covariance Σ, then $Y = WX$ follows $N(0, I)$, where W is a whitening matrix $W^T W = \Sigma^{-1}$. This could be achieved by PCA: the PCs of the input matrix are orthogonal with variance 1.

- ICA model and likelihood: we assume X has be whitened. Let my t-th data point be x_t, then we have:

$$x_t = Wz_t \qquad p(z_t) = \prod_j p_j(z_{tj}) \tag{3.123}$$

  and variance of z_j is 1. Typically in ICA, we ignore the error. With this model we can show that W is orthogonal:

$$\mathrm{Cov}\,(x) = \mathrm{E}\,(xx^T) = W\mathrm{E}\,(zz^T)W^T = WW^T \tag{3.124}$$

  since $\mathrm{Cov}\,(x) = I$. Assuming that we know the non-Gaussian distribution of each z_j (Note: not a bad assumption, PPCA assumes all factors follow standard normal), we can now express the log-likelihood in terms of $V = W^{-1}$. Using the result about transformation of random variables. We can show that:

$$\log p(D|V) = T\log|\det(V)| + \sum_j \sum_{t=1}^{T} \log p_j(v_j^T x_j) \tag{3.125}$$

  where v_j is the j-th row of V. Note that the first term is constant, since det. of orthogonal matrix is 1 or -1, so we need to compute only the second term. We maximize this function subject to the constraint that V is orthogonal (actually orthonomal).

- FastICA algorithm: let $G(z) = -\log p(z)$, and assume it is the same for latent dimensions. Given V, we can compute the objective function above. To maximize this function wrt V, we can do gradident descent or Newton's method (FastICA). Both derivative and second derivative of the objective can be computed.

- Background: k-th central moment of a random variable X is defined as $\mu_k = \mathrm{E}\,(X - \mathrm{E}\,(X))^k$. If X is normal, we can show that $\mu_4/\sigma^4 = 3$. This is based on the moment of chi-square distribution, which can be computed with MGF.

- Modeling source densities with Non-Gaussian distributions: we have so far assumed source distributions are known. In general, we need to choose an appropriate foorm. z is super-Gaussian, if kurtosis of z is positive:

$$\mathrm{kurt}(z) = \mu_4/\sigma^4 - 3 \tag{3.126}$$

  This means z has long tail (spike near mean). Ex. Laplacian distribution. We say z is sub-Gaussian, if $\mathrm{kurt}(z) < 0$. Other possibilities are skewed distribution. In practice, super-Gaussian distribution is common.

- EM algorithm: another strategy is to estimate the source densities, which are assumed to follow a mixture of normal. Key observation of the EM is: we can computer $\mathrm{E}\,(z_t|x_t, \theta)$. And then given z_t's, we can estimate the parameters of mixture normal. These two steps will be alternated.

- Remark: in the analysis, data is assumed to be whitened first, and then W is orthogonal. So the dimension of x is different from the original data. This would affect interpretation of W.

Sparse coding [Murphy, 13.8]

- Motivation: (1) Topic model: each document, a set of words, covers multiple topics. (2) Image analysis: each patch represents one or more topics (content). The number of topics can be very large. (3) Transcriptome of cells: each cell can express any type of programs (or combination).

- Topic model by factor analysis: let x_i be our document/image patch/etc, which is a vector of word counts. Let z_i be the topics of the document. We can model x_{ij}, the word count of document i, as summation of the contribution of each topic on word j, $x_i = Wz_i$. The matrix W is known as Dictionary.

- Sparsity inducing prior on z_i: when the number of topics is large, it may be reasonable to assume z_i is sparse. Our problem is to infer both W and z_i, and this can be approximated by minimizing over W and Z:

$$-\log p(D|W,Z) = \frac{1}{2}\sum_i \|x_i - Wz_i\|^2 + \lambda \sum_i \|z_i\|_1 \tag{3.127}$$

  Optimization by iterative algorithm: When W is given, minimizing Z is just Lasso; when Z is given, minimizing W is least square.

- Connection with basis functions/wavelet analysis: we can view $\{x_i\}$ as a set of data points (in spatial and temporal dimensions), then they can be thought of as linear combination of underlying signals, e.g. wavelets.

- Compressed sensing (CS): e.g. in MRI, we do not observe x_i directly, but rather linear combination $y_i = Rx_i + \epsilon_i$. The goal is to infer x_i from y_i and given R. The idea of CS is that we leverage prior of $x_i = Wz_i$, where W can be a (large) dictionary, and z_i sparse-inducing prior.

- Application in image processing: we use a large database of images to train a Dictionary first: the "topic" of each possible patch. Then for any given image, we can learn the topics of each patch. Image inpainting: removing text in an image. We model image patches with the sparse coding model: some topics are images, and the rest are text. We select only the topics related to image to reconstruct.

Enter the Matrix: Factorization Uncovers Knowledge from Omics [OBrien and Fergit, TIG, 2018]

- Behavior of PCA: tries to explain variations using a small number of factors, while biological pathways may be relatively uniform. So a PC may mix signals from multiple processes.

- Behavior of ICA and NMF: better than PCA in associating factors with processes. ICA: may have both over- and under-representation of genes, while the non-negative constraint of NMF may help avoiding under-representation.

- Hierarchical nature of factors: e.g. ICA on tumor and normal samples, using two factors only separate tumor vs. normal; use more factors may separate subtypes of tumor samples.

- Using factors to find biomarkers of sample patterns (e.g. subtypes): associate factors with subtypes, then associate genes with factors.

Analysis of population structure: a unifying framework and novel methods based on sparse factor analysis (SFA) [Engelhardt and Stephens, PLG, 2010]

- Factor analysis: Let G be $n \times p$ genotype matrix, where $G \in \{0,1,2\}$. Factor analysis can be generally written as:

$$\mathrm{E}(G) = \Lambda F \Leftrightarrow \mathrm{E}(G_{ij}) = \sum_{k=1}^{K} \Lambda_{ik} F_{kj} \tag{3.128}$$

  where Λ is $n \times K$ matrix, calling *factor loading*, and F is $K \times p$ matrix, called *factors*. The term factor loading means: how each factor is loaded to a sample. See (Figure 1) for the dimensions of Λ and F.

- Factor analysis vs. PCA and mixed membership model: The general model reduces to the mixed membership model or PCA depending on the constraints on Λ and F. (1) Mixed membership model:

$$G_{ij} \sim \mathrm{Bin}(2, r_{ij}) \qquad r_{ij} = \sum_k \Lambda_{ik} F_{kj} \tag{3.129}$$

  where Λ_{ik} is the population composition of sample i, and F_{kj} is the frequency of SNP j in population k. (2) PCA: we usually assume standardized genotype matrix G_{ij}, then we have: $G_{ij} \sim N((\Lambda F)_{ij}, \psi^{-1})$. The constraints are: K rows of F are orthonomal and K columns of Λ are orthogonal.

- Sparse factor analysis (SFA): the key idea is to induce sparsity of Λ: each sample is represented as a linear combination of a *small* number of latent factors. Specifically, the model is:

$$G_{ij} = \mu_j + \sum_{k=1}^{K} \Lambda_{ik} F_{kj} + \epsilon_{ij} \tag{3.130}$$

where $\epsilon_{ij} \sim N(0, \psi_i^{-1})$. The model with μ is called SFAm, and with $\mu = 0$ is SFA. The ARD prior encourages sparsity of Λ by:

$$\Lambda_{ik} \sim N(0, \sigma_{ik}^2) \tag{3.131}$$

Intuition: similar to ridge regression, mean 0 encourages small values of Λ_{ik}. In other words, suppose we know σ_{ik}^2, our model will shrink Λ_{ik} towards 0, with the extent of shrinkage determined by σ_{ik}^2. With this model, we can integrate out Λ. We write the genotype of sample i as:

$$G_i = \mu + F^T \Lambda_i + \epsilon_i \tag{3.132}$$

where $\Lambda_i \sim N(0, \Sigma_i)$ (Σ_i is diagonal matrix with diagonal elements σ_{ik}^2), and $\epsilon_i \sim N(0, \psi_i^{-1} I_p)$. Using properties of MVN, we have:

$$G_i \sim N(\mu, F^T \Sigma_i F + \Psi_i^{-1}) \tag{3.133}$$

where $\Psi_i^{-1} = \psi_i^{-1} I_p$.

- Inference of SFA: our unknowns are μ, F, Σ, Ψ, and Λ is missing variable. Note that main variables of interest are F and Σ or Λ. The inference (ECME algorithm) has two parts: (1) Suppose Σ is given, we can do standard EM, to update μ, F, Ψ, treating Λ as missing data. This involves maximizing the expected log-likelihood, where expectation is taken over Λ. The Q function is given by Equation (11) and (12) in the paper. (2) Suppose the other parameters are given, to update Σ, we can maximizing the marginal likelihood, marginalizing Λ.

- Behaviors of three models in discrete populations (admixture): simulations with three populations (Figure 3), the results: SFA and admixture show interpretable factors, with loading close to 0 if a sample does not belong to a population, but PCA does not have the pattern. Remark: PCA requires factors to be orthogonal, which is not the case here.

- Behaviors of three models in continuous populations: (1) 1D isolation-by-distance. PCA: first factor is mean AF, and the second factor is the deviation from the mean - roughly location of a sample relative to the center. SFA and admixture: 2 factors are AFs at either end. (2) 2D isolation-by-distance. PCA: first factor is mean, and the second, third factors more closely capture spatial dimensions (diagonal), similar to SFAm. But SFA and admixture different.

- Remark: can we understand the behavior of PCA in the case of discrete populations? What linear combinations (of the admixed populations) will PCA find in order to satisfy orthogonality?

A Bayesian Framework to Account for Complex Non-Genetic Factors in Gene Expression Levels Greatly Increases Power in eQTL Studies (PEER) [Stegle and Winn, PLCB, 2010]

- Background: (1) PCAsig: complexity control via significance testing of eigenvalues. (2) SVA: similar to PCASig for complexity control, also per-gene noise model and allow for sparse non-orthogonal components.

- VBQTL model ideas: gene-specific noise model; joint inference of genetic effects, known and hidden factors; ARD prior (shrinkage) estimates of the effects of known and hidden factors.

- Model: let y_{gj} be the expression of gene g of sample j. The error is $1/\tau_g$. The expression has three parts, from: genetic effect, known factors and hidden factors, denoted as $y_{gj}^{(1-3)}$. For genetic effects,

consider N SNPs, let s_{nj} be the genotype of SNP n in sample j, and b_{ng} be the indicator of whether SNP n is associated with gene g, and u_{ng} its effect size. We have:

$$\mathrm{E}\left(y_{gj}^{(1)}\right) = \sum_{n=1}^{N} b_{ng} n_{ng} s_{nj} \qquad b_{ng} \sim \mathrm{Bern}(p), u_{ng} \sim N(0,1) \tag{3.134}$$

For known factors, denoted as f_{cj} for factor c, we have:

$$\mathrm{E}\left(y_{gj}^{(2)}\right) = \sum_{c} \nu_{gc} f_{cj} \qquad \nu_{gc} \sim N(0, 1/\alpha_c) \tag{3.135}$$

And Gamma prior for α_c. Hidden factor model, let x_{kj} be the k-th factor in sample j, we have:

$$\mathrm{E}\left(y_{gj}^{(3)}\right) = \sum_{k} w_{gk} x_{kj} \qquad w_{gk} \sim N(0, 1/\beta_k) \qquad x_{kj} \sim N(0,1) \tag{3.136}$$

Similarly Gamma prior for β_k. ARD prior: note that β_k represents the prior importance of factor k (variance explained), when it is large (low effects), it will drive w_{gk} towards 0; and when it is small (large effects), it has less shrinkage towards 0 - hence named ARD.

- Inference: VB (Figure 2). Given other parameters, regress out their effects, and update the effect size parameters for hidden or known factors. Initial values from MLE. Two versions: fVBQTL - single update of the full model. It is appropriate when known and hidden factors are unrelated to genetics. iVBQTL: iterative update.
- Simulation: (1) Include 10 factors, 7 non-genetic and 3 genetic factors. (2) Comparison of recovered hidden factors (MSE). (3) Comparison of eQTL discovery: both cis- and trans. VB-QTL much better at trans-eQTL and also better at cis-eQTL.
- Results in cis-eQTL mapping: improve over PCA even when the same amount of variance is explained (Figure 4c).
- Results in trans-eQTL mapping: in yeast data, adjusting for hidden factors reduces the power (Figure 4d). This means some PCs are heritable, thus mediates genetic effects. Regressing out these PCs reduces the genetic effects.
- Remark: in classical Factor Analysis, W is fixed, and estimated via EM. Here, W is random, and shrinked towards 0 via ARD prior.
- Reference: details of Bayesian SFA at, "Inference algorithms and learning theory for Bayesian sparse factor analysis", Journal of Physics: Conference Series.

Joint Genetic Analysis of Gene Expression Data with Inferred Cellular Phenotypes (FAQTL) [Parts and Durbin, PLG, 2011]

- Motivation: PCA, PEER, SVA mostly capture broad variations in the data. The goal here is to learn latent factors capturing cellular phenotypes.
- Model: let $y_{g,j}$ be expression of gene g in sample j, we express it as:

$$y_{g,j} = \mu_g + \sum_{k=1}^{K} w_{g,k} x_{k,j} + \psi_{g,j} \tag{3.137}$$

where $x_{k,j}$ is the k-th latent factor in sample j, and $w_{k,g}$ the effect of factor k on g. The latent factors $x_{k,j} \sim N(0,1)$. Use spike-and-slab prior for the loading:

$$w_{g,k} \sim (1 - \pi_{g,k}) N(0, \sigma_0^2) + \pi_{g,k} N(0,1) \tag{3.138}$$

where σ_0 is small, use 10^{-4}. Note that it is assumed that y has been normalized, so w is also scaled to have variance 1.

- Note: in the full model, $y_{g,j}$ also depends on genetic effects and interactions.
- Set the prior $\pi_{g,k}$: the factors represent TF activity or pathways. 167 TFs and KEGG pathways. So we know if g is assigned to factor k. The prior is then chosen to reflect the link: if g and k are linked, use a large prior for $\pi_{g,k}$ (use 0.99); otherwise a small number - capture FP rate: 0.06 for ChIP-seq data (likely FP rate in ChIP-seq) and 0.001 for KEGG.
- Orthogonality of factors to experimental covariates: factors are shown mostly to represent pathways or TF activation, and not correlated with batches etc.
- Statistical identifiability: informative priors reduce ambiguity of factors. Shown that the factors are largely independent. Also do 20 random initializations.
- Association and interaction testing of latent variables: test the association of SNPs and latent factors. For each set of inferred latent factors, test association and obtain local FDR Q values using permutations to get empirical null (of all SNPs and all genes). Then average over 20 random starts to get the average q values. This is valid if we interpret local FDR as $P(\beta = 0|D)$ and random starts are interpreted as posterior sample.
- Inferred factors are often genetically driven: (1) TF factors: association with genotypes and environment. Ex: PHO84 locus > PHO4 targets. Another example, some locus is a cis-eQTL of the TF. (2) Pathway factors: LYS2 lous > lyn. biosynthetic pathway.
- Inferred TF activity correlate with expression: only 27/167 factors.
- Interactions of locus and inferred factors in determining gene expression: several TFs involved in metabolism, stress and IRF2 (stress response) show interactions in determining expression of some genes. Note: some interactions represent gene-environment interactions.
- Remark: the hyperparameters of the prior of loading matrix are not learned.

Empirical Bayes Matrix Factorization (FLASH) [Wang and Stephens, arxiv, 2018]

- Motivation: common distribution (to be learned from data) of l (loading of factors on individuals) and f (factors represented in terms of observed variables). The factors can be dense (affecting many variables) or sparse (affecting a small number). Ex: (1) Stock market: factors are not sparse, only factor > stock is sparse. (2) Gene expression: both [TF] and TF > gene are sparse.
- Note: the notations are different from other sources, such as Wiki and [Murphy].
- Single factor model: let Y be $n \times p$ data matrix, let l be the vector of loading (Z in the common notation of factor analysis) and f be the vector of factors (W in the common notation), we have:

$$Y = lf^T + E \qquad l_1, \cdots, l_n \sim g_l \qquad f_1, \cdots, f_p \sim g_f \qquad E_{ij} \sim N(0, 1/\tau_{ij}) \tag{3.139}$$

  where g_f and g_l are common distributions (e.g. ASH or normal). $\tau = (\tau_{ij})$ is unknown but can have some structure, e.g. the same for each column. The role of g_f and g_l is to impose regularization on the factors and loading.
- Variational inference: the goal is to infer the posterior $P(l, f, g_l, g_f, \tau|Y)$. Do this by Variational inference, we approximate the posterior distribution of l and f as $q(l, f)$, assuming independence of each component:

$$q(l, f) = \prod_i q_i(l_i) \prod_j q_j(f_j) \tag{3.140}$$

  The problem is then to maximize ELBO over q: $F(q_l, q_f, g_l, g_f, \tau)$. This is done via an algorithm alternating between l and f. When the distribution of l is given (posterior of l from the previous

iteration), inference of f is reduced to ASH type problem (EB normal mean problem): given the prior of f in g_f, and the data given f, Y_f, follows normal distribution (parameterized by l), we can infer g_f and f posterior. Similarly, we can infer l when the distribution of f is given.

- Reducing FLASH to EBNM problem: first, we state EBNM problem as:

$$Y_i \sim N(\mu_i, s_i^2) \qquad \mu_i \sim g(\cdot) \tag{3.141}$$

where s_i^2 are known. The goal is to infer μ_i. Next, for FLASH, let's assume f is given, since $K = 1$, we have:

$$Y_{ij} \sim N(l_i f_j, s_{ij}^2) \qquad l_i \sim g_l \tag{3.142}$$

where s_{ij}^2 is also known. This is N independent linear regression. We can derive $\hat{l}_i$ as MLE of l_i, which is a function of Y_{ij} and f - this is a sufficient statistic. Then $\hat{l}_l$ follows $N(l_i, s_i^2)$, this reduces to EBNM problem.

- K-factor model: (1) greedy algorithm, start with 1 factor, then add factors 2, 3, etc. (2) backfitting algorithm, iteratively refines the estimates for each factor given the estimates for the other factors. Selection of K: the estimation of g_l and g_f could lead to 0 as MLE. The algorithm then stops.

- Missing data: if "missing at random", in the VB iteration update, simply ignore missing data. This is implemented by setting $\tau_{ij} = 0$ for the missing entries (infinite variance, which leads to flat likelihood).

- Orthogonal cross-validation (OCV): for selection of hyperparameters (other methods) and comparison between methods. E.x. 3-fold OCV: permutation of row and column indices, and create held-out data as balanced part of data matrix - Appendix B. Use the training data to fit the model (treat the rest as missing data), and then use the factors and loading for the held-out part to compute missing entries and compare with the observed values.

- Simulation: start with 1 factor simulation, evaluate results by the low-rank structure $B = lf^T$, compare the estimated vs. the actual values by RRMSE, defined as:

$$RRMSE = \sqrt{\frac{\sum_i (\hat{B}_i - B_i)^2}{\sum_i B_i^2}} \tag{3.143}$$

Simulation setting: $N = 200, p = 300$, l_i ASH prior with $\pi_0 = 0.9, 0.3, 0$, and effect size variance $(0.25, 0.5, 1, 2, 4)$ with equal weights. And $f_i \sim N(0, 1)$. Error variance $\tau = 1, 1/16, 1/25$ under the three values of π_0.

- Assessing performance in real data: compare performance of imputing missing data.

- Remark: in VB iteration, we should be given the distribution (posterior) of f, rather than the values of f, so this is similar to measure-error regression, rather than standard regression?

Covariate-dependent negative binomial factor analysis of RNA sequencing data (dNBFA) [Dadaneh and Qian, Bioinfo, 2018]

- NBFA model: let n_j be expression vector of V genes in sample j. We model it as:

$$n_j \sim NB(\Phi\theta_j, p_j) \tag{3.144}$$

where Φ is the factor loading matrix (factor-to-gene effects), and Θ is factor score matrix. p_j is a parameter to account for overdispersion. Use a Dirichlet prior for Φ:

$$(\phi_{1k}, \cdots, \phi_{vk}) \sim \text{Dir}(\eta, \cdots, \eta) \tag{3.145}$$

where η controls smoothness, with small η favors more specific/sparse factors (most concentrate on a small number of genes).

- dNBFA model: allow factor scores to depend on covariates of sample j, x_j. Model:

$$\theta_{kj} \sim \text{Gamma}(r_k, e^{\beta_k^T x_j}) \tag{3.146}$$

  where r_k is the mean of factor k across all samples, and β_k the coefficients of covariates. Prior of β_k: mixture of normal distribution. Conjugate prior on hyperparameters.

- Inference: Gibbs sampling. 3000 MCMC iterations, where after the first 1000 burn-in iterations. Number of factors $K = 250$ initially, and then only the top 100 factors with non-negligible baseline expressions were kept for further analyses. For evaluting module memebership, use only top 20 genes by ϕ_{vk}.

- TCGA data analysis: BRCA, lung cancer. To evaluate modules: extract eigen-gene of each module, and test association with disease factor by t-test. Comparison with WGCNA. (1) Higher associations of disease factors (cancer vs. normal) with modules - p-value distribution (Fig. 3). (2) Compare factor scores across two sample groups (Fig. 4).

- ASD data analysis: 20-30 ASD and control expression in 3 brain regions. Covariates: age, sex and brain regions. Found stronger associations of modules in dNBFA vs. NBFA (Fig. 5).

3.4.4 Canonical Correlation Analysis (CCA)

CCA:

- Problem: let x be m-dim. and y be n-dim. vector. We believe there is a common structure between x and y. Ex. x, arithmetic speed and arithmetic power are related to y, reading speed and reading power. The goal is to find projections $a^T x$ and $b^T y$, where a, b are m and n-dim. vectors, s.t. $a^T x$ and $y^T y$ have max. correlation. The idea can be extended: once we find the first projection, we would like to find the second projection that is independent of the first projection, and max. the remaining correlation.

- Finding the first canonical correlation vector: see [Anderson, An Introduction to Multivariate Statistics, 3ed]. We first consider the first canonical correlation vectors, denoted as a and b. Our problem is to maximize $\rho(a, b)$. It is given by:

$$\rho(a, b) = \frac{a^T \Sigma_{xy} b}{(a^T \Sigma_{xx} a)^{1/2} (b^T \Sigma_{yy} b)^{1/2}} \tag{3.147}$$

  This is equivalent to solving this problem:

$$\text{maximize } a^T \Sigma_{xy} b, \text{ subject to } a^T \Sigma_{xx} a = 1, b^T \Sigma_{yy} b = 1 \tag{3.148}$$

  We can solve this using Lagrange's Multiplier. Define the function:

$$f(a, b, \lambda, \mu) = a^T \Sigma_{xy} b - \frac{\lambda}{2}(a^T \Sigma_{xx} a - 1) - \frac{\mu}{2}(b^T \Sigma_{yy} b - 1) \tag{3.149}$$

  Taking derivative wrt. a and b:

$$\begin{cases} \Sigma_{xy} b - \lambda \Sigma_{xx} a = 0 \\ \Sigma_{xy}^T a - \mu \Sigma_{yy} b = 0 \end{cases} \tag{3.150}$$

  With some algebra, we can show that $\lambda = \mu = a^T \Sigma_{xy} b$. Plugging in this to the equations can solve λ. To extend the results to next canonical correlation vector: we add the constraint that the vectors must be orthogonal to previous vectors.

- CCA results: Section 16.1 [Hardle et al. Applied Multivariate Statistical Analysis, 4ed]. Let Σ_{xx}, Σ_{yy} be covariance of x and y, respectively, and Σ_{xy} be the covariance between x and y. We define:

$$K = \Sigma_{xx}^{-\frac{1}{2}} \Sigma_{xy} \Sigma_{yy}^{-\frac{1}{2}} \tag{3.151}$$

  And we denote γ_i, δ_i be the i-th eigenvectors of KK^T and K^TK, respectively (assuming eigevalues are sorted from largest to smallest). Then our canonical correlation vectors are given by:

$$a_i = \Sigma_{xx}^{-\frac{1}{2}} \gamma_i \qquad b_i = \Sigma_{yy}^{-\frac{1}{2}} \delta_i \tag{3.152}$$

- Analogy with PCA: we consider the (normalized) covariance between X and Y, denoted as K. The canonical correlation vectors are singular vectors of K and K^T, respectively.

- Probabilistic interpretations of CCA: see [Bach et al, A Probabilistic Interpretation of Canonical Correlation Analysis], `https://www.di.ens.fr/~fbach/probacca.pdf`. Let x_1 and x_2 be data of m_1, m_2 dimensions, respectively. CCA is the MLE of the following model:

$$\begin{array}{ll} z \sim N(0, I_d) & d \leq \min(m_1, m_2) \\ x_1 | z \sim N(W_1 z + \mu_1, \Psi_1) & W_1 \in \mathbf{R}^{m_1 \times d} \\ x_2 | z \sim N(W_2 z + \mu_2, \Psi_2) & W_2 \in \mathbf{R}^{m_2 \times d} \end{array} \tag{3.153}$$

3.4.5 Mixed-Membership Model

Topic models overview:

- Motivation: in mixture model, each object belongs to one of multiple classes, each class with different distributions. However, we may want to model the case, where each object may belong to multiple classes, or each object is a result of "actions/contributions" of multiple classes. Ex. genotype of one individual: may come from multiple ancestral populations.

- Mixed-membership model: our basic data is an object, e.g. a vector of words, a set of pixels, expression of multiple genes, and so on. Each object can be decomposed as resulting from multiple components; or each element of the object can be viewed as a mixture of some underlying distributions. Examples:

  - Text analysis: each document, instead of having a specific model, consists of a mixed set of topics, in varying proportions.
  - Genetics: each individual (a set of genetic polymorphisms) consists of a mixed set of ancestral genotypes - some part of genome may come from one ancestral population, and some other part may be from another ancestral population, etc.
  - Computer vision: an image consists of a mixed set of objects, in varying proportions.
  - Gene expression: expression of a set of cells results from expression of individual types of cells and proportion of cell populations.

  The model is characterized by: (1) component distributions; (2) proportion of components. A simple way of making inference is non-negative matrix factorization (NMF).

- Comparison of mixed-membership and mixture models: in the mixture model, an object belongs to only one class (even though the class membership is not observed); in contrast, in the mixed-membership model, an object itself (usually a group, e.g. document, image) is a mix of multiple classes. Alternatively, in mixed-membership model, each object (document) is a mixture; but different objects (documents) share the same component distributions, with varying mixing proportions.

- Mixed-membership model is a special case of the hierarchical model: the group model is a mixture of distribution, this is similar to hierarchical linear model, where we model the regression coefficient or slope of a group, as a linear function of some other variables of the group.

- Latent variable model: the general idea of the topic model is to use a small set of latent variables to explain the observations. The idea can be applied even if there is no obvious group structure, e.g. all the documents are concatenated.

Latent Dirichlet Allocation (LDA) model

- Ref: [Introduction to Probabilistic Topic Models, Blei, 2011] `https://www.cs.cmu.edu/~mgormley/courses/10701-f16/slides/lecture20-topic-models.pdf`

- Dirichlet-Multinomial Mixture Model: the intuition is for any document, we first sample a topic, and then generate words according to the topic. Formally, suppose there are K topics in total, where each topic model defines a multinomial distribution on the dictionary, ϕ_k, with $\phi_k \sim \text{Dir}(\beta)$. Let θ be the proportion of these topics, $\theta \sim \text{Dir}(\alpha)$. Then for document m, we first sample its topic Z_m according to θ. Then we sample words at each position n, $x_{mn} \sim \text{Mul}(1, \phi_{Z_m})$.

- Latent Dirichlet Allocation: Dirichlet-Multinomial Admixture model. For a document m, the topic proportions are $\theta_m \sim \text{Dir}(\alpha)$. For a word at position n, we first sample its topic, $Z_{mn} \sim \text{Mul}(1, \theta_m)$, and then sample the word $x_{mn} \sim \text{Mul}(1, \phi_{Z_{mn}})$. The model can be simply written as:

$$\alpha \to \theta_m \to (Z_{mn} \to x_{mn}) \leftarrow \phi_k \leftarrow \beta \tag{3.154}$$

  where the parenthesis means repeat for every word of the document. And note that Z_{mn} is usually a K-dim vector (with only one element 1 and the rest 0), instead of a category variable.

- Inference: our unknowns are (Z, θ_m, ϕ_k). The Collapsed Gibbs sampler sample from $p(Z|X, \alpha, \beta)$, where θ and ϕ are marginalized. The algorithm iteratively sample Z_i (topic of a word, for simplicity, use i instead of mn), assuming everything else Z^{-i} is given. The conditional $p(z_i = k|Z^{-i}, X, \alpha, \beta)$ is basically the product of: (1) the probability of topic k is in document based on other words; (2) the probability the topic k contains the word at position i.

- Extensions: in several directions:
  - Structure of documents: documents may be associated with other data, e.g. authorship, time, networks (link), etc.
  - Topic structure: correlation and interaction between topics, tree structure of topics, etc.
  - Topic model representation: instead of bag-of-words, allow sequential order of words, entities, etc.
  - Sparsity: learn sparse set of topics. This may be associated with other tasks.

- Applications of LDA (mixed-membership model): population genetics, computer vision (see above).

Extensions of LDA:

- Author-topic model [Rosen-Ziv & Smyth, UAI, 2004]: each author is associated with a set of topics (with varying proportions), θ_a; and for each document (whose authors are known), its topic proportions is a uniform mixture of the topics of all its authors. Or at the word level: first sample an author (uniformly), then sample the topic from this author, then sample the word.

- Dynamic topic model [Blei & Lafferty, ICML, 2006]: the topic model (words in a certain topic) and the corpus-level topic proportions change over time. The dsitributions α_t (prior parameter of θ) and β_t are modeled as random walk:

$$\alpha_t|\alpha_{t-1} \sim N(\alpha_{t-1}, \sigma^2 I) \tag{3.155}$$

  and similarly for β_t.

- Relational topic model [Chang & Blei, Hierarchical relational models for document networks, AAS, 2010]: the topics of neighboring documents should be close. This is modeled as: the link variable (whether two documents are linked) depends on the topic similarity between the two. Define variable $y_{d,d'}$ for each document pair, and $y_{d,d'}$ is modeled as logistic model:

$$P(Y = 1) = \sigma(\eta^T(\bar{z}_d \circ \bar{z}_{d'})) \tag{3.156}$$

  where $\bar{z}_d$ and $\bar{z}_{d'}$ are the observed topic proportions of d and d', and $\circ$ means inner product or similarity.

Supervised LDA:

- Motivations: why a good way of document classification
  - Unsupervised LDA followed by regression on topics: the problem is the topics discovered may not correspond to class labels. Ex. for classifying movie reviews (positive vs. negative), unsupervised LDA may find topics related to movie genres.
  - Regression using words: the document class depends on topics, and by regression on topics, one actually pool the (small) effects of many words of a topic, thus improve inference.
- Model: suppose Y_d is the label of document d, we modeled it as dependent on the topic proportion of d, $\bar{z}_d$: $Y_d \sim N(\eta^T \bar{z}_d, \sigma^2)$, where η represents the effect of each topic.
- Alternative models:
  - Separate modeling of positive and negative classes: the topics of the two classes are not directly related, while in reality, many topics of the two classes are probably shared with only some topics different.
  - Modeling class-specific models: each topic is assigned to be general or class-specific. The drawback of this strategy is that the assumption may be too strong, i.e. the difference between the two classes lies more in the importance of topics, rather than presence/absence.
  - Modeling Y_d as functions of θ_d (the average topic proportion): this would be equivalent to some regularization on topic proportion variables. However, conceptually, Y_d depends on specific topics discussed in a document.

Topic flow model [TopicFlow model: Unsupervised learning of topic specific influences of hyperlinked documents. AI-STAT, 2011]

- Idea: in a citation/hyperlink network, the topics of the linked documents are shared. This would allow better inference of topics of a document. Develop a flow model that capture the topic relations among network documents.
- Model: parameterize with f to represent the topic influence between linked documents (f perhaps influences the topic distribution of linked documents), then optimize the likelihood as usual.
- Question: is flow is a good metaphor? What does flow balance means? Ex. say $A \to B \to C$, require the flow of $A \to B$ to be equal to that of $B \to C$, however, the topical influences in the two cases are generally independent.

Model based visualization of structure in biological data [Kushal Key, Thesis defense, 2018]

- Part I. cell type compositions in bulk tissue. Topic model for gene expression: p_{ij} is the expression of sample i, gene j. The read counts are related to p_{ij} via multinomial model. The value of p_{ij} is $\sum_k \omega_{ik}\theta_{kj}$, where ω_{ik} and θ_{kj} sum to 1 for k.
- Application to GTEx: fitting all samples together with K=20, clusters learned are generally tissue-specific with relevant biological functions. Also refined structure using only brain expression data.

- Q: Adjusting for batches and population ancestry? Answer: no, difficult with multinomial model. So some of the clusters may capture these covariants.
- Q: Compare the inferred clusters with actual transcriptome data, e.g. scRNA-seq? Is there any way to use the reference transcriptome to guide the search of the factors?
- Part II. Ancient DNA. DNA damage patterns (sequencing errors): ancient, spike of C>T near 5 ends.
- Important features of C>T: two flanking bases, the base immediately before the break point (DNA fragment), distance to the 5 end.
- Model: x_{ij}, for sample i, and position/mismatch j, five features. Model distribution of 5 features (mutational spectrum: distribution of features/contexts of all mutations) as mixed membership model.
- Learn several mutation profiles: one of them is modern DNA, another reflects ancient DNA damage pattern.
- UDG treatment: removing C>T mismatches.
- Q: Mismatches in modern DNA due to DNA damage during lib. prep. or sequencing errors?
- Part III. model bird distributions. Data: species presence x samples (locations). Fit STRUCTURE model: sample x clusters, cluster x species. Results: Ex. blue cluster: coastal birds.
- Lesson: mixed membership model is related to both mixture model and PCA. (1) If we have a mixture model, it may be natural to extend to mixed membership if the mixture weights (but not components) vary. (2) Relation to PCA: our data can be explained as composition (linear combination) of some unobserved structure. Ex. gene expression data across conditions: composition of multiple transcriptional programs (say each corresponds to a TF).
- **Remark**: in cell type deconvolution problem (and general problem of learning structure), it may be important to adjust for known covariates. However, need to consider if the known covariates can influence the factors/cell types.

3.5 Multiple Hypothesis Testing

3.5.1 Frequentist Approach

Reference: [Applied Statistical Genetics, Chapter 4; Storey & Tibshirani, PNAS, 2003; Efron, Chapter 2-4]
Multiple hypothesis testing problem:

- Basic strategy: consider a set of m tests with a fixed procedure (decision rule). Each hypothesis is true or not (but fixed values). We imagine that the data are randomly generated according to the true value of the hypothesis, and the number of TPs, FPs, etc. are random variables (fixed rule on random dataset, and fixed hypothesis). We will formulate the errors, etc. in terms of the distribution of these RVs.
- Problem: suppose we are testing m hypothesis H_{0i} vs. $H_{1i}, 1 \leq i \leq m$. We denote $h_i \in \{0, 1\}$ the truth: it is 1 if H_{1i} is true and 0 if H_{0i} is true (these are parameters, and not considered random in the frequentist interpretation). The test statistic of the i-th hypothesis is T_i, and we decide whether H_{0i} is accepted or rejected based on T_i at level α: we obtain the p-value of T_i, p_i, and define

$$\tilde{h}_i = [p_i \leq \alpha] \tag{3.157}$$

Thus $\tilde{h}_i$ is an estimator of h_i under frequentist interpretation (thus random variable).

- TP, FP, TN, FP: they are defined based on h_i and $\tilde{h}_i$:

$$TN = \#[h_i = 0, \tilde{h}_i = 0] = U = m_0 - V \qquad FP = \#[h_i = 0, \tilde{h}_i = 1] = V \tag{3.158}$$

  where we use V to denote the number of false discoveries, and m_0 is the total number of tests where null model is true. Similarly,

$$FN = \#[h_i = 1, \tilde{h}_i = 0] = T = m_1 - S \qquad TP = \#[h_i = 1, \tilde{h}_i = 1] = S \tag{3.159}$$

  where S is the number of true discoveries, and m_1 is the total number of tests where alternative model is true.

- False discovery proportion (FDP): we call $R = S + V$ the number of positive predictions, and the FDP is defined as:

$$FDP = V/R \tag{3.160}$$

  The expectation of FDP is the false discovery rate (FDR):

$$\text{FDR} = \text{E}\left(\frac{V}{R}\right) \tag{3.161}$$

- Remark: relation to parameter estimation problem. In this case, h_i is a parameter, and T_i or p_i are our data, and $\hat{h}_i$ is our estimator. Instead of assessing individual estimator, $\hat{h}_i$, we evaluate the overall performance over all estimators. Comparing with usual problems, the distribution of $\hat{h}_i$ under $h_i = 0$ is known, but under $h_i = 1$ is unknown.

- Remark: relation to machine learning: evulation of a prediction procedure. In machine learning, we assume we are estimating y_i and a procedure predicts $\hat{y}_i$ for each y_i. The procedure can be evaluated by summing over the errors, with certain loss function (training error). In our problem, we assess $\hat{h}_i$ using a 0-1 loss function.

Family-wide error rate (FWER):

- Definition: the probability that any true null hypothesis is rejected:

$$\text{FWER} = P(V \geq 1) \tag{3.162}$$

- Bonferroni correction: to control for FWED at α, we reject null hypothesis for which $p_i \leq \tilde{\alpha}$,

$$P(V \geq 1) = P\left(\bigcup_{i=1}^{m_0}[\tilde{h}_i = 1]\right) \leq \sum_{i=1}^{m_0} P(\tilde{h}_i = 1 | h_i = 0) = \sum_{i=1}^{m_0} P\left(p_i \leq \tilde{\alpha}\right) = m_0\tilde{\alpha} \tag{3.163}$$

  Clearly, if we choose $\tilde{\alpha} = \alpha/N$, we could control FWER at α. Equivalently, we could say that to control for FWER using Bonferroni correction, we adjust the p-values by: $\tilde{p}_i = mp_i$, and then apply the cutoff at α.

- Sidak correction: the Bonferroni bound can be improved if the N hypothesis are independent (p_i are independent): suppose our rule is $p_i \leq \tilde{\alpha}$, then

$$P(a \geq 1) = P(\forall i \in N_0, \tilde{h}_i = 0) = 1 - (1 - \tilde{\alpha})^{N_0} \leq \alpha \tag{3.164}$$

  Solving this equation, we have:

$$\tilde{\alpha} = 1 - (1 - \alpha)^{1/N} \tag{3.165}$$

  The adjusted p-value can be shown as:

$$\tilde{p}_i = 1 - (1 - p_i)^N \tag{3.166}$$

- Weakness of FWER: in general, control for FWER tends to be very conservative in genome-wide settings: given that a reasonable number of significant findings will be reported, requiring only one FP would be too stringent.

False discovery rate (FDR):

- Definition: the expected proportion of FPs among all features predicted significant, thus it is simply precision in Information Retrieval. Formally: $FDR = E(\frac{V}{R})$. At $R = 0$, define $V/R = 0$. Thus we can write FDR as:

$$FDR = E(V/R|R > 0)P(R > 0) + E(V/R|R = 0)P(R = 0) = E(V/R|R > 0)P(R > 0) \tag{3.167}$$

  We call $E(V/R|R > 0)$ the positive FDR (pFDR). If m is large, we usually have $P(R > 0) \approx 1$, thus $pFDR \approx FDR$.

- Benjamini-Hochberg (B-H) adjustment: (Algorithm 4.21 in the book) adjust p value by: $p_i m/i$, and sort all adjusted p values, and choose those that are below a specified value α. This would control FDR less than α. Intuitively, suppose BH adjustment finds i hypothesis to reject, then at the threshold p_i, the number of FPs is $V \approx m_0 p_i$ and the number of positive predictions is $R = i$. The FDR is thus approximately $V/R \approx m_0 p_i/i \leq \alpha$ (by how BH selects i).

- Calculating FDR: suppose our test statistic is T, and we want to compute FDR at threshold t, assuming $T > t$ would indicate signifcance (rejection of H_0). We have:

$$FDR(t) = E\left[\frac{V(t)}{R(t)}\right] \approx \frac{E[V(t)]}{E[R(t)]} = \frac{m_0 \cdot P(T > t|H_0)}{R(t)} \tag{3.168}$$

  where $R(t)$ is the observed number of significant features and $P(T > t|H_0)$ can be estimated when null distribution of T is known (e.g. P value or permutation test).

- Extrapolation approach to estimate m_0: we draw the empirical distribution and the null distribution of T, and find the point where the two distributions diverge, denoted as d. Then m_0 can be extrapolated from the number of features below d (let it be k):

$$m_0 = \frac{k}{P(T < d)} \tag{3.169}$$

- Estimation of FDR in real data: Suppose in the real data, we have $R(t)$ cases where $T > t$ is true. And the distribution $T|H_0$ can be computed, from random sampling, permutation, or from analytic computation, and assuming $m_0 \approx m$, we have $E(V(t)) = m \cdot P(T > t|H_0)$. Then we have $FDR = V(t)/R(t)$.
  - Simulation to obtain $V(t)$: instead of obtaining $T|H_0$ distribution, we can also obtain $V(t)$ directly by simulation. Ex. sample/permutate data s.t. none of the hypothesis tested is true, and compute T for all tests, and count the cases where $T > t$.
  - In the case when the statistic is P-value: then $T|H_0$ follows uniform distribution, and FDR can be easily calculated [Zhong & Schadt, AJHG, 2010].
  - Examples: (1) binding site prediction: random permutation of motifs and predict motif matches [BLS, GR07]; (2) pathway statistic in GWAS: permutation of phenotype labels to obtain the null distribution of pathway statistic [Wang & Bucan, AJHG, 2007]; (3) de novo mutations, the FDR at 2 de novo events in a single gene: simulate mutations randomly and count the number of genes with 2 hits.

- q value: this is motivated by assessing the significance of any individual feature (just as p value). Define q value of a particular feature as the expected proportion of FPs among all significant predictions (i.e. FDR) if we call that feature significant.

- Remark:
  - Intuition for B-H adjustment: first sort all p values, at $i = 1$, Bonferroni correction. Then at $i = 2$, since $i = 1$ is "safe" (below the threshold), then we could relax by using $2 \cdot q/m$ as threshold, and so on.
  - Both B-H adjustment and q value calculation assumes independence or at least weak-independence of multiple features.

Summary statistics under multiple hypothesis testing: [Laird & Lange, Fundamentals of Modern Statistical Genetics]

- Summary statisics: similar to FDR, we can use some other statistics to summarize the findings under multiple hypothesis testing. Ex. suppose t_i is the test statistic of the i-th hypothesis, we could calculate: (1) $\min t_i$; (2) the number of tests with $t_i < t_C$ for some threshold t_C; etc. Then the significance of these statistics would suggest the validity of the test.
- Null distribution of summary statistics: often difficult to calculate, resampling is a common way of obtaining the P-value of the summary statistics.

Example: search for proteins similar to a query protein in a database. Suppose the score function S between the query and any database protien is given:

- Extreme value theory: suppose the best match in the database search has score $S_{\max}$, we want to test if it is significant. The null distribution of $S_{\max}$ is simply the maximum value of N independent RVs (where N is the database size), each following a distribution under the hypothesis that the subject is unrelated to the query. This distribution of maximum can be approximated by the extreme value theory. The drawback of this approach: only for the best match in the database search.
- Bayesian approach: suppose S represents the log-BF of the comparison of two hypothesis, then S can be transformed to the posterior odds or the posterior probability that the subject is related to the query. Suppose we set a target threshold of posterior odds, then the threshold of S is determined by the prior odds. We could choose a smaller value of prior odds when N gets larger, to penalize for large databases; alternatively, we could fix the prior odds (assuming there is a constant ratio of true positives).

Genome-Wide Significance Levels and Weighted Hypothesis Testing [Roeder & Wasserman, Stat Sci, 2009]

- Weighted testing to control FWER: suppose we are testing m hypothesis, the test statistics $T_j \sim N(\xi_j, 1)$, where ξ_j is the parameter of the alternative model (unknown). The hypothesis being testsed are: $\xi_j = 0$. Let w_j be the weight of the j-th hypothesis, then we control FWER at α if we choose threshold:
$$\frac{p_j}{w_j} \leq \frac{\alpha}{m} \tag{3.170}$$
as long as $\sum_j w_j = m$.
- Theoretically optimal weights: when ξ_js' are known, the optimal weights that maximize the number of discoveries can be derived. The power of a single test is:
$$\pi(\xi_j, w_j) = P\left(T_j > \Phi^{-1}(\frac{\alpha w_j}{m})\right) = \Phi(z_{\alpha w_j/m} - \xi_j) \tag{3.171}$$
where Φ is the normal CDF. We are interested in maximizing the power over all tests, defined as:
$$R(w) = \sum_j \pi(\xi_j, w_j) = \sum_j \Phi(z_{\alpha w_j/m} - \xi_j) \tag{3.172}$$

subject to $\sum_j w_j = m$. The optimal weights are given by: $w = (\rho(\xi_1), \cdots, \rho(\xi_m)$, where

$$\rho(\xi) = \frac{m}{\alpha}\Phi\left(\frac{\xi}{2} + \frac{c}{\xi}I(\xi > 0)\right) \tag{3.173}$$

where c is a normalization constant s.t. sum of w_j is 1. So the optimal weight depends on ξ_j, but also on other ξ's through the constant c. The paper shows figures of the function $\rho()$ at different values of c. Under some values of c, for the tests with large power (large ξ_j), it is better to reduce the threshold (higher weights). Under other values of c, it is better to give higher weights to intermediate values of ξ_j.

- Choosing external weights: a special case of assigning weights is binary scheme, where k hypothesis have weights w_1 and the rest w_0. Let $\epsilon = k/m$ and $B = w_1/w_0$. In practice, we assume ϵ is given, and need to assign B. Show that this can be done with given ξ.

- Estimating weights from data: in practice, ξ_j is unknown. The idea is to create multiple groups of hypothesis, and assume a mixture distribution of T_j (null and alternative), and within a group, the same alternative distribution. Specifically in group k, the i-th test:

$$T_{ik} \sim (1 - \pi_k)N(0, 1) + \pi_k N(\xi_k, 1) \tag{3.174}$$

  And we estimate π_k and ξ_k.

- Remark:
  - In multiple testing problem (or testing problem in general), the goal is to **maximize power while controling for false positives** (types I error or FDR).
  - The power (and for estimation problem, SSE) under frequentiest statistics depends on the true values of parameters, which are unknown. One strategy is to use Empirical Bayes to effectively estimate the prior mean/distribution of parameters.

Optimal Multiple Testing Under a Gaussian Prior on the Effect Sizes [Dobriban & Owen, arxiv, 2015]

- Motivation: in the work by [Roeder & Wasserman, 2009], the optimal weights are determined for known (or estimated) prior mean effect (ξ_j). However, this may be known only approximately, e.g. for testing one GWAS, we may use a different GWAS for weighting SNPs: the second GWAS is related, but effect sizes are probably different.

- Model: let test statistics be $T_i \sim N(\mu_i, 1)$, however μ_i is unknown. Instead, we have its prior $\mu_i \sim N(\eta_i, \sigma_i^2)$. Our goal is to determine the optimal weights using values of η_i and σ_i. To do that, we first integate over μ_i: $T_i \sim N(\eta_i, \sigma_i^2 + 1)$. Let $\gamma_i^2 = \sigma_i^2 + 1$. Use the same objective function (power over all tests), we solve the optimization problem:

$$\max_w \Phi\left[\frac{\Phi^{-1}(qw_i) - \eta_i}{\gamma_i}\right] \quad \text{subject to } \sum_j w_j = m \tag{3.175}$$

  where $q = \alpha/m$ is the threshold. One can show that when $\gamma_i > 1$, the function is not concave.

- Solution to the optimization problem: by maximizing the Lagrangian.

p values for high-dimensional regression [Meinshausen & Buhlmann, JASA, 2009]:

- Motivation: Lasso regression in high-dim., what is the significance of the features selected by Lasso? And how do we control FWER or FDR (for selected features)?

- Method: the procedure consists of several steps: (1) random partition the data into two equal groups (many times); (2) train the Lasso in one group, and computer p-value of any single feature chosen by Lasso in the second group (e.g. t-test); (3) obtain the distribution of the p-value of any feature, and then obtain the value ranked at top γ percentile in this distribution, as the adjusted p-value of this feature.
- Questions:
  - How to choose γ to control FWER or FDR?
  - Alternative approach: e.g. random permutation of data to obtain a null distribution of some Lasso statistic, and then get the p-value. What is the advantage of this approach?

Computationally Efficient Estimation of False Discovery Rate Using Sequential Permutation P-Values [Bancroft and Nettleton, 2013]:

- Goal: in a problem of testing multiple hypothesis where most are null (genome-wide testing), suppose we are using a permutation test. For most genes, we don't need to perform many simulations because after a small number, it will be clear that the observed stat has a large p-value. Thus one may stop earlier. This may produce a sequential permutation, with p-value discrete and not uniformly distributed.
- Idea: BH correction under the non-uniform distribution of p-values.

The functional false discovery rate with applications to genomics (fFDR) [Chen and Storey, Biostatistics, 2019]

- Motivation: quantitative informative variable to capture the prior probability of a hypothesis. Similar to stratified FDR control and IHW method, but is based on EB strategy.
- Model: let p_i be the p-values of i-th test. Let Z be the value of the informative variable, $Z \sim U(0,1)$, e.g. Z can be the quantile of a variable informative of hypothesis. Let $\pi_0(z)$ be the prior probability H_0 is true. Assuming p_i follows $U(0,1)$ under H_0 and $f_1(p|z)$ under H_1. We define the joint density of p and z (indicator) as:
$$f(p,z) = \pi_0(z) + (1-\pi_0(z))f_1(p|z) \tag{3.176}$$
Inference: $\pi_0(z)$ follows GLM or GAM or non-parametric. Joint estimation of $\pi_0(z)$ and $f(p,z)$.
- Application: eQTL testing (favoring close SNPs) and DEG (normalized per-gene read depth).

3.5.2 Efron's Empirical Bayes Approach

Reference: [Efron10, Chapter 2-6]
Bayesian FDR:

- p-values and z values: z values may convey more information than p-values in multiple testing problem. Ex. in a problem with more than 10,000 hypothesis, the departure from the null distribution is more evident with z values, and more details are revealed in the histogram (Figure 3.1).
  - Converting p-values to z-values: two-sided p values tend to be favored.
- Two-group model: suppose we are testing N hypothesis, and z_i is our test statistic for the i-th test. The null model has prior density π_0, and the alternative model π_1. The distribution of the test statistic under H_0 and H_1 are $f_0(z)$ and $f_1(z)$ respectively. And we also assume that π_0 is close to 1. Our goal is to estimate, given a decision region of z, how often do we make a false discovery (reject a null model when it is actually true)?

- Bayesian FDR and local FDR: suppose $\mathbb{Z}$ is our critical region of z. We could define the probability over $\mathbb{Z}$:

$$F_0(\mathbb{Z}) = \int_{\mathbb{Z}} f_0(z)dz \qquad F_1(\mathbb{Z}) = \int_{\mathbb{Z}} f_1(z)dz \tag{3.177}$$

  The mixture density and the mixture distribution are:

$$f(z) = \pi_0 f_0(z) + \pi_1 f_1(z) \qquad F(\mathbb{Z}) = \pi_0 F_0(\mathbb{Z}) + \pi_1 F_1(\mathbb{Z}) \tag{3.178}$$

  The FDR is defined as:

$$\phi(\mathbb{Z}) = P(H_0 | z \in \mathbb{Z}) = \frac{\pi_0 F_0(\mathbb{Z})}{F(\mathbb{Z})} \tag{3.179}$$

  We could also define local FDR at a single point z_0:

$$\phi(z_0) = P(H_0 | z = z_0) = \frac{\pi_0 f_0(z_0)}{f(z_0)} \tag{3.180}$$

  The two FDRs, also written as $\text{Fdr}(\mathbb{Z})$ and $\text{fdr}(z)$ respectively, are related by:

$$\text{E}\left[\phi(z) | z \in \mathbb{Z}\right] = \phi(\mathbb{Z}) \tag{3.181}$$

  The condition expectation of local FDR (over the mixture density $f(z)$) is equal to the FDR.

- Why local FDR? With local FDR, the decision of accepting a hypothesis or not is based on the local FDR. For example, at level α, we simply accept all hypothesis with local FDR below α.

- Hierarchical model approach to multiple hypothesis testing: an alternative way of modeling multiple testing, this is based on directly modeling data instead of through p-values. For example, suppose we want to test $H_{0i} : \mu_i = 0$ vs. $H_{1i} : \mu_i \neq 0$, with the data $z_i | \mu_i \sim N(\mu_i, \sigma^2)$. We could have a mixture model of μ_i, depending on whether H_{0i} is true:

$$\mu_i \sim g(\mu) = \pi_0 \Delta_0(\mu) + (1 - \pi_0) g_1(\mu) \tag{3.182}$$

  where $\Delta_0(\mu)$ is Dirac's delta function centered at 0, and $g_1(\mu)$ is the distribution of μ under H_{1i}. This allows one to estimate prior, once the prior is available, one can estimate the posterior probability that H_{0i} is true.

Empirical FDR estimates:

- Empirical estimate: if the null distribution $f_0(z)$ is known, we could form the empirical estimate. First, we estimate the mixture probability (the fraction of positive predictions)

$$\bar{F}(\mathbb{Z}) = \#\left\{z_i \in \mathbb{Z}\right\} / N \tag{3.183}$$

  Then we estimate the FDR as:

$$\bar{\text{Fdr}}(\mathbb{Z}) = \pi_0 F_0(\mathbb{Z}) / \bar{F}(\mathbb{Z}) \tag{3.184}$$

  When N is large, this is a good estimate of FDR.

- False discovery proportion (FDP): Bayesian FDR is defined as a probability; in a specific dataset, we are interested in knowning the FDP, defined as:

$$\text{Fdp}(\mathbb{Z}) = N_0(\mathbb{Z}) / N_+(\mathbb{Z}) \tag{3.185}$$

  where $N_0(\mathbb{Z})$ is the total number of null z_i falling into $\mathbb{Z}$, and $N_+(N_0(\mathbb{Z}))$ is the total number of z_i falling into $N_0(\mathbb{Z})$.

- Assessing the FDR estimator: under the independence assumption (z_i are independent), one can show that:
$$\mathrm{E}\left(\tilde{\mathrm{Fdr}}(\mathbb{Z})\right) = \mathrm{E}\left(\mathrm{Fdp}(\mathbb{Z})\right) = \phi(\mathbb{Z})[1 - \exp(-e_+(\mathbb{Z}))] \tag{3.186}$$
where $e_+(\mathbb{Z}) = NF(\mathbb{Z})$ is the expected total number of z_i falling in $\mathbb{Z}$. Therefore, $\bar{\mathrm{Fdr}}(\mathbb{Z})$ is an accurate estimator of $\phi(\mathbb{Z})$ when $e_+(\mathbb{Z})$ is large, say, $e_+(\mathbb{Z}) > 10$.

Estimating FDR with theoretical null:

- Estimator of FDR and local FDR: let π_0 be the proportion of H_0, f_0 and f_1 be the PDF of z-scores under H_0 and H_1 respectively, and similarly, F_0 and F_1 for the CDF. The estimator of FDR and local FDR are:
$$\hat{\mathrm{fdr}}(z) = \frac{\hat{\pi_0}\hat{f}_0(z)}{\hat{f}(z)} \qquad \hat{\mathrm{Fdr}}(z) = \frac{\hat{\pi_0}\hat{F}_0(z)}{\hat{F}(z)} \tag{3.187}$$
When the theoretical null is given, then we need to estimate π_0 and f from the data, $z_1, \cdots, z_n$ (for local FDR).

- Estimation of π_0 [Section 4.5]: the idea is to use a region where $f(z) = 0$ to estimate π_0, as in this region, all the observed z_i's are due to H_0 (called "zero assumption'). Suppose the region is denoted as $\mathbb{A}_0$, let $N_+(\mathbb{A}_0)$ be the observed number of points in the region, then we have the simple estimate:
$$\hat{\pi}_0 = \frac{N_+(\mathbb{A}_0)}{N \cdot F_0(\mathbb{A}_0)} \tag{3.188}$$

- Estimation of $f(z)$ [Section 5.2]: Poisson regression estimate. The idea is that suppose we divide the range into K bins, then the observed data points in each bin is a Poisson random variable with rate determined by the density function. Define y_k as the count in the k-th bin, $y_k = \#\{z_i \in \mathbb{Z}_k\}$, and let x_k be the center point of $\mathbb{Z}_k$. Then we have:
$$y_k \sim \mathrm{Pois}(Ndf(x_k)) \tag{3.189}$$
where d is the bin size. To fit $f(x_k)$ for all k, suppose $f(z)$ has the form:
$$f(z) = \exp\left(\sum_j \beta_j z^j\right) \tag{3.190}$$
Then we solve the parameters β_j using the counts. This is a standard Poisson GLM.

Why the theoretical null may fail? [Efron, Chapter 6]:

- Example: a microarray experiment, N genes, expressed in two different conditions (each condition, multiple samples, possibly correlated). Suppose we test the differential expression by a two-sample t-test, and plot the distribution of p_i or z_i of all genes. It is possible that the distribution is over-(more often) or under-dispersed. In the case of over-dispersion, we make more false predictions (i.e. we underestimate the FDR).

- Reason I: failed mathematical assumption. This would happen, for example, when we assume the data are normally distributed while in fact, there are significant outliers.

- Reason II: correlation among subjects/sampling units: in the gene expression example, the correlation between experiment conditions. The correlation between different subjects would also make the test statistic z_i not following theoretical distribution (when there are correlations, t-test may not be applicable).

- Reason III: correlation among test cases, in the gene expression example, the correlation between test units, i.e. genes. In contrast to Reason I and II, which apply to single cases (genes), this correlation would create departure from theoretical null even if individual $z_i \sim N(0,1)$. To see this, suppose z_i are correlated, and suppose our threshold is $z_i > 2.5$. In some cases, because of correlation, the number of points greater than 2.5 is higher than expected by chance, then assuming theoretical null would lead to a lower estimate of FDR.
- Reason IV: unobserved covariates. In the gene expression example, this would be other confounders, e.g. batch effect, that creates the difference of expression between conditions. This is the most common source of the failure of theoretical null.
- Permutation null distribution: for the gene expression experiment, this would be permuting the columns (experiments). This would address Reason I, but not Reason II (since the permutation would destroy the dependency between columns) and IV. This would preserve correlation among test cases (genes), since permutation would preserve this correlation, but of no direct assistance with Reason III.

Estimating FDR when the theoretical null fails: [Efron, Chapter 6]

- Estimating empirical null distribution: once $f_0(z)$ is estimated, we could use the same procedure to estimate π_0 and $f(z)$ as before. So the only problem for estimating FDR is to construct the empirical null distribution. First we note that we need the zero assumption, i.e. in some region $\mathbb{A}_0$, $f_1(z) = 0$, otherwise, the model is not identifable (since π_0 also unknown). Suppose $f_0(z)$ is normally distributed:

$$f_0(z) \sim N(\delta_0, \sigma_0^2) \tag{3.191}$$

  Our goal is to estimate $(\pi_0, \delta_0, \sigma_0)$.
- Center matching: near $z = 0$, assume $f(z) \approx f_0(z)$, thus $\log f(z)$ is a quadratic function. The task is to approximate the quadratic function using the counts z_i near $z = 0$. The simplest strategy is to perform the least square fit of $\log f(z)$.
- MLE: we consider all $z_i \in \mathbb{A}_0$, and assume $f_1(z) = 0$ if $z \in \mathbb{A}_0$. Let I_0 be the indices of those z_i, and N_0 be the size of I_0. Let $\mathbf{z}_0$ be the set $\{z_i : i \in I_0\}$, and $\phi_{\delta_0,\sigma_0}(z)$ be the density function of $N(\delta_0, \sigma_0^2)$. The likelihood function consists of two parts: (1) the probability of having N_0 points in $\mathbb{A}_0$: this is given by a binomial distribution; and (2) the probability of generating $z_i \in \mathbb{A}_0$: this is given by the normal density, conditioned on the fact that the point generated falls in $\mathbb{A}_0$. We have:

$$f(\mathbf{z}_0|\delta_0, \sigma_0, \pi_0) = \binom{N}{N_0} \theta^{N_0} (1-\theta)^{N-N_0} \prod_{i \in I_0} \frac{\phi_{\delta_0,\sigma_0}(z)}{H_0(\delta_0, \sigma_0)} \tag{3.192}$$

  where

$$H_0(\delta_0, \sigma_0) = \int_{\mathbb{A}_0} \phi_{\delta_0,\sigma_0}(z) dz \tag{3.193}$$

  and $\theta = P(z_i \in \mathbb{A}_0) = \pi_0 H_0(\delta_0, \sigma_0)$.
- Problem of the method of estimating FDR using zero-assumption: in general, $f_1(z) = 0$ in the selected region $\mathbb{A}_0$ or the center (for the center matching method) is not realistic. This is particularly true for low-powered studies where one expect a significant overlap between f_0 and f_1. Even if the empirical null can be estimated, the zero assumption would overestimate π_0, leading to an overestimate of FDR.

3.5.3 Extensions of FDR

Covariate-modulated local false discovery rate for genome-wide association studies (cmfdr) [Zablocki and Thompson, Bioinfo, 2014]

- Background: QQ plots of SNPs in different categories (UTR, coding, etc.) show different distributions of SNPs.

- Model: Let z_i be test statistics, and x_i be covariates of i-th hypothesis. The model assumes that both prior probability of i and the test distribution depends on x_i. Let $f_0(z_i)$ be distribution of z_i under H_0, $N(0, \sigma_0^2)$. Let $f_1(z_i|x_i)$ be the distribution of z_i under H_1, with Gamma$(a(x_i), \beta)$. Model assumptions:

$$\text{logit}(\pi_1(x_i)) = x_i^T \gamma \qquad \log a(x_i) = x_i^T \alpha \tag{3.194}$$

  Prior: α, γ are normal with mean 0, and σ_0^2 and β follow Gamma and Inv-Gamma. Inference: Gibbs sampling.

LSMM: a statistical approach to integrating functional annotations with genome-wide association studies [Can Yang, Bioinfo, 2018]

- Model: let p_j be p-value of test j. Let γ_j be indicator, $p_j|\gamma_j = 0 \sim U(0,1)$ and $p_j|\gamma_j = 1 \sim \text{Beta}(\alpha, 1)$. The prior of γ_j depends on genic annotation Z_j (e.g. UTR, CDS) and tissue annotations A_j:

$$\text{logit}(P(\gamma_j = 1|Z_j, A_j)) = Z_j b + A_j \beta \tag{3.195}$$

  where b follows normal prior, and β follows spike-and-slab prior. The effect of Z_j: fixed effect; the effect of β treated as random (marginalize effect size).

- Remark: the model treats all SNPs as independent. This would not lead to correct interpretation of b, β, which would NOT be log-OR of causal variant probability.

Fast and covariate-adaptive method amplifies detection power in large-scale multiple hypothesis testing (AdaFDR) [James Zou, NC, 2019]

- Problem formulation: Let P_i be the p-value of the i-th test, and x_i be covariates. The goal is to determine FDR threshold for a given x, or $t(x)$. Let $D(t)$ be the power at threshold function t. The goal is to maximize $D(t)$ s.t. FDP constraint.

- Model of $t(x)$: a mixture of GLM and K-component Gaussian mixture (with diagonal covariance matrices).

3.5.4 Bayesian Approach

Bayesian decsion theory approach:

- Reference: A Bayesian Measure of the Probability of False Discovery in Genetic Epidemiology Studies [Wakefield, AJHG, 2007], Reporting and interpretation in genome-wide association studies [Wakefield, Int. J Epiderm, 2008]

- False positive report probability (FPRP): The idea is at the probability of achieving $T > t_{\text{obs}}$ under H_0 is p (the p-value), and the probability of achieving that under H_1 depends on the power. So we combine the two with a mixture model, and estimate the probability of H_0.

$$\text{FPRP} = \frac{p\pi_0}{p\pi_0 + (1-\pi_0) \times \text{Power}} \tag{3.196}$$

  This is not a strictly speaking, Bayesian approach, but tries to approximate the it. The drawbacks are: (1) information is lost by considering only $T > t_{\text{obs}}$. (2) does not provide control of FDR because a variable threshold for T is used.

- Bayesian false discovery probability (BFDP) is defined as posterior prob. of null, $P(H_0|D)$. Note that this depends on the prior probability of null π_0.

- The threshold of PPA (posterior probability of association) or BFDP: determined by the cost of false discoveries vs. false non-discoveries (posterior expected loss). It can be shown that the threshold should be chosen s.t.:

$$P(H_0|D) < \frac{C_\beta/C_\alpha}{1 + C_\beta/C_\alpha} \tag{3.197}$$

  where C_α and C_β are costs of a false discovery a false nondiscovery, respectively.

- BFDP and p-value: for single SNP trend test, p-values are equivalent to BFs with a specific prior (exactly the same ranking).

Direct posterior probability approach: [Murphy, Section 5.7.2.4], [Newton, Biostatistics, 2004, PMID:15054023]

- Why need multiple testing correction under Bayesian? We predict a hypothesis by its posterior, $P(H_{1i}|D)$, and could impose a cutoff (e.g. τ) on this posterior. However, it does not tell the false discovery rate, as the rate (average over all predictions) is certainly smaller than $1-\tau$. In other words, if we want FDR at α, use $1-\alpha$ as the posterior cutoff is too stringent.

- FDR of Bayesian multiple testing: let $p_i = P(H_{1i}|D)$, then our predictions are those with $p_i > \tau$. The posterior probability can be written in terms of Bayes factors as:

$$p_i = \frac{(1-\pi_0)\mathrm{BF}_i}{\pi_0 + (1-\pi_0)\mathrm{BF}_i} \tag{3.198}$$

  where π_0 is the prior proability of H_0. The total number of predictions satisfying the threshold τ is:

$$N(\tau, D) = \sum_i I(p_i > \tau) \tag{3.199}$$

  The number of false discoveries among these predictions is given by:

$$FD(\tau, D) = \sum_i (1-p_i) I(p_i > \tau) \tag{3.200}$$

  where $1-p_i$ gives the probability that H_{0i} is true (thus a false discovery). The FDR is given by:

$$\mathrm{FDR}(\tau) = E(\mathrm{FDR}|D) = \frac{FD(\tau, D)}{N(\tau, D)} \tag{3.201}$$

- Procedure of FDR control: suppose we want to control FDR at the rate α. Suppose there exists at least one test with $1-p_i \le \alpha$, then we can choose the largest τ s.t. the FDR is less than α [PMID:19822692].

- Remark: Bayesian FDR is defined conditioned on data. The false positives have different semantics under Bayesian interpretation: given the data, what is the chance that the (unknown) hypothesis is false, as opposed to: given H_0 is true, how often we have a better test statistic.

Issues of Bayesian multiple testing: for both decision theoretical and the direct posterior probability approach, we need:

- Determining π_0: this is specified a priori. For GWAS, it is typically 10^{-4} to 10^{-5} per SNP. Note that in the q-value approach, π_0 is estimated from data: this is relatively simple in the problem of determining differential expression, but much harder in GWAS where π_0 is usually much smaller.

- Assessing false discoveries by simulation: for both approaches, the results (the discoveries) depend on π_0, which is usually not known accurately, we may need to assess the performance empirically through simulation. Specifically, we simulate the data under H_0 (or permutation), and compute the BFs. Then

count the number of tests with BFs above a chosen threshold (expected) vs. the observed number. The enrichment would suggest the FDR:

$$\text{FDR} = \frac{\#\text{Expected}}{\#\text{Observed}} \tag{3.202}$$

Table 3 of [Wakefield, Int. J Epiderm, 2008]. The same analysis can be done via PPA or BFDP, see [Li & Maris, A hidden Markov random field model for genome-wide association studies, Biostatistics, 2009].

- Inflation of BFs: for the BFs to properly behave, they should satisfy certain properties. In parciular, under H_0, the expectation of BFs should be less than or equal to 1. In general, for genomics problems, most of the hypothesis tested are false, so one could check the overall BF distribution to see if there is inflation.
  - Remark: according to Xiaoquan Wen's paper, $\text{E}\,(BF|H_0) = 1$ - need to check if this is true. Intuitively, as sample sizes goes to infinity, $BF|H_0 \to 0$.

Bayes/non-Bayes compromise:

- Reference: Imputation-Based Analysis of Association Studies: Candidate Regions and Quantitative Traits [Servin & Stephens, PLG, 2007]
- FDR estimation through permutation: permutation to obtain the null distribution of BFs, and obtain the cutoff through FDR estimation. This is essentially the same approach used to assess false discoveries by simulation with decision theoretical or direct posterior prob. approaches.
- p-value calculation: one can also obtain the null distribution of BFs, then convert each BF to a p-value, and apply the standard approach of FDR control on the p-values.
- Remark: the permutation approach is less sensitive to prior parameters, so more robust.

Comparison of Posterior probability FDR control and permutation FDR control:

- Problem: suppose we have m tests with BFs, $B_1, \cdots, B_m$, and the corresponding posterior probabilities are $v_1, \cdots, v_m$. Let π_0 be the prior of H_0. The direct posterior probability FDR at the threshold $v_i > t$ is given by:

  $$\text{FDR}_d(t) = \frac{\sum_i I(v_i > t)(1 - v_i)}{\sum_i I(v_i > t)} \tag{3.203}$$

  Suppose we have the null distribution of BFs from permutation, the FDR based on permutation at the threshold b (choose b and t s.t. they are matched) is:

  $$\text{FDR}_d(b) = \frac{m\pi_0 \cdot P(B > b|H_0)}{\sum_i I(B_i > b)} \tag{3.204}$$

  where $P(B > b|H_0)$ is the probability of BF greater than b under H_0. Are these two generally equal?
- Analysis: suppose we choose b and t s.t. the denominators are identical (or we look at the FDR at top K predictions, where K is fixed). The numerator of permutaion FDR is independent of the alternative distribution of BF. On the other hand, the numerator of direct posterior FDR depends on BF distribution under H_1, or the power of the test - when the power is large, the numerator might be very small (all top K BFs are very large).

Conservative estimation of π_0 [Xiaoquan Wen, Robust Bayesian FDR Control with Bayes Factors, arxiv, 2013]

- Motivation: in Bayesian FDR control using posterior probabilities, if π_0 is underestimated, then the posterior v_i will be overestimated, leading to inflation of FDR control. The goal is to provide an upper bound of π_0, then if we replace π_0 in calculating v_i:

$$\hat{v}_i = \frac{(1-\hat{\pi}_0)\text{BF}_i}{\hat{\pi}_0 + (1-\hat{\pi}_0)\text{BF}_i} \tag{3.205}$$

  the FDR control will be guaranteed.

- EBF procedure: the idea is that if we roughly the distribution of BFs under H_0, then we could estimate the proportion from H_0 from the overall distribution of BFs. It is easy to prove that:

$$\text{E}\left(\text{BF}|H_0\right) = \int B f_0(B) dB = \int \frac{P(y|H_1)}{P(y|H_0)} f_0(B) dB = 1 \tag{3.206}$$

  where B is the BF and $f_0(B)$ is the PDF of B under H_0. To see this, at any value of B, consider the value of y s.t. the BF under H_0 near this y is close to B, then at the neighborhood, we have:

$$f_(B) dB = P(y|H_0) dy \tag{3.207}$$

  Plug in this to the above equation we have the integral of B under H_0 is equal to 1. The procedure is: we rank in increasing order of all m BFs, and choose the maximum d s.t. the mean of BF_1 to BF_d is less than 1 (thus these BFs are likely from H_0). Then we choose $\hat{\pi}_0 = d/m$. It can be shown that when m_0 is large enough (the number of tests from H_0), then $\hat{\pi}_0$ provides a conservative estimate of π_0, i.e.

$$P(\hat{\pi}_0 \geq \pi_0 | \pi_0) \to 1 \tag{3.208}$$

- Proof of EBF: we first show that this is true when $\pi_0 = 1$, essentially, we need to show that the mean of all m BFs would be less than 1. First we show that the top BF, BF_m cannot be very large. By Markov's Inequality, for any i, we have:

$$P(\text{BF}_i \geq m^2) \leq \frac{1}{m^2} \tag{3.209}$$

  Thus using the extreme value distribution:

$$P(\text{BF}_i < m^2) = \prod_i P(\text{BF}_i < m^2) \geq \prod_i \left(1 - \frac{1}{m^2}\right) = \left(1 - \frac{1}{m^2}\right)^m \to 1 \tag{3.210}$$

  Now because for each i, BF_i is bounded, the expectation $\text{E}\left(\text{BF}_i|H_0\right) < 1$. Apply WLLN, the mean of all BF_i would be less than 1. To move to the general case of $\pi_0 < 1$, we apply this special case to all BF_i's that are from H_0.

- QBF procedure: similar to EBF, but instead of considering the mean, we consider the γ-quantile of BF under H_0. Specifically, for the i-th test, suppose we have the null distribution of BF_i, and from this we obtain its quantile $q_{i,\gamma}$. Then our estimator of π_0 is:

$$\hat{\pi}_0 = \frac{\sum_i I(\text{BF}_i \leq q_{i,\gamma})}{m\gamma} \tag{3.211}$$

  Roughly speaking, the denomiator is the expected number of tests with BFs higher than γ-quantile; and the numerator is the expected number from H_0.

- Remark:

  - The idea is similar to the procedure of estimating π_0 by Storey et al: if we know the null distribution of the test statistic (either p-value or BF), we could use the fact to estimate the fraction from H_0 (assuming that in some range of statistics, most are from H_0).

- The proof of Proposition 1 (Appendix C) appears to be flawed. In particular, if the mean of a subset is less than 1, we cannot prove that the mean of the entire set is also less than 1 (imagine that the mean of all BFs from H_0 is close to 1, but we have a few from H_1 whose BFs are larger than 1, and this would push the mean of all BFs greater than 1).
- The analysis is not robust to H_1 model specification: when BFs are inflated, the FDR will still be underestimated.

EB normal means with correlated noise [Lei Sun, NHS talk, 2018]

- Problem of correlated noise: comparison of liver expression in GTEx across different groups (randomly partition samples into groups). In some cases, inflation (many more genes with low FDR) and others deflation. Observation: often change of shoulder comparing with standard normal, but not excess or depletion of tails. Existing framework is not adequate:
  - BH correction: correct on average, but in specific case, may fail (over or under-estimate false discovery proportion, FDP).
  - Efron's FDR control: normal mixture could not explain the pattern, e.g. $N(0, 2^2)$ would have a high shoulder, but also a long tail. This would lead to loss of power.
- Idea: consider $Z_i \sim N(0, 1)$ but they may be correlated, we fit the histogram of Z_i's to capture inflation or deflation. To do this, we model the CDF of Z_i (across all tests), $F_i(Z)$. Correlation will influence this distribution, but will not be directly modeled.
- Theory: let $F_i(Z)$ be the CDF of Z_i. Assume Z_i, Z_j are bivariate normal with correlation coefficient ρ_{ij}. Our goal is to approximate $F_i(Z)$. To do that, we note $\mathrm{E}(F_i(Z)) = \phi(Z)$, the CDF of standard normal. The variance depends on l-th moment correlation:

$$\bar{\rho}^l = 1/\binom{2}{n} \sum_{i,j} \rho_{ij}^l \tag{3.212}$$

  Using these results, we can approximate the histogram (PDF) as:

$$f(z) = \phi(z) + \sum_l w_l \phi^{(l-1)}(z) \tag{3.213}$$

  where $\phi(z)$ is the PDF of standard normal, $\phi^{(l)}(z)$ is the l-th derivative of standard normal, and w_l is given by $\bar{\rho}^l$.
- Remark: the correlation is defined as average over all pairs. If there is only local correlation, e.g. LD, it will not create inflation/deflation.
- Application to large-scale multiple testing: our test statistic $x_j = \theta_j + z_j s_j$, where θ_j represents signal, $\theta_j \sim g(\cdot)$, e.g. ASH, and s_j standard error. The term z_j represents correlation of noise and we have $z_j \sim f(\cdot)$, where $f(\cdot)$ is defined as above. To fit the model, ASH parameters π, and w for correlated noise, we solve constrained optimization problem, where we penalize large w_l (expect exponential decay), and $f(\cdot)$ is constrained to be non-negative at specified values (across a large range).
- **Lesson**: even with correlated noise, under null, hard to obtain tail (large Z score). Under signal, could obtain large tail. So the method is able to disentangle the two.
- Lesson: we can study the distribution of test statistics using CDF of test statistic for an individual test. We can study the expectation and variance of the distribution.
- Q: correlation of Z_i's is not directly modeled. How much can we gain if we explicitly model them, e.g. use factor model/low rank approximation?

3.5.5 Post-hoc Analysis

The dangers of post-hoc analysis:

- General idea: when we test a hypothesis that is not specified a priori, instead, the hypothesis is formulated from the data (the parameters of the hypothesis, the explanatory variables included in the hypothesis, and so on), then it is possible that the statistical test of the hypothesis may not have a valid type I error.
- Example: test the difference of frequency in two groups. Suppose we have $x_1 \sim \text{Bin}(n, p_1)$ and $x_0 \sim \text{Bin}(n, p_0)$, and test if $p_1 = p_0$. If p_0 is known, the test statistic in cases is:

$$T = \frac{x_1 - np_0}{\sqrt{np_0(1-p_0)}} \tag{3.214}$$

  which follows $N(0,1)$ as $n \to \infty$. If p_0 is unknow, we could use for example, a chi-square test. However, if we use control only to define $\hat{p}_0 = x_0/n$, and test $p_1 = x_0/n$, our test will be inflated. The new statistic:

$$T' = \frac{x_1 - x_0}{\sqrt{x_0\left(1 - \frac{x_0}{n}\right)}} \tag{3.215}$$

  As $n \to \infty$, the denominator approaches $\sqrt{np_0(1-p_0)}$ (the same as before), however, the numerator has a higher variance than previously:

$$\text{Var}\left(x_1 - x_0\right) = \text{Var}\left(x_1\right) + \text{Var}\left(x_0\right) = 2np_0(1-p_0) \tag{3.216}$$

  under H_0. So using this test, the variance and standard error is higher than 1, thus using the $N(0,1)$ as the null distribution of T will lead to inflated type I error.
- Example: feature selection in linear regression. Suppose we do regression of y over $D = 100$ variables x_j's, the significance of each x_j is derived from a F-statistic. To control FWER at $\alpha = 0.05$, we would demand the significance $P < \alpha/D = 5 \times 10G^{-4}$. Now suppose we use feature selection as a filter first, choose only features with $P < 0.05$ with a simple correlation test (single feature), then apply the same regression analysis, and demand $P < \alpha/m$, where m is the number of features passing the filtering step. Clearly, this test is inflated, as we somehow reduces the multiple testing threshold (from 100 to m, whose mean is about 5) on exactly the same data.
- Example: test the significance of a group of p-values. Suppose we have a set of m p-values, and our goal is to test if the group as a whole departs from the uniform null distribution (pathway association). One test, let T be the minimum p-value, and we test T against extreme-value distribution (the minimum); or we could use the Fisher's method of combining the smallest k p-value as test statistic with fixed k:

$$T = -2\sum_{i=1}^{k} \log p_i \tag{3.217}$$

  Now, suppose our k is not fixed, but chosen s.t. T is the most significant among all k's then the null distribution is not valid. To see this: suppose we want to control type I error at $\alpha = 0.05$, then under H_0, we have probability 0.05 that the minimum p-value is 0.05 (actually somwhat different), however, since we always choose k to make T more significant, the p-value of our T will be lower than 0.05, creating inflation.
- Remark: the general characteristics is that the form of hypothesis is derived from data, thus it appears more siginificant than it actually is. Some specific types may include:

  - Using parameters estimated from the data: another example is the genetic burden test, where the weights of variants are learned from data (and then fixed). The problem is: because the parameters are fixed (they are actually nuisance parameters), we underestimate the variance of the test statistic (part of it comes from the variance of the parameter).
  - Filtering: incorrectly reduce the multiple hypothesis testing burden. We are testing D hypothesis, but with filtering, we test only $m < D$ hypothesis, and fail to pay the cost of multiple testing.

- Reference: [Wiki, Post-hoc analysis], [Wiki, Testing hypotheses suggested by the data].

Solutions of post-hoc analysis:

- Motivation: in some cases, post-hoc analysis would be beneficial, and we want to still use it, but control for type I error. Examples:
  - Feature selection: in the linear regression example, with a very large number of features (e.g. $p > n$), it may be computationally expensive, or numerically unstable.
  - Increased power: in the pathway association example, choosing a fixed k may not be the best strategy: for some pathway, it is more powerful to use only the few top gene; for other ones, it is more powerful to combine multiple weak signals.
- Permutation test: this is the general approach of controling type I error.
- Independence of filtering: if the filtering step is independent of the main test, then it is safe to do the filtering. Example, [Gene-Environment Interaction in Genome-Wide Association Studies, Murcray & Gauderman, Am J Epiderm, 2009]

3.6 Resampling Methods

Permutation tests [Moore, Intro. to Practice of Statistics, Chapter 14]

- Aim: test if some effect could reasonably occur "just by chance".
- Method: suppose we have a test statistic that measures the size of the effect of interest. We test its significance by:
  - Compute the statistic for the original data.
  - Choose permutation resamples from the data without replacement in a way that is consistent with the null hypothesis of the test and with the study design. Construct the permutation distribution of the statistic from its values in a large number of resamples.
  - Find the P-value by locating the original statistic on the permutation distribution.

Applications of permutation tests:

- Two-sample problems: H_0 states the two populations are identical (e.g. same mean). If H_0 is true, any observation will not depend on which group it comes from. So form the permuation sample by randomly reassigning the groups of all data points.
- Matched pairs designs: suppose each observation is associated with a label. If there is no effect (difference of the mean), the observation will not depend on its label, thus randomly switch the two labels for each pair.
- Relationships between two quantitative variables: data $(x_i, y_i), 1 \leq i \leq n$ and use correlation coefficient as the measure of dependence. If the two variables are independent, then the values of Y will not depend on X. So we can form the permutation sample by randomly permuate y_i among all data points.

- Correction for multiple hypothesis testing (when multiple hypothesis are dependent): suppose we test M hypothesis and obtain test statistic $T_1, \cdots, T_M$ (e.g. test multiple markers for association in genetic studies), we are interested in testing whether the most significant one $T_m = \max\{T_1, \cdots, T_M\}$ is truly significant. Since the test statistics are dependent, there is no simple way of correction. We can do permutation test, which will take into account the dependence among T_i's.

Examples of permutation tests:

- Test if the cis-regulatory elements (CREs) are randomly distributed in the genome [Zhang & Gerstein, GR, 2007]: if it is randomly distributed, then where a CRE occurs will not depend on its genomic coordinates (regions), thus we form the sample by randomly permutating the positions of CREs.
- Test if two or more words (or motifs) co-occur more often than by chance (as predicted from the densitiy of individual words) [Sharan & Karp, Bioinfo, 2003]: permute the positions of these words, and count the co-occurrence of the word cluster under each permutation.

Permutation test in the presence of confounding variables [personal notes]:

- Example: suppose we have x: gender and y: income, we want to test if gender has an effect on income (discrimination). But z: education level is a confounding variable: it is associated with both (suppose males tend to be more educated). Suppose our test statistic is: the ratio of male and female income, $T = y(x = 1)/y(x = 0)$, then even if gender is not related to income, we would have $T > 1$ from education effects.
- Idea of permutation test with confounding variables: we need to maintain the association of both x and z and y and z. We can do this by: fix x_i, z_i, pair it with permuted y_j, z_j, but j is chosen s.t. $z_j \approx z_i$.
- Application in the example: we permute y_i to a new individual j s.t. education levels are the same $z_i = z_j$. Then under null model, $T_0 > 1$ as $y(x = 1) > y(x = 0)$ because males tend to have higher education and thus higher income. But T_0 would be lower than the real T if gender does have an effect independent of education, because permutation breaks the relation between gender and income.

3.7 Meta-Analysis

Reference: [BHHR - Borenstein et al, Introduction to Meta-analysis], [Meta-analysis in clinical trials, Der-Simonian & Laird, 1986]

Overview of meta-analysis: [BHHR, Chapter 1]

- Effect size: the research problem is the estimation of certain effect size: generally the relationship between two variables. This could be treatment effect, correlation, etc, but could also be simply some unknown parameters.
- Goal of meta-analysis: combining multiple studies to get a better estimation of effect sizes. This includes: (1) evaluation of heterogeneity of effect size; in particular, if this is heterogenious, what may be additional independent variable; (2) the summary effect.
- General procedure: (1) effect size of each individual study, and the variance of the effect size; (2) the summary effect is usually the weighted average of the indivudal effect size, where the weight often depends on the variance. Intuition: low variance means we have an accurate estimation of the effect size, thus should have a higher weight.

Why we need meta-analysis [BHHR, Chapter 2]

- Meta-analysis vs. narrative review: meta-analsis addresses two issues: (1) whether the effect size is consistent; (2) if yes, estimate effect size; if not, quantify the variance. Narrative review, in contrast, are based on p-values.
- Problems with narrative reviews: not taking power into account. Ex. power is 50%, then even if true effect is consistent, in half of studies, p-values will not be significant. This leads to the wrong conclusion that there are "conflicting" results.

Effect size indices: different ways of defining effect size, depending on the nature of problem and the experimental design (and whether indivudal studies are directly comparable). [BHHR, Chapter 4-5]

- Mean difference between two groups (D): suppose the means of two groups are μ_1 and μ_2 respectively, then the effect size is $\Delta = \mu_1 - \mu_2$. The estimator of Δ is given by:

$$D = \bar{X}_1 - \bar{X}_2 \tag{3.218}$$

  If we assume the variances of the two groups are equal, the variance of D is:

$$V_D = \frac{n_1 + n_2}{n_1 n_2} S_p^2 \tag{3.219}$$

  where S_p^2 is the pooled variance (see the equation of Pooled variance in Math-Physics Notes). If we do not assume equal variance, the variance of D is given by:

$$V_D = \frac{S_1^2}{n_1} + \frac{S_2^2}{n_2} \tag{3.220}$$

  where S_1^2 and S_2^2 are the sample variance of the two groups.

- Standardized mean difference between two groups (d): when individual studies are not directly comparable (difference variances), we need to standarize D across studies. For any given study, suppose the pooled sample variance is S_p, then the estimator of this effect size is:

$$d = \frac{\bar{X}_1 - \bar{X}_2}{S_p} \tag{3.221}$$

  The variance of d is approximated by:

$$V_d = \frac{n_1 + n_2}{n_1 n_2} + \frac{d^2}{2(n_1 + n_2)} \tag{3.222}$$

- Odds ratio between two groups for binary data (OR): suppose we have a 2 by 2 table, with counts A, B, C and D. The odds ratio is estimated by:

$$OR = \frac{AD}{BC} \tag{3.223}$$

  Typically we use log-odds ratio as the index of effect size (LOR), and its variance is given by:

$$V_{\log OR} = \frac{1}{A} + \frac{1}{B} + \frac{1}{C} + \frac{1}{D} \tag{3.224}$$

Fixed-effect model [BHHR, Chapter 11]:

- Model: assume that the true effect size in each study is identical, and the goal is to estimate this true effect size. Suppose Y_i is the effect size of the i-th study, $1 \le i \le k$, and the true effect size is μ. We have: $Y_i \sim N(\mu, V_i)$, where V_i is the variance of the i-th study.

- Summary effect: the likelihood function:

$$P(Y|\mu) = \prod_i P(Y_i|\mu) \propto \exp\left[-\sum_i \frac{(Y_i-\mu)^2}{2V_i}\right] \tag{3.225}$$

The MLE of μ is:

$$\hat{\mu} = \frac{\sum_i w_i Y_i}{\sum_i w_i} \tag{3.226}$$

where w_i is the weight of the i-th study, and given by $w_i = 1/V_i$. Thus the summary effect is the average of the effect sizes of indivudal studies, weighted by the inverse of variance.

- Distribution of summary effect: the estimator is the weighted sum of normal variables (Y_i), and thus has normal distribution:

$$\hat{\mu} \sim N\left(\mu, \frac{1}{\sum_i w_i}\right) \tag{3.227}$$

Random-effect model [BHHR, Chapter 12]:

- Model: assume that the true effect size of each study may be different, and is a sample of the true/population effect size, which is to be estimated. Let Y_i be the effect size of the i-th study, $Y_i \sim N(\mu_i, V_i)$, and $\mu_i \sim N(\mu, \tau^2)$.

- Summary effect: we first note that the distribution of Y_i, marganalizing μ_i is: $Y_i \sim N(\mu, \tau^2 + V_i)$. Then similar to the fixed-effect model, we have the likelihood function:

$$P(Y|\mu,\tau^2) \propto \prod_i \frac{1}{V_i+\tau^2} \cdot \exp\left[-\frac{1}{2}\sum_i \frac{(Y_i-\mu)^2}{V_i+\tau^2}\right] \tag{3.228}$$

Define weight $w_i^* = 1/(V_i+\tau^2)$, the MLE of μ is:

$$\hat{\mu} = \frac{\sum_i w_i^* Y_i}{\sum_i w_i^*} \tag{3.229}$$

Since τ^2 is unknown, in practice, we replace this with the estimation T^2. From [DerSimonian & Laird], we use MOM to estimate τ^2. Define a test statistic that captures the heterogeneity of effects, Cochran's Q:

$$Q = \sum_i w_i(y_i - \hat{\mu})^2 = \sum_i \left(\frac{y_i - \hat{\mu}}{\sqrt{V_i}}\right)^2 \tag{3.230}$$

So Q is a measure of the total dispersion (standarized). The MOM estimator of τ^2 is given by:

$$\hat{\tau}^2 = \frac{Q-(k-1)}{\sum_i w_i - (\sum_i w_i^2 / \sum_i w_i)} \tag{3.231}$$

when it is less than 0, it should be 0.
Proof: The expectation of Q is simple since y_i follows normal distribution (we use μ instead of $\hat{\mu}$ for simplicity):

$$\mathrm{E}(Q) = \sum_i w_i(\tau^2 + V_i) \tag{3.232}$$

Solving the MOM equation:

$$\sum_i w_i(\tau^2 + V_i) = Q \tag{3.233}$$

and the result is:

$$\tau^2 = \frac{Q-k}{\sum_i w_i} \tag{3.234}$$

The difference is due to the difference between $\hat{\mu}$ and μ.

- Distribution of the summary effect: similar to fixed-effect model, the weights are replaced with w_i^*.
- Remark: in random-effect model, MLE of τ^2 does not have a closed form. So use MOM approach, where the test statistic is intuitive: departure of the study effect from the overall mean.

Comparison of two models [BHHR, Chapter 13]

- Smoothing: the two models both use weighted average of individual effects, the difference being the weights. Under the random effect model, $w_i^* = 1/(V_i + \tau^2)$, thus comparing with the fixed-effect model, the influences of extreme studies are smoothed: studies with very small variance would have lower weight under the random effect model, similarly, studies with very large variance would have larger weight.
- Application/selection of model: in general, fixed-effect model is suited to studies where the design, experimental/intervention procedure, etc. are the same across studies (e.g. repeat of the same experiment), wherease random-effect model is more generally applicable to independent studies. One caveat is: when the number of studies is small, the estimation of τ^2 is poor, and the random-effect model may be limited (Bayesian analysis would be better).
- The practice of using fixed-effect first, then switch to random effect if the test of heterogeneity is significant. This should be strongly discouraged: the power of the test is often low. The decision of which model to use should be made before the analysis.

Measuring and testing heterogeneity [BHHR, Chapter 15-16]

- Motivation: in addition to estimating the summary effect, researchers often need to answer questions such as, is there difference in true effect size across studies? So we need to test this and quantify the extent of heterogeneity.
- Intuition: the variation of data has two parts: true variation of effect size (between-study variation) and sampling error (within-study variation). Our goal is to extract heterogenity from the total variation: the idea is that we can compare the total variation with expected variation when there is no heterogeneity.
- Interpretation of Q and testing heterogeneity: from the defintion of Q, when there is no heterogeneity (null hypothesis), it follows chi-square distribution with dof equal to $k-1$. So this allows us to consutrct a chi-square test of Q. In addition, we have:

$$\mathrm{E}\left(Q|H_0\right) = k - 1 \tag{3.235}$$

  However, Q itself depends on the number of studies, so is not interpretable/comparable.
- Measuring heterogeneity using T^2: from the definition of T^2, we know that it is a normalized measure of excess dispersion $Q - df$. It depends on the scale of effect size.
- Measuring heterogeneity using I^2: because $Q - df$ measures the excess dispersion, we can ask what proportion of total dispersion is due to excess dispersion:

$$I^2 = \frac{Q - df}{Q} \times 100\% \tag{3.236}$$

  It is a descriptive measure, ranging from 0-1 and insensitive to effect size scale and number of studies. Low values suggest that heterogeneity is probably low.
- Comparison of Q, T^2 and I^2 in measuring heterogeneity: Q and the p value from Q could test heterogeneity, but does not quantify the extent of heterogeneity. T^2 is sensitive to the scale of effect size and measures the variation of true effect size across studies. I^2 is entirely driven by Q and df, so it does not reflect variation of true effect size (Figure 16.7 of the BHHR book).

Combining p-values: [BHHR, Chapter 36]

- Sign test: comparing the number of studies where the effect is one direction vs. the number of studies of the other direction. Very simple, not use all the information (magnitude of effect).
- Comparison of using p-values and using effect sizes: the effect size meta-analysis is generally preferred because:
  - Effect size is often what is desired in practice, instead of p-values.
  - p values also depend on sample size, and do not fully reflect the effect size. Ex. two studies may have the same p-values, but very different effect sizes (because of different sample sizes).
- When to use p-value based test: (1) the sample size is not known, thus impossible to back-compute effect sizes; (2) the studies are very diverse, and it's meaningless to ask about a single summary effect, whereas one could ask whether any of the effect size is zero.
- Fisher's method of combining p-values: $X^2 = -2\sum_i \ln p_i$, where p_i is one-sided p-value. Under H_0 of no effect in each study, X^2 follows χ^2 distribution with df equal to $2k$.
- Stouffer's method of combining Z-scores: let Z_i be the Z-score computed from p-value (one-sided so that Z is standard normal distribution; if two-sided, Z is always positive), define

$$Z_{\text{Stouffer}} = \sum_i w_i Z_i \tag{3.237}$$

  where $\sum_i w_i^2 = 1$. Then under H_0 of no effect in each study, Z_{Stouffer} follows standard normal distribution (easy to check). In particular, when we combine results of multiple studies of different sample sizes (but the effect is the same across all studies), we have:

$$Z_{\text{Stouffer}} = \sum_i \sqrt{n_i} Z_i / \sqrt{n} \tag{3.238}$$

  where n is the total sample size.
  Proof: we consider the problem of testing $H_0 : \mu = 0$ in normal distribution, and suppose there are only two groups. For the i-th group, the data points are i.i.d. $N(\mu, \sigma^2)$, with sample size n_i. The test statistic is the Z-score:

$$Z_i = \sqrt{n_i}\frac{\bar{x}_i - \mu}{\sigma} \tag{3.239}$$

  while follows $N(0,1)$ under H_0. It is easy to show that:

$$\frac{\sqrt{n_1}Z_1 + \sqrt{n_2}Z_2}{\sqrt{n_1+n_2}} = \frac{n_1\bar{x}_1 + n_2\bar{x}_2 - (n_1+n_2)\mu}{\sigma\sqrt{n_1+n_2}} = \sqrt{n_1+n_2}\frac{\bar{x}-\mu}{\sigma} \tag{3.240}$$

  This is exactly the test statistic we would use if we combine the two datasets. This proves that combining Z-scores with the weighting above (meta-analysis) is equivalent to combining the two datasets (mega-analysis) when testing the mean of normal distributions.

Chapter 4

Regression Analysis

4.1 Overview of Regression Analysis

Some examples of linear regression: [Chatterjee]

- Effect of union law on the cost of living: state-level data of the cost of living (response variable), whether the law was implemented (predictor), other variables such as income level.
- Prediction of domestic immigrant rates: the total immigration into a state (could be negative). The predictors include state wage, unemployment rates, crime rates, etc.
- The water quality at many rivers as a function of agriculture, forest, commercial/industrial use in the neighborhood.

Perpsectives of linear regression:

- Perspective of variation: regression can be viewed as prediction from independent variables. But can be also viewed as the description of how variation of the response variable results from the variation of predictors, or equivalently, comparison of groups defined by values of independent variables. Thus if Y depends on a feature X_j, we interpret as: the groups with different values of X_j will have different Y.
- Predictive vs counterfactual interpretations: (1) predictive: the difference of the outcome variable between two groups that differ by 1 on average in the relavant predictor (on average: the effect that other variables are held constant); (2) counterfactual: everything else fixed, changing the predictor of one object results in the change of outcome (Note: on individual objects instead of groups).
- Linear regression vs. grouping/ANOVA: to study if X influences Y, we can build a linear model of Y vs. X; or study the groups defined by X, and see if their Y's are different (or effects are different - this allows controlling other covariates). The difference is that with the grouping approach: no assumption of linearity (i.e. incorporating interactions).
  - Example: salary survey data, the question is whether education and management affects salary. Define groups of subjects by education $\times$ management, and test if the salaries of the groups are different, while controlling for other variables.

General assumptions of linear model:

- Linearity: of coefficients. When the relation of Y over X_j is not linear, do transformation on X_j's. This assumption means that the effect of X_j on Y (could be linear or not) does not depend on other explanatory variables.

- Error: normality and independence.
- No measurement error.

Modeling procedure for regression:

- Formulate a problem: whether the goal is to test one particular variable (e.g. the union law example) or prediction.
- Collecting data: selecting control variables, variation of the main variable to be tested.
- Statistical model: choose a set of predictor variables, choose a form of model. May need to transform the variables, e.g. when Y is a linear function of X_1^2, we will need to transform X_1 s.t. the model is linear.
- Model fitting.
- Criticism and analysis: check assumptions and model dignostics. Examples: (1) outliers in the data: in the domestic immigration example, Alaska and Hawii are outliers that should be removed. (2) For prediction tasks, whether the variables fall into the range of the training data. Specific steps may include: residual plot, outlier detection, sensitivity analysis.

Setting up linear regression:

- Choose features/predictors: in general, a feature X should be chosen if the groups defined by X have different values of Y.
- Control variables: in many problems, the goal is to investigate the effect of one treatment variable on the outcome (some type of causal inference). In these cases, it is important to remove the effect of other variables that may influence the outcomes: the control variables.
- Post-treatment variables: should not be used as control variables when investigating the causal effect of treatments. They should be correlated with treatment variables, thus when doing regression, the effect of the treatment variable may be masked by the post-treatment variables.
- Feature expansion: may be functions of the independent variables - basis expansion.
- Interaction between features: two features interact if the effect (coefficient/slope) of one feature depends on the value of another feature. Ex. the child IQ as an outcome variable of the mother's number of high school years ($mom.hs$), and the mother's IQ ($mom.IQ$). The effect (slope) of $mom.IQ$ depends on $mom.hs$: when $mom.hs$ is small (no high school), $mom.IQ$ plays a large effect; however when $mom.hs$ is large, the effect of $mom.IQ$ is considerably smaller (education makes up for the deficiency of mother IQ).

General strategy for fitting a linear model: the idea is to find a model whose predictions agree with the observations. This can be done in two ways, broadly speaking

- Summary statistics: most commonly the moment of the data, the histogram, and so on. The summary statistics under the model (expected values) should match the observed values.
- Conditional distribution/expectation: intutively, we should have $y_i \approx \mathrm{E}\left(Y_i | x_i\right)$.

Model diagnosis: any statistical model is based on some assumptions of the data distributions, which may not be true. So it is important to inspect these assumptions. For example, for linear model, we will need to check the linearity assumption and the normality of error assumption. The insights gained can be used to improve the model.

- Plotting: either by plot y against x, or often, plot residual e_i against x_i, one can explore the linearity of the relationship, and whether the error is constant at different x. More generally, plot can reveal the unexpected relationship among variables. If some non-linear relationship is found, one may transform the variables to make it more linear.
- Measuring quality-of-fit: we can quantify how good a model fits the data. Two very general ideas are (1) The agreement of observations and predictions. (2) How much variation in the data is explained by our model.

Regression model with known properties of variables:

- Motivation: in high-dim. regression problem, we may have information of the properties of variables. It may be desirable to include such properties to improve estimation of the effects.
- Strategy 1: variable selection prior. The idea is that each variable has an indicator variable Z_j, whose prior depends on the properties through a regression model (or a similar model).
- Strategy 2: variable grouping. The idea is that we can divide the variables by their properties (within a group, variables would have similar effect sizes - a common prior distribution), and then estimate these group-level effect sizes.
- Example: to test if some annotation makes a SNP more likely to be causal SNP for a trait.
  - Variable selection prior: define annotations of SNPs as features, do regression model on the SNP prior.
  - Variable grouping: define groups of SNPs, e.g. promoters, tissue-specific enhancers. Within a group, all SNPs have the same prior of effect sizes. Estimation of the effect sizes of each group: this would allow one to control other groups when estimating the effect of one group (important when LD is present).
- Comparison of the strategies: similar to the situation where we can study the effect of X on Y through either regression or ANOVA (compare means of groups defiend by X).
  - Non-linear effects: variable grouping can more easily accommodate non-linear effects (e.g. groups are defined by the product of two features, or clusters of features).
  - Overlapping groups: when the variable groups overlap, it is easier to model with variable selection prior. With the grouping approach, one needs extra assumption about the effect sizes of variables belonging to multiple groups.

4.2 Analysis of Variance (ANOVA)

1. Introduction to ANOVA [KNNL, Applied Linear Statistical Models, 5ed, Chapter 15]

   Experimental and observation studies:

   - Problem: the effect of some "treatment" of interest on some experimental units. The difficulty is that there are often confounding/nuisance factors that cause variation of the response variables (but the interest is only in the treatment effect). All those variables that may influence the response variables are called factors, including treatment and confounding factors.
   - Example: compare yield of varieties 1 and 2. Need to compare in multiple blocks. However, each block itself has effect on the yield. Need to compare the two varieties in an unbiased fahsion.
   - Experimental studies: randomization is employed to assign a set of treatments to the experimental units. The causal relationship can be established, as the differences between the treatment and control groups are averaged out, and the only difference is thus due to treatment.

- Observational studies: random samples are obtained from multiple populations defined by the levels of one or more explanatory factors, referred to as observational factors. Usually, need external evidence to estabilish causal relationship.

Experimental designs:

- Complete randomized design: the treatments are randomly assigned. The linear statistical model for the response is:
$$Y = [\text{Constant}] + [\text{Treatment Effect}] + [\text{Error}] \tag{4.1}$$
- Factorial design: multiple factors, and the treatment combinations are randomly assigned. Ex. two factors, one has two levels and the other three levels, lead to the 2×3 factorial design. The linear model:
$$Y = [\text{Constant}] + [\text{First-order Treatment Effect}] + [\text{Interaction Effect}] + [\text{Error}] \tag{4.2}$$
- Randomized complete block design: the experimental units can be grouped into blocks, according some factors, and within the blocks, randomization of treatments is applied. The linear model:
$$Y = [\text{Constant}] + [\text{Treatment Effect}] + [\text{Block Effect}] + [\text{Error}] \tag{4.3}$$

Observational studies:

- Cross-sectional studies: Measurements of one or more subpopulations at a single time point or time interval. It provides a "snapshot" of the factors and the outcome variable.
- Prospective studies: one or more groups are formed according to the levels of a hypothesized causal factor, and these groups are observed over time wrt. an outcome variable of interest.
- Retrospective studies: groups are defined on the basis of an observed outcome, and the differences among the groups at an earlier time point are identified as potential causal effects.
- Matching: similar to blocking, treatment is assigned to a pair of matched units, which are identical in all aspects except treatment.

Overview of ANOVA strategy:

- Strategy: the basic goal is to explain how the observed variables vary with treatments, and other groups.
  - Model of factor effects (means of groups): suppose there is only one treatment, then we could have a model like:
  $$Y_{ij} = \mu + \tau_i + \epsilon_{ij} \tag{4.4}$$
  where Y_{ij} is the j-th observation of the i-th group, τ_i is the effect of the i-th group. If we view the treatment also as random, then we could have the relation between random variables:
  $$Y = \bar{Y} + X + \epsilon \tag{4.5}$$
  where X is the variation due to the (random) treatment.
  - Variance partition: a consequence of the factor effect model is that the total variance of the data can be partitioned according to groups. For example, from the equation above, we see that:
  $$\operatorname{Var} Y = \operatorname{Var} X + \sigma^2 \tag{4.6}$$
  We replace the variance with the sample variance, and we have the variance partition relation.
  - Inference of factor effects: if there is no treatment effect, we have the sample variance across the treatment group ($\operatorname{Var} X$) close to 0. This suggest that we could assess the variance (in comparison with σ^2) to test if the treatment effect is 0.

- The idea of ANOVA (variance partition) can be applied in many cases:
  - Quantitative genetics: the phenotype $P = G + E$, where G and E represent the independent influence of genotype and environment, respectively. Thus we have the partition: $V_P = V_G + V_E$. The variance partition could also be understood through regression of phenotype on genotype, and the SST of phenotype is the sum of SSR (genotype) and SSE (environment).
  - PCA: the variance is partitioned into orthogonal principal components
  - Fisher's LDA: the variance is partitioned into within-class and between-class variances.

  Note that when applying ANOVA, it may be that the group/factor is treated as a RV, instead of constant.
- Limitation of variance partition: (1) if some factor is important, but does not vary in an observational study, then its importance cannot be quantified in this way. Ex. essential genes will not vary in the population, thus the real importance not detectable. (2) orthogonality of sources is important, need correction if this does not hold.

2. One way ANOVA [KNNL, Applied Linear Statistical Models, 5ed, Chapter 16]

   One way ANOVA:

   - Cell means model: r groups/levels of the treatment factor, the mean of i-th group is μ_i (where the sample size is n_i), and the response variable can be expressed as:

   $$Y_{ij} = \mu_i + \epsilon_{ij} \qquad 1 \le j \le n_i \tag{4.7}$$

   where Y_{ij} is the outcome variable of the j-th sample in the i-th group.
   - Assumptions: $E(\epsilon_{ij}) = 0$, independent and normally distributed. Also in one way ANOVA, assume equal variance in groups, i.e. $\sigma_i^2 = \sigma^2$. The classical ANOVA hypothesis is:

   $$H_0 : \mu_1 = \mu_2 = \cdots = \mu_k \tag{4.8}$$

   - Factor effects model: the ANOVA model can be written equivalently as:

   $$Y_{ij} = \mu + \tau_i + \epsilon_{ij} \tag{4.9}$$

   where τ_i the effect of the i-th factor level, and ϵ_{ij} are independent $N(0, \sigma^2)$. To ensure identifiability of the model, we require that: $\sum_i \tau_i = 0$.
   - Relation to linear regression: ANOVA model is equivalent to linear regression model, where the factor levels (groups) are treated as indicator variables. The main diffrence between ANOVA and regression is: when the predictor variables are quantitative, ANOVA does not make any assumption about the nature of the statistical relation.

   Analysis of variance:

   - Partitioning of total sum of squares: we first note that the total variation (sum of square) $SSTO$ can be partitioned as the between group variation or treatment variation ($SSTR$) and the within group variation, or the error sum of squares (SSE):

   $$\sum_i \sum_j (Y_{ij} - \bar{Y})^2 = \sum_i n_i (\bar{Y}_{i\cdot} - \bar{Y})^2 + \sum_i \sum_j (Y_{ij} - \bar{Y}_{i\cdot})^2 \tag{4.10}$$

   Or written equivalently:

   $$SSTO = SSTR + SSE \tag{4.11}$$

   The term $SSTR$ measures the extent of differences between the estimated factor level means, and SSE measures the random variation of the observations around the estimated factor level means. The degree of freedom: $SSTO$ - $n_T - 1$; $SSTR$ - $r - 1$; SSE - $n_T - r$.

- Mean sum of squares: Then the mean sum of square defined as:

$$\begin{aligned} MSTR &= SSTR/(r-1) \\ MSE &= SSE/(n_T - r) \end{aligned} \tag{4.12}$$

The expected values of $MSTR$ and MSE are:

$$\begin{aligned} E[MSE] &= \sigma^2 \\ E[MSTR] &= \sigma^2 + \frac{\sum_i n_i(\mu_i - \mu)^2}{r-1} \end{aligned} \tag{4.13}$$

where $\mu = \sum_i n_i \mu_i / n_T$.

- F test: Note that when μ_i's are equal, we have $E[MSTR] = E[MSE]$, and if not, $E[MSTR]$ would be larger. The ratio of the two can thus indicate how different μ_i's are. The F test statistic is simply:

$$F = \frac{MSTR}{MSE} \tag{4.14}$$

the average between-group variation (difference of factor level means), normalized by the within group variation.

Alternative approach to ANOVA test: [Casella, Chapter 11]

- Goal: for some constants $a = (a_1, \cdots, a_k)$, test the hypothesis:

$$H_0 : \sum_{i=1}^{k} a_i \theta_i = 0 \qquad \text{vs.} \qquad H_1 : \sum_{i=1}^{k} a_i \theta_i \neq 0 \tag{4.15}$$

- Pooled estimator of within group variance: for a group i, the estimator of σ^2 is the sample variance:

$$S_i^2 = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{i\cdot})^2 \tag{4.16}$$

where $\bar{Y}_{i\cdot}$ is the mean of group i. Since σ^2 is shared, the pooled estimator is:

$$S_i^2 = \frac{1}{N-k} \sum_{i=1}^{k} (n_i - 1) S_i^2 = \frac{1}{N-k} \sum_{i=1}^{k} \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{i\cdot})^2 \tag{4.17}$$

- t-test: $\sum_i a_i \bar{Y}_{i\cdot}$ follows normal distribution with mean $\sum_i a_i \theta_i$, thus use t test. We reject H_0 if:

$$\left| \frac{\sum_i a_i \bar{Y}_{i\cdot}}{S_p \sqrt{\sum_i a_i^2 / n_i}} \right| > t_{N-k,\alpha/2} \tag{4.18}$$

The coefficient in the denominator is needed to account for a_i.

- Contrast: a special case of the above test is for a s.t. $\sum_i a_i = 0$. Ex. when $a = (1, -1, 0, \cdots, 0)$, this is the pairwise test: $\theta_1 = \theta_2$: reject H_0 if

$$\left| \frac{\bar{Y}_1 - \bar{Y}_2}{S_p \sqrt{1/n_1 + 1/n_2}} \right| > t_{N-k,\alpha/2} \tag{4.19}$$

- F-test: for any given a, let T_a be the test statistic (t-test above), i.e. we reject H_{0a} if $T_a > k$ for some k. Then the classical ANOVA hypothesis is equivalent to for any contrast a, $\sum_i a_i \theta_i = 0$.

Therefore if for some contrast a, the hypothesis H_{0a} is reject, H_0 will be rejected. We use the maximum of T_a^2 as the test statistic:

$$F = \frac{\frac{1}{k-1}\sum_i n_i(\bar{Y}_{i\cdot} - \bar{Y})^2}{S_p^2} \sim F_{k-1,N-k} \tag{4.20}$$

where $\bar{Y}$ is the population mean. To see F follows F distribution, note that both the numerator and denominator follow χ^2 distribution (independent).

3. Two-way ANOVA [KNNL, Applied Linear Statistical Models, 5ed, Chapter 19]

Two-way ANOVA model:

- Factor effects model: suppose we have two factors A and B, where A has a levels (indexed by i) and B has b levels (indexed by j), then we can write the outcome of the k-th sample in the ij cell as:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk} \tag{4.21}$$

where α_i and β_j are constants subject to the constraints: $\sum_i \alpha_i = 0$, $\sum_j \beta_j = 0$ and $(\alpha\beta)_{ij}$ are constants subject to the constraints:

$$\begin{array}{ll} \sum_i (\alpha\beta)_{ij} = 0 & j = 1, 2, \cdots, b \\ \sum_j (\alpha\beta)_{ij} = 0 & i = 1, 2, \cdots, a \end{array} \tag{4.22}$$

ϵ_{ijk} are independent $N(0, \sigma^2)$.

- Interpretation of parameters: α_i and β_j are main effects. Ex. $\alpha_1 = \mu_{1\cdot} - \mu$, where $\mu_{1\cdot}$ is the mean of all samples with A factor at level 1. The interaction effect $(\alpha\beta)_{ij}$ is defined as:

$$(\alpha\beta)_{ij} = \mu_{ij} - (\mu + \alpha_i + \beta_j) \tag{4.23}$$

Thus it is the difference between μ_{ij} and the value that would be expected if the factors are additive (defined as $\mu_{ij} = \mu + \alpha_i + \beta_j$). The interaction effect is not equal to 0 if the effect of one factor depends on the level of another factor.

ANOVA table: consider the case where each cell ij has the same sample size n:

- Partitioning of variances: we have:

$$SST = SSA + SSB + SSAB + SSE \tag{4.24}$$

where SSA and SSB are computed from the variation of the A and B factor, respectively, and $SSAB$ from the combination of two factors:

$$\begin{array}{ll} SSA & = nb\sum_i (\bar{Y}_{i\cdot} - \bar{Y})^2 \\ SSB & = na\sum_j (\bar{Y}_{\cdot j} - \bar{Y})^2 \\ SSAB & = n\sum_{i,j} (\bar{Y}_{ij} - \bar{Y}_{i\cdot} - \bar{Y}_{\cdot j} + \bar{Y})^2 \end{array} \tag{4.25}$$

- Mean squares: these are defined as:

$$\begin{array}{l} MSA = \frac{SSA}{a-1} \\ MSB = \frac{SSB}{b-1} \\ MSAB = \frac{SSAB}{(a-1)(b-1)} \end{array} \tag{4.26}$$

- F test: to test if all interaction effects are equal to 0:

$$F = \frac{MSAB}{MSE} \tag{4.27}$$

To test for factor A and B main effects (all 0): $F = \frac{MSA}{MSE}$ and $F = \frac{MSB}{MSE}$.

- Remark: the same test can be derived from regression approach (see the section "Analysis of variance approach to regression"). In the regression approach, for any factor A, we could write $SSA = SSR(A)$ (variation of group means defined by A).

4. Random and mixed effects models [KNNL, Applied Linear Statistical Models, 5ed, Chapter 25]

   Random effects model:

   - Two types of factors: some factors are properties of objects and are of intrinsic interest (i.e. to know their effects would be interesting); some other factors, however, are only random samples and of no intrinsic interest. Ex. to test the effect of a manufacturing procedure, multiple plants are selected and outcomes measured (with each plant: the outcome of different procedures for different samples). In this case, the selected plants are random, but they need to be treated as an experimental factor as their individual variation is important.
   - Random effects model: the second type of factors have effects that can be viewed as random samples of a larger population. Also called ANOVA model II. The questions of interest are generally about the larger population, not individual samples.
   - Mixed effects model: if there are multiple factors, some may have fixed effects and other random effects.

   Random effects model of 1 factor: random cell means model:

   - Model: the j-th sample of the i-th group has outcome:
   $$Y_{ij} = \mu_i + \epsilon_{ij} \tag{4.28}$$
   where μ_i are independent RVs of $N(\mu, \sigma_\mu^2)$ and ϵ_{ij} are independent $N(0, \sigma^2)$.
   - Questions: we are typically interested in: (1) whether there is significant variation across groups, i.e. $\sigma_\mu^2 = 0$; (2) estimate the average population mean μ and the variances σ_μ^2 and σ^2.
   - Important features of the model: the expected value of Y_{ij}:
   $$E(Y_{ij}) = \mu \tag{4.29}$$
   The variance of Y_{ij}:
   $$\text{Var}(Y_{ij}) = \sigma_Y^2 = \sigma_\mu^2 + \sigma^2 \tag{4.30}$$
   The covariance:
   $$\text{Cov}(Y_{ij}, Y_{ij'}) = \sigma_\mu^2 \tag{4.31}$$

   ANOVA test and estimation of random 1-factor model:

   - Mean sum of squares: the main test statistics are still $MSTR$ and MSE. Assuming the treatment sample size is n for all cells:
   $$\begin{aligned} E[MSE] &= \sigma^2 \\ E[MSTR] &= \sigma^2 + n\sigma_\mu^2 \end{aligned} \tag{4.32}$$
   - Test whether $\sigma_\mu^2 = 0$: this could be formed by the F test:
   $$F = \frac{MSTR}{MSE} \tag{4.33}$$
   - Estimation of μ: an unbiased estimator of μ is: $\hat{\mu} = \bar{Y}$. The variance of this estimator is:
   $$\text{Var}(\bar{Y}) = \frac{\sigma_\mu^2}{r} + \frac{\sigma^2}{rn} \tag{4.34}$$
   And the unbiased estimator of the variance of $\bar{Y}$ is:
   $$s^2(\bar{Y}) = MSTR/(rn) \tag{4.35}$$
   Thus $(\bar{Y} - \mu)/s(\bar{Y})$ follows t distribution with df $r - 1$.

- Estimation of σ^2 and σ_μ^2: for σ^2, we have:

$$\frac{r(n-1)MSE}{\sigma^2} \sim \chi^2_{r(n-1)} \tag{4.36}$$

For σ_μ^2, our unbiased estimator is:

$$s_\mu^2 = \frac{MSTR - MSE}{n} \tag{4.37}$$

4.3 Linear Regression

4.3.1 Simple Linear Regression

Reference: [KNNL, Applied Linear Statistical Models, 5ed, Chapter 1-2], [Chatterjee & Hadi, Regression analysis by example, 4ed, Chapter 2]

Linear model with one predictor variable:

- Model: for $1 \le i \le n$:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \tag{4.38}$$

where X_i is a known constant, Y_i are independent RVs, and ϵ_i is a RV with mean 0 and variance $\text{Var}(\epsilon_i) = \sigma^2$, where ϵ_i and ϵ_j are uncorrelated for any i, j. It is often assumed that $\epsilon_i \sim N(0, \sigma^2)$.

- Alternative models: a dummy variable $X_0 = 1$:

$$Y_i = \beta_0 \cdot 1 + \beta_1 X_i + \epsilon_i \tag{4.39}$$

Another model is:

$$Y_i = \beta_0^* + \beta_1(X_i - \bar{X}) + \epsilon_i \tag{4.40}$$

where $\beta_0^* = \beta_0 + \beta_1 \bar{X}$.

- Remark: in a linear model, we assume X_i's are constants. In some other situations, it may be easier to view X_i's also as RVs, e.g. in quantitative genetics: total variance is the sum of variance in predictor variable (genotype) and variance in environment (error term).

Least square parameter estimation:

- Nomral equations: minimizing the sum of squared error:

$$Q = \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 X_i)^2 \tag{4.41}$$

leads to the normal equations that the estimated parameter (b_0 and b_1) should satisfy:

$$\begin{array}{rcl} \sum_i Y_i & = & nb_0 + b_1 \sum_i X_i \\ \sum_i X_i Y_i & = & b_0 \sum_i X_i + b_1 \sum_i X_i^2 \end{array} \tag{4.42}$$

- LS estimators: solving the normal equations:

$$b_1 = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sum (X_i - \bar{X})^2} \tag{4.43}$$

$$b_0 = \bar{Y} - b_1 \bar{X} \tag{4.44}$$

Intuition of the estimator: the numerator $\sum (X_i - \bar{X})(Y_i - \bar{Y})$ is the sample covariance, when $X_i - \bar{X}$ and $Y_i - \bar{Y}$ have the same signs, the term is positive. Thus, the sign of the sample covariance captures the relationship: when one is larger, whether the other is large too.

- Geometry of least square estimation: suppose X and y are centered (vectors) in n-dim. space, the objective function is to minimize the distance from y to any point in the direction of X. This is solved by the projection of y on X:

$$\hat{\beta_1} = \frac{X^T y}{\|X\|^2} \tag{4.45}$$

  And $\hat{\beta_0} = 0$. When X and y are not centered, simply replace X by $X - \bar{X}$ and y by $y - \bar{y}$.

- Relation to MOM estimation: given $Y = \beta_0 + \beta_1 X + \epsilon$, we take the expectation:

$$\mathrm{E}(Y) = \beta_0 + \beta_1 \mathrm{E}(X) \tag{4.46}$$

  And consider the covariance between X and Y:

$$\mathrm{Cov}(X, Y) = \beta_1 \mathrm{Var}(X) \tag{4.47}$$

  Solving the two equations gives the MOM estimator of β_0 and β_1, which are the same as LS estimators.

- Properties of estimators of coefficients: both b_0 and b_1 are linear estimators, i.e. they are linear combinations of Y_i's:

$$b_1 = \sum_i k_i Y_i \qquad \text{where} \qquad k_i = \frac{X_i - \bar{X}}{\sum (X_i - \bar{X})^2} \tag{4.48}$$

  And b_0 and b_1 are unbiased estimators of β_0 and β_1, respectively:

$$E(b_1) = \beta_1 \qquad E(b_0) = \beta_0 \tag{4.49}$$

- Properties of fitted lines: we define residuals $e_i = Y_i - \hat{Y}_i$, where $\hat{Y}_i$ is the point estimator. We have the properties of residuals: (these can be proved easily)

$$\sum_{i=1}^{n} e_i = 0 \tag{4.50}$$

$$\sum_{i=1}^{n} X_i e_i = 0 \tag{4.51}$$

  Furthermore: (1) $\sum e_i^2$ is minimized by b_0 and b_1 (from LS estimation); (2) the regression line passes through $(\bar{X}, \bar{Y})$.

- Estimator of σ^2: define sum of square error as:

$$SSE = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 = \sum_{i=1}^{n} e_i^2 \tag{4.52}$$

  The SSE has two dof. (two parameters), the appropriate mean square is:

$$MSE = \frac{SSE}{n-2} = \frac{\sum e_i^2}{n-2} \tag{4.53}$$

  It can be shown that:

$$E(MSE) = \sigma^2 \tag{4.54}$$

  The intuition is that: $Y_i - \hat{Y}_i$ is normally distributed with variance σ^2, so using MOM estimation of normal distribution, the MSE is an estimator of σ^2.

Sampling distributions of b_0, b_1:

- b_1 as a linear function of Y_i: we could write $b_1 = \sum k_i Y_i$, where

$$k_i = \frac{X_i - \bar{X}}{\sum (X_i - \bar{X})^2} \tag{4.55}$$

The coefficients k_i have the properties:

$$\begin{array}{lcl} \sum k_i & = & 0 \\ \sum k_i X_i & = & 1 \\ \sum k_i^2 & = & \frac{1}{\sum (X_i - \bar{X})^2} \end{array} \tag{4.56}$$

- Sampling distribution of b_1: b_1 follows a normal distribution with:

$$E(b_1) = \beta_1 \qquad \mathrm{Var}(b_1) = \frac{\sigma^2}{\sum_i (X_i - \bar{X})^2} \tag{4.57}$$

Proof: since b_1 is a linear combination of Y_i's and Y_i are independent normal RVs, b_1 must follow normal distribution. Its mean:

$$E(b_1) = E(\sum_i k_i Y_i) = \sum_i k_i (\beta_0 + \beta_1 X_i) = \beta_0 \sum_i k_i + \beta_1 \sum_i X_i k_i = \beta_1 \tag{4.58}$$

And its variance:

$$\mathrm{Var}(b_1) = \mathrm{Var}(\sum_i k_i Y_i) = \sum_i k_i^2 \mathrm{Var}(Y_i) = \sum_i k_i^2 \sigma^2 = \frac{\sigma^2}{\sum (X_i - \bar{X})^2} \tag{4.59}$$

We can estimate the variance of b_1 by using the unbiased estimator of σ^2, MSE:

$$s^2(b_1) = \frac{MSE}{\sum (X_i - \bar{X})^2} \tag{4.60}$$

- Intuition of $\mathrm{Var}\,(b_1)$: larger σ^2 (intrinstic error) leads to a larger $\mathrm{Var}(b_1)$, and smaller $\sum_i (X_i - \bar{X})^2$ (small variation of X) also leads to a larger $\mathrm{Var}(b_1)$. Intuitively, one could think of $\sum_i (X_i - \bar{X})^2$ as "effective sample size": when $X_i = \bar{X}$, sample i has no information of β_1. To see why large $X_i - \bar{X}$ is more preferred (reducing the variance), we consider this example:

$$Y_i \sim N(\beta t, \sigma^2) \tag{4.61}$$

then $\hat{\beta} t = \bar{Y}$ is the estimator of β. Its variance is $\mathrm{Var}\,\hat{\beta} = \sigma^2/(nt^2)$.

- Sampling distribution of SSE: SSE/σ^2 is independent of b_1 and b_0, and follows χ^2 distribution with dof $n-2$.
Proof: similar to the proof that for normal distribution, sample variance divided by σ^2 follows χ^2 distribution with dof $n-1$. The difference now is that in SSE, we have $y_i - b_1 x_i - b_0$, thus we have two parameters (vs. normal distribution: only one parameter, the mean).

- t-test of b_1: $(b_1 - \beta_1)/s(b_1)$ follows t-distribution with dof $n-2$.
Proof: divide $\sigma(b_1)$ in both numerator and denomiator, the numerator follows the standard normal distribution, and denominator follows χ^2 distribution, so the ratio is t-distribution.
This distribution can be used to construct t test for the parameter value b_1.

- Sampling distribution of b_0: b_0 is also a linear combination of Y_i, thus follows normal distribution. Its mean:

$$\mathrm{E}\,(b_0) = \mathrm{E}\,(\bar{Y}) - \mathrm{E}\,(b_1)\bar{X} = \beta_1 \bar{X} + \beta_0 - \beta_1 \bar{X} = \beta_0 \tag{4.62}$$

To find out its variance, we first show that $\bar{y}$ and b_1 are independent. Both are linear functions of Y_i, and they are independent if the covariance is equal to 0:

$$\text{Cov}\left(b_1, \bar{y}\right) = \text{Cov}\left(\sum_i k_i Y_i, \frac{1}{n}\sum_i Y_i\right) = \sum_i k_i \cdot \frac{1}{n}\text{Var}\left(Y_i\right) = \frac{\sigma^2}{n}\sum_i k_i = 0 \tag{4.63}$$

Next we find the variance:

$$\text{Var}\left(b_0\right) = \text{Var}\left(\bar{Y}\right) + \bar{X}^2\text{Var}\left(b_1\right) = \sigma^2\left[\frac{1}{n} + \frac{\bar{X}^2}{\sum_i (X_i - \bar{X})^2}\right] \tag{4.64}$$

And an estimator of the variance of b_0:

$$s^2(b_0) = MSE\left[\frac{1}{n} + \frac{\bar{X}^2}{\sum_i (X_i - \bar{X})^2}\right] \tag{4.65}$$

- t-test of b_0: $(b_0 - \beta_0)/s(b_0)$ follows t-distribution with dof $n-2$.
- Covariance between b_1 and b_0: we use the fact that $\bar{Y}$ and b_1 are independent.

$$\text{Cov}\left(b_1, b_0\right) = \text{Cov}\left(\bar{Y} - b_1\bar{X}, b_1\right) = -\bar{X}\text{Var}\left(b_1\right) = -\sigma^2\frac{\bar{X}}{\sum_i (X_i - \bar{X})^2} \tag{4.66}$$

Point estimation of response:

- Point estimation of mean response: for any value of X, the estimator of the respose variable is given by:

$$\hat{Y} = b_0 + b_1 X \tag{4.67}$$

- Sampling distribution of the point estimator of response: at a point X_h, the estimator of Y_h is given by:

$$\hat{Y_h} = b_0 + b_1 X_h \tag{4.68}$$

  $\hat{Y_h}$ follows normal distribution with:

$$E(\hat{Y_h}) = Y_h \qquad \text{Var}(Y_h) = \sigma^2\left[\frac{1}{n} + \frac{(X_h - \bar{X})^2}{\sum_i (X_i - \bar{X})^2}\right] \tag{4.69}$$

  To prove the latter equation, first show that $\bar{Y}$ and b_1 are independent (since b_1 and b_0 are dependent), and express the variance as a sum of $\bar{Y}$ and b_1, then apply the variances of the two. The estimator of the variance of $\hat{Y_h}$ can be obtained by replacing σ^2 with MSE. Note that at X_h close to $\bar{X}$, the variance of the response estimator is smaller.

- Remark: the prediction problem can be studied in a way similar to parameter estimation: construct the estimator and the confidence interval (which often involves the standard error).

Dignostics and assessing the quality of fit:

- Covariance measures the linear dependency between two variables. Anscombe's quartet illustrates that when the linearity does not hold, very different relationships could result in the same correlation. It's important to use (scatter) plot to examine the relationship.
- Analysis: we can theoretically analyze how the results change when the assumptions are violated. In the linear model, two mains assumptions are: linearity and normality of errors. Suppose the second assumption is invalid, e.g. there are several outliers in the data, then from this equation: $\text{Var}\left(\beta_1\right) = \hat{\sigma}^2/\sum_i (x_i - \bar{x})^2$, the numerator will be inflated. We will then overestimate the variance of the estimator, which reduces the power of testing β_1.

- Measures of quality of fit: two ideas
  - The correlation between $\hat{Y}$ (the predicted value) and Y measures how prediction agrees with the observation. Furthermore, the measure can be easily generalized to multiple linear regression.
  - Coefficient of determination: measures how much variance of Y is explained by the variance of the predictor, $R^2 = SSR/SST$ (see the section of "ANOVA approach to regression").
- **Lessons**: both measures of quality of fit of a linear model are based on very general ideas (1) The agreement of observations and predictions. (2) How much variation in the data is explained by our model.

Normal correlation models:

- Bivariate normal model: in the regression model, X is considered as constants. But we could also view X as a RV, and consider the joint distribution of (X, Y). According to the conditional distribution of $Y|X$ (the section "bivariate normal distribution"), we have:

$$E(Y|X) = \left(\mu_Y - \mu_X \rho_{YX} \frac{\sigma_Y}{\sigma_X}\right) + \rho_{YX} \frac{\sigma_Y}{\sigma_X} X \tag{4.70}$$

  Thus $Y|X$ follows normal distribution and $E(Y|X)$ is a linear function of X. Therefore the linear regression model is equivalent to the conditional distribution of the bivariate normal distribution. In particular, we have the following relations under two models:

$$\beta_1 = \rho_{YX} \frac{\sigma_Y}{\sigma_X} = \frac{\sigma_{XY}}{\sigma_X^2} \tag{4.71}$$

$$\beta_0 = \mu_Y - \mu_X \rho_{YX} \frac{\sigma_Y}{\sigma_X} = \mu_Y - \beta_1 \mu_X \tag{4.72}$$

- Inference on ρ: the MLE of ρ is given by the Pearson product-moment correlation coefficient:

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}} \tag{4.73}$$

  The test of $\rho = 0$ is equivalent to the test of $\beta_1 = 0$, and it can be shown that the t test of β_1 can be expressed in r as:

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \tag{4.74}$$

- Fisher z transformation: the distribution of r when $\rho \neq 0$ is complicated, so define

$$z' = \frac{1}{2} \ln \frac{1+r}{1-r} \tag{4.75}$$

  With large n, z' is approximately normally distributed with:

$$E(z') = \frac{1}{2} \ln \frac{1+\rho}{1-\rho} \qquad \sigma^2(z') = \frac{1}{n-3} \tag{4.76}$$

- Relation between R^2 and r: (see ANOVA approach below) plug in the equation of b_1:

$$R^2 = \frac{SSR}{SST} = b_1^2 \frac{\sum (X_i - \bar{X})^2}{\sum (Y_i - \bar{Y})^2} = \frac{\left[\sum (X_i - \bar{X})(Y_i - \bar{Y})\right]^2}{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2} = r^2 \tag{4.77}$$

4.3.2 Multiple Linear Regression

Reference: [RABE, 5ed; Hastie, Section 3.2; KNNL, Applied Linear Statistical Models, 5ed]

Linear models:

- Definition: the models are linear wrt. the parameters. The function form for predictors can be non-linear.
- Model: p independent variables $X_j, j = 1 \cdots p$, and dependent variable Y are related by:

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \epsilon \tag{4.78}$$

  Suppose the error follows normal distribution $N(0, \sigma^2)$. Suppose there are n data points, $(x_i, y_i), 1 \leq i \leq N$ and each $x_i = (x_{i1}, \ldots, x_{ip})$ is a vector of features. We want to estimate $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^T$.

Least square parameter fitting:

- Notation: Denote by $\mathbf{X}$ the N by $(p+1)$ matrix with each row a data point and each column a feature (with 1 at the first position of each row: a dummy feature), and $\mathbf{y}$ the response vector (column vector). β is assumed to be a column vector. X is called the design matrix. Note: without this notation, we would have $X - \bar{X}$, and $y - \bar{y}$ in the equations below, instead of X and y.
- Least square: To maximize the log likelihood is equivalent to minimizing the residue sum of squares:

$$\text{RSS}(\beta) = \sum_{i=1}^{N} (y_i - f(x_i))^2 = \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \tag{4.79}$$

  Then we can write:

$$\text{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^{\mathbf{T}} (\mathbf{y} - \mathbf{X}\beta) \tag{4.80}$$

  Take derivative and solve the equatoin:

$$\hat{\beta} = (\mathbf{X^T X})^{-1} \mathbf{X^T y} \tag{4.81}$$

  And the predicted value for a new data point $\mathbf{x_0}$ is simply $\mathbf{x_0}\hat{\beta}$, and the predicted values for the training input:

$$\hat{\mathbf{y}} = X\hat{\beta} = X(X^T X)^{-1} Xy \tag{4.82}$$

  The $N \times N$ matrix $X(X^T X)^{-1} X$ is called the hat or projection matrix.

- Estimator of σ^2: The unbiased estimator of σ^2 is given by the sample variance of residuals:

$$\hat{\sigma^2} = \frac{1}{N - p - 1} \sum_{i=1}^{N} (y_i - \hat{y_i})^2 = \frac{SSE}{N - p - 1} \tag{4.83}$$

  It is also called mean squared error (MSE).

Deriving parameters by method-of-moment (MOM) approach:

- Method: given the linear model, we consider the covariance beween Y and X_k:

$$\text{Cov}(X_k, Y) = \sum_j \beta_j \text{Cov}(X_k, X_j) \tag{4.84}$$

  Write in matrix form:

$$\text{Cov}(X, Y) = \text{Cov}(X)\beta \tag{4.85}$$

  This describes the relationship between expected variance and covariance and β. To estimate β, we use the sample variance and covariance, and have the estimator of β in Equation 4.81.

- Intepretation of the coefficients: $\hat{\beta}$ is determined by the covarances between X_j vectors and y (X matrix is given): a special case, when X_js are independent, $\hat{\beta}_j$ is proportional to $\text{Cov}(X_j, y)$ - this is what we expect by intuition.
- Remark:
  - This approach assumes centering of variables. Alternatively, when we have dummy variable $X_0 = 1$, we do not need centering.
  - The equation of $\hat{\beta}$ suggests a relationship between the rank of $X^T X$ and the variance of $\hat{\beta}$: when the matrix $X^T X$ is not full ranked, it is hard to estimate β, or the variance of β is high.

Interpretation of regression coefficients:

- Geometric interpretation: we could write:

$$\text{RSS}(\beta) = \|\mathbf{y} - X\beta\|^2 \tag{4.86}$$

  In the N-dim. space, y is a vector, $X\beta$ is a point in the subspace created by $X_1, \cdots, X_p$ (linear combination of these p vectors), thus the objective function is the distance from y to some point in the subspace. We choose $\hat{\beta}$ s.t. the residual vector $\mathbf{y} - X\hat{\beta}$ is orthogonal to the column space of X. The projection of Y on the subspace is:

$$\sum_j \beta_j X_j = [X_1 \cdots X_p][\beta_1 \cdots \beta_p]^T = X\beta \tag{4.87}$$

  The orthogonality implies that $Y - X\beta$ is orthogonal to any X_j, i.e.

$$X_j^T(Y - X\beta) = 0 \qquad 1 \le j \le p \tag{4.88}$$

  This is an important property of residuals: they are independent of X_j's. This can be written in the matrix form (one equation above per row):

$$X^T(Y - X\beta) = 0 \tag{4.89}$$

  This is called "*normal equation*". The intuition is that the residues should be independent of X_j's - an important property of residues. Solving it gives $\hat{\beta} = (X^T X)^{-1} X^T y$, which can be simply understood as inner product divided by the squared norm of X.

- Relation to simple regression/condtional regression: e.g. β_2 in a regression involving three explanatory variables should be the regression coefficient of Y on X_2 after adjusting all other variables, for both Y and X_2. First, we do regression of Y on X_1 and X_3, and let residuals be $e_{Y \cdot X_1 X_3}$; next we do regression of X_2 on X_1 and X_3, and let residuals be $e_{X_2 \cdot X_1 X_3}$. Then the regression coefficient of $e_{Y \cdot X_1 X_3}$ on $e_{X_2 \cdot X_1 X_3}$ would give β_2. Thus regression coefficients in a multiple regression model are the *partial regression coefficents*.

Statistical significance of parameters:

- The estimator of β: according to Equation 4.81, $\hat{\beta}$ is a linear combination of $\mathbf{y}$. Since y_i follows normal distribution $N(x_i\beta, \sigma^2)$, thus y follows a multivariate normal distribution with mean $X\beta$, and covariance matrix: $\sigma^2 I$. Using the results of linear function of multivariate normal RVs, the mean of $\hat{\beta}$ is:

$$(X^T X)^{-1} X^T X\beta = \beta \tag{4.90}$$

  Thus $\hat{\beta}$ is unbiased. And the covariance matrix of $\hat{\beta}$ is:

$$(X^T X)^{-1} X^T \cdot \sigma^2 I \cdot ((X^T X)^{-1} X^T)^T = \sigma^2 (X^T X)^{-1} X^T X (X^T X)^{-1} = \sigma^2 (X^T X)^{-1} \tag{4.91}$$

We use the fact that $X^T X$ is a symmetric matrix. This leads to:
Theorem: $\hat{\beta}$ follows normal distribution:

$$\hat{\beta} \sim N(\beta, (\mathbf{X^T X})^{-1}\sigma^2) \tag{4.92}$$

Its marginal distribution $\hat{\beta}_j \sim N(\beta_j, v_j\sigma^2)$ where v_j is the j-th diagonal element of $(X^T X)^{-1}$. In practice, we often do not know σ^2, so we replace it with its estimator:

$$s^2(\hat{\beta}) = (N-p-1)^{-1}(X^T X)^{-1}(y - X\hat{\beta})^T(y - X\hat{\beta}) \tag{4.93}$$

- Interpretation of the variance of β: it is proportional to the sample precision matrix of X: $\hat{\text{Cov}}(X)^{-1}$. Intuitively, when X_i is linearly dependent on other variables, σ_{ii} (the partial covariance of i) is small, and thus its inverse is large, so $\text{Var}(\hat{\beta}_j)$ is large.

- The estimator of σ^2: similar to the case of sample variance of the univariate Gaussian distribution:

$$(N-p-1)\hat{\sigma^2} \sim \sigma^2\chi^2_{N-p-1} \tag{4.94}$$

In addition, $\hat{\beta}$ and $\hat{\sigma^2}$ are statistically independent. We could also state this as the distribution of the error SSE:

$$\frac{SSE}{\sigma^2} \sim \chi^2_{N-p-1} \tag{4.95}$$

- Testing individual coefficients: often we are interested in testing a particular coefficient while controlling for all other variables. To test the hypothesis that a particular coefficient $\beta_j = 0$, compute the standarized coefficient:

$$t_j = \frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{v_j}} \tag{4.96}$$

where v_j is the j-th diagonal element of $(\mathbf{X^T X})^{-1}$. Under the null hypothesis $\beta_j = 0$, t_j is distributed as t_{N-p-1}, and if σ is known, t_j follows normal distribution. As sample size increasese, the difference between normal and t distribution becomes negligible.

- Testing a group of parameters simultanesouly: suppose given p_0 coefficients, want to test if more parameters, p_1, gives a significant better fit (i.e. test the significance of extra $p_1 - p_0$ parameters). The F-test statistic:

$$F = \frac{(\text{RSS}_0 - \text{RSS}_1)/(p_1 - p_0)}{\text{RSS}_1/(N - p_1 - 1)} \tag{4.97}$$

See the section on AVOVA below.

Likelihood approach:

- Likelihood function:

$$L(\beta, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - X_i\beta)^2\right] \tag{4.98}$$

The log-likelihood function is:

$$l(\beta, \sigma^2) = -\frac{1}{2}\left[\log(2\pi) + \log\sigma^2 + \frac{1}{\sigma^2}\sum_{i=1}^{N}(y_i - X_i\beta)^2\right] \tag{4.99}$$

- MLE: maximizing likelihood is equivalent to minimize squared error, so we have $\hat{\beta} = \hat{\beta}_{\text{LS}}$, and

$$\hat{\sigma^2} = \frac{\sum(y_i - \hat{y_i})^2}{n} \tag{4.100}$$

where $\hat{y_i} = X_i\hat{\beta}$.

- Likelihood ratio test: suppose we compare two models, one reduced model (R) and one full model (F). The dof. of the two models are d_R and d_F respectively (n minus the number of free parameters), and the difference of the number of parameters in two models is thus $d_R - d_F$. To test if the parameters in the full model are equal to 0, we form the LRT, assuming σ^2 is known:

$$-2[l(\hat{\theta}|R - l(\hat{\theta}|F))] = \frac{\text{SSE}_R - \text{SSE}_F}{\sigma^2} \tag{4.101}$$

  which follows χ^2 distribution of $d_R - d_F$. However, σ^2 is not known, so this test is not directly applicable (could use MLE of σ^2, a different test). But we use the fact that: for the full model:

$$\frac{\text{SSE}_F}{\sigma^2} \sim \chi^2_{d_F} \tag{4.102}$$

  The ratio of the two χ^2 distribution follows the $F(d_R - d_F, d_F)$ distribution:

$$F = \frac{\frac{\text{SSE}_R - \text{SSE}_F}{\sigma^2}/(d_R - d_F)}{\frac{\text{SSE}_F}{\sigma^2}/d_F} = \frac{(\text{SSE}_R - \text{SSE}_F)/(d_R - d_F)}{\text{SSE}_F/d_F} \tag{4.103}$$

  This is exactly the same F-test above (also see the section on AVOVA approach on regression).

Prediction and residuals [RABE, Chapter 4]: the predictions and residuals are also random variable and we want to determine their distribution. This would allow us to estimate the accuracy of predictions and the distribution of residuals can be used to check for model violations.

- Predictions: for sample i, its fitted value is $\hat{y}_i = x_i\hat{\beta}$, or in matrix form:

$$\hat{y} = X\hat{\beta} \tag{4.104}$$

  To write it in terms of y_i's, we plug in $\hat{\beta}$:

$$\hat{y} = X(X^TX)^{-1}X^Ty = Py \tag{4.105}$$

  where $P = X(X^TX)^{-1}X^T$ is the projection matrix. Or:

$$\hat{y}_i = p_{i1}y_1 + p_{i2}y_2 + \cdots + p_{in}y_n \tag{4.106}$$

  For simple regression, we have:

$$p_{ij} = \frac{1}{n} + \frac{(x_i - \bar{x})(x_j - \bar{x}))}{\sum(x_i - \bar{x})^2} \tag{4.107}$$

  In particular, the term p_{ii} is called leverage. Intuitively, when $x_i - \bar{x}$ is large, the leverage is bigger.

- Properties of the projection matrix: first, it is symmetric

$$P^T = P \tag{4.108}$$

  This is easy to check. Next, it is idempotent:

$$P^2 = X(X^TX)^{-1}X^T \cdot X(X^TX)^{-1}X^T = X(X^TX)^{-1}X^T = P \tag{4.109}$$

  From this, it is easy to show that $(I - P)^2 = I - P$.

- Distribution of residuals: once a model is fit, we could compute the residuals:

$$e_i = y_i - \hat{y}_i = y_i - x_i\hat{\beta} \tag{4.110}$$

To derive the distribution of e_i, we have $e = y - \hat{y} = (I - P)y$. Since y is MVN, then clearly e also follow MVN, and its variance is:

$$\text{Var}\,(e) = (I - P)\text{Var}\,(y)(I - P)^T = \sigma^2(I - P)^2 = \sigma^2(I - P) \tag{4.111}$$

where we used the properties of the projection matrix above. So the variance of the i-th residual is given by:

$$\text{Var}(e_i) = \sigma^2(1 - p_{ii}) \tag{4.112}$$

So the variance of all residuals are not equal: when x_i is far from $\bar{x}$, we have larger leverage, then the variance of e_i is smaller - easier to predict. Ex. when x_i is 0 (or close to 0), then we cannot predict y_i, we have small leverage and large variance of e_i.

- Standardized and studentized residuals: to address the problem that e_i are not comparable across samples, we standarize the residuals by:

$$r_i = \frac{e_i}{\sigma\sqrt{1 - p_{ii}}} \tag{4.113}$$

It has mean 0 and variance 1. We can use unbiased estimator of σ^2:

$$r_i = \frac{e_i}{\hat{\sigma}\sqrt{1 - p_{ii}}} \tag{4.114}$$

where $\hat{\sigma}$ is the estimated σ. It follows t-distribution.

- Properties of residuals: $\text{RSS}(\beta)$ can be written in terms of residuals:

$$\text{RSS}(\beta) = \sum_i e_i^2 \tag{4.115}$$

Thus from minimization of RSS, we have: $\partial\text{RSS}(\beta)/\partial\beta_j = 0$ for $\hat{\beta}_j$. This leads to the following results:

$$\sum_i e_i = 0 \tag{4.116}$$

$$\sum_i e_i x_{ij} = 0 \qquad j = 1, 2, \cdots, p \tag{4.117}$$

Gauss-Markov Theorem: suppose we want to estimate $\theta = a^T\beta$ (this is predicting a new data point, if $a = x_0$). Let $\hat{\beta}$ be the least square estimator of β. The theorem states:

- Unbiased estimator: $E(a^T\hat{\beta}) = a^T\beta$.
- Minimum variance: among all estimators that are linear to $\mathbf{y}$, $\tilde{\theta} = c^T\mathbf{y}$, we have:

$$\text{Var}(a^T\hat{\beta}) \leq \text{Var}(c^T\mathbf{y}) \tag{4.118}$$

Alternative notations of linear regression: (often in machine learning literature)

- Notation: a data point $\mathbf{x}$ will first be mapped to a feature space, using basis functions $\phi_j(\mathbf{x})$. Then a data point will be represented by a column vector $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \cdots, \phi_p(\mathbf{x}))^T$. The parameters will be denoted by a column vector $\mathbf{w}$. The data matrix will be denoted as $N \times p$ matrix (design matrix), Φ, where the i-th row is the i-th data point $\phi(\mathbf{x_i})^T$.
- Least square solutions: the solution can now be written as:

$$\hat{\mathbf{w}} = (\Phi^T\Phi)^{-1}\Phi^T y \tag{4.119}$$

And the prediction for a new example $\mathbf{x}$ is: $\hat{y} = \hat{\mathbf{w}}^T\mathbf{x}$.

4.3.3 Generalized Least Square

Generalized least square (GLS): [Wiki]

- Motivation: in ordinary linear model, we assume that the errors are iid. When the errors are independent but not identical, or are correlated, we need to generalized the model.
- Model: let X be the $N \times p$ design matrix, and y be the response variable ($N \times 1$ vector), we have the linear model
$$y = X\beta + \epsilon \quad \epsilon \sim N(0, \Sigma) \tag{4.120}$$
The log-likelihood function of β is:
$$l(\beta) = -\frac{1}{2}(y - X\beta)^T \Sigma^{-1}(y - X\beta) + \text{const} \tag{4.121}$$
The derivative is:
$$\frac{\partial l(\beta)}{\partial \beta} = -\frac{1}{2} \cdot 2(y - X\beta)^T \Sigma^{-1}(-X) \tag{4.122}$$
This leads to the estimating equation: $X^T\Sigma^{-1}(y - X\beta) = 0$, and the solution is:
$$\hat{\beta} = (X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}y \tag{4.123}$$
- GLS approach: to generalize, we do not have to assume that the data follows MVN distribution. Rather, the errors only need to follow: $\mathrm{E}(\epsilon) = 0$ and $\mathrm{Var}\,\epsilon = \Sigma$, we minimize the generalized squared error, defined as:
$$GSE = (y - X\beta)^T \Sigma^{-1}(y - X\beta) \tag{4.124}$$

Weighted least square (WLS): This is a special case of GLS with Σ a diagnoal matrix. Let σ_i^2 be the variance of the error of the i-th observation, then Σ^{-1} is the diagnoal matrix, with the diagnoal term $1/\sigma_i^2$.

Diagnonalization of GLS model [personal notes]

- Method: we do Spectrum Decomposition $\Sigma = UDU^T$, then we multiply U^T to both sides of the regression model:
$$U^T y = U^T X\beta + U^T \epsilon \tag{4.125}$$
Then the error follows $U^T\epsilon \sim N(0, U^T\Sigma U) = N(0, D)$. This means that we do variable substitution: let $y' = U^T y$ and $X' = U^T X$, then we have $y' = X'\beta + N(0, D)$ where the errors are independent.
- Remark: the same method can be applied to factor analysis. Suppose we have x_j, $N \times 1$ vector:
$$x_j = \sum_k Z_k W_{jk} + \epsilon_j, \qquad \epsilon_j \sim N(0, \Sigma) \tag{4.126}$$
We can use the same trick to diagonalize. This has been used in RSSp, where the errors of nearby SNPs are correlated because of LD; and in GWAS factor analysis of multiple traits, where the errors are correlated b/c sample overlapping.

4.4 Analysis of Variance Approach to Regression

Reference: [KNNL, Applied Linear Statistical Models, 5ed, Sections 2.7-2.8, 6.5, 7.1-7.4]

Partitioning of total sum of squares:

- Definitions: we define the total sum of squares of the response variable (total variation):

$$SST = \sum_{i=1}^{n} (Y_i - \bar{Y})^2 \tag{4.127}$$

The measure of the variation of Y_i after X is taken into account is the error sum of squares:

$$SSE = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{4.128}$$

The regression sum of squares measures the deviation of the predicted Y_i from $\bar{Y}$:

$$SSR = \sum_{i=1}^{n} (\hat{Y}_i - \bar{Y})^2 \tag{4.129}$$

- Partitioning: we could write, for every observation:

$$Y_i - \bar{Y} = \hat{Y}_i - \bar{Y} + Y_i - \hat{Y}_i \tag{4.130}$$

Take the square in both sides and sum over i:

$$\sum_i (Y_i - \bar{Y})^2 = \sum_i (\hat{Y}_i - \bar{Y})^2 + \sum_i (Y_i - \hat{Y}_i)^2 + 2\sum_i (\hat{Y}_i - \bar{Y})(Y_i - \bar{Y}) \tag{4.131}$$

It can be shown that the last term is 0, by replacing $(Y_i - \bar{Y})$ with e_i, $\hat{Y}_i$ with $x_i\hat{\beta}$, and applying the properties of residuals. This leads to the partition:

$$SST = SSR + SSE \tag{4.132}$$

Thus we could view SSR as the explained variation and SSE as the unexplained variation.

- Degree of freedom (df): the df of SST is $N - 1$, the lost degree comes from the constraint that the sum of $Y_i - \bar{Y}$ must sum to 0. The df of SSR is p, the number of free parameters (β_0 is not counted as we are only interested in the deviation from the mean). The df of SSE is $N - p - 1$, where $p + 1$ comes from the number of constraints of residuals. The df's also satisfy:

$$df(SSE) = df(SSR) + df(SSE) \tag{4.133}$$

- Mean squares: defined as the ratio of sum of squares and the df:

$$MSR = \frac{SSR}{p} \tag{4.134}$$

$$MSE = \frac{SSE}{N - p - 1} \tag{4.135}$$

The expected value of MSE is simply:

$$E[MSE] = \sigma^2 \tag{4.136}$$

The expected value of MSR is σ^2 plus a nonnegative number, e.g. for $p = 1$:

$$E[MSR] = \sigma^2 + \beta_1^2 \sum_i (X_i - \bar{X})^2 \tag{4.137}$$

- Coefficient of determination: we could define a measure of goodness-of-fit as the fraction of explained variation:
$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} \tag{4.138}$$
Also note that it equals the square of the correlation coefficient between the y and $\hat{y}$.

ANOVA approach to univariate regression:

- Idea: if β_1 is large, then X and Y are highly correlated, and we expect a significant part of the variance of Y can be explained by X. So we could use the variance partitioning to infer/test the parameter of the linear model.

- F-test of $\beta_1 = 0$: from expected MSE and MSR, we see that if $\beta_1 = 0$, they have the same expectation; and would be different if $\beta_1 \neq 0$. This motivates the F-test:
$$F = \frac{MSR}{MSE} \tag{4.139}$$
F statistic follows the $F(1, n-2)$ distribution, where 1 and $n-2$ are df. of SSR and SSE respectively.
Proof: we know that SSE/σ^2 follows χ^2 distribution with df equal to $n-2$. For SSR, we know that it is equal to b_1^2 times constant. b_1 follows normal distribution, under $H_0 : \beta_1 = 0$, we have:
$$\frac{b_1}{\sigma / \sqrt{\sum_i (X_i - \bar{X})^2}} \sim N(0, 1) \tag{4.140}$$
Its square, which is SSR/σ^2 thus follows χ^2 distribution with df 1. The ratio of the two χ^2 distribution (divided by d.f.) thus follows F-distribution.
When $H_A : \beta_1 \neq 0$, F follows the noncentral distribution.

- Equivalent of F and t test of β_1: use $SSR = b_1^2 \sum (X_i - \bar{X})^2$, and $s^2(b_1) = MSE / \sum (X_i - \bar{X})^2$, we could prove that:
$$F = \frac{b_1^2}{s^2(b_1)} = t^2 \tag{4.141}$$
where t is the test statistic of the Student t-test.

- Descriptive measure of linear association: the partition of sum of square motivates the following measure, the coefficient of determination, of how good the linear model explains the data:
$$R^2 = \frac{SSR}{SST} \tag{4.142}$$
It is interpreted as the fraction of variance explained by the predictor variable X. Generally, higher linear assocition means larger R^2.
Caution: if X and Y may not be linearly associated, R^2 may have very limited use. Ex. it is possible that X and Y are strongly dependent (but nonlinear), but R^2 is close to 0.

- Remark: approach is a way of selecting models: for a variable X to be selected, need to reject $H_0 : \beta_1 = 0$. This only happens if adding X explains a significant amount of variation (SSR term), and simpler models are preferred because our prior belief is H_0 is more likely (the conservative nature of classical hypothesis testing). Comparing with approaches that explicitly penalize complex models such as Lasso and Bayesian model selection, ANOVA approach has a few drawbacks:
  - Penalization is implicit: thus the prior belief of how H_0 is likely vs H_A is never specified, and not tested.

– Multiple hypothesis testing: when selecting among multiple models, we are testing multiple hypothesis. Not clear how correction should be done.

Partition using extra sum of squares: in multiple regression, we are interested in the question of marginal reduction of error when extra variables are introduced.

- Extra sum of squares: suppose we want to know the marginal effect of adding X_3 to the regression model which already contains X_1 and X_2, we have two equivalent forms:

$$SSR(X_3|X_1, X_2) = SSE(X_1, X_2) - SSE(X_1, X_2, X_3) = SSR(X_1, X_2, X_3) - SSR(X_1, X_2) \quad (4.143)$$

  The equality of the two forms: the increase of explained variation should be equal to the decrease of unexplained variation.

- Decomposition of sum of squares: in general, we have many different ways of paritioning SSR or SST, e.g.:

$$\begin{aligned} SSR(X_1, X_2, X_3) &= SSR(X_1) + SSR(X_2|X_1) + SSR(X_3|X_1, X_2) \\ SSR(X_1, X_2, X_3) &= SSR(X_2) + SSR(X_1, X_3|X_2) \end{aligned} \quad (4.144)$$

  The df of each conditional SSR is simply the number of extra parameters. And this allows one to define the conditional MSR, e.g.

$$MSR(X_2, X_3|X_1) = \frac{SSR(X_2, X_3|X_1)}{2} \quad (4.145)$$

- Coefficient of partial determination: this is defined as the fraction of explained variance by the extra variables over all the variance in the current model (with some variables already in the model). Example:

$$R^2_{X_1|X_2,X_3} = \frac{SSR(X_1|X_2, X_3)}{SSE(X_2, X_3)} \quad (4.146)$$

  This could also be explained as: suppose we regress Y on X_2 and X_3 and obtain residuals, $e_i(Y|X_2, X_3)$; and we regress X_1 on X_2 and X_3 and obtain residuals, $e_i(X_1|X_2, X_3)$, then the coefficient of determination R^2 between the two sets of residuals is $R^2_{X_1|X_2,X_3}$. The square root of the coefficient of paritial determination is the partial correlation coefficient.

F test of general linear models:

- Full model vs. reduced model: our test is whether the full model is significantly better than a resrictied/reduced model, e.g. some $\beta_k = 0$ where β_k is the extra parameter (or parameters). Suppose $SSE(F)$ is the SSE of the full model, and $SSE(R)$ is the SSE of the reduced model, then the test statistic is a function of $SSE(R) - SSE(F)$:

$$F = \frac{SSE(R) - SSE(F)}{df_R - df_F} / \frac{SSE(F)}{df_F} \quad (4.147)$$

  The F test statistic follows the F distribution with df $df_R - df_F$ and df_F.

  – Ex. to test if $\beta_2 = \beta_3 = 0$ in a regression of 3 variables, we have: $SSE(F) = SSE(X_1, X_2, X_3)$, $SSE(R) = SSE(X_1)$, and $SSE(R) - SSE(F) = SSR(X_2, X_3|X_1)$.

  – Also note that, the constraints in the reduced model do not have to be some $\beta_k = 0$, it could also be, e.g. $\beta_1 = \beta_2$. In this case, the F test still applies, but not the extra sum of square (conditional SSR).

- Proof of the F test: to show that F under H_0 follows F distribution, we first note that the denominator divided by σ^2 follows χ^2 distribution. For the numerator, we note that it is the variance explained by the additional parameters in the F model. Similar to our proof in the univariate case (SSR/σ^2 follows χ^2 distribution), we can show that it also follows χ^2 distribution.

– Remark: a rigorous proof could use Cochran's Theorem.

- Example: dealing with confounding variables. Suppose we want to test the effect of A, in the presence of a confoudning variable B. Thus the full model is $F = (A, B)$ and the reduce model is $R = B$. We have: $SSE(F) = SSE(A, B)$, and

$$SSE(R) - SSE(F) = SSE(B) - SSE(A, B) = SSR(A|B) \tag{4.148}$$

  Therefore, the F statistic is determined by $SSR(A|B)$, the extra variation explained by A conditioned on B. This relates to the idea of "stratification": $SSR(A|B)$ is computed on each stratum defined by B.

Relationship between R^2 and regression coefficients [personal notes]

- Motivation: in statistical genetics, we want to estimate heritability (or proportion of variance explained or PVE) from the effect size estimates (and other related quantities, including the standard error).

- Simple regression: suppose the model is $y = x\beta + \epsilon$, we want to estimate R^2 from $\hat{\beta}$, $\text{Var}\,(\beta)$ and $\text{Var}\,(x)$. We take the variance of y:

$$\text{Var}\,(y) = \beta^2 \text{Var}\,(x) + \sigma^2 \tag{4.149}$$

  To determine R^2, we need σ^2. We use this equation:

$$\text{Var}\,(\hat{\beta}) = \frac{\sigma^2}{\text{Var}\,(x)} \tag{4.150}$$

  In summary, our estimate of R^2:

$$\hat{R^2} = \frac{\hat{\beta}^2 \text{Var}\,(x)}{\hat{\beta}^2 \text{Var}\,(x) + \hat{\sigma}^2} \qquad \hat{\sigma}^2 = \text{Var}\,(\hat{\beta}) \text{Var}\,(x) \tag{4.151}$$

- Multiple regression when the explanatory variables are independent: the variance of y is given by:

$$\text{Var}\,(y) = \beta^2 \text{Var}\,(x) + \sigma^2 = \sum_j \beta_j \text{Var}\,(x_j) + \sigma^2 \tag{4.152}$$

  And the standard error of $\hat{\beta}$ is given by:

$$\text{Var}\,(\hat{\beta}) = \sigma^2 (X^T X)^{-1} \tag{4.153}$$

  This leads to the estimate:

$$\hat{R^2} = \frac{\sum_j \hat{\beta_j}^2 \text{Var}\,(x_j)}{\sum_j \hat{\beta_j}^2 \text{Var}\,(x_j) + \hat{\sigma}^2} \qquad \hat{\sigma}^2 = \text{Var}\,(\hat{\beta})(X^T X) \tag{4.154}$$

- Logistic model: when Y is binary, we can still talk about the same questions, using a liability model. An alternative may be to use conditional distributions, when X is also discrete. For example, to see how important X is: $P(Y = 1|X = 1) - P(Y = 1)$ estimates how many cases of $Y = 1$ is due to $X = 1$.

4.4.1 Linear Regression with Categorical Variables

Reference: [RABE, 5th ed, Chapter 5]

Motivation:

- Qualitative variables: often we have such variables, such as gender or occupation, that may influence the response variables. Therefore we need to incorporate these variables in the analysis. But their model is different from continuous variables as we need to "code" them, and the value of the code itself does not have a numerical meaning.
- Groups: often the samples may have some group structure, and the response variables may differ across groups. So we could encode the groups as qualitative variables. The point here is that even if there is explicit/natural way of grouping (and naming it with a group variable), the group structure may still exist.
- Non-linear effects: sometimes the effect of a variable depends on some other conditions - one way to model this is to create groups where the effect is homogeneous, then the interaction between the variable and the group membership variable models the non-linear effect.

Basic model of dealing with categorical variables (factors):

- Factor: if we have a cateorical variable, then we call it a "factor". A factor can have multple levels. The basic strategy of modeling factors in regression is: define one level as reference/base/control level, and compare the other levels with this one. If we have K levels, we have $K-1$ indicator variables, one for each level other than the reference.
- Salary survey data: we study the relationship Y - salary, and some variables including X - experience (continuous), E - educator, HS (high-school) or BS (bachelor) or advanced (base level), and M - management (1 or 0). Our model:

$$Y = \beta_0 + \beta_1 X + \gamma_1 E_1 + \gamma_2 E_2 + \delta_1 M \tag{4.155}$$

  where E_1, E_2 are the two levels (HS and BS). The interpretation of coefficients: the effect of a variable (over base level), when controlling for all other variables. Effectively, controlling for categorical variables does not matter here: because the effect does not depend on the categorical variables, so we can simply assume all categorical variables are at the base level (coded as 0).

Interaction model:

- Detecting interactions: residual plot could help. Plot the residuals against the categorical variables and see if they are homogeneous against differnet groupings.
- Model: for the salary example, we found that there is an interaction between E and M, so the new model:

$$Y = \beta_0 + \beta_1 X + \gamma_1 E_1 + \gamma_2 E_2 + \delta_1 M + \alpha_1 (E_1 \cdot M) + \alpha_2 (E_2 \cdot M) \tag{4.156}$$

- The interpretations of coefficients:
  - Marginal coefficients: The effect of a variable (vs. base) when controlling for all other variables (categorical variables at the base level). Example: β_1 means the effect of HS education when $M = 0$ and X at the mean (or 0).
  - Interaction coefficients: The differential effect of one variable when the other variable is at the non-base level, and controlling for all other variables. Example: α_1 is the difference of the effect of HS education when $M = 1$, vs. the effect of HS education when $M = 0$. In other words, β_1 is the effect of E_1 when $M = 0$, and $\beta_1 + \alpha_1$ is the effect of E_1 when $M = 1$ (adjusting for the effect of M itself).

- Alternative model: we can also encode interaction simply by treating each combination as one group. In the salary example, E could take three values and M two values, then there are 6 possible groups. We can simply model them as one variable of 6 levels. We can also replace the intercept with the mean of the base group, then the model is simple (symmetric). For example, let G_i be the i-th group, our model:

$$y = \beta_1 G_1 + \beta_2 G_2 + \cdots \beta_6 G_6 \tag{4.157}$$

  The coefficients now have simple interpretation: mean of a group. Statistical problem may become testing if coefficients are equal.

Comparing two groups: systems of regression equation

- Example: suppose we study the relationship between X (test score) and Y (job performance) on subjects of different races. The relationship between Y and X can be different in different races. To model this, we can create two regression models, one for subjects of each race.

- Model: we could create a single regression model, and it would be easier to test hypothesis. The basic idea is to model interaction bwteen X and the group variable. Example, let Z be the race variable, our model is:

$$Y = \beta_0 + \beta_1 X + \gamma Z + \delta(X \cdot Z) \tag{4.158}$$

  So if $\gamma \neq 0$, it means that the race could affect the performance; if $\delta \neq 0$, it means that the effect of X depends on the race. To test the hypothesis that there is discrimanation against race, we test: $H_0 : \gamma = 0, \delta = 0$. This can be done with F-test.

- Special case 1: same slope different intercepts. This is the special case above, and we do not have the interaction terms. We test $H_0 : \gamma = 0$.

- Special case 2: same intercepts different slopes. We do not have the term γZ, and we test $H_0 : \delta = 0$.

ANOVA: a special case of regression model with categorical variable

- Problem: suppose we test if the mean is different across K groups $H_0 : \mu_1 = \cdots = \mu_K$.

- Model: we create group variables $X_1, \cdots, X_{K-1}$, which takes $K - 1$ values, and we test $H_0 : \beta_1 = \cdots = \beta_{K-1} = 0$.

Analysis of one example: study treatment effects across subjects [personal notes]

- Problem: we want to study the treatment effects (potentially many kinds of treatments/drugs), but the effect may differ across subjects. We ask questions such as: is there any treatment effect (averaging across subjects)? How often the effect varies across individuals?

- Model: We have a factor of treatment with three levels: control, low dosage and high dosage. We define two variables T_1 for the effect of low dosage vs. control; and T_2 for high dosage vs. control. Then our basic model is :

$$Y = \beta_0 + \beta_1 T_1 + \beta_2 T_2 + \epsilon \tag{4.159}$$

  where β_1 and β_2 describe the effects of low and high dosage, and β_0 is the baseline (control) of the response variable.

- Paired design: when we do not care about subject difference, we assume it is the same, and we only need to test the effect, but adjusting for the background difference across subjects. So we have a paired design, where the treatment and control are paired in each subject. Our model:

$$Y = \beta_0 + \beta_1 T + \beta_2 S + \epsilon \tag{4.160}$$

  where T is treatment and S subject. If we have two subjects, then S is the difference between the two (0 for one, 1 for the other).

- Differential effect: to answer the question about variation of effects across subjects, we include interaction in the model. Suppose we have two subjects (S_1 is the referene), our model:

$$Y = \beta_0 + \beta_1 T + \beta_2 S + \gamma(S \cdot T) + \epsilon \tag{4.161}$$

  Then β_1 is the treatment effect in S_1 (reference subject), and β_2 the baseline of S_2 (the subject effect) and γ the difference of the treatment effect in S_2 (so $\beta_1 + \gamma$ gives the treatment effect in S_2).

- Interpretation of coefficients: four cases
  - $\beta_1 = 0, \gamma = 0$: no effect in both subjects.
  - $\beta_1 \neq 0, \gamma = 0$: same effect in both subjects.
  - $\beta_1 = 0, \gamma \neq 0$: no effect in S_1, effect in S_2.
  - $\beta_1 \neq 0, \gamma \neq 0$: effect in both, but different sizes (including the case: effect in S_1, but no effect in S_2).

- Using alternative encoding: we could also model $S \times T$ as groups. Suppose there are four combinations, we have:

$$Y = \beta_1 G_1 + \beta_2 G_2 + \beta_3 G_3 + \beta_4 G_4 \tag{4.162}$$

  Suppose G_1 is $S_1 \times$control, G_2 is $S_2 \times$treatment, and similarly for G_3 and G_4. Then $\beta_2 - \beta_1$ is the treatment effect in S_1, and $\beta_4 - \beta_3$ the effect in S_2, and $(\beta_4 - \beta_3) - (\beta_2 - \beta_1)$ the difference of effect between S_2 and S_1.

- Estimating the fraction of differential effects: sometimes we may want to estimate how often each of the scenarios occur (among many treatments, e.g. in genomics, could be many genes). Ideally, we should use a Bayesian approach: mixture prior on the effect sizes, then estimate the proportion.

4.5 Linear Regression in Practice

Reference: [Gelman07, chapter 2, 3; Hastie, chapter 3; KKNL, Applied Linear Statistical Models], [Chatterjee, Regression analysis by example (RABE), 4ed]

Assumptions of linear model and possible violations [RABE, Chapter 4]:

- Linearity assumption: about the coefficients. Data transformation is often used to make sure this assumption holds.
- Normality assumption of the errors: this can be assessed with appropriate graphs of residuals.
- Constant variance assumption of the errors (homoscedasticity).
- Independent-error assumption: auto-correlation problem.
- Collearity problem: when X_j's are not linearly independent.

Graphical methods [RABE, Chapter 4]

- Motivation: Anscombe quarter (Figure 4.1). Plot Y against X in scatter plot. The same correlation cofficient and regression model, but very different patterns. In (b) quadratic relation; (c) outlier; (d) influential point.
- Overview: use graphical methods to explore the relation between variables, recognize patterns such as clusters, detect errors, detect outliers and highly influential observations.
- Explorative analysis of data: include the distribution of variables and relationship between variables.

  - 1D graphs: such as histogram, dot plot, box plot for the distribution of X's or Y's. May suggest necessary transformation, outlier, etc.
  - 2D graphs: scatter plot for Y against X. In the mutiple regression case, plot matrix. However, note that, it is possible that even if the model is linear, the pairwise relationship can differ dramatically (e.g. no correlation). So one should control for other variables when creating the plots.

- Checking linearity and normality assumptions using residuals: if the linear model is a reasonable fit, then one should expect the standardized residuals to be $N(0,1)$ distributed.
  - Normal probabability plot: directly check if the standardized residuals follows standard normal.
  - Index plot of the standardized residuals.
  - Standardized residuals against predictor variables
  - standardized residuals against the fitted values ($\hat{Y}_i$).

- **Lesson**: to check for the validaty of a model, one can:
  - Directly examine the assumption of the model: e.g. linear relationship between Y and X.
  - Examine model predictions: in the linear model case, the predicted values should be close to the actual values of the response variables, more precisely, the errors should follow normal distribution.

Outliers and influential poins: [RABE, Chapter 4]

- Outliers: in response variables. They can be detected by the residual plots.
- Influential points: outliers in predictors. These points have large leverage p_{ii}. High-leverage points may not be detected by residual plot because these points often dominate the model fitting thus have small residuals.
- Potential-residual plot: define the potential function as a function of leverage (monotically increasing) and residual function as a function of normalized residual. Then for each point, plot its potential and residual. The plot can aid in classifying unusual observations as high-leverage points, or outliers or both.
- What to do with outliers? They should not be automatically discarded. Instead, they may be very informative (model assumptions, etc.). One should examine these outliers, then take appropriate actions such as: down-weighting or deleting outliers, transforming data, considering a different model, etc.

Adding variables to a regression equation [RABE, Chapter 4] After adding a variable, there are four cases

- The new variable has a insignicant coefficient and not change other parameters: the new variable can be ignored.
- The new variable has a significant coefficient and no change other parameters: the new variable should be included.
- Whenever the new variable substantially change other variables, should check for colinearity.

Feature transformation:

- Rationale: for choosing and transforming features are two fold:
  - Scale/linearity: the scale of a feature X should be chosen s.t. when X increases by two-fold, we expect its contribution to Y als increases by two fold. This is very important consideration, for example, we consider a model where X is inverse of the allele frequency of a SNP, and Y is phenotype. The bigger X, we expect the bigger Y, however, the scale of X is incorrect: there is no simple linearity between Y and X.

  - Additivity: important for combining features. Ex. the phenotype depends on all alleles of a gene (some strong, some weak), the additivity question is essentially the question of whether one strong site is equivalent to how many weak sites.

- Centering: often helps simplifies the model, $z_j = x_j - \bar{x}$.
- Standarization of features: often helps interpretation. Ex. the intercept β_0 is now the average outcome variable when the data point is average (if not standardized, the feature value at 0 may be meaningless, e.g. body height). It particularly helps interaction. Ex. consider the regression problem:

$$\texttt{earn} \sim \texttt{height} + \texttt{male} + \texttt{height} \cdot \texttt{male} \tag{4.163}$$

  where `male` is the binary variable of sex (1 if male, 0 female). The coefficients now have interpretation: (1) intercept: the earning of females of average height; (2) coefficient of `height`: effect of height in females; (3) coefficient of `male`: the average difference of male vs females; (4) coefficient of `height`·`male`: the increase of the effect of height in males, relative to females.

- Unit length scaling: alternative way is to scale the features: $z_j = x_j/\|x\|$, then $\|z_j\| = 1$.
- Logarithmic transformation: when the variables are positive, often do log. transformation. Suppose we do log. transformation on the response variable Y, we have:

$$\log Y = f(X) \qquad \log Y' = f(X + \Delta X) \Rightarrow \frac{Y'}{Y} = \exp[f(X + \Delta X) - f(X)] \tag{4.164}$$

  Thus the coefficient of a feature X means: the increase of Y (in multiplicative terms) due to the increase of X by a unit. Thus if coefficient is 0.081, it means Y is increase by $\exp(0.081) = 1.084$, i.e. Y increases 8.4%. The same interpretation is useful in logistic regression.

- Categorical variables: if there are multiple categories, need to define multiple indicator variables, one for each category. But since the indicators variables are not independent (sum to 1), one category is chosen as the reference, and the effect of any other category is always relative to this reference category.

Rank deficiency and feature correlation:

- When X is not of full rank, then $X^T X$ is singular, and the least square fit $\hat{\beta}$ is not well-defined. This could result from: (1) correlation of features; (2) $p > N$: more features than the number of data points.
- Correlated features: if multiple features are correlated, then a true causal feature (assume it exists) may not be chosen, if its effect has alreay be explained by other correlated features. Alternatively speaking, there may be multiple ways of choosing the model (features) that explain the data equally well. Ex. [Hastie] in heart disease risk data, the feature, blood pressure is not chosen.

Model diagnosis: the goal is to check if the model is sufficient to model the variation of data points.

- Residual plot: we could test the assumptions by
  - Scatter plot: r_i should be uncorrelated with $\hat{y}_i$ or x_i, thus we could draw scatter plot of r_i vs. $\hat{y}_i$ or x_i.
  - Probability plot: r_i should follow $N(0, 1)$ distribution, thus we could draw normal Q-Q plot and compare the observed r_i distribution with the normal distribution.
- Outliers: several ways to deal with outliers in the residual plot:
  - Sometimes they hint that our model assumptions are wrong.
  - Down-weight or delete outlying data points.

Robustness of linear model:

- Problem: suppose we are testing the effect of X on Y, but we need to control for Z. If the true relation between Z and Y is not-linear, e.g. Y increases with larger Z, but reaches a plateau when Z is large enough. Then fitting a linear model between Z and Y may overcorrect Z when it is large, and undercorrect Z when it is small. As a result, even if Y is similar for different X, its residual (after subtracting the effect of Z) may show correlation with X.
- Example: [Large-Scale Psychological Differences Within China Explained by Rice Versus Wheat Agriculture, Science, 2014] Study how invention (number of patents) varies with rice/wheat agriculture, controlling GDP per capita. If GDP has a non-linear effect on patents, then one may not control for it correctly, and this can lead to false correlation between patents and rice.

4.6 Generalized Linear Models

1. Introduction to generalized linear models (GLM)

   Reference: [Agresti, Introduction to Categorical Data Analysis, 2ed, Chapter 3] [McCullagh & Nelder, Section 2.2, Chapter 6], [KNNL, Chapter 14]

   Problems of GLM:

   - Relation to contingency table analysis.
   - Extensions of GLM: dependency between samples, different variance of errors.

   Lessons:

   - When developing a GLM, ask if the assumption made by the model fit the characteristics of the data. For example, logistic function describes the sigmoid relation between π and x. When the situation is: π initially increases (linearly) with x when x is small, but approaches to 1 as x increases, then sigmoid model may not fully capture this situation.
   - Checking model assumptions: e.g. Poisson regression assumes that sample mean and sample variance are equal - this assumption can be tested in real data.
   - Model checking using residuals (or more generally model predictions): we assess the model predictions at given samples, and compare those with observations. If the model fit is good, the residuals should follow some distributions and generally small.

   GLM: the three components.

   - The random component: the distribution of Y given X. This may depend on some dispersion parameter ϕ.
   - The systematic component: the linear predictor

   $$\eta = \sum_{j=1}^{p} x_j \beta_j = X\beta \tag{4.165}$$

   - The link function between the two components: let $\mu = \mathrm{E}(y|X)$:

   $$g(\mu_i) = X_i\beta \tag{4.166}$$

   The function $g(\cdot)$ should be a monotonically differentiable function. Equivalently, we could have: $\mu_i = g^{-1}(X_i\beta)$. For linear regression, the link function is identity link. When the link function is log, we have log-linear model; when it is logit function, it is the logistic regression model.

Common GLM for binary data: [Agresti, 3.2], [GCSR, Chapter 16]

- Logistic regression model: the response variable follow Bernoulli distribution with parameter $\pi(x)$, and it increases with the independent variable x. The link function is logit:

$$\log \frac{\pi(x)}{1-\pi(x)} = \alpha + \beta x \tag{4.167}$$

  The parameter β describes the direction and the magnitude of change of $\pi(x)$ with x, and α is similar to the "threshold" - when it is large (negative value), need large value of x to overcome it.

- Probit model: when the response variable is between 0 and 1, we could view Y_i as a probability. We can convert it to the Z-scores and model Z-scores as linear functions of X. More generally, using the CDF, we could relate a probability to the value of some random variable, let it be $X\beta$ for our regression purpose:

$$\mathrm{E}\,(Y_i|X_i) = \Phi(X_i\beta) \tag{4.168}$$

  Thus this is a GLM with link function $g(\cdot) = \Phi^{-1}(\cdot)$. When the response variable is binary, we have:

$$P(Y_i = 1|X_i) = \Phi(X_i\beta) \tag{4.169}$$

  This is similar to the logistic regression model, except that the logit function is replaced by the CDF of standard normal distribution.

- Interpretation of discrete-data model in terms of latent continuous data: sometimes a model with discrete data may be equivalent to a model with latent continuous variable. Ex. for the probit model above, we imagine some latent variable u_i corresponding to $X_i\beta$: $u_i \sim N(X_i\beta, 1)$, and y_i is determined by u_i via:

$$y_i = \begin{cases} 1 & \text{if } u_i \geq 0 \\ 0 & \text{if } u_i < 0 \end{cases} \tag{4.170}$$

  The advantage of this model is: the first step is a simple distribution, and the second step is deterministic, so it allows a simple Gibbs sampler.

  - Remark: this is a special case of data augmentation, which facilitates Gibbs sampling (or EM).

Common GLM for count data [Agresti, 3.3], [GCSR, Chapter 16]

- Poission regression: we often need to model the count data, which is assumed to follow Poisson distribution. Suppose Y given X follows Poisson distribution with mean μ, and the link function is log, we have, $\log \mu_i = X_i\beta$, or,

$$y_|X_i \sim \text{Poisson}(\exp(X_i\beta)) \tag{4.171}$$

  In the simple case (one covariate), we have:

$$\log \mu_i = \alpha + \beta x_i \tag{4.172}$$

  The meaning of β is: a one-unit increase of x has a multiplicative effect of e^β on μ. The log-likelihood function is given by:

$$l(\beta) = \sum_i y_i X_i\beta - \sum_i e^{X_i\beta} - \sum_i \log y_i! \tag{4.173}$$

  Note that the last term is constant.

- Often x is continuous, then μ at a particular value of x is not well-defined. We may need to divide x into bins for this analysis.

- Overdispersion problem: Often the Poission assumption is violated because of overdispersion (the variance is larger than expected from Poisson distribution). This could be caused by for example, fail to include other independent variables (this would not be a problem for normal model, because it has a separate parameter for variance) - then there is additional heterogeneity between samples.
- Negative Binomial regression: we define the negative binomial by two parameters μ (mean) and D (dispersion parameter): $y_i \sim \text{NB}(\mu_i, D)$, and the parameters are:

$$\text{E}(Y) = \mu, \quad \text{Var}(Y) = \mu + D\mu^2 \tag{4.174}$$

  Negative binomial GLM is then similar to Poisson regression: using the log link function:

$$\log \mu_i = \alpha + \beta x_i \tag{4.175}$$

- Count regression with rate data: in some problems, our data is rate: e.g. number of murders in a city of certain population size. So for each sample Y_i, we have an index t_i (such as population size), the log-linear model has the form:

$$\log(\mu_i/t_i) = \alpha + \beta x_i \tag{4.176}$$

  Or equivalently:

$$\mu_i = t_i \exp(\alpha + \beta x_i) \tag{4.177}$$

- Identity vs log link function: the log link function is often used. Need to decide which one is a better model. When x is binary, the form of link function does not make a difference though (simple transformation of parameter).

Model checking:

- Explorative analysis of the relation between y_i and x_i: plot y_i and x_i to check if there is any correlation. Do log-transformation to see if that improves linearity: choose log-link function if $\log(y_i)$ is more linear to x_i.
- Examining the assumptions of the model: e.g. for to check if Poisson regression is a good fit, we collect all samples Y_i under the same values of x, and see if the sample mean and the sample variance are equal.
- Model comparison/selection by the deviance: to select between two models of different complexity, calculate the deviance, defined as the -2 times the difference of log-likelihood. Two types of deviance:
  - Null deviance: Null Deviance = 2(LL(Saturated Model) - LL(Null Model)) on df = df(Sat) - df(Null), where Saturated model uses one parameter for each observation. It should roughly follow chi-square distribution with df specified above.
  - Residual deviance: Residual Deviance = 2(LL(Saturated Model) - LL(Proposed Model)).

  Small deviance of the proposed model means the model fits the data relatively well. One can define the "deviance residual", which characterizes how good the model predicts a particular observation using the LL function. This is used in R `glm()` (standarized, thus roughly following normal distribution).
- Model diagnosis using residuals: similar to linear model, we can assess the model fit by $y_i - \hat{\mu}_i$ (under the normal linear model, this should follow normal distribution). For count regression, however, the variance of the residual depends on y_i, so we "standardize" the residual by dividing the standard error:

$$e_i = \frac{y_i - \hat{\mu}_i}{\sqrt{\hat{\text{Var}}(y_i)}} \tag{4.178}$$

  For Poisson GLM, the Pearson residual is $e_i = (y_i - \hat{\mu}_i)/\sqrt{\hat{\mu}_i}$. The residuals follow approximately normal distribution when μ_i is large. Overdispersion under Poisson GLM can be detected if residuals tend to be larger at higher values of independent variables (or mean of y_i).

Statistical inference of GLM:

- Fitting GLM: the Fisher scoring algorithm, effectively Newton-Raphson algorithm for maximizing the log-likelihood function.
- Inference of parameter β: Wald, LRT or score tests are commonly used. Wald test is simpler, but LRT is more trustworthy.

2. Logistic regression

   Reference: [Hastie, Section 4.4], [KNNL, Chapter 14], [GCSR, Chapter 16]

   Logistic regression model:

   - Background: logit function. It is also called log-odds function:

     $$\text{logit}(p) = \log \frac{p}{1-p} \tag{4.179}$$

     Logit function is the inverse of the logistic function, let $x = \text{logit}(p)$, then:

     $$p = \frac{1}{1+e^{-x}} \tag{4.180}$$

   - Model: some explanatory variables increase the chance that the event occurs ($Y = 1$); others decrease it; and yet others have no effect. Thus model the probability that the event occurs as a logistic function:

     $$P(Y = 1|X, \beta) = \frac{1}{1+e^{-X\beta}} \tag{4.181}$$

     where $X\beta = \beta_0 + \sum_{j=1}^{p} \beta_j X_j$. Alternatively, let $\mu = P(Y = 1|X, \beta)$, and $P(Y = 0|X, \beta) = 1 - \mu$ (the probability that the event does not occur), the model can be equivalently defined using the logit as the link function:

     $$\text{logit}(\mu) = X\beta \tag{4.182}$$

   - Intepretation of parameters: for the parameter β_j, suppose every other explanatory variable is fixed, and we increase x_j by one unit, the log odds-ratio (OR) will change by β_j. In particular, when x_j itself is also binary, the coefficient β_j is the difference of log-odds of the group $x_j = 1$ vs. $x_j = 0$.
   - Classification: To classify an object, we only need to test if $P(Y = 1|X, \beta) > P(Y = 0|X, \beta)$, or simply: $Y = 1$ if $X\beta > 0$, and $Y = 0$ otherwise. To apply to the multi-class case: for every two class, the log-odds follows a linear function (the coefficients depend on the class). Typically one class is chosen as the reference class.
   - Assessing a model: to assess a fitted model, let the predicted probability of $Y = 1$ be:

     $$\hat{\pi} = \frac{1}{1+\exp(-X\hat{\beta})} \tag{4.183}$$

     Then we could test the model by comparing $\hat{\pi}_i$ and the observed y_i.

   Parameter estimation and inference:

   - Maximum-likelihood parameter estimation: let π_i be the $P(Y = 1|x_i, \beta)$, we have the log likelihood function:

     $$l(\beta) = \sum_{i=1}^{N} [y_i \log \pi_i + (1 - y_i) \log(1 - \pi_i)] \tag{4.184}$$

Plug in $\pi_i = P(y_i = 1|X_i, \beta)$, we have:

$$l(\beta) = \sum_{i=1}^{N} [y_i X_i \beta - \log(1 + \exp(X_i \beta)] \tag{4.185}$$

To maximize:

$$\frac{\partial l(\beta)}{\partial(\beta)} = \sum_{i=1}^{N} (y_i - \pi_i) X_i = 0 \tag{4.186}$$

No closed form solution. Typically use Newton's method or gradient ascent to find $\hat{\beta}$.

- Wald test: the distribution of the MLE can be approximated by a normal distribution, thus the significance of coefficient β_j is estimated by a Z score.
- Likelihood ratio test: a special case is to test if some $\beta_j = 0$. This can be formulated as a nested test where H_A has one more parameter, so the standard LRT can be applied. It is not, however, as general as the Wald test.
- Group comparison test: this is an analogy of F test. To test a feature X_j, test if the groups defined by the value of X_j have different frequencies of the two classess. If X_j is discrete, and has K different values, then the test of whether K groups have equal frequencies is a test of $K \times 2$ table, and χ^2 test or Fisher's exact test can be applied.
- Testing mode of feature action: the above test only tests if all groups have equal frequencies. More generally, we may want to know exactly how a feature acts: e.g. which group has the highest risk (frequency of the positive class), whether the effect of a feature is monotonic, etc.

Extension of the basic model:

- Feature selection and model improvement: to improve the model, we can do this repeatedly: remove the least significant feature (by Z score) and retrain the model, until no feature can be removed.
- Regularization: maximize the objective function:

$$\max_{\beta_0, \beta} \left[l(\beta_0, \beta) - \lambda \sum_{j=1}^{p} |\beta_j| \right] \tag{4.187}$$

where $l(\beta_0, \beta)$ is the log-likelihood function.

3. Introduction to nonlinear regression

Reference: [Gallant, Am Stat, 1975], [Seber & Wild, Nonlinear regression], [KNNL, Chapter 13]

Non-linear regression:

- Model: the response variable is a function of (multiple) predictors, where the functional form wrt. parameters is non-linear:

$$Y = f(X; \theta) + \epsilon \tag{4.188}$$

where ϵ is error of distribution $N(0, \sigma^2)$. Given n observations $(x_i, y_i), 1 \leq i \leq n$, the goal is to estimate θ and assess its significance.

- Example: exponential regression model. Suppose the response variable is given by:

$$Y_i = \gamma_0 + \gamma_1 \exp(\gamma_2 X_i) + \epsilon_i \tag{4.189}$$

where $\epsilon_i \sim N(0, \sigma^2)$. This models the case where the response may increase with X ($\gamma_1 > 0$)), but is bounded as X becomes very large ($\gamma_2 < 0$).

- Linearization: sometimes it is possible to do variable transformation s.t. the resulting model is linear. However, the error term after transformation may no longer be normally distributed. For the example above:
$$\log(Y_i - \gamma_0) = \log \gamma_1 + \gamma_2 X_i + \log(\epsilon_i) \tag{4.190}$$
If we define $Y_i' = \log(Y_i - \gamma_0)$, the model is linear, however, the error term is not normal.
- Paramter estimation: the estimator of θ is obtained by minimizing sum of square error:
$$SSE(\theta) = \sum_i [y_i - f(x_i; \theta)]^2 \tag{4.191}$$
The minimization is generally performed by the numerical method. The estimator of the variance is given by:
$$s^2 = \frac{1}{n-p} SSE(\hat{\theta}) \tag{4.192}$$

Significance test of nonlinear regression:

- Linear approximation: suppose $\hat{\theta}$ is MLE of θ, the idea is to linearize the function $f(x;\theta)$ in the neighborhood of $\hat{\theta}$ so that we can apply the results from linear regression to obtain the distribution of the estimator of θ. Suppose θ^* is the true value of the parameter, then we have, for each x_i:
$$f(x_i; \theta^*) \approx f(x_i; \hat{\theta}) + \nabla f(x_i, \hat{\theta}) \cdot (\theta^* - \hat{\theta}) \tag{4.193}$$
Let $z_i = y_i - f(x_i, \hat{\theta})$, then we have:
$$z_i = \nabla f(x_i, \hat{\theta}) \cdot (\theta^* - \hat{\theta}) + \epsilon_i \tag{4.194}$$
This is a linear regresion with response variable z_i and predictors $(\theta^* - \hat{\theta})$. Apparently the estimator of θ^* is $\hat{\theta}$.
- Inference: Let $F(\theta)$ be the matrix $[\partial f(x_i, \theta)/\partial \theta_j]$, then following the results from linear regression, $\hat{\theta}$ follows normal distribution with mean θ^* and variance-covariance matrix $\sigma^2(F^T F)^{-1}$. Define:
$$\hat{C} = [F\hat{(\theta)}^T F\hat{(\theta)}]^{-1} \tag{4.195}$$
Then the confidence interval of θ_i is $\hat{\theta_i} \pm t_{0.025}\sqrt{s^2 \hat{c_{ii}}}$, where $t_{0.025}$ is the critical value for t distribution of $n-p$ d.o.f. The test is applicable if the function $f(x;\theta)$ satisfies some regularity conditions: most notably, second partial derivative must be continuous.

4.7 Linear Mixed Model

Reference: [McCulloch & Searle, Generalized, linear and mixed models]
Review of ANOVA: concepts [McCulloch, 1.1-1.2]

- Factors: in ANOVA, our goal is to assess the effect of explanatory variable(s). If the variable is discrete, we call it a *factor*, and the values of a factor are *levels*. The ANOVA problem is to compare mean across different levels.
- With multiple factors: they could be nested or crossed. Two factors could also have interaction effect.
- Balanced data: if we have equal number of observations/samples in each cell.

Fixed and random effects: [McCulloch, 1.3-1.4]

- Fixed effect model: e.g. study the drug effect on blood pressure, at different dosages. The response of the j-th subject at dosage i:
$$\mathrm{E}(y_{ij}) = \mu + \alpha_i \tag{4.196}$$
where α_i is the effect of the dosage i - fixed effect.

- Random effect model: e.g. study the drug effect at different clinics. The response of the j-th subject at the i-th clinic is:
$$\mathrm{E}(y_{ij}) = \mu + a_i \tag{4.197}$$
where a_i is the effect of the drug at the i-th clinic. Different clinics may be different in some ways - doctors, equipments, etc., thus the effect may vary. On the other hand, each clinic represents just one sample of the population (all putative patients), so we treat a_i as random effect: $a_i \sim N(0, \sigma_a^2)$.

- Variance component: under a pure random effect model, we have $\mathrm{Var}(y_{ij}) = \sigma_a^2 + \sigma^2$, which has two components: random effects and error.

- Choose fixed or random effect? Generally we choose random effect if we are interested in the population effect, and each group represents only one sample of the population.

Introduction to linear mixed model: [McCulloch, 1.5-1.7]

- ANOVA form: suppose for the drug treatment problem, we have both multiple clinics (i) and multiple dosages (j), the response can be written as:
$$\mathrm{E}(y_{ij}) = \mu + a_i + \beta_j + c_{ij} \tag{4.198}$$
where a_i is the effect of clinic (random), β_j is the effect of dosage (fixed) and c_{ij} is the interaction effect.

- Regression form: suppose we are testing a drug on subjects, with longitudinal data. For the i-th patient, its j-th measurement of response is y_{ij}, and we have dosage d_{ij}. We are interested in the effect of the drug. Furthermore, the drug effect might depend on age, so we control for age well. The age at the j-th time point is x_{ij}. Our model is:
$$\mathrm{E}(y_{ij}) = a_i + b_i d_{ij} + \gamma x_{ij} \tag{4.199}$$
where a_i is the base-level of the i-th subject (random), b_i is the response of the i-th subject (again random) and γ is the age effect (fixed). We could write $b_i = \beta + b_i'$, then β is the effect we want to estimate, and b_i' is the individual variation.

- Inference: generally, we will need to marginalize the random effects. To do that, we consider the marginal distribution of main random variables, typically, considering their variance or covariance.
  - REML (restricted ML): we remove all the fixed effects, and do ML estimation.
  - Quasi-likelihood: use mean and variance instead of the full distribution to estimate parameters.

Normal random effect model: [McCulloch, 2.2]

- Model: let μ_i be the mean of the i-th group ($1 \le i \le m$, it is $\mu + a_i$, where a_i is the random effect of the i-th group
$$y_{ij}|a_i \sim N(\mu + a_i, \sigma^2) \tag{4.200}$$
where $1 \le j \le n$. And
$$a_i \sim N(0, \sigma_a^2) \tag{4.201}$$
The ANOVO H_0: $\sigma_a^2 = 0$.

- Variance components: the idea of inference is that the variance and covariance encode information of σ^2 and σ_a^2:

$$\mathrm{Var}\left(y_{ij}\right) = \mathrm{E}_{a_i}\left[\mathrm{Var}\left(y_{ij}|a_i\right)\right] + \mathrm{Var}_{a_i}\left[\mathrm{E}\left(y_{ij}|a_i\right)\right] = \mathrm{E}_{a_i}(\sigma^2) + \mathrm{Var}_{a_i}(\mu + a_i) = \sigma^2 + \sigma_a^2 \tag{4.202}$$

$$\begin{aligned} \mathrm{Cov}\left(y_{ij}, y_{ik}\right) &= \mathrm{E}_{a_i}\left[\mathrm{Cov}\left(y_{ij}, y_{ik}|a_i\right)\right] + \mathrm{Cov}_{a_i}\left[\mathrm{E}\left(y_{ij}|a_i\right), \mathrm{E}\left(y_{ik}|a_i\right)\right] \\ &= \mathrm{E}_{a_i}(0) + \mathrm{Cov}_{a_i}(\mu + a_i, \mu + a_i) \\ &= \sigma_a^2 \end{aligned} \tag{4.203}$$

  Intuitively, the covariance of two samples within a group is not 0 because they share the same random effect in that group.

- Likelihood: within the i-th group, the covariance matrix is given above, written in the matrix form: $V_i = \sigma^2 I_n + \sigma_a^2 J_n$ where I_n is identity matrix and J_n is $n \times n$ matrix of 1's. Thus the vector $\mathbf{y_i}$ follows MVN:

$$\mathbf{y_i} \sim N(\mu \mathbf{1}_n, V_i) \tag{4.204}$$

  When the data is balanced, the MLE is: $\hat{\mu} = \bar{y}$,

$$\hat{\sigma}^2 = \mathrm{MSE} = \frac{1}{m(n-1)} \sum_{i,j} (y_{ij} - \bar{y})^2 \tag{4.205}$$

$$\hat{\lambda} = \hat{\sigma}^2 + n\hat{\sigma}_a^2 = \frac{1}{n}\mathrm{SSA} = \frac{1}{m} \sum_{i=1}^{m} n(\bar{y}_i - \bar{y})^2 \tag{4.206}$$

  Note that it's possible that $\hat{\sigma}_a^2 < 0$ in practice.

Random intercept model: [McCulloch, 3.5]

- Model: the j-th subject ($1 \le j \le n$) of the i-th group ($1 \le i \le m$) follows:

$$y_{ij}|a_i \sim N(\mu + a_i + \beta x_j, \sigma^2) \tag{4.207}$$

  Note that we assume x_{ij} only depends on j, thus droping x. And $a_i \sim N(0, \sigma_a^2)$. The distribution can be written in the matrix form. First the vector of response variable in group i:

$$\mathrm{E}\left(\mathbf{y_i}|a_i\right) = [\mathbf{1x}][\mu\beta]^T + \mathbf{1}a_i \tag{4.208}$$

  where $x = [x_1, \cdots, x_n]^T$. And $\mathrm{Var}\left(\mathbf{y_i}\right) = \sigma^2 I_n + \sigma_a^2 J_n = V_0$. From this we can obtain the full matrix form of all y_{ij}'s using Kronecker production (see the text).

- MLE: $\hat{\mu} = \bar{y} - \hat{\beta}\bar{x}$, and

$$\hat{\beta} = \frac{S_{xy}}{S_{xx}} = \frac{\sum_j x_j \bar{y}_{\cdot j} - n\bar{x}\bar{y}}{\sum_j x_j^2 - n\bar{x}^2} \tag{4.209}$$

$$\hat{\sigma}^2 = \frac{SSR}{m(n-1)} \tag{4.210}$$

  where SSR is the residual sum of square. Note that $\hat{\mu}$ and $\hat{\beta}$ do not depend on the unknown variance σ_2 and σ_a^2, and are exactly the same when a_i effects are fixed or random or no effect.

Introduction to Linear Mixed Model (LMM)

- Ref: Chapter 2. [West and Welch, Linear Mixed Models, a practical guide, 2007]. `https://stats.idre.ucla.edu/other/mult-pkg/introduction-to-linear-mixed-models/`. `http://www2.stat.duke.edu/~sayan/Sta613/2018/lec/LMM.pdf`

- Motivation: dependency in the data. Ex. (1) Grouping structure: patient data from multiple doctors. The patients from the same data are expected to share something in common. (2) Repeated measurements of the same subjects. (3) Longitudinal data: the same individual is measured multiple times.

- Fixed and Random factors: fixed factors are given groups, e.g. by gender. Random factors are random sample of groups, e.g. patients from a doctor. Fixed and random effects: coefficients of a covariates that are the same for all samples; or associated with the levels of a random factor.

- Note: in the grouping structure case, the sharing of individuals within a group can be modeled in different ways: they could have the same mean (which is different from population mean); or the effect of a certain covariate is the same in a group, but vary across groups. These lead to random intercept or random slope models.

- About multiple random effects: even if we model only random intercept, it is possible to have multiple random effects. Basically, a sample may come from different ways of grouping: e.g. in patient study, doctor is one factor, and study cohort (multiple studies) could be another factor.

- LMM with a single random effect as random intercept: we have N samples, and p covariates (fixed effects). In addition, we have m groups, with n_i the size of group i. Note: $\sum_i n_i = N$. For each group i, its group mean may deviate from population mean, we have:

$$y_i = X_i\beta + Z_i u_i + \epsilon_i \tag{4.211}$$

  where y_i is $n_i \times 1$ and X_i is $n_i \times p$. Z_i here is simply a n_i-dim. vector of 1 (we only look at samples in group j) and u_i is a scalar (mean of group i). Across all groups, we use Z to denote the $N \times m$ "design matrix", with $Z_{ij} = 1$ if sample j belongs to the group i and 0 o/w. Now we have

$$y = X\beta + Zu + \epsilon \tag{4.212}$$

  where u is $m \times 1$ vector. Generally, we do not estimate u, but rather treat u as random: $u_i \sim N(0, G)$. For random intercept model, $u_i \sim N(0, \sigma^2)$. More generally, we write $G = G(\theta)$, where θ is the parameter of G. Often we use $\epsilon \sim R$, and $R = \sigma_e^2 I$. Our problem is to infer β, θ, R given y and X, Z.

- General LMM: we could have q random effects. Let u_i be the vector of q random effects, with i group index, $1 \le i \le m$. Our model for group i is:

$$y_i = X_i\beta + Z_i u_i + \epsilon_i \tag{4.213}$$

  where Z_i is $n_i \times q$ design matrix, $u_i \sim N(0, D)$ is q dim. random vector, and D is the covariance matrix of q random effects. We have $\epsilon_i \sim N(0, R_i)$ is n_i-dim. random vector, and R_i captures across-sample correlation within group i. We can combine the model of all m factors into a single model. Let Z be the block diagonal matrix:

$$Z = \begin{pmatrix} Z_1 & 0 & \cdots & 0 \\ 0 & Z_2 & \cdots & 0 \\ & \cdots & & \\ 0 & \cdots & 0 & Z_m \end{pmatrix} \tag{4.214}$$

  Now Z is $N \times mq$ dimension. u is now concatenation of all u_i vectors, so it is $mq \times 1$ vector of q random effects in m groups. And ϵ is N-dim. random vector. We can define the prior distribution of u and ϵ as: $u \sim N(0, G)$ and $\epsilon \sim N(0, R)$, where

$$G = \begin{pmatrix} D & 0 & \cdots & 0 \\ 0 & D & \cdots & 0 \\ & \cdots & & \\ 0 & \cdots & 0 & D \end{pmatrix} \qquad R = \begin{pmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & 0 \\ & \cdots & & \\ 0 & \cdots & 0 & R_m \end{pmatrix} \tag{4.215}$$

- Marginal linear model: to make inference, we can marginalize u_i (consider single group for now). We define $\epsilon_i^* = Z_i u_i + \epsilon_i \sim N(0, V_i)$, where

$$V_i = Z_i D Z_i^T + R_i \tag{4.216}$$

  Assuming D and R are given, we can obtain log-likelihood as function of β. More generally, let θ be the parameters of $D(\theta)$ and $R_i(\theta)$, we can compute:

$$l_i(\beta, \theta) = P(y_i | X_i, Z_i, \beta, \theta) \tag{4.217}$$

  We can also collapse all the m groups, and have a single model with G and R are covariance matrix.

- ML and REML estimation of θ : for a given θ, we can do MLE of β, $\hat{\beta}(\theta)$, we can then plug in this estimate in the log-likelihood of θ. This estimator is biased, because it does not take into account the uncertainty of $\hat{\beta}$. REML is preferred: it marginalizes β using uniform prior. It maximizes this objective function:

$$l(\theta) = \ln \int L(\beta, \theta) d\beta \tag{4.218}$$

- Hypothesis testing: suppose we test parameter $\hat{\beta}$. If we use the variance of the estimator V, where θ (random effect parameters) are fixed from REML estimation, we ignore the uncertainty of $\hat{\theta}$, this leads to underestimated error and inflation of type I error.

- Estimation of fixed and random effects β and u: Once θ is estimated, we can treat as known and estimate β, this is Best Linear Unbiased Estimator (BLUE) - Equation (2.19) of the LMM book. We can also estimate u, assuming G and R are given. The result is Best Linear Unbiased Predictor (BLUP). The result is given by:

$$\hat{u} = G Z^T V^{-1} (Y - X\hat{\beta}) \tag{4.219}$$

  Intuitively, this is a linear combination of the residual terms, with weights determined by Z and other terms. Assuming a simple model where there is a single random effect and m groups, the BLUP of u of a group is given by the group mean, but also the prior.

- Correlation of BLUP with errors: see " Residual analysis for linear mixed models" `https://www.ime.usp.br/~jmsinger/MAE0610/Mixedmodelresiduals.pdf`. The BLUP of the random effect part is a linear function of the errors. Intuition: suppose we have a model with a single random factor with many levels/subgroups. Suppose in one subgroup, by chance (errors), the group mean is somewhat high, we will learn a higher group mean in this group - we attribute larger errors to large random effects. This leads to correlation of predicted random effects and errors. To address this problem, use cross-validated prediction (use all data except one to train the parameters).

Linear mixed model [GCTA paper, personal notes]

- Model: let X be covariates (fixed effect) and G be genotypes (p variants). Let β be the fixed effect parameters, and γ_j be the effect size of variable $G_j, 1 \leq j \leq p$. In other words: we have p random factors - each individual is assigned randomly to one of two (or three) groups for each factor. The effect size for a variant is the difference of the mean of the two groups. We have $\gamma_j \sim N(0, \sigma_a^2)$. The model:

$$Y = X\beta + G\gamma + \epsilon \tag{4.220}$$

  where $\epsilon \sim N(0, \sigma_e^2)$.

- Remark: in this setting, we have a large number of factors, with each factor only 2 or 3 levels. We do not learn a different random effect for each factor (SNP), rather, we assume that the random effects are shared across all factors.

- Inference: the marginal model is given by $Y = X\beta + \epsilon^*$, where $\epsilon^* \sim N(0, V)$, where

$$V = G\sigma_a^2 I G^T + \sigma_e^2 I = \sigma_a^2 GG^T + \sigma_e^2 I \tag{4.221}$$

where GG^T is the GRM (up to a constant). We can then solve σ_a^2, σ_e^2 by REML. Let $\hat{V} = \hat{\sigma_a}^2 GG^T + \hat{\sigma}_e^2 I$ be the estimated covariance matrix, the BLUP of random effect component is given by:

$$y_{\text{BLUP}} = \sigma_a^2 GG^T \hat{V}^{-1}(y - X\hat{\beta}) \tag{4.222}$$

If we use $\sigma_g^2 = M\sigma_a^2$ as the heritability, where M is the number of variants, we can write it in terms of GRM $K = GG^T/M$:

$$y_{\text{BLUP}} = \sigma_g^2 K \hat{V}^{-1}(y - X\hat{\beta}) \tag{4.223}$$

Ref: GBAT paper [Xuanyao Liu].

- Alternative view: viewing random effects as correlated error terms. We can also write $u = G\gamma$ as the total contribution of genetic background (over all SNPs)

$$Y = X\beta + u + \epsilon \tag{4.224}$$

where $u \sim N(0, \sigma_a^2 A)$ is given by the GRM. Under this view, the genetic random effect can be thought of an error term that correlates across samples, with correlation given by the GRM.

Variance component model with score test (SKAT):

- Model: we have $y = \mu + X\beta + \epsilon$, and the prior $\beta_j \sim N(0, \tau)$, where μ is the intercept, and assume know. Our goal is to test $H_0 : \tau = 0$. We test this using the score test. We write the model in matrix form:

$$y|\beta \sim N(X\beta + \mu, \sigma^2 I) \qquad \beta|\tau \sim N(0, \tau I) \tag{4.225}$$

where I Is the identity matrix. Using the property of MVN, Equation 3.24, the marginal of y is:

$$y|\tau \sim N(\mu, \sigma^2 I + \tau XX^T) \tag{4.226}$$

The covariance is exactly the same equation we had before for LMM: XX^T is the kinship matrix, and τ corresponds to heritability.

- Score test: the log-likelihood function is

$$l(\tau) = -\frac{1}{2}(y - \mu)^T [\sigma^2 I + \tau XX^T]^{-1}(y - \mu) \tag{4.227}$$

The score is:

$$S(\tau) = -\frac{1}{2}(y - \mu)^T [\sigma^2 I + \tau XX^T]^{-2}(XX^T)(y - \mu) \tag{4.228}$$

At $\tau = 0$, we have:

$$S = -\frac{1}{2\sigma^4}(y - \mu)^T XX^T (y - \mu) \tag{4.229}$$

4.8 Bayesian Linear Regression

Reference: [Gelman04, chapter 14; Bishop, 3.3]

Bayesian simple linear regression: posterior and BF [personal notes; SuSiE paper]

- Model: $y = x\beta + \epsilon$, where $\beta \sim N(0, \sigma_0^2)$ and $\epsilon \sim N(0, \sigma^2)$. The MLE and s.e. are given by:

$$\hat{\beta} = (x^T y)^{-1} x^T y \qquad s^2 = \frac{\sigma^2}{x^T x} \tag{4.230}$$

The Z-score is given by: $Z = \hat{\beta}/s$.

- BF: Wakefield formula:

$$B = \sqrt{1-r} \cdot \exp\left(\frac{Z^2 r}{2}\right) \text{ where } r = \frac{\sigma_0^2}{\sigma_0^2 + s^2} \tag{4.231}$$

- Posterior distribution: $\beta|y \sim N(\mu_1, \sigma_1^2)$, where

$$\sigma_1^2 = \frac{1}{1/s^2 + 1/\sigma_0^2} \qquad \mu_1 = \hat{\beta} \cdot \frac{\sigma_1^2}{s^2} \tag{4.232}$$

- Alternative calculation of BF in terms of posterior parameter:

$$B = \sqrt{\frac{\sigma_1^2}{\sigma_0^2}} \cdot \exp\left(\frac{\mu_1^2}{2\sigma_1^2}\right) \tag{4.233}$$

  Proof: we first consider the exponential part, using $\hat{\beta} = Zs$:

$$\mu_1 = Zs \cdot \frac{\sigma_1^2}{s^2} = Z \cdot \frac{\sigma_1^2}{s} \Rightarrow \frac{\mu_1^2}{2\sigma_1^2} = Z^2 \frac{\sigma_1^4}{s^2 \sigma_1^2} = Z^2 \frac{\sigma_1^2}{s^2} = Z^2 r \tag{4.234}$$

  And the proof of the quadratic part is simple.

Ordinary linear regression with noninformative prior:

- Model: let X be the data matrix, y be the response variables, β be the parameters (intercept and effects), and σ^2 be the sampling variance. The model:

$$y|\beta, \sigma^2, X \sim N(X\beta, \sigma^2 I) \tag{4.235}$$

  where I is the $n \times n$ identity matrix. The prior is assumed to be uniform on β and $\log(\sigma)$:

$$p(\beta, \sigma^2) \propto \sigma^{-2} \tag{4.236}$$

  Note that log. has the effect of "compression": for instance, if σ belongs to 1 to 100, at the original scale, it only has 10% of being less than 10, but at the log10 scale, 50%.

- Likelihood function:

$$p(y|\beta, \sigma^2, X) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - X_i\beta)^2\right] \tag{4.237}$$

- Posterior distribution: plug in the prior, we have:

$$p(\beta, \sigma^2|y) \propto \left(\frac{1}{\sigma^2}\right)^{\frac{n}{2}+1} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - X_i\beta)^2\right] \tag{4.238}$$

  We write the exponential in quadratic form of β:

$$(y - X\beta)^T(y - X\beta) = \beta^T X^T X\beta - 2\beta^T X^T y + y^T y \tag{4.239}$$

  Apply the quadratic form of $x^T Ax + x^T b + c$, we have:

$$(y - X\beta)^T(y - X\beta) = (\beta - \hat{\beta})^T V_\beta^{-1}(\beta - \hat{\beta}) + C \tag{4.240}$$

  where C is some constant and

$$\hat{\beta} = (X^T X)^{-1} X^T y \qquad V_\beta = (X^T X)^{-1} \tag{4.241}$$

To see what C is (important for the distribution of σ^2), we write the quadratic form in a different way:

$$(y - X\beta)^T(y - X\beta) = [y - X\hat{\beta} - X(\beta - \hat{\beta})]^T[y - X\hat{\beta} - X(\beta - \hat{\beta})] \tag{4.242}$$

Expand this, and we can show that:

$$(y - X\beta)^T(y - X\beta) = (\beta - \hat{\beta})^T X^T X(\beta - \hat{\beta}) + (y - X\hat{\beta})^T(y - X\hat{\beta}) \tag{4.243}$$

Let s^2 be the mean squared error:

$$s^2 = \frac{1}{n-p}(y - X\hat{\beta})^T(y - X\hat{\beta}) \tag{4.244}$$

Then the posterior distribution can be written as:

$$p(\beta, \sigma^2|y) \propto \left(\frac{1}{\sigma^2}\right)^{\frac{n}{2}+1} \exp\left[-\frac{1}{2\sigma^2}(\beta - \hat{\beta})^T X^T X(\beta - \hat{\beta})\right] \exp\left[-\frac{(N-p)s^2}{2\sigma^2}\right] \tag{4.245}$$

- Conditional posterior distribution of β given σ^2: follows normal distribution:

$$\beta|\sigma^2, y \sim N(\hat{\beta}, V_\beta \sigma^2) \tag{4.246}$$

Interpretation: β has mean $\hat{\beta}$, the least-square estimate of β, and covariance matrix $(X^T X)^{-1}\sigma^2$, which is the covariance matrix of the least square estimator under classical statistics. σ^2 has mean s^2, the MLE of σ^2, and dof. equal to $N - p$.

- We could understand/prove the results using relationship between posterior and estimator distribution (see notes in the Introduction of Bayesian). Suppose σ^2 is given, the MLE of β has the distribution:

$$\hat{\beta}|\beta \sim N(\beta, (X^T X)^{-1}\sigma^2) \tag{4.247}$$

Treat $\hat{\beta}$ as given, β follows the distribution $N(\hat{\beta}, (X^T X)^{-1}\sigma^2)$ if $P(\beta)$ is uniform.

- Marginal posterior distribution of $\sigma^2|y$: we integrate out β in the posterior distribution. Using Multivariate Gaussian integral:

$$\int_\infty^\infty \exp\left[-\frac{1}{2\sigma^2}(\beta - \hat{\beta})^T X^T X(\beta - \hat{\beta})\right] d\beta = \frac{(2\pi\sigma^2)^p}{|X^T X|^{1/2}} \tag{4.248}$$

Thus:

$$p(\sigma^2|y) \propto \left(\frac{1}{\sigma^2}\right)^{\frac{n-p}{2}+1} \exp\left[-\frac{(n-p)s^2}{2\sigma^2}\right] \tag{4.249}$$

This is the scaled inverse-χ^2 distribution:

$$\sigma^2|y \sim \text{Inv} - \chi^2(n-p, s^2) \tag{4.250}$$

- Proper posterior distribution: for posterior to be proper (finite integral), two conditions must be satisfied: (1) $N > p$; (2) the rank of X equals p, i.e. the variables (columns) of X must be linearly independent.

- Sampling from posterior distribution:

  1. Computation of $\hat{\beta}$, V_β and s^2.
  2. Draw σ^2 from the inverse-χ^2 distribution.

3. Draw β from the multivariate normal distribution, this is often done by first sampling independent standard normal then using the Cholesky decomposition.

Note that computation of $\hat{\beta}$ and V_β involves the inverse of $X^T X$. This is typically done via QR factorization. See the book.

- Posterior predictive distribution: for a given $\tilde{x}$, given σ^2, we have:

$$E(\tilde{y}|\sigma^2, y, \tilde{x}) = E(E(\tilde{y}|\beta, \sigma^2, y, \tilde{x})|\sigma^2, y, \tilde{x}) = E(\tilde{x}\beta|\sigma^2, y, \tilde{x}) = \tilde{x}\hat{\beta} \tag{4.251}$$

And variance:

$$\text{Var}(\tilde{y}|\sigma^2, y, \tilde{x}) = (I + \tilde{x}V_\beta \tilde{x}^T)\sigma^2 \tag{4.252}$$

The posterior predictive distribution has mean $\tilde{x}\hat{\beta}$, and variance consisting of sample variance σ^2, and the uncertainty due to β.

- Example: estimate incumbency advantage in congressional election. Let y_i be the proportion of votes, and R_i, the main variable of interest, be whether incumbent or not. Also control for vote proportion in the previous election and the incumbent party. The analysis:
  - Data transformation: this is not needed here (typically we should make y normally distributed).
  - Posterior inference: obtain the posterior intervals (quantiles) of the variables and offset. See how this changes with election years.
  - Model checking: (1) Plot standardized residual, $(y_i - X\hat{\beta}_i)/s$, it should be standard normal. Check if there are outliers. (2) Predictive distribution: use existing data, we sample y_i from X_i, and obtain the residuals, and compare the proportion of high residuals vs. observed proportion.

Weighted linear regression:

- Model: a slight generalization of the ordinary linear regression model is the errors with unequal variance (but still independent). Suppose the error of the i-th observation has variance $\sigma_i^2 = \sigma^2/w_i$. Then the likelihood function is:

$$p(y|\beta, \sigma^2, X) \propto \sigma^{-n} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n} w_i(y_i - X_i\beta)^2\right] \tag{4.253}$$

- Weighted least square: the MLE is thus minimization of least square, with weights w_i. Note that if we multiply each row X_i and y_i by $\sqrt{w_i}$, then it becomes the standard least square. So equivalently, we could say each observation has weight $\sqrt{w_i}$, with observations with low errors higher weights.

- Bayesian inference: we perform variable transformation $y_i' = \sqrt{w_i}y_i$ and $X_i' = \sqrt{w_i}X_i$. In matrix form, this is:

$$X' = W^{1/2}X \qquad y' = W^{1/2}y \tag{4.254}$$

where $W = \text{diag}(w_1, \cdots, w_n)$. Then in terms of X' and y', it is the ordinary linear regression. The posterior distribution of β in terms of the orignial X and y:

$$\hat{\beta} = (X^T W X)^{-1} X^T W y \qquad V_\beta = (X^T W X)^{-1} \tag{4.255}$$

Linear regression with known covariance matrix:

- Model: the covariance matrix is given Σ_y, and we have:

$$y \sim N(X\beta, \Sigma_y) \tag{4.256}$$

- Posterior distribution: following our approach of weighted linear regression, we use variable transformation s.t. the regression becomes ordinary. Let $y' = \Sigma_y^{-1/2} y$, we have (linear map of normal RV):

$$y' = \Sigma_y^{-1/2} y \sim N(\Sigma_y^{-1/2} X\beta, \Sigma_y^{-1/2} \Sigma_y (\Sigma_y^{-1/2})^T) = N(\Sigma_y^{-1/2} X\beta, I) \tag{4.257}$$

  Thus we could define $X' = \Sigma_y^{-1/2} X$ and solve the resulting ordinary linear regression. The posterior distribution in terms of original data:

$$\hat{\beta} = (X^T \Sigma_y^{-1} X)^{-1} X^T \Sigma_y^{-1} y \qquad V_\beta = (X^T \Sigma_y^{-1} X)^{-1} \tag{4.258}$$

Linear regression with unknown covariance matrix:

- Model: we assume prior $\beta|\Sigma_y$ is uniform, and the prior $p(\Sigma_y)$. The conditional posterior of β given Σ_y is already solved. To obtain the joint posterior sample, we need the marginal posterior distribution $p(\Sigma_y|y)$. See Equation (14.14) in the book.
- Remark: Σ_y is $N \times N$ matrix, and in general, cannot be estimated (any sample point has its own parameter). Need strong informative prior or some structure of Σ_y (e.g. diagonal, or grouping).
- Independent errors with the variance dependent on some (unknown) constant:

$$\Sigma_{ii} = \sigma^2 v(w_i, \phi) \tag{4.259}$$

  where v is a function, e.g. $v(w_i, \phi) = (1 - \phi) + \phi/w_i$, a mixture of constant variance and weighted variance. The inference problem is to find the posterior $p(\beta, \sigma^2, \phi|y)$.
- Group-specific errors: n observations form I groups, with all observations in one group the same error $\sigma_i^2, i = 1, \cdots, I$. The inference problem is to sample from the posterior $p(\beta, \sigma_1^2, \cdots, \sigma_I^2|y)$. Suppose we use the noninformative prior: $p(\beta, \Sigma_y) \propto \prod_i \sigma_i^{-2}$.
  - The complete posterior distribution:

$$p(\beta, \sigma_1^2, \cdots, \sigma_I^2|y) \propto \left(\prod_i \sigma_i^{-n_i - 2} \right) \exp\left[-\frac{1}{2} (y - X\beta)^T \Sigma_y^{-1} (y - X\beta) \right] \tag{4.260}$$

  - Posterior mode of $p(\sigma_1^2, \cdots, \sigma_I^2|y)$: using the EM algorithm. Note that in the log-posterior distribution, for the E-step, we only need to consider the term that depends on β (missing parameters), thus E-step involves computing:

$$\mathrm{E}_{\text{ old}} \left[(y - X\beta)^T \Sigma_y^{-1} (y - X\beta) \right] \tag{4.261}$$

    averaging over $\beta|\Sigma_y^{\text{old}}, y$. We could solve this posterior of β using weighted linear regression. Then evaluating the expectation above is equivalent to evaluting the expectation of quadratic forms of MVN random variables.
  - Gibbs sampling: the distribution $p(\beta|\Sigma_y, y)$ is simply weighted linear regression. The distribution $p(\sigma_i^2|\beta, y)$ (σ^i are independent) is also simple: scaled inverse-χ^2 distribution from the Bayesian inference of normal distribution (with known mean).

Linear regression with conjugate prior: [Banerjee, Bayesian linear models: gory details; Bishop; GCSR 14.8]

- Prior distribution: use the conjugate prior, where σ^2 follows inverse-Gamma distribution:

$$p(\sigma^2) = IG(\sigma^2|a_0, b_0) \propto \left(\frac{1}{\sigma^2} \right)^{a_0 + 1} \exp\left(-\frac{b_0}{\sigma^2} \right) \tag{4.262}$$

$$p(\beta|\sigma^2) = N(\beta|\beta_0, \sigma^2 V_0) \tag{4.263}$$

We call the joint distribution as Normal-Inverse-Gamma (NIG) distribution:

$$p(\beta, \sigma) = NIG(\beta_0, V_0, a_0, b_0) \propto \left(\frac{1}{\sigma^2}\right)^{a_0+\frac{p}{2}+1} \exp\left(-\frac{b_0}{\sigma^2}\right) \exp\left[-\frac{1}{2\sigma^2}(\beta-\beta_0)^T V_0^{-1}(\beta-\beta_0)\right] \tag{4.264}$$

- Lkelihood function: this is the same as before:

$$p(y|\beta, \sigma^2) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left[-\frac{1}{2\sigma^2}(y-X\beta)^T(y-X\beta)\right] \tag{4.265}$$

- Conditional posterior distribution $\beta|\sigma^2, y$: we first obtain this distribution in order to get the joint posterior. Both β and $y|\beta$ follow normal distribution under given σ^2:

$$\beta|\sigma^2 \sim N(\beta_0, \sigma^2 V_0) \tag{4.266}$$

$$y|\beta, \sigma^2 \sim N(X\beta, \sigma^2 I) \tag{4.267}$$

Using the properties of MVN, we have:

$$\beta|\sigma^2, y \sim N(\hat{\beta}_n, \sigma^2 V_n) \tag{4.268}$$

where

$$\hat{\beta}_n = V_N(V_0^{-1}\beta_0 + X^T y) \tag{4.269}$$

$$V_N = (V_0^{-1} + X^T X)^{-1} \tag{4.270}$$

- Posterior distribution $\sigma^2|y$: we first obtain the distribution $y|\sigma^2$. From the property of MVN:

$$y|\sigma^2 \sim N(X\beta_0, \sigma^2(I + XV_0X^T)) \tag{4.271}$$

We have the posterior:

$$p(\sigma^2|y) \propto \left(\frac{1}{\sigma^2}\right)^{a_0+\frac{n}{2}+1} \frac{1}{|I+XV_0X^T|^{1/2}} \exp\left\{-\frac{1}{\sigma^2}\left[b_0 + \frac{1}{2}(y-X\beta_0)^T(I+XV_0X^T)^{-1}(y-X\beta_0)\right]\right\} \tag{4.272}$$

Ignoring the constant term, we recognize that this is Inverse-Gamma distribution $IG(a_n, b_n)$, where

$$a_n = a_0 + n \qquad b_n = b_0 + \frac{1}{2}(y-X\beta_0)^T(I+XV_0X^T)^{-1}(y-X\beta_0) \tag{4.273}$$

- Posterior distribution: in summary, we have $\sigma^2|y$ follows inverse-gamma and $\beta|\sigma^2, y$ follows normal distribution, so the joint distribution follows $NIG(\hat{\beta}_n, V_n, a_n, b_n)$, or:

$$p(\beta, \sigma^2|y) \propto \left(\frac{1}{\sigma^2}\right)^{a_n+\frac{p}{2}+1} \exp\left(-\frac{b_n}{\sigma^2}\right) \exp\left[-\frac{1}{2\sigma^2}(\beta-\hat{\beta}_n)^T V_n^{-1}(\beta-\hat{\beta}_n)\right] \tag{4.274}$$

- Linear regression with conjugation prior and general error term: slightly more general form:

$$\beta|\sigma^2 \sim N(\beta_0, \sigma^2 V_0) \tag{4.275}$$

$$y|\beta, \sigma^2 \sim N(X\beta, \Sigma) \tag{4.276}$$

where Σ is the error covariance matrix (weighted or correlated errors). The solutions are given by: Using the properties of MVN, we have:

$$\beta|\sigma^2, y \sim N(\hat{\beta}_n, V_n) \tag{4.277}$$

where

$$\hat{\beta}_n = V_N(V_0^{-1}\beta_0 + X^T\Sigma^{-1}y) \tag{4.278}$$

$$V_N = (V_0^{-1} + X^T\Sigma^{-1}X)^{-1} \tag{4.279}$$

It is easy to see that in the extreme case (data dominates the prior), $\hat{\beta}_n$ converges to the MLE: $(X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}y$.

Linear regression with semi-conjugate prior: [Banerjee, Bayesian linear models: gory details; Bishop; GCSR 14.8]

- Prior distribution: we have independent prior (suppose we use noninformative prior for σ^2):

$$p(\sigma^2) \propto 1/\sigma^2 \qquad p(\beta) = N(\beta|\beta_0, V_0) \tag{4.280}$$

- Conditional posterior distribution of $\beta|\sigma^2, y$: similar to the derivation before, we have:

$$\beta|y, \sigma^2 \sim N(\hat{\beta}_n, V_N) \tag{4.281}$$

The mean and covariance matrix are given by:

$$\hat{\beta}_n = V_n(V_0^{-1}\beta_0 + \frac{1}{\sigma^2}X^T y) \tag{4.282}$$

$$V_N^{-1} = V_0^{-1} + \frac{1}{\sigma^2}X^T X \tag{4.283}$$

In the special case where $\beta_0 = 0$ and $V_0 = \alpha^{-1}I$ (used below), we have:

$$\hat{\beta}_N = \frac{1}{\sigma^2}V_N X^T y \tag{4.284}$$

$$V_N^{-1} = \alpha I + \frac{1}{\sigma^2}X^T X \tag{4.285}$$

- Posterior distribution $\sigma^2|y$: we first obtain $y|\sigma^2$ as normal distribution:

$$y|\sigma^2 \sim N(X\beta_0, \sigma^2 I + XV_0X^T) \tag{4.286}$$

The posterior:

$$p(\sigma^2|y) \propto \sigma^{-2}\frac{1}{|\sigma^2 I + XV_0X^T|^{1/2}}\exp\left[-\frac{1}{2}(y - X\beta_0)^T(\sigma^2 I + XV_0X^T)^{-1}(y - X\beta_0)\right] \tag{4.287}$$

This is not a standard parametric distribution, but given any σ^2, it can be numerically evaluated.

- Alternative approach to informative prior [GSCR]: suppose we have the prior $\beta \sim N(\beta_0, \Sigma_\beta)$. We can treat the prior distribution as p prior "data points": for the j-th point, the explanatory variable is equal to the unit vector, and the response variable $\beta_{0,i}$. Thus this is reduced to the problem before with non-informative prior.

- MAP estimator: the log of the posterior distribution is given by:

$$\ln p(\beta|y, \sigma^2) = -\frac{1}{2\sigma^2}\sum_i (y_i - x_i\beta)^2 - \frac{\alpha}{2}\beta^T\beta + \text{const} \tag{4.288}$$

Maximizing this function leads to ridge regression, where the second term corresponds to L_2 penalty of coefficients. The interpretation: the prior $\beta_0 = 0$, thus favor small parameter values (penalty), and α controls the strength of penalty: if α is small, the prior distribution is diffused, thus little penalty.

- Posterior predictive distribution: given $\tilde{x}$, the distribution is given by:

$$\tilde{y}|\tilde{x}, y, \sigma^2 \sim N(\tilde{x}\hat{\beta}_n, \sigma^2 + \tilde{x}V_n\tilde{x}^T) \tag{4.289}$$

  The mean follows the classical results, and the variance has two parts: the sampling variance and variance due the uncertainty of β.

Linear regression with constraints and variable selection:

- Inequality constraints [GCSR, 14.8]: e.g. $\beta \geq 0$ or $\beta_2 \leq \beta_3$. A simple way to handle constraints in is to ignore constraints in posterior sampling, and in the end, simply discard those samples that violate the constraints.
- Variable selection [GCSR, 15.5]: the idea is to put an informative prior on β_j, s.t. it has a significant probability of being 0. Ex. each variable is probability unimportant, but if it has an effect, it could be large, one could use a t or other wide-tailed distribution for $p(\beta)$.

Equivalent kernel: [Bishop]

- Alternative interpretation of posterior predictive mean: we consider the linear basis function, $\phi(x_i)$ instead of x_i. Then the posterior predictive mean can be written as:

$$\phi(\tilde{x})\hat{\beta_N} = \frac{1}{\sigma^2}\phi(\tilde{x})S_N\phi(X)^T y = \frac{1}{\sigma^2}\sum_{i=1}^{N}\phi(\tilde{x})S_N\phi(x_i)^T y_i = \frac{1}{\sigma^2}\sum_{i=1}^{N}k(\tilde{x}, x_i)y_i \tag{4.290}$$

  where the equivalent kernel is defined as:

$$k(x, x') = \frac{1}{\sigma^2}\phi(x)S_N\phi(x')^T \tag{4.291}$$

  Thus the prediction is the weighted average of y_i of the training data. Note that, for any point x, we have:

$$\sum_{i=1}^{N}k(x, x_i) = 1 \tag{4.292}$$

- Interpretation of kernel: $k(x, x')$ is concentrated on the neighborhood of the x', thus in the prediction, the points in the training data close to $\tilde{x}$ will make a large contribution than distant points. Furthermore, for posterior predictive mean of two points: $\tilde{y}(x)$ and $\tilde{y}(x')$, we have:

$$\text{Cov}[\tilde{y}(x), \tilde{y}(x')] = \text{Cov}[\phi(x)\beta, \phi(x')\beta] = \phi(x)S_N\phi(x')^T = \sigma^2 k(x, x') \tag{4.293}$$

  where S_N is the covariance matrix of $\beta|y$ is used. Thus, the predictive mean at nearby points will be highly correlated.

- Kernel regression: the kernel interpretation of predictions leads to a general idea of regression: use a localized kernel directly and use this to make predictions for new input vector.

4.9 Bayesian Hierarchical Linear Models

Reference: [Gelman07, chapter 11-13; Gelman04, Chapter 15]

Motivations of multi-level regression: compromise between complete pooling and no pooling, thus allows to model group variation and let one borrow information from one group to another.

- Group structure: generally, we have group structure in the data. In regression problem: often at the sample level; but could also be at the parameter level. Different groups: share some common characteristics, but also different (unexplained by known factors). Use complete pooling: we ignore between-group variation; use no pooling, we ignore the common shared distribution.

- Use all the data to perform inference for groups with small sample size - pooling. Caveat: the effect paramters of groups are modeled by a common distribution, based on the assumption that the groups share certain aspects (e.g. in the Radon example, the effect of uranium is constant). If this is not the case, i.e. group variation is so large, then there is no benefit.
- Learn about treatment effects that vary with groups. If the interest lies in the effect on groups, then multi-level modeling is the natural appraoch.
- Simple regression on all predictors (including both group-level and individal level ones) is not as good: it does not account for the additional between-group variation (not explained by the group-level predictors). In other words, different groups may differ (in their means) in some aspects not explained by the group-level explanatory variables (if not accounted for, effectively complete pooling). Thus, the missing variation from existing predictors is the important consideration when developing multi-level models.
  Ex. in presidential election, the election year affects the Dem. vote, in addition to other effects alreay modeled such as national economy (this additional variation may be: e.g. the events, the popularity of the candidates, etc.).

Problems of multi-level modeling:

- Example: Radon data in housing sampled from multiple counties. Individual level data: y_i is the Radon level in the i-th house, and x_i is floor level (the individual-level predictor). Group level data: $j[i]$ is the county of the i-th house (group predictor), and u_j is the county-level soil uranium level (the group-level predictor).
- Common problems:
  - Problems about specific groups: e.g. what is the average Radon level of each group?
  - Group level problem: e.g. how uranium level affects Radon level?
  - Population level problem: e.g. what is the effect of the floor level on Radon (suppose this effect is independent of the county)?

A special case of hierarchical linear regression model: random effect model:

- A special case of the random effect model is the hierarical normal model: the mean of each group is a random effect. To model this as a regression problem, for each data point, we treat the group membership as explantory variables (J groups - J variables), then the X is simply the $J \times J$ identity matrix. And β_j in the linear model is the mean of the j-th group.
- Model: more generally, the j-th coefficient: $\beta_j \sim N(\alpha, \sigma_\beta^2)$, or in vector form:

$$\beta \sim N(\alpha\vec{1}, \sigma_\beta^2 I) \tag{4.294}$$

  where I is the identity matrix.
- Relation with the MVN with intra-class correlation: suppose we have for the i-th group: $y_i \sim N(\beta_i, \sigma^2)$ and $\beta_i \sim N(\alpha, \sigma_\beta^2)$. Then we could derive the joint distribution of $y_1, \cdots, y_n$ as MVN. The variance of a sample in the i-th group is:

$$\text{Var}(y_i) = \text{E}[\text{Var}(y_i|\beta_i)] + \text{Var}[\text{E}(y_i|\beta_i)] = \sigma^2 + \text{Var}(\beta_i) = \sigma^2 + \sigma_\beta^2 \tag{4.295}$$

  The covariance of two samples in different groups is clearly 0, and the covariance of y_{i1} and y_{i2} from the i-th group is:

$$\text{Cov}(y_{i1}, y_{i2}) = \text{E}[\text{Cov}(y_{i1}, y_{i2}|\beta_i)] + \text{Cov}[\text{E}(y_{i1}|\beta_i), \text{E}(y_{i2}|\beta_i)] = 0 + \sigma_\beta^2 = \sigma_\beta^2 \tag{4.296}$$

  Therefore, hierarchical normal model is equivalent to a MVN with group structure. And similarly, the MVN with certain group structure can be modeled as a hierarchical normal model.

Classical regression approach to multi-level data:

- Assumptions: y_i of a house is determined by the floor level and the county effect (plus individual variation). The county effect is a function of the uranium level (however, the uranium level cannot explain all the effects of counties). We denote α_j as the county effect on the average Radon level (not observed).
- Complete pooling: pool all the samples from all groups and do regression (treating group-level predictors as individual predictors, and no group indicator). Ex. to answer the question of how uranium level affects Radon by complete pooling:

$$y_i = \beta x_i + \gamma u_{j[i]} + \epsilon \tag{4.297}$$

  The average Radon level of each county is different (unexplained variation from county uranium level), and this additional group variation is ignored when performing individual-level regression.

- No pooling: treat each group independently and perform regression on each group (assuming each group has its own parameter). This is equivalent to introducing group indicator as individual-level predictors:

$$y_i = \beta x_i + \alpha_{j[i]} + \epsilon \tag{4.298}$$

  This approach ignores the fact that groups may be related, thus one group may carry information on another group (overestimate the group variation).

- Two-step analysis: first do analysis on each group, and then perform group-level analysis. In this example, first estimate α_j for each group, then do: $\alpha_j \sim u_j$. Again, the problem is that the individual group analysis may alreay overestimate the group variation.

Multi-level modeling approach with varying intercept:

- Partial pooling: pooling the data from multiple groups (s.t. information from different groups can be used), but only partially s.t. group variation is still accounted. Ex. in the case of estimating average level of each group, partial pooling amounts to a weighted average between: population-level average and group average (the weight depends on the size of the group and the population).
- Varying intercept model: in the Radon example, we model the effect of the county (intercept) on house Radon level:

$$y_i \sim N(\alpha_{j[i]} + \beta x_i, \sigma_y^2) \tag{4.299}$$

$$\alpha_j \sim N(\gamma_0 + \gamma_1 u_j, \sigma_\alpha^2) \tag{4.300}$$

- Equivalent formulation using indicator variables: in general, the group-specific parameters can also be specified with group indicators. In the above example, the first equation can be written as:

$$y_i \sim N(\sum_j \alpha_j I(j[i] = j) + \beta x_i, \sigma_y^2) \tag{4.301}$$

  where $I(j[i] = j)$ is the indicator variable.

Multi-level modeling with varying slopes and intercepts:

- Varying slope model: the group membership can affect the effect parameter (interaction between feature and group indicator). Aagain consider the Radon example, now assume that the effect of floor level (in addition to average Rado level) depends on the county, we have:

$$y_i \sim N(\alpha_{j[i]} + \beta_{j[i]} x_i, \sigma_y^2) \tag{4.302}$$

$$\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} \sim N\left(\begin{pmatrix} \gamma_0^\alpha + \gamma_1^\alpha u_j \\ \gamma_0^\beta + \gamma_1^\beta u_j \end{pmatrix}, \Sigma \right) \tag{4.303}$$

  where Σ is the covariance matrix (see below why it is needed).

- The varying slope model under classical regression: the varying slope effect can be captured using interaction between group indicator and invidual-level predictor. In the Radon example, we define invidual level predictor $v_i = u_{j[i]}$, and express y_i in terms of only invidual-level predictors, including group indicator, and their interactions:

$$y_i = a + bv_i + c_{j[i]} + dx_i + ev_i x_i + f_{j[i]} x_i + \epsilon_i \tag{4.304}$$

- Correlation between group-level intercepts and slopes: when the data points are not centered, larger α_j means smaller β_j (as the line has to pass through the center). Centering the data can alleviate the problem.

Relation to modeling interaction in regression: in general, interaction can be modeled with a multi-level model, however, this may be different from traditional approach.

- Tradiational model: suppose Y is a function of X_1, and X_1 effect depends on X_2, then our regression:

$$Y = \beta_1 X_1 + \beta_2 X_1 \cdot X_2 + \beta_0 + \epsilon \tag{4.305}$$

- Multi-level model: we assume the regression:

$$Y = \beta_1 X_1 + \beta_0 + \epsilon \tag{4.306}$$

  To model the dependence of X_1 effect on X_2, we assume $\beta_1 \sim N(\gamma_1 X_2 + \gamma_0, \sigma^2_{\beta_1})$. The difference is: (1) traditional model: the dependence of β_1 is fixed on X_2; (2) multi-level model: the dependence on X_2 itself is random. Thus, the variance of Y given X_1 and X_2 is constant given the traditional model, but depends on the value of X_1 under the multi-level model.

General hierarchical regression model:

- Model: the likelihood:

$$y|X, \beta, \Sigma_y \sim N(X\beta, \Sigma_y) \tag{4.307}$$

  The population distribution: given the explanatory variables of β, denoted as X_β, and the coefficients α, the distribution of β:

$$\beta|X_\beta, \alpha, \Sigma_\beta \sim N(X_\beta \alpha, \Sigma_\beta) \tag{4.308}$$

  Finally, we have the hyperprior distribution of α:

$$\alpha|\alpha_0, \Sigma_\alpha \sim N(\alpha_0, \Sigma_\alpha) \tag{4.309}$$

- Equivalent to a single linear regression: by modeling the prior distribution of β as additional "data points", we could treat the hierarchical regression model as a single linear regression model. See Equation (15.3) in [GCSR].

Inference of multi-level regression:

- MCMC: the parameters are β (individual level regression coefficients), α (group level regression coefficients) and variance parameters Σ_y and Σ_β. The update rules are: we use Gibbs sampling to sample:

$$\forall j : \beta_j|\alpha, \Sigma_\beta, \Sigma_y, y \tag{4.310}$$

  This is regression of a single group: the prior of β_j is determined by α and Σ_β. Next we use Gibbs sampling:

$$\alpha|\beta, \Sigma_\beta \tag{4.311}$$

  This is a single regression for the higher-level parameters using β (β_j for each group is a data point). For the variance parameters, we can use Scaled-inverse-χ^2 or MH update:

$$\Sigma_y|y, \beta, \qquad \Sigma_\beta|\alpha, \beta \tag{4.312}$$

  To initialize: regression with noninformative priors can be used to sample the initial values.

- Traps in Gibbs sampling: the sampler could be slow when the group variance parameter is close to 0. Then the group parameters will all be forced close to the population mean, and in the next round, the group variance parameter will be sample close to 0 (as the group parameters are close), and so on.
- Alternative Gibbs sampling:
  - All-at-once Gibbs sampler: treat the hierarchical model as the single-level regression model, and alternatively update regression coefficients and the variance parameters.
  - Scalar Gibbs sampler: update one parameter a time. In particular, for the regression coefficient, this is similar to the stepwise regression in non-Bayesian approach.

More complex multi-level models:

- Non-nested models: e.g. regression of earnings on ethnicity, age and height. The individuals are grouped by ethnicity and age, denoted as $j[i]$ and $k[i]$, respectively. Let z_i be the height, the regression:

$$y_i \sim N(\alpha_{j[i],k[i]} + \beta_{j[i],k[i]} z_i, \sigma_y^2) \tag{4.313}$$

$$\begin{pmatrix} \alpha_{j,k} \\ \beta_{j,k} \end{pmatrix} = \begin{pmatrix} \mu_0 \\ \mu_1 \end{pmatrix} + \begin{pmatrix} \gamma_{0j}^{\text{eth}} \\ \gamma_{1j}^{\text{eth}} \end{pmatrix} + \begin{pmatrix} \gamma_{0k}^{\text{eth}} \\ \gamma_{1k}^{\text{eth}} \end{pmatrix} + \begin{pmatrix} \gamma_{0jk}^{\text{eth}\times\text{age}} \\ \gamma_{1jk}^{\text{eth}} \times \text{age} \end{pmatrix} \tag{4.314}$$

  The last three terms can be modeled as normal distribution with mean 0.
- Structure or modeling of regression predictors:
  - Regression coefficents of classical models: whether predictors are in the model can be viewed as a special case of the multi-level model. When the variance of a coefficient is 0, the predictor is out; if ∞, the predictor is in.
  - Grouping regression predictors: impose structures/grouping in the predictors. E.x. modeling presidential election outcome (Dem. vote), the explanatory variables include a number of economic measures: suppose they are all in the scale, we may assume β_j are from a common distribution, thus making all β_j's close togeter.
  - Modeling regression coefficients: e.g. regression of cancer risk on the food consumption (362 individuals with 87 foods). Each food can be characterized by the level of 35 nutrients, thus the data can be used to infer the effect of nutrient on cancer risk. Let β_j be the effect of food j, it can be modeled as:
$$\beta_j \sim N(z_j \gamma, \sigma_\beta^2) \tag{4.315}$$
    where z_j is the vector of nutrient level, and γ is the effect of nutrients.
- Network structure: group memberships are not always disjoint.

4.10 Bayesian Generalized Linear Models

Bayesian inference of GLM: [GCSR, Section 16.4].

- Model: the model is specified by $g(\mu) = X\beta$, where $\mu = \mathrm{E}(y|X)$ and $g(\cdot)$ is the link function. Sometimes we have a dispersion parameter ϕ (e.g. for Negative Binomial regression).
- General procedure: generally the posterior distribution of β does not an analytic form, so need approximation/sampling. The general procedure is:
  1. Obtain posterior mode $(\hat{\beta}, \hat{\phi})$.

2. Normal approximation about the posterior mode as the starting point for simulation: $p(\beta|\hat{\phi}, y) \approx N(\beta|\hat{\beta}, V_\beta)$ where V_β is determined by the asymptotic approximation (second derivative of log-likelihood function). This is weighted regression problem.
3. Sample posterior by MH.

- Normal approximation: let $\eta_i = X_i\beta$ be the predictor. Let $L(y_i|\eta_i, \phi)$ be the log-likelihood function. We use the second order Tayler expansion as approximation of L (normal approximation):

$$L(y_i|\eta_i, \phi) \approx -\frac{1}{2\sigma_i^2}(z_i - \eta_i)^2 + \text{const} \tag{4.316}$$

- Posterior mode (iterative regression): the problem is essentially linear regression with prior equal to the estimation in the previous round. Thus Newton's method becomes iterated linear regression.

Bayesian logistic regression [Bishop, Section 4.5]:

- Normal approximation of posterior distribution: suppose we have the logistic regression:

$$P(Y = 1|X, \beta) = \sigma(X\beta) \tag{4.317}$$

where σ is the sigmoid function. The prior is $\beta \sim N(\beta_0, S_0)$. The posterior is given by:

$$\log P(\beta|y) = \log P(\beta) + \log P(y|\beta) = \log N(\beta|\beta_0, S_0) + \sum_{i=1}^{N} [y_i X_i \beta - \log(1 + \exp(X_i\beta)] \tag{4.318}$$

Take the second derivative wrt. β, we have:

$$-\frac{\partial^2}{\partial\beta^2} \log p(\beta|y) = S_0^{-1} + \sum_i \pi_i(1 - \pi_i) X_i^T X_i \tag{4.319}$$

where $\pi_i = P(y_i = 1|X_i, \beta)$. Using the normal approximation of posterior distribution, we have: $\beta|y \sim N(\hat{\beta}, S_n)$ where $\hat{\beta}$ is the MAP estimator of β and S_n satisfies:

$$S_n^{-1} = S_0^{-1} + \sum_i \pi_i(1 - \pi_i) X_i^T X_i \tag{4.320}$$

- Predictive distribution and model evidence: in both cases, $p(\beta)$ follows normal distribution: prior normal for model evidence and posterior normal approximation for posterior predictive distribution. Let $\beta \sim N(\mu_\beta, \Sigma_\beta)$. We need to solve a problem of integration: a convolution between sigmoid and normal functions:

$$P(y = 1|X) = \int \sigma(X\beta) p(\beta) d\beta \tag{4.321}$$

We first show that in the integral, the dimension of variables can be reduced. The idea is to integrate over the variable $X\beta$ (but cannot directly apply Change of Variable Theorem because of the difference of dimensionality), over the region defined by $X\beta$. Let $a = X\beta$, we have (by Fubini's Theorem):

$$\int \sigma(X\beta) p(\beta) d\beta = \int \left(\int \delta(a - X\beta)\sigma(a) da \right) p(\beta) d\beta = \int \sigma(a) p(a) da \tag{4.322}$$

where

$$p(a) = \int \delta(a - X\beta) p(\beta) d\beta \tag{4.323}$$

It can be shown that $p(a)$ is simply the multivariate normal distribution of β subject to the linearity constraint, $X\beta = a$, and this marginal distribution is also normal with mean and variance:

$$\mu_a = X\mu_\beta \tag{4.324}$$

$$\sigma_a^2 = X\Sigma_\beta X^T \tag{4.325}$$

The integral can be approximated by using the probit function. For the sigmoid function, we have:

$$\sigma(a) = \Phi(\lambda a) \tag{4.326}$$

where Φ is the CDF of standard normal, and $\lambda^2 = \pi/8$. The result is given by (see [Bishop]):

$$P(y = 1|X) \approx \int \sigma(a)p(a)da \approx \sigma\left(\frac{\mu_a}{\sqrt{1 + \pi\sigma_a^2/8}}\right) \tag{4.327}$$

4.11 Shrinkage Methods and Variable Selection

Reference: [Hastie, Section 3.4]

Motivations for shrinkage:

- Large p, small N problem: when the number of features is large, or in the case of categorical variables, the number of categories is large, the number of data points would be small to learn regression coefficients.
- Parameter shrinkage: prefer models where most parameters are small or zero.
- Structure of predictors: impose additional structure on predictors. Most commonly: (1) group predictors s.t. each group of predictors have the same parameter, e.g. haplotype regression where haplotypes are clustered; (2) random effects of predictors of the same group.

Feature standardization: it is often necessary to standardize features for regression procedures that penalize complex models by shrinkage of parameters. Without standardization, the parameters of different features are not comparable. Standardization consists of:

- Feature standardization: for the j-th feature, let $\bar{x_j}$ be its mean, and $\text{sd}(x_j)$ be its standard deviation, then we have:
$$x'_{ij} = \frac{x_{ij} - \bar{x_j}}{\text{sd}(x_j)} \tag{4.328}$$
- Residual: substract mean from the response variable:
$$y'_i = y_i - \bar{y} \tag{4.329}$$
- Intercept: after standardization, the intercept would be 0, so for the original model, we would have: $\beta_0 = \bar{y}$.
- Covariance matrix: after standardization, the $p \times p$ matrix $X^T X/N$ represents the sample covariance matrix of the features $X_1, \cdots, X_p$.

Subset selection:

- The best subsets are not necessarily nested: the best subset of size 2 does not always contain the best subset of size 1. To see an example, consider a function: $Y = 0.9X_1 + 1.2X_2$. Suppose there is a feature $X_3 = X_1 + X_2$. At size 1, the best subset is X_3; at size 2, the best subset is (X_1, X_2) with coefficients $(0.9, 1, 2)$.

- Forward stepwise selection: start with the intercept, sequentially add the predictor that most improves the fit. The improvement of fit is often based on F-statistic. Stops when the improvement is no longer significant based on F distribution.
- Backward stepwise selection: start with the full model and sequenntially delete predictors with the smallest Z-score.

Ridge regression:

- Minimize a penalized residue sum of squares:

$$\hat{\beta}^{\text{Ridge}} = \text{argmin}_\beta \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{4.330}$$

  The parameter λ is the complexity parameter. Also to apply the method, all input features need to be standarized. The analytic solution can be found:

$$\hat{\beta}^{\text{Ridge}} = (\mathbf{X^T X} + \lambda \mathbf{I})^{-1} \mathbf{X^T y} \tag{4.331}$$

- Bayesian perspective of Ridge regression: If assume the prior distribution $\beta \sim N(0, \tau I)$, then the Ridge regression is effectively maximizing the posterior distribution of β; and the parameter λ effectively corresponds to the variance of the prior.
- Benefit of ridge regression: when the variables are correlated, the coefficients can be poorly determined and exhibit high variance: a large positive coefficient can be canceled by a large negative coefficient on the correlated feature. By imposing the size constraint on the coefficients, this problem is alleviated.
- Principal component interpretation of ridge regression: let SVD of X be:

$$X = UDV^T \tag{4.332}$$

  where U is eigenvectors of XX^T and V eigenvectors of X^TX. The least square solutions are:

$$\begin{array}{rcl} X\hat{\beta}^{\text{ls}} & = & X(X^TX)^{-1}X^T\mathbf{y} = UDV^T \cdot VD^{-2}V^T \cdot VD^TU^T\mathbf{y} \\ & = & U \cdot \text{diag}(1, 1, \cdots, 1, 0, \cdots, 0) \cdot U^T\mathbf{y} = \sum_{j=1}^{p} u_j u_j^T \mathbf{y} \end{array} \tag{4.333}$$

  Note that $u_j^T\mathbf{y}$ are projections of $\mathbf{y}$ on the orthogonal basis U. Also note that only p dimensions are used in this equation (if all N directions are used, we would have the RHS equal to y). Similarly, the ridge solutions can be written as:

$$X\hat{\beta}^{\text{Ridge}} = \sum_{j=1}^{p} u_j \frac{d_j^2}{d_j^2 + \lambda} u_j^T \mathbf{y} \tag{4.334}$$

  Thus the effect of ridge regression is shrinkage of the y coordinates by $d_j^2/(d_j^2 + \lambda)$. This shrinkage is strongest for those j's of small PCs: in these directions, data points have smaller variance, thus it would be more difficult to determine the gradient of y in these directions.

Lasso regression:

- Lasso objective function: the Lasso estimator is defined by:

$$\begin{array}{rcl} \hat{\beta}^{\text{Lasso}} & = & \text{argmin}_\beta \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \\ & & \text{subject to} \sum_{j=1}^{p} |\beta_j| \le t \end{array} \tag{4.335}$$

We note that this is the convex optimization problem with Slater's condition satisified, thus strong duality holds. We show that the problem is equivalent to minimizing the following objective function with L_1 penalty:

$$\hat{\beta}^{\text{Lasso}} = \text{argmin}_\beta \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{4.336}$$

To see this, we note that the Lagrangian of the constrained optimization problem is:

$$L(\beta, \lambda) = \|y - X\beta\|^2 + \lambda \left(\sum_j |\beta_j| - t \right) \tag{4.337}$$

The primal optimal β^* should minimize $L(\beta, \lambda^*)$ at the dual optimal λ^*. Ignoring the constant term $\lambda^* t$, β should minimize the objective function defined in Equation 4.336 (the constant only affects the minimum of the objective function, but not $\hat{\beta}$). Note that t is not given, thus we do not have to know how the dual optimal λ depends on t.

- L_1 regularization from Bayesian perspective [Murphy, Section 13.3]: we use Laplace prior for β, $p(\beta|\lambda) \propto \exp(-\lambda\|\beta\|)$. The MAP estimator is then given by:

$$\hat{\beta} = \text{argmin}_\beta RSS(\beta) + \lambda\|\beta\| \tag{4.338}$$

  This is the equivalent to Lasso.

- Geometric intuition of lasso: as for constrained optimization problems in general, we visualize the feasibility set and the coutour line of the objective function. In this case, the feasibility set is a square $\sum_j |\beta_j| \leq t$, and the contour line is $\|y - X\beta\|^2 = C$, an ellipsid. The solution of lasso is thus the intersection of the ellipsid at with the square. To see why the intersection often occurs in the corners (thus one of β_j is equal to 0):
  - Depending on the slope of the axis of the contour line of the ellipsid: at some range, the intersection may occur at the line of the square; but for the other cases (e.g. the slope is parallel to the x-axis in 2D case), the intersection occurs at the corner.
  - In general, because of the discontinuity at the corner, the intersections tend to occur at the corner (the boundary point, as opposed to an interior point when the constraint is smooth).

- The tuning parameter of Lasso: the shrinkage factor is defined as:

$$s = t / \sum_{j=1}^{p} |\hat{\beta}_j| \tag{4.339}$$

  where $\hat{\beta}_j$ are least square estimates. As $s = 1.0$, the least square estimate will automatically satisfy the constraint, so there is no effect of shrinkage. As $s \to 0$, the parameters decrease to 0.

- Optimization: the objective function is a quadratic function, thus can be solved with quadratic programming. Efficient algorithms exist for solving the entire Lasso path (as λ varies) - least angle regression (LAR) (see below for cyclic coordinate descent algorithm). The basic idea is that as λ changes, the selected variables change only at a few critical points, which can be determined.

- Remark: learning sparse models. Note that regularization itself (constraints on parameters) does not necessarily lead to sparse models, e.g. L_2 norm. The sparsity of the learned model is a consequence of the non-differentiability of the constraint (L_1 norm). In general, design the constraint s.t. the solution is a sparse model.

- Remark: Lasso implicitly penalize large regression coefficients. This may not be desired in practice. Ex. Lasso model for association test with both common and rare variants. Rare variants generally have large effects, but small explanatory power, so Lasso could penalize rare variants too much.

- Memory usage of Lasso [personal note]: glmnet can be memory expensive. One strategy is to use variable selection: from univariate analysis. However, this may not be safe as a general strategy, because the true effects of a variable is learned by adjusting all others. In GWAS problem, this is probably fine b/c most of SNPs are largely independent.

Comparison of subset selection, Ridge and Lasso regression:

- Shrinkage effect [Murphy 13.3.3]: all methods can be viewed as shrinkage of parameters (smaller number of parameters lead to simpler model, thus lower variance). Let $\hat{\beta}^{OLS}$ be least square estimator of β, the difference (using the case where the data input is orthonomal matrix, i.e. uncorrelated features as an example):
  - Subset selection: drop all variables whose coefficients are ranked lowest. A form of "hard thresholding".
  - Ridge: proportional shrinkage, the estimator is $\hat{\beta}^{OLS}/(1+\lambda)$.
  - Lasso: A form of "soft thresholding". Truncate parameters by a constant $\lambda/2$: the lasso estimator is:
    $$\hat{\beta}^{Lasso} = \text{sign}(\hat{\beta}^{OLS}) \left(|\hat{\beta}^{OLS}| - \frac{\lambda}{2} \right)_+ \tag{4.340}$$
    where the last term is 0 if $|\hat{\beta}^{OLS}| < \frac{\lambda}{2}$.
- Subset selection vs Lasso or Ridge: subset selection is a discrete method, which tends to have high variance: in two independently generated datasets, two subsets may be chosen because of noises in the data.

- Lasso vs Ridge: (Figure 3.11) the constraint in Lasso regression has corners, thus with Lasso, there are many opportunities for the estimated parameters to be zero.

Limitations of Lasso:

- Correlation of variables: [Friedman, Regularization Paths for Generalized Linear Models via Coordinate Descent, 2009] when some explanatory variables are highly correlated, lasso will choose one arbitrarily. In contrast, ridge regression will split the weights among these variables, a preferred choice. Mixing the two (elastic net) may be a better option.

- Consistency of Lasso [Murphy, 13.3.5]: the estimators are always biased. Because of the penalty, it will not converge to the true β even as $N \to \infty$ (not "model selection consistent"). Ideally, we want our estimator of a variable j to be close to its true value, if the effect of j is large.

- Addressing consistency problem: (1) adaptive lasso, where the penalty λ can be different for different variables. (2) Debiasing (Murphy Figure 13.9): use Lasso only to select variables, then do least-square estimator of selected variables.

- Statistical inference of Lasso: Lasso describes an algorithm, but does not directly permit inference. For example, what is the FDR of the features. Boostrap lasso (bolasso): approximate posterior inclusion probabilities, boostrap samples, and choose a variable if it occurs in at least 90% of the sets returned by Lasso, for a given λ.

- Bayesian ideas of addressing the limitations of Lasso: for adaptive Lasso, instead of having a weight for each variable, we could have a hierarchical model of β_j s.t. it has shrinkage property but the extent of shrinkage is also specific to each variable.

General form and elastic net: [Friedman, Regularization Paths for Generalized Linear Models via Coordinate Descent, 2009]

- The general form of the penalty term is: $\lambda \sum_j |\beta_j|^q$. The case $q = 1$ is Lasso, and $q = 2$ is ridge regression. However, in general, with $q > 1$, the penalty does not share the ability of Lasso to set many coefficients exactly to 0 (often preferred: simpler model and interpretation).

- Elastic net: the penalty provides a compromise between ridge and lasso:

$$\frac{1}{2N} \sum_i (y_i - x_i\beta)^2 + \lambda \sum_{j=1}^{p} \left(\frac{\alpha}{2} \beta_j^2 + (1-\alpha)|\beta_j| \right) \tag{4.341}$$

  For simplicity, we also assume x_{ij} are standardized, i.e. $\sum_i x_{ij} = 0$ and $1/N \sum_i x_{ij}^2 = 1$.

- Cyclic coordinate descent algorithm: (conditional minimization algorithm) iteratively update the parameters. At the step j, we assume all parameters $\tilde{\beta}_l, l \neq j$ are known, and we find the optimal β_j. Note that the objective is a quadratic function of β_j. Denote by $R(\beta)$ the objective function defined above, the solution of β_j should satisify the condition that the derivative is equal to 0 if $\beta_j \neq 0$. Suppose $\beta_j > 0$, we have:

$$\frac{\partial R}{\partial \beta_j} = -\frac{1}{N} \sum_i x_{ij}(y_i - \tilde{\beta}_0 - \sum_{l \neq j} x_{il}\tilde{\beta}_l - x_{ij}\beta_j) + \lambda(1-\alpha)\beta_j + \lambda\alpha \tag{4.342}$$

  Solving this equation (using the fact that $1/N \sum_i x_{ij}^2 = 1$) , we have:

$$\beta_j = \frac{\frac{1}{N} \sum_i x_{ij}(y_i - \tilde{\beta}_0 - \sum_{l \neq j} x_{il}\tilde{\beta}_l) - \lambda\alpha}{1 + \lambda(1-\alpha)} \tag{4.343}$$

  Let the first term in the numerator be z, then if $z \geq \lambda\alpha$, the function is minimized at β_j defined above; if $z < \lambda\alpha$, it is minimized at 0. The similar condition exists for $\beta_j < 0$ and $\beta_j = 0$.

- Interpretation of the update rule in cyclic coordinate descent algorithm: first, the simple least square fit of β_j is obtained, between the j-th explanatory variable and the partial residual (fitting y_i using all other features and the current estimates of all other coefficients). Next, we decide if the coefficient should be 0 or not (lasso constraint), by comparing the coefficient with $\lambda\alpha$. Finally, we apply the proportional shrinkage, $1 + \lambda(1-\alpha)$, for the ridge penalty.

- Computational efficiency: see "Covariant update" in the paper. Basically the computation of terms in the update can be simplified by storing the reused terms, the inner product of x_i and y.

- Positivity constraint (not verified): if we want $\beta_j \geq 0$, we simply assume that there is an implicit constraint that $\beta_j \geq 0$ in the previous problem. Every step above would be the same, except that we need to change the update rule: suppose z is defined as before, we have (using the fact that the objective function is quadratic of β_j):

$$\tilde{\beta}_j = \begin{cases} \frac{z - \lambda\alpha}{1 + \lambda(1-\alpha)} & \text{if } z > \lambda\alpha \\ 0 & \text{otherwise} \end{cases} \tag{4.344}$$

Fused lasso: [Variable fusion: a new adaptive signal regression method, 1996; Sparsity and smoothness via the fused lasso; 2005]

- Motivation: the explanatory variables can be ordered, and in the correct model, the coefficients of the nearby variables should be close to each other. Ex. classification of cancer status with mass-spec. data of many compounds, clearly the nearby variables (compounds with similar m/z ratio) have similar chemical properties and thus should have similar effect on cancer status.

- Fusion: our constraint is that the sum of the (absolute difference) of the coefficents of nearby features should be small:

$$\sum_{j=2}^{p}|\beta_j - \beta_{j-1}| \leq t \tag{4.345}$$

  The constraint can be geometrically represented as stripes. To see why this penalty leads to sparse solutions, we can assume we do variable substitution with $\gamma_j = \beta_j - \beta_{j-1}$, and the problem is formulated in terms of γ. This is then a Lasso regression and the solution would encourage $\beta_j = \beta_{j-1}$.

- Fused lasso: to encourage both sparseness, and the closeness of nearby coefficients, we solve this problem:

$$\text{Minimize } \sum_i (y_i - x_i\beta)^2 \qquad \text{subject to } \sum_j |\beta_j| \leq s_1 \text{ and } \sum_j |\beta_j - \beta_{j-1}| \leq s_2 \tag{4.346}$$

Group lasso [Model selection and estimation in regression with grouped variables, JRSSB, 2006]:

- Motivation: suppose the explanatory variables can be grouped, and we expect that the variables tend to be selected as a group. Ex. in the association of multiple genetic markers and phenotype, the markers form groups (e.g. of the same pathway), and there should be only a few groups that are relevant to a phenotype. However, within the group, there is no additional constraint/preference (i.e. no sparsity within the group).

- Group lasso penalty: suppose we have J groups of input variables, our objective is to minimize the least square function subject to:

$$\sum_j \|\beta_j\|_2 \leq t \tag{4.347}$$

  where

$$\|\beta_j\|_2 = \sqrt{\sum_k \beta_{jk}^2} \tag{4.348}$$

  Note that the L_2 component above performs ridge penalty within a group, and the L_1 component (e.g. imagine for some groups, there is only one variable, then it effectively becomes Lasso) encourages sparse group selection.

- Example: consider three variables where the first two form one group, the constraint is then:

$$\sqrt{\beta_{11}^2 + \beta_{12}^2} + |\beta_2| \leq t \tag{4.349}$$

  To see what this constraint set look like, we fix one parameter, and check the 2D picture of the constraint region of the other two. When β_2 is fixed at c, the feasibility set:

$$\sqrt{\beta_{11}^2 + \beta_{12}^2} \leq t - c \tag{4.350}$$

  This is a circle, and we have ridge penalty within the group. When β_{11} (or β_{12}) is fixed at c, the feasibility set:

$$\sqrt{c^2 + \beta_{12}^2} + |\beta_2| \leq t \tag{4.351}$$

  The boundary of this set consists of two pieces of parabolas (the top and the bottom piece) joined together. The points where the two pieces joined (at the x-axis) are not differentiable, creating corners. This allows opportunity of learning sparse models, similar to Lasso.

Lasso of multiple related response variables [CMU CS 10-170 lecture 18]:

- Motivation: suppose we have multiple response variables that are related, i.e. two related traits are likely to also share the same explanatory variables. The general idea is similar to fused lasso, where we penalize the difference between coefficients.

- Graph-guided fused lasso: we penalize the difference between coefficients (of the same explanatory variable) on highly correlated response variables. Let β_{jm} be the coefficient of the j-th explanatory variable on the m-th response variable, then our constraint in addition to lasso is:

$$\sum_{(m,l)\in G} f(r_{ml}) \sum_j |\beta_{jm} - \text{sign}(r_{ml})\beta_{jl}| \leq s \tag{4.352}$$

  where G is the graph representing the relation/similarity between response variables, and (m, l) denotes an edge in G, r_{ml} is the correlation and $f(r_{ml})$ is some monotic function of r_{ml}, e.g. $f(r) = 1$ (unweighted) or $f(r) = |r|$.

- Temporarlly-smoothed lasso: suppose Y_t are response variables over time t. Let $\beta_{j,t}$ be the coefficient of the j-th explanatory variable on Y_t, our constraint in addition to lasso constraint is:

$$\sum_j |\beta_{j,t+1} - \beta_{j,t}| \leq s \tag{4.353}$$

Tree-guided group lasso [CMU CS 10-170 lecture 18]:

- Motivation: a generalization of group lasso (in response variables). Suppose we have multiple response variables (traits) related by a tree: for the traits that are really close, we should choose them as a group (i.e. the same explanatory variables likely influence all); for the traits that are far aways, they should be unrelated (sparsity only).

- Idea: consider two subtrees of traits T_L and T_R. Our penalty consists of two parts: (1) if the two subtrees are very close, then we should choose both subtrees (hence all offspring nodes) as a group (group penalty); (2) if the two subtrees are distant, then we should choose either one of them (lasso penalty). The final penalty should be a mix of the two types of penalty with the weight dependent on the distance between T_L and T_R.

- Example: two leaf nodes (Y_1, Y_2). Let h be the height of the tree (the distance between Y_1 and Y_2), the penalty is:

$$\lambda \sum_j \left[(1-h)\sqrt{\beta_{j1}^2 + \beta_{j2}^2} + h(|\beta_{j1}| + |\beta_{j2}|) \right] \tag{4.354}$$

- Example: three leaf nodes $((Y_1, Y_2), Y_3)$. Let h_1 be the height of the subtree (Y_1, Y_2) and h_2 be the height of the remaining tree. The penalty

$$\lambda \sum_j \left[(1-h_2)\sqrt{\beta_{j1}^2 + \beta_{j2}^2 + \beta_{j3}^2} + h_2(|C_1| + |\beta_{j3}|) \right] \tag{4.355}$$

  where C_1 is the penalty of the two node tree (defined recursively):

$$C_1 = (1-h_1)\sqrt{\beta_{j1}^2 + \beta_{j2}^2} + h_1(|\beta_{j1}| + |\beta_{j2}|) \tag{4.356}$$

- Remark: how the height of the tree is define? Ex. for the three-node example, h_2 should only include the distance from the common ancestor of Y_1 and Y_2 vs. Y_3.

4.11.1 Bayesian Variable Selection

Bernoulli-Gaussian model and l_0 regularization [Murphy, 13.2.2]

- Model: $y = wx + \epsilon$, we can rewrite spike-and-slab prior of w as:

$$\gamma_j \sim \text{Ber}(\pi) \qquad w_j \sim N(0, \sigma_w^2) \tag{4.357}$$

  With this prior, we can write our model as: $y = \sum_j w_j \gamma_j x + \epsilon$. Under this model, only $w_j \gamma_j$ is identifiable.

Automatic relevance determination (ARD) prior [Murphy, 13.7]

- Model: we have $y = wx + \epsilon$, we use a normal prior for w, $w_j \sim N(0, 1/\alpha_j)$ and $\epsilon \sim N(0, 1/\beta I)$. ARD approach would do EB estimation of α, and then obtain the posterior of w. The estimation can be done by EM. The procedure is simpler than spike-and-slab prior.
- How the prior leads to sparsity? See Figure 13.20. We note that when $\alpha_j \to \infty$, we have $w_j \approx 0$, so this leads to variable selection. We claim that EB optimization of α would lead to $\alpha_j \approx \infty$ if a feature j is irrelevant/independent of y. Consider a simple case of one independent variable x. We consider the distribution of y, marginalizing over β. This is given by:

$$y|x, \alpha \sim N\left(\frac{1}{\alpha} x x^T + \frac{1}{\beta} I\right) \tag{4.358}$$

  When y is independent of x, we would expect y to be independent of x, so the distribution of y should be spherical. However, when α is finite, the marginal distribution would not be spherical, but this "wastes" probability mass.

  - Remark: similar to heritability analysis: the covariance of y depends on GRM and heritability. When y and x are independent, REML should lead to $h_g^2 = 0$.

Bayesian Variable Selection in Structured High-Dimensional Covariate Spaces With Applications in Genomics [Li & Zhang, JASA, 2010]:

- Motivations: when selecting the true predictors, there is a certain dependency s.t. if one variable is selected, another related one is likely selected too. Eamples:
  - Gene expression modeling: from promoter composition (words as features) to expression level. The true predictors are correlated: if one word is selected, then its neighbor (defined by Hamming distance) has a higher probability of being related to gene expression too.
  - Cancer CNVs to survival outcome: predictors are CNVs, and there is a linear/spatial dependence.
  - fMRI data to behavior traits: the predictors are voxel intensities, but there is spatial smoothness in the selection of true predictions - true signals usually represent connected regions in the brain.
- Idea: use MRF prior for variable selection indicators. Computationally, the advantage is tha the MRF prior can structure the MCMC moves because effectively MCMC searches a smaller set of configurgations based on the MRF prior (e.g. only smooth configurations for spatially-motivated examples). Phase transition problem: the configruation selected may be sensitive to some hyperparameters.
- Model: consider a linear model $Y = X\beta + \epsilon$, let γ_i be an indicator of whether X_i is selected. The distribution of $\beta_i, 1 \le i \le p$ is:

$$\beta_i|\gamma_i = 0 \sim I_0 \qquad \beta_i|\gamma_i = 1 \sim N(0, \sigma^2 \nu^2) \tag{4.359}$$

where I_0 is a point mass at 0, σ^2 is the residual variance, and ν^2 is the variance of β_i (in the unit of σ^2). The prior of σ^2 follows the standard inverse-gamma conjugate prior. For the prior of γ, suppose we have a graph G representing the dependency of variables, then we have the prior:

$$P(\gamma) \propto \exp(a^T\gamma + \gamma^T B\gamma) \tag{4.360}$$

where $a = (a_1, \cdots, a_p)^T$ is a vector, and $B = (b_{ij})$ is $p \times p$ matrix. Usually, we assume $a_i < 0$, thus any $\gamma_i = 1$ will introduce penalty to $P(\gamma)$, and this leads to *sparsity* of the model. For B, we assume $b_{ij} = 0$ if $(i, j) \notin G$, and $b_{ij} > 0$ otherwise. Then any edge (i, j) s.t. $\gamma_i = \gamma_j = 1$ will be favored by the model, leading to *smoothness* of the model. We also assume a single constant a for all a_i's, and similarly another constant b for all b_{ij}'s.

  - The intuition of the Ising prior is that: we have a certain budget of $\gamma_i = 1$ (due to sparsity), and we want to allocate it s.t. γ is smooth (the neighbors have the same γ_i).

- Inference/Gibbs sampling: we are searching the configruation space of γ. Using Gibbs sampling, we update γ_i at each step, and need to compute $P(\gamma_i|\gamma_{-i}, y)$ where y is all the data. Because $\gamma_i = 1$ or 0, this conditional probability can be computed as (similar to Bayesian model selection):

$$\frac{P(\gamma_i = 1|\gamma_{-i}, y)}{P(\gamma_i = 0|\gamma_{-i}, y)} = \frac{P(\gamma_i = 1|\gamma_{-i})}{P(\gamma_i = 0|\gamma_{-i})} \times \frac{P(y|\gamma_i = 1, \gamma_{-i})}{P(y|\gamma_i = 0, \gamma_{-i})} \tag{4.361}$$

  So the posterior odds is the prior odds multiplied by the Bayes factor. The computation of BF under a given γ follows from standard Bayesian regression analysis, by integrating out β and σ. The computation time for one iteration (p variables) is $O(pp_i^2)$, where p_i is the model size.

- Phase transition of the Ising prior: the proportion of $\gamma_i = 1$ is sensitive to the hyperparameters (a, b). In simulations assuming a regular graph (equal degree), the proportion can change sharply from all 0's to nearly all 1's when one varies the value of b, near the phase transition boundary. The intuition is some kind of positive feedback: as we increases b, it is favored to have more 1's, but as we have more 1's, at some point, it will favor even more 1's (if a lot of a node's neighbors are 1's, then this node should be 1 too).

  - For the Ising prior, the phase transition can be analyzed using the mean field theory: how the proportion of 1's depends on the parameters (temperature).
  - The posterior model has phase transition, but cannot be studied analytically. Some heuristics are offered about how to choose a and b.

- Lessons:

  - In many problems, we favor a certain smoothness in the model (sequence data, spatial/image data, structure ...), and this can be encoded by an Ising prior.
  - Analysis of the model behavior/sensitivity to hyperparameters is important in Bayesian inference: in this case, how the results (proportion of $\gamma_i = 1$) depends on the smoothness parameter.

- Remark: the prior of γ can be modified to incoprorate penalty for $(1, 0)$ edges.

Incorporating biological information into linear models: A Bayesian approach to the selection of pathways and genes [Stingo & Vannucci, AAS, 2011]

- Problem: given gene expression data and responses (e.g. survival outcomes), the goals are (1) a predictive model from expression to response; (2) identification of the relevant genes. The main motivation here is to incorporate pathway information. Methods such as GSEA can only identify genes, but not predict responses.

- Background: some relevant works. Doing dim. reduction on the pathways (PCA) and use the PC (the "supergene") as explanatory variables. Priors in regression that incorporate gene-gene relationship.
- Idea: pathway activities as explantory variables (similar to PCA); within pathways, the selection of genes can be enhanced by an Ising prior representing the network.
- Model: suppose we have K pathways, let θ_k be the indicator of the k-th pathway: whether it is selected. For each gene, we have γ_i as gene indicator. The pathway level activity is the Partial Least Square Regression (PLS) of gene expression vs. response, using only genes in the pathway whose $\gamma_i = 1$. For the k-th pathway, we use $k(\gamma)$ to indicate the subset of genes that are selected. The linear model is:

$$Y = \alpha + \sum_k T_{k(\gamma)}\beta_{k(\gamma)} + \epsilon \tag{4.362}$$

  where $T_{k(\gamma)}$ is the activity of the pathway derived from PLS. The prior of θ follows Bernoulli distributions. The prior of γ is given by the Ising prior: $P(\gamma) \propto \exp(\mu \mathbf{1}\gamma + \eta\gamma^T R\gamma)$ where R is the graph representing the gene relationship. To make the model identifiable, there are also constraints on θ and γ: no orphan gene - a gene cannot be selected if none of its pathway is selected. No empty pathway. A subset of genes may belong to multiple pathways: need to be resolved.
- Inference: the regression parameters will be integrated out. Main parameters to be learned are: (θ, γ, η). Use Gibbs sampling. For $P(\theta, \gamma|\eta, D)$ where D is the data, use MH algorithm. The MH moves are structured to implement the constraints of θ, γ. For the posterior of η, it only depends on γ, so we sample $P(\eta|\gamma) \propto P(\eta)P(\gamma|\eta)$.
- Remark: a main motivation of modeling pathway is that the genes can share information: if some genes in a pathway are chosen, then other genes in the same pathway are more likely to be selected as well. So γ should depend on θ, but this is not explicitly modeled. The dependence is only modeled as extra constraints that θ and γ must satisfy.

Bayesian variable selection regression for genome-wide association studies and other large-scale problems [Guan and Stephens, AAS, 2011]

- Model: let τ^{-1} be the variance of the error (environmental effect) of the phenotype:

$$y = \mu + X\beta + \epsilon \qquad \epsilon \sim N(0, \tau^{-1}) \tag{4.363}$$

  We assume a sparse prior for β. Let γ be the indicators of all SNPs:

$$\gamma_j \sim \text{Ber}(\pi) \qquad \beta_j|\gamma_j = 0 \sim \delta_0 \qquad \beta_j|\gamma_j = 1 \sim N\left(0, \frac{\sigma_a^2}{\tau}\right) \tag{4.364}$$

  where σ_a is the effect size, measured in unit of τ. Ex. a SNP with $\sigma_a = 0.1$ means that the SNP changes y by 0.1 standard deviation (note not the sd. from phenotypic variance).
- Prior of π: we use $\log \pi \sim U(a, b)$, where a is a small number, say $1/p$, where p is the number of SNPs, and b the upper bound (at most 1). Comparing with $\pi \sim U(a, b)$, this prior puts more probability mass on smaller values (at log scale, 0 to 0.001 becomes $-\infty$ to -3, clearly most probability mass are far from -3).
- Prior of σ_a^2: if we set the prior as a constant, the issue is that the more variants we have, the larger PVE will be. This is undesirable. Instead, we assume a prior on PVE, and use the PVE to set the value of σ_a^2. Suppose we know γ, the variance explained by genotypes is:

$$V_G(\gamma) = \frac{\sigma_a^2}{\tau} \sum_{j:\gamma_j=1} s_j \tag{4.365}$$

where s_j is the variance of the SNP ($p_j(1-p_j)$). PVE is related to V_G by $h = V_G/(V_G + 1/\tau)$. This allows to have:

$$\sigma_a^2 = \frac{h}{1-h}\frac{1}{\sum_{j:\gamma_j=1} s_j} \tag{4.366}$$

Note that the constant term τ is canceled out. So in practice, we specify the prior on PVE: $h \sim U(0,1)$, and once h and γ is given (from Bernoulli prior), we can compute σ_a^2.

- Relationship between expected PVE and effect size (personal notes): let s_a be the average variance of SNPs, and p be the number of SNPs, we have:

$$V_G = p\pi s_a \sigma_a^2/\tau \tag{4.367}$$

And the PVE is given by:

$$h = \frac{p\pi s_a \sigma_a^2}{p\pi s_a \sigma_a^2 + 1} \tag{4.368}$$

This shows that at a given h, the more causal SNPs we have, the smaller effect size per SNP. This allows us to estimate PVE due to a single SNP. Suppose we have a SNP with effect σ_j (in the unit of τ) and variance s_j, the PVE of this SNP is:

$$\text{PVE}_j = \frac{\sigma_j^2 s_j/\tau}{V_G + 1/\tau} = (1-h)\sigma_j^2 s_j \tag{4.369}$$

Ex. a SNP with effect 0.2 sd, and AF 0.2, and $h = 0.5$, its PVE is $0.0032 = 0.32\%$. The PVE explained by a single causal SNP on average is simply $h/(p\pi)$, the total PVE divided by the number of causal SNPs.

- Inference: we use MCMC to sample the key parameters h and π, and the configuration γ.

$$P(h, \pi, \gamma|y) \propto P(y|h,\gamma)P(h)P(\gamma|\pi)P(\pi) \tag{4.370}$$

Note that $P(y|h,\gamma)$ integrates out β and τ, which has a closed form. Some key ideas of MCMC: (1) mostly local move by MH, sometimes change many γ_j's at once. (2) Sample γ with large marginal association statistic.

- Estimation of PVE, mapping causal variants and phenotype prediction:
  - PVE: once we sample β and γ, we can obtain the actual PVE explained by the causal SNPs.
  - Mapping: estimate $P(\gamma_j = 1|y)$, this uses Rao-Blackwellisation. Intuitively, this is testing a causal SNP by conditioning on all other causal SNPs.
  - Prediction: use $\mathrm{E}(y_{n+1}|y) = x_{n+1}\mathrm{E}(\beta|y)$.

- Simulation procedure:
  1. Sample 10k SNPs with AF sampled from $U(0.05, 0.5)$.
  2. Sample $h \sim U(0,1)$.
  3. Sample causal SNPs (30).
  4. From h, γ and AF, determine σ_a^2.
  5. Sample phenotypes.

- PVE estimation: Figure 1: scatter plot of Estimated PVE vs. True PVE in simulation. Results: With a large number of causal SNPs, estimation of PVE is difficult (hard to distinguish null with a large number of variants with very small effects).

- Identification of causal SNPs: Figure 3: vary threshold of different methods, and estimate TP and FP rates using ROC. For single-SNP test: vary single SNP BF. For BVSR: vary PIP. For Lasso: first compute solution path as λ varies, and then compute TP and FN rates as λ varies. In simulation with LD, compare region level statistics. Results: multi-SNP methods, BVSR and Lasso, better than single-SNP, due to controlling SNPs.

- Prediction of phenotype: BVSR better than Lasso. This is due to problem with Lasso: single λ controls both sparsity and shrinkage (Elastic Net would be better).

- Calibration of posterior inclusion probability (PIP): Figure 5: proportion of True Positives vs. PIPs. Knowing π, σ_a helps calibration. Note that the BFs are relatively insensitive when σ_a is larger than the true value.

- Application in real data: evaluation using region-based analysis, e.g. prob. that a region contains at least 1 causal SNP.

Scalable variational inference for Bayesian variable selection in regression and its accuracy in genetic association studies [Carbonetto and Stephens, Bayesian Analysis, 2012]

- Model: linear regression

$$y = \beta_0 + \sum_{k=1}^{p} X_k \beta_k + \epsilon \qquad \epsilon \sim N(0, \sigma^2) \tag{4.371}$$

  Use spike-and-slab prior for β_k: $\gamma_k \sim \text{Bern}(\pi)$, and $\beta_k | \gamma_k = 1 \sim N(0, \sigma_\beta^2 \sigma^2)$. The main unknowns are β, γ and the hyperparameters are $\theta = (\pi, \sigma_\beta^2, \sigma^2)$. Setting the prior of hyperparameters: encourage sparsity, e.g. in Crohn's GWAS data, a normal prior on $\log \pi/(1-\pi)$ s.t. 95% prob. mass are in the range of 0 to 70 causal variants.

- VB inference of β, γ given hyperparameters: we approximate $p(\beta, \gamma | y, X, \theta)$ by $q(\beta, \gamma) = \prod_k q(\beta_k, \gamma_k)$. This is valid when X_j's are independent, but not in general. VB inference means that we should update β_k, γ_k by the equation:

$$\log q(\beta_k, \gamma_k) = \text{E}_{q(\beta_{-k}, \gamma_{-k})} [\log p(\beta, \gamma | y, X, \theta)] \tag{4.372}$$

  where we take expectation over $q(\cdot)$ of other parameters. We note that the log posterior has three components:

$$\log p(\beta, \gamma | y, X, \theta) = \log p(\gamma | \pi) + \log p(\beta | \gamma, \theta) + \log p(y | X, \beta, \gamma) \tag{4.373}$$

  We can expand these terms and take expectations. For the first one, we have: $\log \pi \sum_k \gamma_k + \log(1-\pi) \sum_k [p - \sum_k \gamma_k]$. For the last term, we have $-1/(2\sigma^2)[(y - X\beta)^T (y - X\beta)]$, whose expectation over β can be determined analytically. This leads to the VB iterative update (Equations 8-10) in terms of: α_k, the probability that γ_k is 1; and μ_k, s_k^2 the mean and variance of β_k if $\gamma_k = 1$. This is equivalent to solving univariate regression problem, where all other coefficients are given by their posterior mean:

$$y = X_k \beta_k + \sum_{j \neq k} X_j \text{E}(\beta_j | D) + \epsilon \tag{4.374}$$

  - Equation (8): for s_k^2, this is posterior variance of β_k in the univariate regression. Since β_j's are given, this is also the same as the simple regression: $y = X_k \beta_k + \epsilon$.
  - Equation (9): for μ_k, this is the posterior mean of the regression above.
  - Equation (10): the posterior ratio is the product of prior ratio and BF. Note: the log-BF has the term μ_k^2 / s_k^2, which is roughly the chi-square of variable k.

- Averaging over hyperparameters by importance sampling: PIPs of variable k should be averaged over all hyperparameters θ. However, we should weigh them by their posterior density $w(\theta)$. This is done by the approximation of marginal likelihood (summing over β, γ) by ELBO from VB inference.

- VB algorithm: Figure 1. Outer loop: over 100-1000 hyperparameters, with each weighted by ELBO. Inner loop: compute β, γ given θ. Final results: average over hyperparameters.
- Behavior of VB in simple simulations: two variable in different degrees of correlation. Tend to overetimate the mode. The independence of posterior assumption leads to problem. Ex. two perfectly correlated variables, in the true posterior, we should have γ_1, γ_2 highly correlated: PDF ellipse along the diagonal line. However, in the posterior, because the two are independent, we need only one variable γ_1 or γ_2 (either variable is sufficient to explain the data, and are two equal modes), so VB posterior is horizontal or vertical ellipse.
- Behavior of VB in real data of genomic regions: (1) Accurate estimation of hyperparameters. (2) In regions of high LD: VB shows single SNPs, while MCMC captures uncertainty. However, VB still correctly calculates the expected number of causal SNPs in the block (Figure 9).
- Comparison of MCMC vs. VB in real data: WTCCC, 4000 samples, 500K SNPs. Full VB takes a day. Some disagreement of VB and MCMC: however, possible that MCMC miss some regions because of convergence issues.
- Extensions: can be used for binary traits, and other priors of effects.

Bayesian structured sparsity from Gaussian fields [Engelhardt & Adams, arxiv, 2014]

- Background: Bayesian approach to sparsity
  - Spike-and-slab prior (two group): β_j follows a mixture prior, with one component point mass 0. The challenge is to search in an exponential space.
  - Continuous relaxation (one group): e.g. Laplacian prior. Often apply a threshold after learning β_j's - this is called zero assumption (if an effect is very small, its true value is probably 0).
- Motivation: linear model where predictors are correlated. Ex. association analysis, SNPs in LD are correlated. The goal is to encourage "dense within-group" sparsity: the closely related predictors should be all 0's or all 1's.
- Model: the idea is to represent the dependency between z_j's (indicator variables) using MVN as an underlying distribution. Let Γ be the indicator matrix, a diagonal matrix with $\Gamma_{j,j} = z_j$, where z_j is the indicator of the j-th predictor. The prior can be written as:

$$\beta|\Gamma \sim N(0, (\nu\lambda)^{-1}\Gamma) \tag{4.375}$$

  where ν is residual precision and λ the precision of β in the unit of ν. For Γ, instead of using independent Bernoulli distributions, we assume there is an underlying latent distribution (Gaussian field):

$$\gamma \sim N(0, \Sigma) \tag{4.376}$$

  where Σ is positive definite matrix. And $\Gamma_{j,j} = I(\gamma_j > \gamma_0)$.
- Application to eQTL: use posterior probability of inclusion (PPI) to select predictors (SNPs). The method has two properties: (1) It encourages sparsity at the group level (spike-and-slab prior); (2) it encourage dense-within-group sparsity: so if a SNP is chosen, another SNP in high LD may be chosen as well. The results show > 10 times increase of significant cis-eQTL, but smaller number of genes with at least one cis-eQTL.
- **Lesson**: model the dependency of discrete/binary RVs using a latent MVN distribution.

Bayesian Variable Selection for Binary Outcomes in High Dimensional Genomic Studies Using Non-Local Priors [Nikooieneja & V.E Johnson, Review for ASA, 2015]

- Problem of existing Bayesian variable selection: the problem is that a prior of coefficient $f(\beta)$ that has mode at 0 would be hard to distinguish from 0. Suppose we compare two models M_1 and M_2 where M_1 has β, but M_2 does not. Suppose the variable does not actually influence y. Then $P(D|M_1)$ and $P(D|M_2)$ are similar.
- Model idea: a model is specified by variables with non-zero coefficients; and we use the prior densities of each variable s.t. the density is 0 at $\beta = 0$.
- Non-local prior densities: see Figure 1. Note that the model puts smaller penalty on large coefficients comparing with alternative models (decays quadratically, instead of exponentially).
- Model prior: choose a prior form s.t. (1) the same for models with the same number of variables; (2) decreases with more variables; (3) marginal probability of a variable being selected is given by Beta(a, b) for some parameters a and b. Choose a at approximately $\log(p)$, and $a + b = p$, where p is the total number of explanatory variables.
- Learning hyperparameters: two parameters r and τ. r controls the tail behavior, and τ is similar to shrinkage parameters - it controls the penalty and determines the minimum value that the regression coefficient must have in order to be selected. The idea of choose r: based on how likely we will have very large effects. Choosing τ: use simlulations, control the number of false positive variables to be included. We simulate data under null, and compare the null distribution of MLE of parametes vs. the prior densities.

Regression with Summary Statistics (RSS) [Xiang Zhu and Stephens, 2016]

- Goal: from the estimated effect size, its standard error, and LD, learn about the underlying distribution of effect sizes. The idea is that: we treat the estimated effect size as data. Its mean is given by the true effect size, and its uncertainty by the observed standard error.
- Related work: GCTA-COJO, "Conditional & joint analysis of GWAS summary statistics without individual level genotype data".
- The scale of $\hat{\beta}$: since we do not have genotype/phenotype data, we do not know the exact scale. But we can assume that β is in the scale of log-OR for binary traits, and the effect on phenotypic standard deviation for quantitative traits.
- Posterior of effect sizes from summary statistics: our main goal is to determine $P(\beta|\hat{\beta}, S, R)$ where β is the effect size, S its standard error (vector) and R the LD (matrix). The definition of R is the LD matrix (this assumes that genotypes are centered):

$$R_{ij} = \frac{X_i^T X_j}{\sqrt{X_i^T X_i}\sqrt{X_j^T X_j}} \tag{4.377}$$

  To simplify, represent S as a diagonal matrix with diagonal element s_j being the standard error of $\hat{\beta}_j$. The posterior is given by:

$$P(\beta|\hat{\beta}, S, R) \propto P(\hat{\beta}|\beta, S, R)P(\beta) \tag{4.378}$$

  So we will mainly need to specify the prior and determine $P(\hat{\beta}|\beta, S, R)$, which is similar to the likelihood (distribution of some statistic of data).
- Single SNP summary statistics: using the standard results from linear model:

$$\hat{\beta}_j = (X_j^T X_j)^{-1} X_j^T y \quad s_j^2 = (n X_j^T X_j)^{-1} (y - X_j\hat{\beta}_j)^T (y - X_j\hat{\beta}_j) \tag{4.379}$$

  We make the assumption that the effect size is small (or total variance explained by each individual loci is small). So we have $s_j^2 = (X_j^T X_j)^{-1}\sigma^2$, where σ^2 is the variance of y. We can also show that by: when σ^2 is known, $s_j^2 = \sigma^2/(X_j^T X_j)$, from simple linear regression.

- Rewriting $X_j^T X_j$ and $X_j^T X_k$ in terms of s_j, R and σ^2: because $\hat{\beta}_j$ is expressed as functions of these covariance terms. We have:

$$X_j^T X_j = \frac{\sigma^2}{s_j^2} \qquad X_j^T X_k = \frac{\sigma^2}{s_j s_k} R_{jk}. \tag{4.380}$$

- Relationship of $X_j^T X_j$, R, S and population standard deviation of genotypes: let $\sigma_{X,j}$ be the standard deviation of SNP j in the population. Define $D = \text{diag}(\sigma_{X,j})$, and $S = \text{diag}(s_j)$. We have $X_j^T X_j = n\sigma_{X,j}^2$, or

$$\text{diag}((X_j^T X_j)^{-1}) = \frac{1}{n} D^{-2} \qquad S = n^{-\frac{1}{2}} \sigma D^{-1} \tag{4.381}$$

  See Proposition 2.2. And we can also link covariance matrix of X with R by:

$$\frac{1}{n} X^T X = DRD \tag{4.382}$$

  See the beginning of Section 2.4.

- Summary statistics of SNPs in LD: we can show that

$$\hat{\beta} | \beta, S, R \sim N(SRS^{-1}\beta, SRS) \tag{4.383}$$

  The mean vector is:

$$\text{E}\,(\hat{\beta}_j) = s_j \sum_{i=1}^{p} R_{ij} s_i^{-1} \beta_i \tag{4.384}$$

  The intuition is that $\hat{\beta}_j$ is the weighted sum of β_i where the weight is given by $R_{ij} s_j / s_i$. The covariance of the summary statistics:

$$\text{Cov}\,(\hat{\beta}_j, \hat{\beta}_k | \beta, S, R) = s_j s_k R_{jk} \tag{4.385}$$

  depends on the LD between j and k. Or the correlation between the effects of two SNPs is simply R_{jk}. When $j = k$, $\text{Var}\,(\hat{\beta}_j) = s_j^2$.

- Proof of RSS likelihood: our idea is in the expression of $\hat{\beta}$, we replace y with $X\beta + \epsilon$. From the equation of $\hat{\beta}_j$,

$$\hat{\beta}_j = (X_j^T X_j)^{-1} X_j^T y = (X_j^T X_j)^{-1} X_j^T X \beta + (X_j^T X_j)^{-1} X_j^T \epsilon \tag{4.386}$$

  we have,

$$\text{E}\,(\hat{\beta}_j) = (X_j^T X_j)^{-1} X_j^T X \beta \tag{4.387}$$

  Plug-in Equation 4.380, we have:

$$\text{E}\,(\hat{\beta}_j) = \frac{s_j^2}{\sigma^2} [X_j^T X_1 \cdots X_j^T X_p][\beta_1 \cdots \beta_p]^T = \frac{s_j^2}{\sigma^2} \sum_{i=1}^{p} (X_j^T X_i) \beta_i = \frac{s_j^2}{\sigma^2} \sum_i \frac{\sigma^2}{s_j s_i} R_{ji} \beta_i \tag{4.388}$$

  Now σ^2 cancels out, and we have Equation 4.384. Next we prove the covariance of observed effects, we only need to consider the random terms (those related to ϵ):

$$\text{Cov}\,(\hat{\beta}_j, \hat{\beta}_k) = \frac{s_j^2}{\sigma^2} \frac{s_k^2}{\sigma^2} \text{Cov}\,(X_j^T \epsilon, X_k^T \epsilon) = \frac{s_j^2}{\sigma^2} \frac{s_k^2}{\sigma^2} \sigma^2 \cdot X_j^T X_k \tag{4.389}$$

  where we use the fact that Covariance of ϵ (vector) is diagonal with diagnoal entry σ^2. Now we plug in Equation 4.380.
  Remark: even if β_j's are independent, $\hat{\beta}_j$'s are not if SNPs are in LD. This is due to the fact that X_j and X_k are correlated if they are in LD.

- Proof of RSS using matrix form: from the expression of $\hat{\beta}_j$, we can write the vector $\hat{\beta}$ in matrix form as:

$$\hat{\beta} = \begin{bmatrix} (X_1^T X_1)^{-1} X_1^T \\ \cdots \\ (X_p^T X_p)^{-1} X_p^T \end{bmatrix} X\beta + \begin{bmatrix} (X_1^T X_1)^{-1} X_1^T \\ \cdots \\ (X_p^T X_p)^{-1} X_p^T \end{bmatrix} \epsilon \tag{4.390}$$

  Simplifying this, we have:

$$\hat{\beta} = \text{diag}((X_j^T X_j)^{-1}) X^T X\beta + \text{diag}((X_j^T X_j)^{-1}) X^T \epsilon \tag{4.391}$$

  where $\epsilon \sim N(0, \sigma^2 I)$ is a random vector. Thus β is a linear function of a random vector, and we can derive its mean and variance.

$$\text{E}(\hat{\beta}) = \text{diag}((X_j^T X_j)^{-1}) X^T X\beta = \frac{1}{n} D^{-2} X^T X\beta = D^{-1} RD\beta = SRS^{-1}\beta \tag{4.392}$$

  The variance is given by:

$$\text{Var}(\hat{\beta}) = \text{Var}\left(\frac{1}{n} D^{-2} X^T \epsilon\right) = \frac{1}{n^2} D^{-2} X^T \cdot \sigma^2 I \cdot X D^{-2} = \frac{\sigma^2}{n} D^{-1} R D^{-1} = SRS \tag{4.393}$$

- **Remark**: the key of proof is (1) Write $\hat{\beta}$ in terms of linear function of ϵ, the errors as random vector. (2) Express quantities in the expression $(X_j^T X_j)$ and $X_j^T X$ in terms of D, the genotype standard deviation matrix, and R. Once we have expression in terms of D, we can relate to the standard errors S.

- Distribution of Z scores: if we define $Z_j = \hat{\beta}_j / s_j$, we can obtain the distribution of Z_j's. Let $Z = S^{-1}\hat{\beta}$, using property of MVN, it is easy to show:

$$Z | S, R, \beta \sim N(RS^{-1}\beta, R) \tag{4.394}$$

- RSS model under polygenic prior: suppose we have $\beta_j \sim N(0, \sigma^2)$, we can integrate out β analytically. Write in matrix form:

$$\beta | \sigma^2 \sim N(0, \sigma^2 I) \qquad \hat{\beta} | \beta, S, R \sim N(SRS^{-1}\beta, SRS) \tag{4.395}$$

  Let $M = SRS^{-1}$ we have:

$$\hat{\beta} | \sigma^2, S, R \sim N(0, \sigma^2 MM^T + SRS) \tag{4.396}$$

- Priors of β: first, Bayesian sparse linear mixed model (BSLMM) prior:

$$\beta_j \sim \pi N(0, \sigma_B^2 + \sigma_P^2) + (1 - \pi) N(0, \sigma_P^2) \tag{4.397}$$

  where π is the fraction or causal variants. This prior would induce sparsity because it will try to fit a Gassuain prior for even non-risk SNPs. Next is adaptive shrinkage (ASH) prior given by:

$$\beta_j \sim \sum_k \omega_k N(0, \sigma_k^2), \quad \omega \sim \text{Dir}(\lambda, \cdots, \lambda) \tag{4.398}$$

  Under this prior, we choose a certain number of effect sizes beforehand; but the bad ones will be effectively removed by the model (fit a small ω_k).

- Specifying BSLMM prior: similar to the BSLMM model before, we specify the prior using PVE h and PGE ρ (uniform). To relate the effect sizes in RSS with PVE and PGE, we first note the relationship between genotype variance, $\sigma_{x,j}^2$ and the std error in effect size s_j:

$$s_j^2 = \frac{\sigma_y^2}{n \sigma_{x,j}^2} \tag{4.399}$$

where σ_y^2 is the residual variance of SNP j - which is effectively the phenotypic variance because of the assumption RSS makes. Next, we have the phenotypic variance explained by the sparse effects:

$$V_B = \sigma_B^2 \pi \sum_j \sigma_{x,j}^2 = \sigma_y^2 \sigma_B^2 \pi \sum_j \frac{1}{n s_j^2} \tag{4.400}$$

Similarly, we have the polygenic component:

$$V_P = \sigma_P^2 \sum_j \sigma_{x,j}^2 = \sigma_y^2 \sigma_P^2 \sum_j \frac{1}{n s_j^2} \tag{4.401}$$

Following the definitions: $h = (V_B + V_P)/\sigma_y^2$ and $\rho = V_B/(V_B + V_P)$, we can solve σ_B and σ_P as:

$$\sigma_B^2 = h\rho \left(\pi \sum_j \frac{1}{n s_j^2} \right)^{-1} \qquad \sigma_P^2 = h(1-\rho) \left(\sum_j \frac{1}{n s_j^2} \right)^{-1} \tag{4.402}$$

Remark: comparing with BVSR paper, we are using σ_y^2 as total phenotypic variance, whereas the BVSR paper uses $1/\tau$ as residual variance.

- Inference: we parameterize by π, h. We use MCMC to sample both the parameters and γ, the indicator variables for all SNPs. The posterior is given by:

$$P(\gamma, \pi, h | \hat{\beta}, S, R) \propto P(\pi) P(h) P(\gamma|\pi) P(\hat{\beta} | S, R, \gamma, \pi, h) \tag{4.403}$$

  The likelihood conditioned on γ has a closed form under BVSR and BSLMM. The proposal distribution of γ uses rank-based strategy: SNPs with small single-point p-values are sampled with higher probabilities.

- Results:
  - Estimation of PVE: when the true PVE is large, an upward bias by RSS. Likely due to the problem with the assumption (each SNP explains a small heritability).
  - Detecting causal variants: compare BVSR-RSS with BVSR on individual level data, the results are highly correlated, and the AUC is almost the same.

- Related work: one simple idea is that $\hat{\beta}$, S, and R all depend on $X^T X$, $X^T y$ and $y^T y$, so we solve $X^T X$, $X^T y$ and $y^T y$ using MOM, then use them in the common linear model framework. The LD score regression approach converts SNP statistics into χ^2 and solve the regression:

$$\mathrm{E}\left(\chi_j^2 | R\right) = a_0 + a_1 \sum_k r_{kj}^2 \tag{4.404}$$

  This is similar to RSS in that the SNP summary statistics is a linear function of the true effects over multiple SNPs in LD, but the error term is not IID. The PAINTOR approach uses Z-scores, using non-centrality parameter λ. Under the alternative model:

$$Z | R, \lambda \sim N(R\lambda, R) \tag{4.405}$$

  But the semantics of λ is not well-defined.

- Application in other domains: e.g. image analysis, the observation at one pixel is a linear function of the true "effect" at adjacent pixels. One can use RSS kind of analysis to denoise the images.

- Remark:

- For the prior model of β_j, it is independent, while we expect that within one LD block, there is usually 1-2 causal SNPs. Does this cause any problem?
- The model may not work well for molecular QTL, where effect sizes could be quite large.

A simple new approach to variable selection in regression, with application to genetic fine-mapping (SuSiE) [Wang and Stephens, biorxiv, 2019]

- Motivation: in VB approach to BVSR [Carbonette, 2012], the posteriors are independent for each variable/SNP. This does not work well for groups of highly correlated variables. Intuitively, if we can group highly correlated variables, and define posterior on them, then the posterior are roughly independent. Use the "single effect" to capture such highly correlated variables.
- Credible set: defined on single effects. Different from CAVIAR.
- Single effect regression (SER): Equations (2.4) - (2.8). Let γ be the indicator vector (1/0 for each variable), the model assumes $\gamma \sim \text{Mult}(1, \pi)$, where π is the prior of all variables (vector). Then for the selected variable, its effect $b \sim N(0, \sigma_0^2)$. Posterior under SER: for variable j, we compute its BF (using only X_j):

$$B_j = \frac{P(y|X_j, \sigma_0^2, \gamma_j = 1)}{P(y|X_j, \gamma_j = 0)} \tag{4.406}$$

  The PIP of variable j is then:

$$\alpha_j = \frac{\pi_j B_j}{\sum_j \pi_j B_j} \tag{4.407}$$

  The posterior of effect size $b_j|\gamma_j = 1$ would follow normal distribution $N(\mu_{1j}, \sigma_{1j}^2)$, defined as:

$$\mu_{1j} = \frac{\sigma_{1j}^2}{s^2}\hat{b}_j \qquad \sigma_{1j}^2 = \left(\frac{1}{s^2} + \frac{1}{\sigma_0^2}\right)^{-1} \tag{4.408}$$

  Note that the posterior mean is shrinked towards 0: as $\sigma_{1j}^2 > s^2$ because of prior. The posterior can be summarized as $(\alpha, \mu_1, \sigma_1^2)$. The method also needs first and second moment of b_j:

$$\text{E}\left(b_j|y, X, \sigma^2, \sigma_0^2\right) = \alpha_j\mu_{1j} \qquad \text{E}\left(b_j^2|y, X, \sigma^2, \sigma_0^2\right) = \alpha_j(\sigma_{1j}^2 + \mu_{1j}^2) \tag{4.409}$$

  It is also easy to obtain credible set: just rank variables by PIPs.
- EB approach to SER: we can compute the marginal likelihood of σ_0^2 as:

$$p(y|X, \sigma_0^2, \sigma^2) = p(y|\sigma^2)\sum_{j=1}^{p}\pi_j B_j(X_j, y; \sigma^2, \sigma_0^2) \tag{4.410}$$

  where σ_2 is the error of y. One can then estimate σ_0^2 and potentially decide if there is a single effect or not.
- SuSiE model: suppose we have L effects, we can then write the effect size vector as the sum of L single effect models:

$$\mathbf{b} = \sum_{l=1}^{L}\gamma_l b_l \qquad \gamma_l \sim \text{Mult}(1, \pi) \qquad b_l \sim N(0, \sigma_{0l}^2) \tag{4.411}$$

  where γ_l is a p-vector.
- Iterative Bayesian Stepwise Regression (IBSS): in the simpler version, not fitting the parameters σ_{0l}^2. In the l-th step, solve SER model for l given other q's. Accounting for other q's by regressing them out (posterior mean) and obtain the residual in the regression model (as response). Results: in iteration $l = 1, \cdots, L$, we obtain α_l, μ_{1l} and σ_{1l} (vector for each SNP). However, a SNP would not be selected in multiple steps, so in reality, we should consider only the union of SNPs chosen in multiple steps.

- VB inference: approximate the posterior of $\mathbf{b_l}$ as:

$$q(\mathbf{b_1}, \cdots, \mathbf{b_L}) = \prod_l q(\mathbf{b_l}) \tag{4.412}$$

Formulate a more general model (additive effect model):

$$y = \sum_{l=1}^{L} \mu_l + e \qquad \mu_l \sim g_l(\cdot) \tag{4.413}$$

where $g_l(\cdot)$ is the prior of μ_l. For SuSiE, we have $\mu_l = X\mathbf{b}_l$. The posterior approximation is then defined on μ_l's. The ELBO function is given by Eq (B.6):

$$F(\mathbf{q}, \mathbf{g}, \sigma^2; y) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\mathrm{E}_q\|y - \sum_l \mu_l\|^2 + \sum_l \mathrm{E}_{q_l}\left[\log\frac{g_l(\mu_l)}{q_l(\mu_l)}\right] \tag{4.414}$$

Our goal is to optimize this function over $\mathbf{q}$ and $\mathbf{g}$. To show that the IBSS algorithm optimizes this function, we prove in two steps: (1) Show that coordinate descent leads to optimization at each step, a simpler problem Eq (B.11). The key of this step is to show that the expectation of SSE (sum of square error) over q_l is the SSE where we the explanatory variables are replaced by their posterior mean. (2) Show that the solution of Eq. (B.11) leads to the solution of SER.

- Remark: it may be possible to derive the IBSS algorithm using mean field equation, i.e. directly computing $\log q_l(\mu_l)$ by taking expectation of ELBO over q_i's, $i \neq l$.

- Estimating hyperparameters σ_{0l}^2: see Algorithm 4, an optional step before fitting SER model, estimate σ_{0l}^2 using the EB approach (marginal likelihood).

- Determining L: the concern is that when $L >$ number of true effects, PIPs may get inflated. Two possible ideas (1) use the size of credible set: if there is no effect, the PIPs would be very diffused, so the credible set becomes large. However, this may not work well for small regions. (2) Use EB estimation of σ_{0l}^2. In practice, this number is either positive or 0, so it is easy to determine L.

- Remark: the idea of "group selection" with single effect may be applicable to more general settings, e.g. factor analysis (for correlated expression), and HMRF (gene clusters).

- Evaluation of SuSiE performance: comparison with DAG, FINEMAP and CAVIAR. Use 574 real genotypes from GTEx. Setting 1: S (number of causal signals) from 1 to 5, and PVE of all variants from 0.05 to 0.4, 1000 SNPs. Setting 2: $S = 10$ and PVE $= 0.3$, 3,000 to 12,000 SNPs. To run SuSiE: use $L = 10$ in setting 1, and $L = 20$ in setting 2.

- Results of simulations: (1) Calibration of PIPs: actual true positives vs. PIPs (Figure S1), most methods are calibrated. (2) Compare PIPs of individual variables between two methods - scatter plot (Figure 2A). Also color the variables by whether it has a true effect (red) or not. Overall correlated, but SuSiE is better: more red dots below the diagonal line. (3) Power vs. FDR (Figure 2B).

- SuSiE-RSS: directly work on summary statistics, so use LD matrix that is a sum of reference LD and LD inferred from Z-scores $R' = R + \lambda Z^T Z$, where Z is the p-dim. vector of Z scores. If all SNPs are null, then $Z^T Z$ should give the LD in the data.

- Remark: SuSiE can be used with prior, which can be derived from e.g. logistic prior model. However, the hyperparameters can have large estimation error. It may be advantageous to accounting for uncertainty of prior/hyper-parameters, similar to Var-BVS.

- Remark: about calibration of PIPs. The simulation setting mimics eQTL, the per SNP PVE is very large. So PIP calibration may be OK. In GWAS setting, the "boundary case" may lead to inflation.

A fast and flexible Empirical Bayes approach for prediction in multiple regression (MR.ASH) [Youngseok Kim and Stephens, 2020]

- Background: prediction accuracy of Lasso is limited by its tendency to overshrink estimates of the large effects. Elastic net: two tuning parameters. MCMC methods (e.g. BSLMM): convergence can be difficult to diagnose.

- Background: EB methods, with spike-and-slab prior, CML method: conditions on a single best model (i.e. which predictors have non-zero coefficients) instead of summing over all models as a conventional likelihood.

- Idea: similar to Var-BVS with two modifications: (1) In updating q_j of individual variables, use ASH prior. (2) After updating all q_j's, update the ASH prior: for a given component, k, π_k is just the expected number of data points from k.

- Model: we have $y = Xb + \epsilon$, where $b_j \sim g(\cdot)$ and $\epsilon \sim N(0, \sigma^2)$. We assume that the effect size prior $g(\cdot)$ is scaled by σ^2, and parameterized by π_k for component k. Also for simplicity, assume x_j is normalized with variance 1. Let q_j be the VB approximation of posterior of variable b_j. It is parameterized by $\phi_{jk} = P(\gamma_j = k|D)$, the probability that b_j is from component k, and μ_{jk} the posterior mean. The VB optimizes:

$$\max_{q,g,\sigma^2} F(q, g, \sigma^2) \tag{4.415}$$

  This is done by Algorithm 1: iteratively update $q_j, 1 \leq j \leq p$ assuming other variables q_{-j} and g, σ^2 given; and update g and σ^2 given all q_j's are given. When g, σ^2 and q_{-j} are given, the update for q_j is equivalent to solving the regression problem. Let $\bar{r}_j = y - X_{-j}\bar{b}_{-j}$ be the residual, where $\bar{b}_{-j}$ is the mean of all other coefficients. Then we solve this problem:

$$\bar{r}_j = X_j b_j + \epsilon \qquad b_j \sim g(\cdot) \tag{4.416}$$

  We can view this as Empirical Bayes Normal Mean (EBNM) problem. Consider the OLS estimator of this regression, $\tilde{b}_j = X_j^t \bar{r}_j$ (Note: variance of X_j is 1). Then we have this EBNM problem:

$$\tilde{b}_j \sim N(b_j, \sigma^2) \qquad b_j \sim g(\cdot) \tag{4.417}$$

  This gives the $\phi_{jk} = \phi_k(\tilde{b}_j; g, \sigma^2)$, the component mixture proportions, and $\mu_{jk} = \mu_k(\tilde{b}_j; g, \sigma^2)$, component means. Once q_j's are given, we can update π_k in the ASH prior $g(\cdot)$ as:

$$\hat{\pi}_k = \frac{1}{p} \sum_{j=1}^{n} \phi_{jk} \tag{4.418}$$

  Finally we can update σ^2.

- Implementation: (1) the actual implementation does not require X_j to have unit variance. (2) The program does not recompute $\bar{r}_j$ at each step: instead, it only computes the residual $\bar{r} = y - X\bar{b}$ once, then at each step, it adds back the variable j being considered. This saves computation time.

- Space and time complexity: running time is $O(n + K)p$ per iteration, and space is $O(n + p)$ (however, need $O(np)$ for storing X).

- Practical issues: (1) Intercept term: usually centering the data. (2) Initialization: use Lasso (CV) to initialize b_j's.

- Connection with penalized regression methods: the penalized regression problem is often optimized by coordinate descent. The update rule is similar to MR.ASH with a different "shrinkage operator". With different priors, MR.ASH can mimic various shrinkage methods (Figure 1).

- Simulation setting include varying n - sample size. p - number of variables, s - the sparsity level and h - the signal distribution. Baseline setting: $n = 500, p = 2000, s = 20$, PVE $= 0.5$ (used to set error variance) and $h = N(0, 1)$. Also vary X: independent, or correlated or real genotype. Evaluation metric: RMSE in prediction (scaled by error variance).
- Performance evaluation: (1) Several designs (dimensions, signal shape, X): show varying levels of s vs. RMSE (Figure 2, 3). Results: Ridge does not work well in sparse signals and Lasso is worst among all other methods. (2) Varying n, PVE or h: the performance does not vary greatly.
- Computational efficiency: comparable to Lasso with CV, and much faster than Elastic net. Ex. $n = 500, p = 10000$, about 10s.
- **Remark**: design simulation schemes to answer specific questions. Ex. Ridge is expected to perform worse with sparse signals. SuSiE is most advantagenous when x_j's are correlated.
- Remark: one main advantage of MR.ASH is flexible prior, this is however not extensively evaluated.

4.12 Extensions of Linear Models

Statistical Methods with Varying Coefficient Models [Fan & Zhang, Stat Interface, 2008]:

- Motivation: given a regression problem, there may be considerable heterogeneity in the effect of predictors on response variable, e.g. the effect may change over time or space. Thus it is desirable to relax the constraint that coefficients are constants in regression models.
- Model: given data (U, X, Y) where U is some index variable (e.g. time). The model can be written as:

$$Y = \sum_j a_j(U) X_j + \epsilon \tag{4.419}$$

  where $a(U)$ is a vector function of U, with unknown functional form.
- Inference: given the value of u, we want to estimate $a(u)$. The objective is: suppose $a(u)$ is the true model, and we apply it to all data points, the loss (error) over all data points. Since we are estimating only at $a(u)$, our model should perform well at points close to u, but not have to be good for distant points, thus the loss should be discounted for the distant sample points. The general form of the objective function:

$$L = \sum_i L(y_i, \hat{y}_i) K_h(u_i - u) \tag{4.420}$$

  where $\hat{y}_i$ is the predicted value of y at i, and K_h is the kernel function. Suppose our model at u is $X^T a(u)$, then at u_i, the prediction when we apply the $a(u)$ model, should be:

$$\hat{y}_i = x_i^T a(u_i) = x_i^T \left[a(u) + a'(u)(u_i - u) \right] \tag{4.421}$$

  Plug in the prediction y_i and use the L_2 loss, we have the objective function to be minimized at u:

$$L(a, b) = \sum_i \left[y_i - x_i^T a - x_i^T b(u_i - u) \right]^2 K_h(u_i - u) \tag{4.422}$$

- Application in longitude analysis: the model can be written as:

$$Y(t) = \beta_0(t) + X(t)^T \beta(t) + \epsilon(t) \tag{4.423}$$

  In some special models, only β_0 or β is a function of t, but not both.

- Remark: other strategies of dealing with heterogeneity of effects (coefficients) may include: hierarchical model (the coefficients follow a random distribution) or HMM (the coefficients follow a mixture distribution and switch over the index variable). These strategies, however, impose additional constraints on the data that may not be desired.
- Remark: the general idea that underlying heterogeneity there exists some local structure can be exploited in other contexts. Ex. to model DNA evolution, the rate at different positions are different, but the rates at adjacent positions should be similar.

Bayesian factor analysis for testing interactions [Federico Ferrari, interview, 2020]

- Data: 56 blood or urine metabolites, 10K samples and outcome (e.g. BMI). Goal: detect interactions of metabolites.
- Quadratic regression for modeling interactions: let y_i be health outcome and x_i be exposure, we have:

$$y_i = x_i^T \beta_x + x_i^T \Omega_x x_i + \epsilon_i \tag{4.424}$$

  where Ω_x is the interaction coefficients. Most methods use sparsity assumptions for interactions.
- Motivation: the metabolites are often correlated, and they are products of some chemical agents that are likely the causal factors. Interactions probably happen at the level of causal factors. So the interaction terms may not be sparse.
- Model: model exposures as function of factors, and outcome also function of factors, allowing interactions.

$$x_i = \Lambda \eta_i + N(0, \Psi) \qquad y_i = \eta_i^T \omega + \eta_i^T \Omega \eta_i + \epsilon_i \tag{4.425}$$

  Inference: use Gibbs sampling, sample η_i explicitly. Note: cannot marginalize η_i, because y_i is product of two normal RVs.
- Induced regression and interaction: the expectation and variance of y_i have analytic form

$$\mathrm{E}\left(y_i | x_i\right) = \mathrm{tr}\left(\Omega V\right) + (\omega^T A) x_i + x_i^T (A^T \Omega A) x_i \tag{4.426}$$

  where V and A are some matrices. This helps interpretation of coefficients.
- Dealing with missing data: in particular, measures below level of detection (LOD). Use Truncated Normal distribution to impute missing data.
- Adding effect modifiers: let Z_i be other covariates, e.g. age. Then we add interaction terms in y_i: $\eta_i^T \Delta z_i$.
- Using sparse prior for Λ, the factor to exposure effects. Dirichlet-Laplace prior.
- Discussion: improve power over quadratic regression? Model does not need normal assumption of x_i, however, need for z_i.

4.12.1 Linear discriminant analysis (LDA)

Ref: [Hastie, Section 4.3]

Class density approach:

- Class prediction: let $f_k(x)$ be the class density of k, and π_k be the prior probability of class k, then:

$$P(G = k | X = x) = \frac{f_k(x) \pi_k}{\sum_l f_l(x) \pi_l} \tag{4.427}$$

- Discriminant function: for a multi-class problem, functions $\delta_k(x)$ are discriminant functions if any x is classified by the class with the largest value for its discriminant function. A class density approach may be formulated as discriminant function.

LDA with class density:

- Model: suppose each class density is multivariate Gaussian, $N(\mu_k, \Sigma_k)$ for class. Then the log likelihood ratio between any two classes is a linear function of x if $\Sigma_k = \Sigma$ for all k; or a quadratic function of x if not equal variance. The discriminant function:

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k \tag{4.428}$$

  Note: x and μ_k are considered column vector in this equation.

- Inference: the parameters are estimated as: $\hat{\pi_k} = N_k/N$, $\hat{\mu_k}$ as sample mean of class k, and $\hat{\Sigma}$ as the pooled estimator of Σ using all classes.

- Quadratic discriminant function (QDA): not assume equal variance among classes. The discriminant is a quadratic function of x.

- Compromise bewteen LDA and QDA: this is similar to complete pool vs. no pooling in multi-level problems. A regularized covariance matrix may have the form:

$$\hat{\Sigma_k}(\alpha) = \alpha \hat{\Sigma_k} + (1 - \alpha)\hat{\Sigma} \tag{4.429}$$

  where $\hat{\Sigma}$ is te pooled covariance matrix.

Fisher's LDA (reduced-rank LDA):

- Motivation: the class prediction is entirely determined by the (probability) distance to the class centroids. With K classes, the centroids lie in at most $K - 1$ dim. hyperplane, thus, the distance comparison can be performed in this hyperplane. This dimensionality-reduction can be achieved by Fisher's LDA.

- Fisher's problem: start with the two-class case, the intuition that the points in the positive class are closer to μ_+ than to μ_- is the same as: most positive examples will occur in one side of the decision boundary, and negative examples in the other side. Consider the direction perpendicular to the decision bounday, then the projection of points in this direction have maximum spread between two classes, relative to the within-class spread. This can be formulated as: find the direction a that maximizes the ratio of between-class variation to the within-class variation:

$$\max_{a, \|a\|=1} \frac{a^T B a}{a^T W a} \tag{4.430}$$

  where B is the between-class covariance matrix, defined as the covariance of the class centroids, and W is the within-class covariance matrix, defined as (in the two class case) $(W_+ + W_-)$, the sum of the covariance matrix of positive and negative points, respectively. The variance calculation follows Equation 3.59.

- Solution: The problem is solved by the generalized Rayleigh quotient, and the solution is the eigenvector of the maximum eigenvalue of $W^{-1}B$. All the eigenvectors define a vector subspace containing the variability between features (the smaller eigenvalues can be ignored), and the projections are called discriminant (or canonical) coordinates. Thus the distance comparison/classification of data points can be entirely performed in terms of discriminant coordinates (if only the top eigenvalues considered, then reduced dimension).

4.12.2 Generalized additive models and structural regression

Motivations:

- The linear models can be generalized in different ways through, e.g.:
  - More general basis functions (but still additive);
  - Form of the basis functions: allow higher-order terms, such as multiplication;
  - General non-linear functions.
- Example: predict some property of a sequence, the basis functions could be word counts, or some basic properties of the sequence.

Generalized additive model:

- Idea: why a non-linear model can be learned from the data? When all other variables are given, the relationship $Y \sim X_j$ is 1D, and can be approximated through, e.g. splines.
- Model: it has the form:

$$E(Y|X_1, X_2, \cdots, X_p) = \alpha + f_1(X_1) + f_2(X_2) + \cdots + f_p(X_p) \tag{4.431}$$

  More generally, the LHS could be $g(\mu(X_1, X_2, \cdots, X_p))$, where g is a link function, such as logit function.

- Fitting additive models: the objective function is minimize error, with penality of non-smooth functions:

$$\text{PRSS}(\alpha, f_1, \cdots, f_p) = \sum_{i=1}^{N} \left(y_i - \alpha - \sum_{j=1}^{p} f_j(x_{ij}) \right)^2 + \sum_{j=1}^{p} \lambda_j \int f_j''(t_j)^2 dt_j \tag{4.432}$$

  where the second term penalizes non-smooth functions. Fitting can be achieved in an iterative scheme: suppose all other $\hat{f}_k, k \neq j$ are known, then $\hat{f}_j$ can be estimated through a 1D nonlinear fitting (e.g. cubic spline) of the j-th feature and the residual terms (after removing all other variables):

$$x_{ij} \sim y_i - \hat{\alpha} - \sum_{k \neq j} \hat{f}_k(x_{ik}), 1 \leq i \leq N \tag{4.433}$$

- **Remark**: in the problem of learning the effect of one predictor, need to control the other predictors. This can be achieved through regression of the predictor of interest and the residual terms (after applying other predictors). And this can be applied in an iterative fashion. The general idea can be applied in any regression type problems, not just linear models.

Structured regression functions: [Hastie, Section 6.4]

- A general form of the structured regression:

$$f(X_1, X_2, \cdots, X_p) = \alpha + \sum_j g_j(X_j) + \sum_{k<l} g_{kl}(X_k, X_l) + \cdots \tag{4.434}$$

  The additive model assume only the main effect terms (i.e. no higher-order terms).

- Varying coefficient model: a special form of structured regression. It generalizes the regression model: the parameters/coefficients are not constant, instead, they depend on the values of (at least some) variables. Let $X_1, X_2, \cdots, X_q$ be the main predictors, and Z be the variables that may influence the effect of these main predictors. We could write regression as:

$$f(X) = \alpha(Z) + \beta_1(Z)X_1 + \cdots \beta_q(Z)X_q \tag{4.435}$$

Chapter 5

Probabilistic Graphical Model and Causal Inference

5.1 Overview of Causal Inference

Experimental and observational studies [Pearle, Causality, 2000]: consider an example where we want to test if smoking (S) has an causal influence on disease (D), e.g. lung cancer, and want to estimate the causal effect. We have three possible experimental stategies:

- Experimental study with control: treatment (smoking) on the same or identical individuals (e.g. twins). Then any difference in the response variable must be due to the treatment.
- Experimental study with randomized control: treatment on two groups - case and control, where any confounding variables have been completely randomized in the two groups. Then the confounding variables do not statistically bias case or control group, and any difference is due to treatment.
- Observational study: the association between explanatory variable and response variable cannot be taken as causation. Ex. association bewteen smoking and disease can be explained by:
  - Disease $\rightarrow$ Smoking (disease affects the smoking habit).
  - Genotype $\rightarrow$ Smoking and Genotype $\rightarrow$ Disease, then the same genotype that causes one person to smoking may also predispose the person to the disease. In other words, the sample of individuals with smoking is not completely random: they are biased towards certain genotypes.

  In general, failure to account for confounding variables would lead to false causality discoveries; and if the regression corrects for variables in the causal path from exposure to outcome (*mediator*), it may cause over-adjustment.

Regression from the graphical model perspective [personal notes]:

- Theorem: given variables X, Y, Z, if
$$X \leftarrow Y \rightarrow Z \tag{5.1}$$
then X and Z are dependent, but $X \perp Z|Y$. If we have the collider case:
$$X \rightarrow Y \leftarrow Z \tag{5.2}$$
Then $X \perp Z$, but X and Z are dependent given Y. Intuitively, given Y, then large X may mean smaller Z.

- Motivation: suppose we want to estimate the effect of X on Y via regression. But we have other covariates that may be associated with Y, denoted as Z. Shall we include Z? What happens if we do not include Z (e.g. when Z is actually hidden)? Consider a linear model $y = \beta x + \gamma z$. Ex. x is the genotype, y is phenotype, and z is a possible confounding variable.

- Z not associated with X but associated with Y (**Covariate**): e.g. z is an environmental exposure (e.g. air pollution) in the genetics example, but independent of genotype. Incorporating Z will regress out some variance of Y, thus we should include Z to increase the power of detecting X effect. In other words, once Z is included, the total variance of Y (after adjusting for Z) is smaller, and the proportion explained by X would be bigger. If Z is not adjusted, it will not create false positives - only losing power.

  - Even if Z does not has a direct effect on Y, Z may be a marker of some hidden variables that affect Y.

  Analysis: when z is not included, our estimated β is:

$$\hat{\beta} = \frac{\hat{\text{Cov}}(x, y)}{\hat{\text{Var}}(x)} \approx \frac{\text{Cov}(x, \beta x + \gamma z)}{\text{Var}(x)} = \frac{\beta^2 \text{Var}(x)}{\text{Var}(x)} = \beta^2 \tag{5.3}$$

  We could replace the approximation with expectation in the above argument to be rigorous. Therefore, the estimator is still about right (unbiased). However, it will have a bigger confidence interval:

$$\text{Var}(\hat{\beta}) = \frac{\sum_i e_i^2/(n-2)}{\text{Var}(x)} > \frac{\sum_i \sigma_i^2/(n-2)}{\text{Var}(x)} \tag{5.4}$$

  where e_i is the residual when only x is included, and σ_i is the true redisual. Obviously we have on average $e_i > \sigma_i$ since e_i now is the sum of σ_i and γz_i. The intuition is that: we will have to explain the missing z through larger error terms, leading to lower power.

- **Confounding**: if Z has an effect on X, and also independent effect on Y, conditioned on X,

$$X \leftarrow Z \rightarrow Y \tag{5.5}$$

  we should adjust for Z, otherwise false positive association between X and Y. Ex. z represents population ancestry (which may be associated with disease through culture, history, sampling bias, etc), then it can create false association.

- Covariates affected by X: We have two cases: (1) If X has a true effect on Y,

$$Y \leftarrow X \rightarrow Z \qquad Z \sim Y \tag{5.6}$$

  Then adjusting for Z will eliminate some signal of X, and is undesirable. This is **Mediation**. Ex. z represents smoking habit, but is influenced by genetics. (2) In the second case, X has no effect on Y, but Z is associated with Y, then it may create false positive association. We can call this **Confounding via Mediation**. Ex. suppose a hidden variable U affects both Z and Y:

$$Y \leftarrow U \rightarrow Z \leftarrow X \tag{5.7}$$

  Then because U and X are now "colliders", and they become dependent conditioned on Z. So adjusting for Z will create correlation of U and Z, and U would be a confounder. Ex. Y is risk of kidney disease and Z is obesity, U diet (sugar content), then a SNP of obesity may be falsely associated with kidney disease because of confounding variable diet. The optimal strategy in this case may be: (1) Adjusting for X in Z, i.e. define $Z' = Z - X\delta$, and then adjusting for Z' when regressing Y on X; (2) Joint analysis of Y and Z.

- Application to genetic association: suppose X is SNP, Y is expression of a gene tested and Z is another gene or PC from expression. Z cannot be a confounder now (Z cannot affect X). It is possible that X affects Z (hertiable PC, or SNP is eQTL of Z). Generally, we will not adjust for Z, or we should adjust for only the part of Z that is independent of X.
- Multivariate regression (association): suppose we consider X on Y_1 and Y_2, and Y_1, Y_2 are correlated (independent of X). Then testing regression of Y_1 and Y_2 on X independently would not create false positive associations.
- Reference: [Stephens, PLoS ONE, 2013], [Aschard, Musical Chairs paper, 2016]

Unobserved confounding variable in multi-trait association [personal notes]

- Problem: suppose we have a QTL Q of a gene G_1. Suppose G_1 highly correlates with another gene G_2. Could it be possible that we'll find Q is also a QTL of G_2 because of the correlation?
- Three cases: first, the correlation between G_1 and G_2 is created by an unobeserved variable Z:

$$Q \to G_1 \quad G_1 \leftarrow Z \to G_2 \tag{5.8}$$

  It is easy to see that $Q \perp G_2$ even if we do not control for Z. Essentially, G_2 only depends on Z, which is independent of Q. In the second case,

$$Q \to G_1 \quad G_1 \to Z \to G_2 \tag{5.9}$$

  It is clear that G_2 would depend on Q through Z. In the third case,

$$Q \to G_1 \quad G_1 \to Z \leftarrow G_2 \tag{5.10}$$

  Then Q would also depend on G_2; but if we control for Z, they will be independent.
- Summary: whether the unobserved confounding variable creates problem depends on the nature of the correlation between G_1 and G_2.

Mediation analysis [personal notes]:

- Mediation analysis [Wiki]: we are testing if a variable M mediates the effect of X on Y: $X \to M \to Y$. The analysis has three steps:
  1. Establish that $X \to Y$ by regression of Y on X.
  2. Establish that $X \to M$ by regression of M on X.
  3. Mediation: $Y = \beta_0 + \beta_1 X + \beta_2 M + \epsilon$, test if $\beta_2 \neq 0$ and β_1 should be smaller than the coefficient in step (1). Mediator should have some independent effect on Y (β_2 significant), and after including M, the effect of $X \to Y$ should be reduced.

  All three steps are needed. Ex. if we have only step 3, the true model could be $X \to Y \leftarrow M$. Application to genetic association: X is SNP, M and Y are two traits. Then conditions 1 and 2 mean that X needs to be QTL of both M and Y.
- Sobel test [Wiki]: we test the reduction of $X \to Y$ effect after controlling for the mediator. Suppose we estimate total effect first:

$$X \xrightarrow{\tau} Y \tag{5.11}$$

  Next we estimate the mediated effect and the direct effect:

$$X \xrightarrow{\alpha} M \xrightarrow{\beta} Y \qquad X \xrightarrow{\tau'} Y \tag{5.12}$$

where the second part is the effect of X on Y not through M. Write this as:

$$M = X\alpha + \epsilon_M \qquad Y = X\tau' + M\beta + \epsilon_Y \tag{5.13}$$

We can see now how Y depends on X by plugging in M equation:

$$Y = X\tau' + (X\alpha + \epsilon_M)\beta + \epsilon_Y = X(\tau' + \alpha\beta) + (\beta\epsilon_M + \epsilon_Y) \tag{5.14}$$

We test $\tau - \tau'$, but note that $\tau - \tau' = \alpha\beta$. This says: τ has two parts, mediated effects $\alpha\beta$ and direct effect τ'. So the test statistics is $\alpha\beta$, and the SE is $\sqrt{\alpha\sigma_\beta^2 + \beta\sigma_\alpha^2}$. Roughly, the test statistics of α is the regression of M against X, and β is the regression of Y against M when conditioned on X.

- Does Mediation imply causality? For simplicity, assume steps 1 and 2 are causal (e.g in the case of genetics when X is SNP). Consider a simple model of independent effect:

$$X \to M, X \to Y \tag{5.15}$$

Then conditioning on M is as if we condition on part of X, so $Y \sim X$ would have reduced effect. The same can be said when there is a confounder U acting on M and Y:

$$X \to M \qquad M \leftarrow U \to Y \qquad X \to Y \tag{5.16}$$

So in summary, we could have mediation without any causal effect.

- Does Mediation tell the direction of causality? Suppose there is a causal effect, of $M \to Y$ or $Y \to M$. We can show that this model where Y affects M also show significant mediation of M on Y:

$$X \to Y \to M \tag{5.17}$$

It's easy to verify that $M \sim X$ and $Y \sim X$. Additionally, conditioned on M would remove some of the effect of X on Y, so $Y \sim X + M$ would have reduced effects. The results would hold if there is U acting on M and Y.

- Interpretation of β in Mediation: given the analysis above, we can think of β as total association of M with Y, through the causal effect and the confounder-induced correlation. With this interpretation: $\tau - \tau' = \alpha\beta$ is always true even when there is confounder.

Lessons and questions for IV approach [personal notes]

- How to combine information from multiple (weak) IVs of the same exposure?
- A variable may not be a valid IV for X, e.g. it affecting another variable X', which could have a causal effect on Y. But can we include X' as well to infer the causal effect of X?

5.2 Graphical Models

5.2.1 Directed Graphical Models

Reference: [Pearl, Causality, 2000;], [Bishop, Pattern Recognition and Machine Learning, Chapter 10], [Murphy, 2012, Chapter 8]

Modeling with Bayesian networks (BN):

- Constraints of graph structure: this is often important to limit the model complexity. In addition to some appropriate priors, the common ideas are: mixture models or hierarchical models that group variables. Examples:

  - Bi-clustering of gene expression data: the genes and the conditions are clustered into groups, and the expression in the same gene-condition group follows the same distribution.
  - Module networks: members of the same modules share the parents.

  Sometimes, we can introduce additional variables to impose some structure. Ex. if we know X affects Y_j's, and Y_js' are correlated, but we don't know their relationship. We can introduce U as a latent factor, and $X \to U$ and $U \to Y_j$.

- Application of BN in prediction problems: when the features X_j's have certain known structure, it may be best to model X_j's as well as Y within a BN, treating Y as just another variable while modeling the dependence of features. Examples:
  - Prediction of gene interaction: the features are co-expression, PPIs and genetic interaction. However for each type of data, there may be multiple features, which should be highly correlated. Inference could be done through a hierarchical BN [Troyanskaya & Botstein, PNAS, 2003].
  - Prediction of interacting AA of a protein: the features are AA conservation, hydrophobicity, etc. A naive Bayesian model to infer AA state (interaction or not) [Needham & Westhead, PLCB, 2007].
- Directionality in modeling: this may not be obvious (corresponding to causal relations), and the BN may allow both directions. This is true, for instance, when some variables are introduced to model variable grouping (as in mixture or hierarchical models), thus having no or ambiguous physical meaning. Example: features of gene promoters (X) and gene cluster assignment (Y). $X \to Y$: the promoter determines which cluster a gene belongs to; $Y \to X$: which cluster a gene belongs to determines the characteristics of promoters (evolutionary consequence).
- Interventional vs. observational data: both can be used for learning causal models. However, the interventional data will generally be more informative than observational data in constructing causal networks. This should be taken into account when learning the causal model. Ref: [Sachs & Nolan, Science, 2005].

Conditional independence (CI) [Wiki]:

- Motivation: capture the structure of data (Occum's razor). Ex. the future does not depend on the past given the future, this is modeled by a Markov chain.
- Intuition: the relation, $X \perp Y|Z$, can be understood from different perspective. Take an example of working in a chemical factory (X) and cancer (Y), where Z indicates chemical exposure.
  - Stratification: given Z, i.e. if the data are stratified by Z, then X and Y are independent. This means that for people with the same chemical exposure, then whether working for the chemical factory is not related to cancer risk.
  - Explanation of correlation: the variables X and Y are correlated, but the correlation is caused by their common relationship with Z; if we take Z out, the correlation between X and Y will be explained away. In our example, the correlation of working status and cancer risk is explained away by the chemical exposure.
- Properties of CI: we ignore the condition for simplicity of notations.
  - Symmetry: $X \perp\!\!\!\perp Y \Leftrightarrow Y \perp\!\!\!\perp X$.
  - Decomposition: if $X \perp\!\!\!\perp A, B$, then $X \perp\!\!\!\perp A$ and $X \perp\!\!\!\perp B$.
  - Contraction: if $X \perp\!\!\!\perp A|B$, and $X \perp\!\!\!\perp B$, then $X \perp\!\!\!\perp A, B$. The intuition is if $X \perp\!\!\!\perp A|B$, but B is independent of X, thus cannot explain the relation between X and A, thus we must have X independent of A.

– Intersection: if the probabilities of X, A and B are all positive, then $X \perp\!\!\!\perp A|B$ and $X \perp\!\!\!\perp B|A$ implies that $X \perp\!\!\!\perp A, B$.

Markov model:

- Definition: a probability distribution defined on a directed acyclic graph (DAG) is Markovian if:

$$P(v) = \prod_i P(x_i|\pi(x_i)) \tag{5.18}$$

  where x_i is a node and $\pi(x_i)$ is the parent of x_i.

- Markov condition: the factorization and the joint condition is equivalent to the following Markov condition: for every variable W,

$$W \perp \tilde{W}|\pi(W) \tag{5.19}$$

  where $\tilde{W}$ denotes all the other variables except the parents and descendants of W (it may include co-parents of W).
  Example: $X \to Y \to Z$, then it is easy to prove that $p(x, z|y) = p(x|y)p(z|y)$ from the factorization of $p(x, y, z)$.

- Graphical notation for representing models: (1) multiple independent RVs: by plate; (2) deterministic parameters: small solid circles; (3) hidden and observed variables: different colors or open/solid state.

- Causal interpretation of a graphical model: a model only represents a factorization of joint probability distribution, thus may not correspond to causality at all. Ex. the model $X \to Y \to Z$ can also be written as: $Z \to Y \to X \leftarrow Z$. Therefore, it is the interpretation and the attempt at modeling that gives a model the causal semantics.

Directed Gaussian graphical model (Directed GGM):

- Model: conditional distribution of a node, given its parents, follow the linear model. Suppose t is a node, we have:

$$x_t - \mu_t = \sum_{s \in \pi(t)} w_{ts}(x_s - \mu_s) + \sigma_t \cdot z_t \tag{5.20}$$

  where w_{ts} is the weight coefficient of $s \to t$ edge, and $z_t \sim N(0, 1)$ is the error term. We could write the relation in matrix form:

$$x - \mu = W(x - \mu) + Sz \tag{5.21}$$

  where S is the diagonal matrix with terms σ_t. We define $e = Sz$ as the error term for all variables.

- Joint distribution: we write the above relation as:

$$x - \mu = (I - W)^{-1}e = USz \tag{5.22}$$

  where $U = (I - W)^{-1}$. Thus the joint distribution is normal, with mean equal to μ, and the covariance matrix:

$$\Sigma = \mathrm{Cov}\,(x - \mu) = \mathrm{Cov}\,(USz) = US\mathrm{Cov}\,(z)SU^T = US^2U^T \tag{5.23}$$

Conditional independence in Markov model:

- The semantics of a Markov model lies in the conditional independence (CI): it is the lack of arrows (conditional independence) that makes a certain model specific. Ex. for the model $X \to Y \to Z$, we can factorize the joint distribution in different ways, e.g. $P(Z)P(Y|Z)P(X|Y, Z)$, but then X will depend on both Y and Z, no CI.

- Simple examples of d-separation and d-connection (not d-separation):

- Non-collider: including chains and forks. E.g. $X \to Y \to Z$, or $X \leftarrow Y \to Z$, X and Z are d-connected, but d-separated given Y.
- Collider: e.g. $X \to Y \leftarrow Z$, X and Z are d-separated, but d-connected given Y. Also note that for collider, $X \perp Z$, and this independence is true if conditioned on, e.g. predecessor of X or Z. Ex. $X \to Y \to Z \leftarrow W$, then we have $Y \perp W|X$.
- Collider descendant: in $X \to Y \leftarrow Z$, also $Y \to W$, the same relation holds: X and Z are d-separated, but d-connected given Y or W. Thus a descendant of a collider has the same role in determining CI.

- Rules of d-separation and d-connection: X and Y are d-separated by a set of nodes S if S blocks all paths from X to Y (path is undirected). S blocks a path p if either of the two conditions satisfy:
  - p contains a chain ($i \to m \to j$) or a fork ($i \leftarrow m \to j$) s.t. the middle node m is in S.
  - p contains a collider ($i \to m \leftarrow j$) s.t. the middle node m as well as its descendant is not in S (they will make i and j d-connected). Note that a node Z may block a path between X and Y, even if Z lies outside the path, due to the fact that colliders (alone, without conditioning on the middle node) create d-separation (thus conditioning on some variable outside the path is equivalent to not conditioning).

  Example: $X \to W \leftarrow V \to Y \leftarrow Z$. X and Y are d-separated by V (from rule 1) and Z (from rule 2), but not by W.

- Theorem: let A, B and C be disjoint sets of vertices, then $A \perp B|C$ iff A and B are d-separated by C.

- Remark: this is closely related to backdoor criterion for adjusting in confounding. If we adjust all variables s.t. X and Y are conditionally independent (ignoring $X \to Y$ edge), then regression of Y on X estimates the causal effect.

- Markov blanket: for a given variable x_i, its Markov blanket is the set of variables Y, s.t.

$$P(x_i|x_{-i}) = P(x_i|Y) \tag{5.24}$$

  where x_{-i} denotes all other variables in the graph. In other words, the Markov blanket isolates a variable from the rest of graph. It comprises the set of parents, children and co-parents (other parents of its children) of the node:
  - Children need to be included: parent of a variable X does not block the path from X to its children.
  - Co-parents: a co-parent and X are d-connected because the common child is a collider. So if we include the child, we must include its other parent(s).

  The probability $P(x_i|x_{-i})$ is called the *full conditional*.

Markov equivalence:

- Definition: two graphs with the same set of CI relations.

- Condition of Markov equivalence: Two DAGs G_1 and G_2 are Markov equivalent iff skeleton(G_1) = skeleton(G_2) (skeleton: undirected graph by replacing all arrows with undirected edges) and G_1 and G_2 have the same set of v-structures, that is, two colliders whose tails are not connected by an arrow. Thus the structure of a graphical model can be uniquely determined only up to Markov equivalent class.

- To assess Markov Equivalence, we only need to see if collider structure are identical. Example: consider the standard MR model, with $G \to X \to Y$ and U is a confounder. Consider the model where U is a mediator of X to Y. In the first case, X is a collider, but in the second, X is not.

- Directionality of edge: can be identified only if changing the direction will change the collider structure. Example: in the model $X \to Y \to Z$, the direction of $Y \to Z$ can be determined, but not the first edge.

Inference of graphical models [Murphy, Chapter 20]

- Inference vs. learning: inference is about estimating hidden variables.
- Learning (parameter estimation) from complete data [Murphy, 10.4]: MLE or MAP. The likelihood can be factorized into distributions of a node and its parents. With discrete RV for each node (multinomial distribution), the parameters can be analytically determined: posterior still independent, following Dirichlet distribution.
- Forward-backward algorithm for HMM: as message passing algorithm. $P(z_t|x[1..T])$, two parts, belief of z_t using past data up to t, and conditional likelihood of future $x[t+1..T]$, written as $P(x[t+1..T]|z_t)$. Forward pass: update the believe from left to right; backward pass: using likelihood information from right to left.
- Belief propagation (BP) in tree: from root to leaf (forward) and leaf to root (backward).
- Variable elimination (VE) algorithm: pushing sums into products. A version is Peeling algorithm in genealogy trees.

5.2.2 Tree Model

Reference: [Ronen, Parameter Estimation of Dependence Tree Models Using the EM Algorithm, 1995; Kazemian & Sinha, Quantitative analysis of the Drosophila segmentation regulatory network using pattern generating potentials, PLos Bio, 2010]

Tree model:

- Likelihood: given n RV's related by a tree, $\{X_1, \cdots, X_n\}$, where X_1 is the root node. Let t_i be the length of the branch leading to X_i. Also assume that X_1 has a uniform distribution (prior). For a node i, we denote $\pi(i)$ as the parent of i, and $C(i)$ as the set of child nodes of i. The complete likelihood of the model is defined by:
$$P(X) = P(X_1) \prod_{i=2}^{n} P(X_i|X_{\pi(i)}) \tag{5.25}$$
- Model with missing variables: assume that only the variables in the leaf nodes are observed (e.g. in phylogenetic analysis). We use O_i to denote the observations at the subtree rooted at i, thus O_1 is our data. We are interested in two problems: (1) parameter estimation; and (2) estimation of the missing variables (internal nodes). For the former, let θ be model parameters, we compute the likelihood $P(O_1|\theta)$; and for the latter, we need to compute $P(X_i = m|O_1)$.

Upward/downward algorithm:

- Recurrence variables for the likelihood and missing variables: we have the decomposition for the likelihood:
$$L(\theta) = P(O_1|\theta) = \sum_m P(X_1 = m|\theta) \cdot P(O_1|X_1 = m, \theta) \tag{5.26}$$
We also have the decomposition for the latent varialbes:
$$P(X_i = m|O_1) = \frac{P(X_i = m, O_1)}{P(O_1)} = \frac{P(O_i|X_i = m)P(X_i = m, O_{1\setminus i})}{P(O_1)} \tag{5.27}$$

where we use the notation $O_{i\setminus j}$ to denote the leaf nodes that are part of the subtree rooted at i, but not part of the subtree rooted at j. The two equations suggest that we need the following two types of variables:

$$\beta_i(m) = P(O_i|X_i = m) \tag{5.28}$$

$$\alpha_i(m) = P(X_i = m, O_{1\setminus i}) \tag{5.29}$$

Using these recurrence variables, we have: $L(\theta) = \sum_i \beta_1(m)$ and

$$P(X_i = m|O_1) = \frac{\alpha_i(m)\beta_i(m)}{\sum_{m'} \alpha_i(m')\beta_i(m')} \tag{5.30}$$

We could compute the two types of variables using the upward-downward algorithm.

- Upward algorithm: this computes the upward variable $\beta_i(m)$. If i is the leaf node, we have $\beta_i(m) = \delta_{m,x_i}$ (delta-function). If i is an internal node, $\beta_i(m)$ can be decomposed using the child nodes of i:

$$\beta_i(m) = \prod_{j\in C(i)} P(O_j|X_i = m) \tag{5.31}$$

We define $\gamma_i(m) = P(O_i|X_{\pi(i)} = m)$, thus $\beta_i(m)$ is a product of $\gamma_j(m)$ over the child nodes of i. We next consider the recurence of $\gamma_i(m)$, if i is a leaf node:

$$\gamma_i(m) = P(m \to x_i|t_i)\beta_i(x_i) \tag{5.32}$$

If i is an internal node, we have the recurrence:

$$\gamma_i(m) = \sum_{m'} P(m \to m'|t_i)\beta_i(m') \tag{5.33}$$

To implement the algorithm, we first compute the recurrence at the leaf nodes, then move up the tree.

- Downward algorithm: this computes the downward variable $\alpha_i(m)$. If i is the root node, we have $\alpha_1(m) = P(X_1 = m)$, this is given by the prior distribution. If i is an internal node, we have the recurrence:

$$\alpha_i(m) = \sum_{m'} \alpha_{\pi(i)}(m')P(m' \to m|t_i)P(O_{\pi(i)\setminus i}|X_{\pi(i)} = m') \tag{5.34}$$

For a binary tree, we have $O_{\pi(i)\setminus i} = O_{\mathrm{Sib}(i)}$, where $\mathrm{Sib}(i)$ is the sibling of i. We can then write the recurrence:

$$\alpha_i(m) = \sum_{m'} \alpha_{\pi(i)}(m')P(m' \to m|t_i)\gamma_{\mathrm{Sib}(i)}(m') \tag{5.35}$$

A Spectral Algorithm for Latent Tree Graphical Models [ICML, 2011]

- Motivation: in the graphical models, if there are latent variables, need to do EM to do parameter estimation, which is local optima.
- Idea: do transformation s.t. the inference is based on transformed variables on observable data only. Suppose O is the observed vector, R is the root (hidden), then $P(O)$ (a vector) can be written as a product of $P(O|R)$ (a matrix) and $P(R)$ (a vector). In general, this is the message passing equation.

5.2.3 Markov Random Fields (MRF)

Reference: [Hastie, Chapter 17; Bishop, Section 8.3]

MRF idea:

- Motivation: need a flexible way of modeling probability distribution of multiple random variables with certain dependencies. Ex. we may know that two RVs are correlated, but not the exact order of causality of all RVs (thus directed graphical models are not applicable).
- A general strategy of modeling probability distribution: simply define the energy of a system of multiple RVs, and the probability distribution follows Boltzmann distribution. How the energy of the system is defined encodes our knowledge/belief of the likely states of the system.
- Examples: (1) Spatial data: where the variable at a grid should be similar to the value of its adjacent grids. Model this as an energy function that rewards neighboring interaction/correlation. (2) Network data: where two linked nodes tend to be similar (e.g. social network, PPI network). Model this as an energy function that rewards correlated network neighbors.

MRF model:

- Motivation: Ising model. Consider a lattice, and each site in the lattice has a spin state (+1 or −1), σ is the assignment of spin states of all sites. The energy of the system is given by:

$$H(\sigma) = -\Sigma_{i,j} J_{ij}\sigma_i\sigma_j - \Sigma_j h_j\sigma_j \tag{5.36}$$

  where the first sum is over pairs of adjacent spins (every pair is counted once) and h_j is the external field at j. The probability of σ is then given by the Boltzmann distribution. The interesting statistical questions to ask are all in the limit of large numbers of spins:

  - In a typical configuration, are most of the spins +1 or −1, or are they split equally?
  - If a spin at any given position i is 1, what is the probability that the spin at position j is also 1?
  - If β (temperature) is changed, is there a phase transition?

- Probability distribution: suppose a potential function is defined for every clique of the graph (some type of mutual interaction), then the joint distribution:

$$f(x) = \frac{1}{Z}\prod_C \Psi(x_C) \tag{5.37}$$

  where C is over all maximum cliques of the graph, x_C is the RVs at C and $\Psi(x_C)$ is the potential function of x_C. Z is the partition function, defined as:

$$Z = \sum_x \prod_C \Psi(x_C) \tag{5.38}$$

  It is common to parameterize $\Psi(x_C)$ with an energy term, $E(x_C)$, using Boltzmann distribution:

$$\Psi(x_C) = \exp[-E(x_C)] \tag{5.39}$$

- Conditional independence: three sets of nodes A, B, C, we have: if C separates A and B, then $A \perp B|C$. CI allows one to decompose a graph into maximal cliques, thus the factorization above can also be expressed as a set of CI statements.
- Pairwse Markov graph: a potential function for each edge, and at most second-order (pairs) interactions are represented. This has the benefit of fewer parameters and easier to work with. Ex. for a three-node complete graph, we simply have:

$$f(x, y, z) = \frac{1}{Z}\Psi(x, y)\Psi(y, z)\Psi(x, z) \tag{5.40}$$

Continuous MRF (Gaussian MRF): multivariate normal distribution

- Idea: suppose the set of RVs can be treated as multivariate normal distribution, then an associated pairwise Markov graph encodes the additional constraints of the covariance structure of these RVs. This would allow one to better estimate this distribution if the graph is known; or learn a simpler model if the graph is not known.
- Background: let $\Theta = \Sigma^{-1}$ be the precision matrix, if the ij component of Θ is 0, then i and j are CI given the other variables.
- Learning the covariance matrix given the associated pairwise Makrov graph. Suppose the potential function of the state x is given by:

$$\Psi(x) = \sum_i \theta_{ii} x_i^2 + \sum_{i \neq j} \theta_{ij} x_i x_j = x^T \Theta x \tag{5.41}$$

  So the pdf is given by: $P(x) \propto \exp(-x^T \Theta x)$. When x is centered, we see that this is just the pdf of MVN, with $\Theta = \Sigma^{-1}/2$. The log-likelihood function is:

$$l(\Theta) = \log\det\Theta - \mathrm{tr}(S\Theta) \tag{5.42}$$

  where S is the sample covariance matrix. The maximization is subject to the equality constraint that $\Theta_{ij} = 0$ if i and j are not linked in the Markov graph. It can be shown that the parameter estimation can be done through a series of regression of the variable i on j, conditioned
- Learning the Markov graph: when the graph is unknown, then effectively we want to learn a model with possibly as few edges as possible. This can be done by maximizing the penalized log-likelihood:

$$\log\det\Theta - \mathrm{tr}(S\Theta) - \lambda\|\Theta\|_1 \tag{5.43}$$

  where $\|\Theta\|_1$ is the L_1 norm. One can adapt the lasso to solve this problem.

Graphical lasso: [Sparse inverse covariance estimation with the graphical lasso]

- Motivation: to estimate a multivarate normal distribution, the correlation structure should be relatively sparse, specifically, most variables should be conditionally independent (given all other variables). This translates to: the number of non-zero terms in the precision matrix should be small.
- Model: let $\Theta = \Sigma^{-1}$ be the precision matrix (non-negative definite), and S be the sample covariance matrix, our goal is to maximize the penalized log-likelihood:

$$\log\det\Theta - \mathrm{tr}\left(S\Theta\right) - \rho\|\Theta\|_1 \tag{5.44}$$

  where $\|\Theta\|_1$ is the L_1 norm of the matrix: the sum of the absolute values of the elements of Θ.
- Optimization: block coordinate descent algorithm. One can show that to solve the conditional maximization problem is equivalent to the dual problem, which resembles a lasso regression (least square with L_1 constraint).
- The issues to be solved/proved:
  - The interpretation of the elements of Σ^{-1}: conditional independence of variables.
  - Convexity of the optimization problem: prove the convexity in any line restriction.
  - Optimization: block coordinate descent and the dual problem.
- Remark: penalized log-likelihood method, where the penalty is based on the non-zero coefficients.

Disrete MRF: most common when the variables are binary. Also called Ising model, Boltzmann machine.

- Potential function: each edge of the variables X_j and X_k has the energy, $-\theta_{jk}X_jX_k$, which means favorable interaction when $X_j = X_k$. This leads to the probability of a state X:

$$p(X) = \frac{1}{Z}\exp\left[\sum_{(j,k)\in E}\theta_{jk}X_jX_k\right] \tag{5.45}$$

- Application: Image de-noising. Let x_i be the value of the i-th grid in the true image (hidden variables), and y_i be the corresponding variable in the noisy image (data). Then the solution x should satisfy that: x generally has similar values in the neighboring grids; and x_i should be generally equal to y_i. Also we could favor x_i to certain class (e.g. more $+1$ over -1). This can be expressed as the energy function:

$$E(x, y) = h\sum_i x_i - \beta\sum_{i,j}x_ix_j - \eta\sum_i x_iy_i \tag{5.46}$$

  where i and j are neighboring grids. The goal is to find/sample x_i according to the distribution specified by this energy function ((effectively minimize the energy).

Remark: The alignment problem can serve as a general strategy for capturing the similarity/matching structure in a problem, and this can be used for finding the hidden variables (see examples below).

- The alignment problem is often charcterized by: the missing variables/alignment should (1) the counterparts of different objects should match each other; (2) the missing variables should be consistent with their "neighbors" within each object.

- In particular, MRF is a general modeling framework for alignment problems; it is more general than directed models, e.g. for sequence alignment, one may have the situation where some sequences are similar but no direction (temporal) is known (example below).

Some examples of MRF:

- Finding missing sequences: suppose we have multiple sequences from a group, each with part of sequence missing/error. Suppose the sequences are related to each other, e.g. some are from the same family, some from the same region, and we have a matrix characterizing mutual relationship of sequences. Then we have a generative model of each sequence (such as HMM), but also considering the similarity between sequences.

- Markov random topic models [Daume, ACL, 2009]: suppose documents are related to each other through a graph G, the topic proportions of documents should be consistent with the graph, i.e. the neighbors should have similar topics. Define a MRF on topic proportions θ, the potential of a pair of documents d, d' is $\psi(\theta_d, \theta_{d'}) = \exp[-l_{d,d'}\rho(\theta, \theta')]$, where $l_{d,d'}$ measures the strengh of link, and ρ is a distance function.

- Machine translation (word alignment): suppose given a sentence in L_1, we want to find the best alignment of the words in L_2. This alignment should: (1) conform to the syntax/semantics of L_2; (2) match the sentence in L_1.

Time-varying Indian Buffer Process [A Dynamic Relational Infinite Feature Model for Longitudinal Social Networks, AI-STAT, 2011]

- Idea: modeling the change of IBP over time. Define latent variables (Z) and weight matrix W that models dependence of Z. Only the latent variables change over time, while W remains constant.

Learning Scale Free Networks by Reweighted L1 regularization [AI-STAT, 2011]:

- Favoring scale-free networks: by regularizing parameters.
- Optimization: minorize-maximization (MM) algorithm, which is a generalization of EM. The idea is for a non-convex function to be max'ed, approximate with a series of convex function, which provides a lower-bound of the max at each step.

5.2.4 Graphical Model Structure Learning

Graphical model structure learning [Murphy, Chapter 26]

- Heuristic approach: relevance network with MI. Dependency network: use sparse regression for full conditional (one variable a time).
- Learning tree structure: equivalence of directed and undirected trees (26.3.1). It is easier to use undirected tree (simple factorization of probability).
- ML tree structure: Chow-Liu algorithm, the likelihood function is the score of a tree. Finding maximum weight spanning tree.
- Mixture of tree model for general graph: to apply the tree algorithm to general graph. Remark: standard MR causal diagram is not a tree.
- Learning DAG structure with complete data: one can only find graphs with Markov equivalence. The results are PDAG. For a graph with no missing variable, we marginalize parameters:

$$P(D|G) = \int P(D|G,\theta)P(\theta)d\theta \tag{5.47}$$

  This will produce simpler models than ML (which would give complete graph). The model requires hyperparameters.

- Example: college plan network. Figure 26.7, college decision depends on sex, IQ, SES, and parental encouragement (PE). Learns the graph from data: one model is much preferred than the rest with large BF.
- Learning DAG with latent variables: BIC often gives overly simplified model. A better approach is VB-EM: let z represent hiden variables

$$p(\theta, z_{1:N}|D) \approx q(\theta)\prod_i q(z_i) \tag{5.48}$$

- Example: college plan network with hidden variable - a common cause of SES and IQ. This model has much better BF.
- Discovering hidden variables: use latent variables to explain dense clusters of variables. Could also extend to hierarchical latent variable model. Example: Googles Rephil, to fit word-document model, several levels of latent variables
- SEM: all the relationships are linear, let w_{ij} be the effect of x_j on x_i, then we have:

$$x_i = \mu_i + \sum_{j\neq i} w_{ij}x_j + \epsilon_i \tag{5.49}$$

  where $\epsilon \sim N(0, \Psi)$. The model can be written in the matrix form:

$$x = Wx + \mu + \epsilon \Rightarrow x = (I - W)^{-1}(\mu + \epsilon) \tag{5.50}$$

The joint distribution of x is thus given by $x \sim N(\mu, \Sigma)$, where

$$\Sigma = (I - W)^{-1} \Psi (I - W)^{-T} \tag{5.51}$$

Note that in SEM, the graph can be cyclic.

Learning causal DAG structures [Murphy, 26.6]

- Example: Treatment $\rightarrow$ Effects, Gender as a confounder, or Blood pressure as mediator.
- Learning from observational data: up to PDAG (Markov equivalence class).
- Analysis: need assumptions of no missing data. Ex. given two nodes, $X \rightarrow Y$ and $X \leftarrow Y$ are Markov equivalent, however, the true model may be U affects both.
- Analysis: often we include hidden variables to have a DAG with known structure, and only estimate parameters. Inference of PDAG with missing data may be difficult/un-identifiable. Special case: if we can bound some effects of hidden variables, we may be able to infer causal effects. Smoking-gene example.
- Learning from interventional data: (1) Do graph surgery in interventional data. (2) Inference.

5.3 The Book of Why [Judea Pearl]

Chapter 1: The ladders of causation

- Ladder 1: association, $P(Y|X)$.
- Ladder 2: causation, $P(Y|do(X))$. Ex. what is the chance we will sell Y if we set the price of X?
- Ladder 3: counterfactual, imagined world, the reason for observed events. Ex. what is the probability that a customer who bought toothpaste would still have bought it if we had doubled the price?
- Firing squad example: causal diagram (Figure 1.4) with Court Order to Captain to two soldiers A and B to death D. Intervention: A decides to shoot, we set $A =$ true, and remove all edges pointing to A. Counterfactual: suppose we have seen D, we ask if A is responsible. We imagine the world where $A =$ false and remove all edges pointing to A. Given Court Order is true, we conclude that D will still be true, so A should not be responsible.
- Small pox vs. vaccine: after vaccination, more people died from innoculation than small pox. Can we conclude that we should ban vaccine? Counterfactual: how many people will die if we set vaccination rate to be 0 percent? Use causal diagram to answer.
- **Lesson**: to answer counterfactual questions, what would happen to Y if some condition about X had happened? We set X in the causal diagram with $do(X)$ operation, and then estimate the distribution of Y.

Chapter 2. From Buccaneers to Guinea Pigs: The Genesis of causal inference

- Regression to the mean by Francis Galton: one can predict the height of father from son, and height of son from father by the regression line. The situation is symmetric (no causal implication). Stability of the population? Answer: HWE.
- Abandoning causality by Karl Pearson: "the ultimate scientific statement of description of the relation between two things can always be thrown back upon ... a contingency table".

- Problems of Karl Pearson: some correlations are just silly, called "spurious". However, how do we know which ones are meaningful, which spurious? Discovery of Simpson's paradox: skull length and breath are uncorrelated if analyzing males and females separately; but correlated if together. Explanation: if shorter, likely come from female, thus breath smaller.

- Path Diagram of Sewall Wright: Figure 2.7, coat color in guinea pigs. D: developmental factors, E: environmental factors, G: genetic factors. Show how these factors pass through generations and influence the trait. Path diagram allows one to derive the correlation in terms of path coefficients (thus estimating unobserved path coefficients).

- Application of path diagram: Observed that one more day in the womb leads to gain of 5.66g weight. Does it mean that it grows 5.66g per day? No, because longer gestation period means the growth condition is more favorable (smaller litter size). To estimate: birth weight X depends on P gestation period and Q prenatal growth rate (unobserved). Both depends on L litter size. Derive the total correlation of P and X: which is the sum of direct effect of P to X, and the correlated induced by L (L affects both P and Q and Q affects X).

- Debate between Sewall Wright and Samuel Karlin in AJHG: about path analysis (1) Karlin: one can adopt an essentially model-free approach, seeking to understand the data interactively by using a battery of displays, indices and contrasts". (2) Wright: "There can be no such analysis without a model".

- **Lesson**: in path diagram, one can derive the correlation of variables, which sum over all "relevant" paths: any path that can induce correlations (backdoor paths other than causal paths). This includes: chain and confounding, but not collider.

Chapter 3: From evidence to causes

- Bayesian networks: **Belief propagation**. Mimic how a neural network works. Suppose we want to infer a variable X, let's say we have information of its child or parent Y, and we can update our belief of X. When Y is child: we update X by likelihood; when Y is parent: we update X by prior. Proof that belief propagation algorithm eventually converges.

- Application of Bayesian networks: (1) Genetic relationship in pedigrees. (2) Error-correcting code: need several codes to encode a single word. BN of hidden information bit, codeword and noisy code words.

- Building blocks of BNs: (1) Chain. (2) Fork: e.g. children with larger shoes tend to read at higher levels. (3) Collider: two independent variables affecting a common one (multiple causes of the same thing). Ex. beauty $\rightarrow$ celebrity $\leftarrow$ talent, then given a celebrity, usually negatively correlated.

Chapter 4: Confounding and deconfounding: or, slaying the lurking variable

- Why RCTs work? Fisher's problem: how yield may depend on various factors of interest. Challenge: many possible confounders. Experimental design can reduce but not remove all possible confounders. Solution: RCT. What it does in the causal diagram is: for X, remove all edges pointing to X, and replace it be a "random card". This eliminates all back-door paths.

- What is confounding? Anything that makes $P(Y|do(X))$ different from $P(Y|X)$. The most important case is a fork: $X \leftarrow Z \rightarrow Y$. In general, if there is any non-causal path from X to Y, then there is confounding: the variables in this path leads to confounding.

- Back-door criteria to deconfound: block back-door path from X to Y (edge pointing to X) by adjusting variables. Basic rules:

  - Control for a mediator closes the back-door path;

  - Control for a collider opens the back-door path;
  - Control for descendant of a variable is like "partially" controlling for the variable itself.

- Remark: one example, suppose we have a gene candidate of a trait (e.g. expression correlation), but the effect may be confounded by e.g. [TF], which is not observed. We can use the descendant, mRNA levels of TF target genes. In general, we may formulate this as **missing data problem**, where we adjust for some unobserved variables but using their posterior distributions.

Chapter 5: The smoke-filled debate: clearing the air

- How to prove causality of smoking to lung cancer in the absence of RCT? (1) Cornfield's Inequality: suppose we have an unobserved confounder, Smoking Gene. Show that to explain the strong association of smoking and cancer, the confounder needs to have an effect so large that is unrealistic. (2) Hill's criteria: consistency, dosage effect, temporal order, coherence with other data. None of this by itself is sufficient.

- Birth-weight paradox: smoking usually leads to low birth-weight, which increases mortality. So we expect that low birth-weight infants of smokers have higher mortality. But the data is opposite, why? Causal model: birth defect also affects birth weight, and usually have more severe effects on mortality. This creates a collider:

$$\text{Smoking} \to \text{Birth weight} \leftarrow \text{Birth defect} \tag{5.52}$$

  So conditioned on Birth weight, smoking and birth defect are anti-correlated. Thus low birth-weight infants from smokers are less likely to have birth defect, thus lower mortality.

Chapter 6. Paradoxes Galore!

- Monte Hall problem: The decision of which door to open (by the host) depends on both the door you opened and the true location of the car. So our causal diagram is: Your Door $\to$ Door opened $\leftarrow$ Car location. So Door opened is a collider.

- Berkson's Paradox: also collider bias. Even if two diseases are not associated in general population, they may be associated in hospitals. Causal diagram: Disease 1 $\to$ Hospitalization $\leftarrow$ Disease 2.

- Simpson's Paradox: a drug is associated with higher risk in males and in females, but putting all data together, it is associated with lower risk. (1) Purely numerical explanation. (2) Explanation by causal diagram: gender can influence the risk, and it also has an effect on whether one takes a drug. So our causal model:

$$\text{Drug} \to \text{Heart attack} \qquad \text{Drug} \leftarrow \text{Gender} \to \text{Heart attack} \tag{5.53}$$

  So Gender is a confounder. If we want to study the effect of drug, we should adjust for Gender (stratify). However, if our covariate is Blood pressure, then it may mediate the effect of drug, and we should NOT adjust for it.

$$\text{Drug} \to \text{Heart attack} \qquad \text{Drug} \to \text{Blood pressure} \to \text{Heart attack} \tag{5.54}$$

- Simpson's Paradox in pictures: Figure 6.6. Cholesterol (Y) vs. Exercise (X): age is a confounder. So conditioned on age, exercise reduces Cholesterol; but across all samples, positive correlation of the two.

Chapter 7. Beyond adjustment: the conquest of Mount Intervention

- Backdoor adjustment: suppose our model is $X \to Y$ with Z a confounder. Then to compute the causal effect of X on Y: (1) Regression model: $Y \sim X + Z$, with partial correlation of Y on X conditioned on Z.
  (2) Probabilities:

$$P(Y|do(X)) = \sum_z P(Y|X, Z = z)P(Z = z) \tag{5.55}$$

  Interpretation: conditioned on Z, how Y depends on X. Note that we need to average over $P(Z = z)$, not $P(Z = z|X)$.

- Frontdoor criterion: suppose we have an measured confounder U on X and Y. We cannot do backdoor adjustment, but if we can measure the mediator of X on Y, say Z, we can adjust the confounder by frontdoor criterion:

$$X \to Z \to Y \qquad X \leftarrow U \to Y \tag{5.56}$$

  One example: to study smoking on cancer, we use tar as Z. The idea is: (1) We can learn the effect of X to Z (smoking to tar) with Y being a collider; (2) We can learn the effect of Z to Y (tar to cancer) by adjusting for X (smoking). Putting the two together, we can get X to Y effect. In probabilistic terms:

$$P(Y|do(X)) = \sum_z P(Z=z|X) \sum_x P(Y|Z=z, X=x)P(X=x) \tag{5.57}$$

- Generalization: axioms of do calculus. Three rules:
  - Rule 1: if Z blocks all paths from W to Y, then

$$P(Y|do(X), Z, W) = P(Y|do(X), Z) \tag{5.58}$$

  - Rule 2: if Z blocks all backdoor paths from X to Y, then we have:

$$P(Y|do(X), Z) = P(Y|X, Z) \tag{5.59}$$

  - Rule 3: if there is no causal path from X to Y:

$$P(Y|do(X)) = P(Y) \tag{5.60}$$

  How this leads to the front door criterion (Figure 7.4).

- Algorithm and completeness theorem: three rules are enough to decide if the do-operation has a solution (in terms of conditional distributions). Also a polynomial time algorithm exists to find the solution. Extensions: when the problem is not solvable, can we find a variable Z s.t. we can solve the problem.

- IV approach: the story of Dr. Snow. Cholera epidemic. Hypothesis: caused by water purity (correlation with infection rate), however, many confounders such as poverty. IV: which water company provided the service.

$$\text{Water company} \to \text{Water purity} \to \text{Cholera} \qquad \text{Water purity} \leftarrow \text{Poverty, etc.} \to \text{Cholera} \tag{5.61}$$

  More generally, we can understand the validity of the IV approach by causal diagram:

$$G \to X \to Y \qquad X \leftarrow U \to Y \tag{5.62}$$

  The $G \to X$ effect can be learned from regression of X on G. The $G \to Y$ effect is only due to the causal path from X to Y, because the backdoor path $G \to X \leftarrow U \to Y$ has a collider X in it.

- Caution about MR: IV (SNPs) represents cumulative life-time impact, while drugs cannot mimic that effect.

- **Lesson**: in practice, if a given causal diagram does not allow one to estimate causal effects, one can choose what additional variables to measure to circumvent the problem. Ex. surrogate of some confounders, creating frontdoor conditions, etc. Most important cases: to learn $X \to Y$, we can use two strategies (1) Use IV of X. (2) Measure mediator of X to Y.

- Remark: limitation of frontdoor criterion in practice. If there are multiple mediators, and one only measure one (part) of them, the approach would not work. This is relevant in genomics, where the effect of a risk factor may be mediated via multiple paths.

Chapter 8. Counterfactuals: Mining worlds that could have been

- Difference of causal inference and counterfactuals: causal inference concerns the average effect of one on another, so we use probabilistic models. Counterfactuals concerns the cause of a specific instance, so the influences of all variables on a quantity of interest are modeled in a deterministic fashion. Ex. phenotype depends on genotype and environment: in modeling average effects, we treat environmental effects as random (errors); in modeling an individual, the environmental effect is deterministic.
- David Hume on causality: "We may define a cause to be an object followed by another, and where all the objects, similar to the first, are followed by objects similar to the second. Or, in other words, where, *if the first object has not been, the second never had existed*". The second definition is a **counterfactual**.
- General approach to counterfactuals: use Structural Causal Models (SCM). Similar to SEM, the difference is that in SCM, the relationship may not be linear.
- Potential outcomes: salary estimation problem, salary S depends on one's education ED and one's experience (number of years) EX. What would the salary of someone had he have a different education degree? Notations: what is the value of Y for individual u, had X been assigned the value x: $Y_{X=x}(u)$, or simply $Y_x(u)$. Ex.
- Imputation approach: why is this wrong? Suppose we are interested in the salary of Alice, had she had a college degree. We can match employees with Alice in every aspect, except the degree, and then estimate from those with college degree. Intuitively, ED and EX are not independent: if Alice has a higher degree, she will likely have less EX. So matching gives misleading results.
- Solving the counterfactual problem by SCM: first, the causal diagram

$$\text{Education} \to \text{Experience} \to \text{Salary} \qquad \text{Education} \to \text{Salary} \tag{5.63}$$

  We can fit the data to have the SCM equations as:

$$EX = 10 - 4 \times ED + U_{EX} \qquad S = 65,000 + 2,500 \times EX + 5,000 \times ED + U_S \tag{5.64}$$

  where U_S are U_{EX} are idiosyncratic factors (unique to each individual). Note this is a deterministic equation. With this, we solve the problem in three steps:

  - Abduction: determine U_S are U_{EX} for Alice.
  - Action: apply do operation. This means setting $ED = 1$ for Alice (eliminating any edges to ED).
  - Prediction: estimate S using the causal diagram and idiosyncratic factors. This means estimate the effect of $ED = 1$ on EX, and then on S.
- Why Durbin's approach is flawed? Potential outcome approach makes key assumption of "ignorability". However, this is hard to verify.
- Necessary cause and sufficient cause: Let $X = 1$ be treatment and $Y = 1$ be observed outcome. Suppose we observe $X = 1, Y = 1$, we would like to ask if X causes Y. There are two ways: (1) Probability of necessity, PN. It captures "but for" cause. Someone would not die but for some condition. PN is defined as:

$$PN = P(Y_{X=0} = 0 | X = 1, Y = 1) \tag{5.65}$$

  (2) Probability of sufficiency, PS. It captures proximate cause, defined as:

$$PS = P(Y_{X=1} = 1 | X = 0, Y = 0) \tag{5.66}$$

  It involves imagining $X = Y = 0$ (no death, no treatment), then if we have treatment, would it lead to the outcome? Ex. fire broke out after someone struck a match. Does oxygen cause the fire? It has high PN, but low PS.

- Climate change: can we attribute one extreme weather event to climate change?

Chapter 9: mediation: the search for a mechanism

- The story of Barbara Burks: Nature vs. Nurture. Estimate the effect of parental intelligence on children's IQ (total genetic effect). The causal diagram:

$$\text{Parental IQ} \to \text{Child IQ} \qquad \text{Parental IQ} \to \text{Social status} \to \text{Child IQ} \tag{5.67}$$

  So one should not adjust for social status, which is a mediator. Also suppose we have unknown factor affecting both child IQ and social status:

$$\text{Social status} \leftarrow X \to \text{Child IQ} \tag{5.68}$$

  Then Social status is a collider from parental IQ to social status to X to child IQ. Adjusting it leads to collider bias.

- Why we need mediation analysis? Suppose we have a causal diagram, and we estimate the effect of X on Y. When we perform $do(X)$, the effect will be propagated to other variables, which may change Y. Our goal is to make statements of what are the "mechanisms" of X effect on Y: does it go through a particular variable M or directly?

- Direct effects: Berkeley admission example. One wants to estimate if gender affects admission decision. However, gender may influence which department one applies, and the admission rates differ across departments.

$$\text{Gender} \to \text{Admission} \qquad \text{Gender} \to \text{Department} \to \text{Admission} \tag{5.69}$$

  Our goal is to estimate the direct effect from Gender to Admission.

- Defining direct effects: our model is:

$$X \to Y \qquad X \to M \to Y \tag{5.70}$$

  To define direct effects of X to Y, we will use $do(X)$, and control for M (s.t. the effect must be from X to Y directly). The problem is that when we change X, M will be also, so at what value of M shall we control? Natural Direct Effect (NDE) is defined as M is set at the level it had been when there is no perturbation of X. Let's say M_0 is the value of M for a sample whose $X = 0$. We have:

$$\text{NDE} = P(Y_{M=M_0} = 1|do(X) = 1) - P(Y_{M=M_0} = 1|do(X) = 0) \tag{5.71}$$

  where the subscript indicates counterfactual.

- Defining indirect effects: we want to know the effect of X on Y that is mediated by M, so we should control for X (do-operation on X), and assess the effect of M. But we should not just use do-operation for M: we would learn the effect of M to Y, which may have nothing to do with X. So we should assess what happens to Y, if we change M based on the effect of X on M, while controlling X. This leads to Natural Indirect Effect:

$$\text{NIE} = P(Y_{M=M_1} = 1|do(X = 0)) - P(Y_{M=M_0} = 1|do(X = 0)) \tag{5.72}$$

  where M_1 is the value of M when $X = 1$, and M_0 is the value when $X = 0$. The formulat:

$$\text{NIE} = \sum_m [P(M = m|X = 1) - P(M = m|X = 0)] \times P(Y = 1|X = 0, M = m) \tag{5.73}$$

  In words, to assess the indirect effect of changing $X = 0$ to $X = 1$ on the probability of $Y = 1$: imagine we change X from 0 to 1, how the values of M are changed? Then given that value of M and $X = 0$ (we do not want direct effect), what is the probability of $Y = 1$?

- Smoking gene: increases the risk of cancer, estimation of direct vs. indirect effects. The indirect effect through smoking is very small: the gene makes people smoke only one more cigarette per day (small clinical effect). However, the effect of smoking gene on lung cancer is large only on individuals who smoke - an example of interaction.
- Tourniquet saving soldiers: Tourniquet uses are not found to save lives. However, this is because doctors only collect who survived long to reach hospitals, thus some of Tourniquet effect is through pre-administrative survival, which is not measured.

Chapter 10: big data, artificial intelligence and the big questions

- Data fusion: e.g. merging several datasets from different states. The causal diagrams are different, e.g. in some state we have RCT data, and in different states, we need to adjust for different confounders. But causal model and do-calculus allow us to estimate the same causal effect across studies.
- Selection bias: e.g. we only have data from hospitalized samples, then in the model, we may need to capture this aspect, by having a link from Hospital to the variable of interest.
- Free will: this is basically an illusion. But what is the benefit? Possible explanation: voluntary actions are recognized by a trace they leave in short-term memory. This leaves the impression of "consciousness".
- **How do we encode causal relations in our mind**? [Personal notes] Ex. we know that a car crash is bad even if we have never seen or experienced it. This is based on analogy: we know that crash is bad in general from our experience, and we can make reasonable predictions in new situations. In other words, we use the same pattern recognition mechanism for vision, speech, etc. to make causal predictions.

5.4 Causality: Models, Reasoning and Inference [Judea Pearl]

Problems of causal inference:

- Learn causal model from the data: a model may not be identifiable, e.g. the simplest case, two associated variables (X, Y), is not identifiable.
- Prediction: what would a variable Y be given another variable X?
- Intervention/causal effect: given a causal model, estimate the causal effects of interest (or other causal aspects).
- Counterfactuals: e.g. given that a person who smokes developed lung cancer, would the person avoid lung cancer had he not smoke?

Limitations of regression approach to observational studies:

- Simple regression approach: Suppose we want to infer if X has a causal influence on Y, or estimate the effect of X on Y. Then we do regression of Y on X, with other possible confounding variables, say Z, and we conclude: X has an influence on Y iff the coefficient, $\beta_X \neq 0$; and β_X measures the strength of causation. Why is this approach insufficent?
- Learning causal model:
  - Backward edge: if $Y \to X$, then regressin of Y on X will have non-zero coefficient for X (the parent variables of Y generally cannot explain all the variations of Y).
  - Missing indirect effect: if X acts indirectly on Y through Z, and Z is included in the features, then the effect of X on Y will not be seen.

  - Unobserved confounding variables: will create false association, e.g genotype that influences both smoking and disease.

- Estimating causal effect: suppose the causal model is known, the the coefficient suggests causal effect only when all features influence the response independently. Example: the effect of G on D (1) direct effect of $G \to D$; (2) indirect effect: $G \to S \to D$. The regression of D on G and S will miss the indirect effect.

Structural/functional causal models:

- Structural model: specifies the deterministic relations among variables of interest, plus the disturbances (exogeneous) random variables. In general, a variable x_i can be written as:

$$x_i = f_i(\pi(x_i), u_i) \tag{5.74}$$

  where f_i is a deterministic function, $\pi(x_i)$ represents the variables that directly influences x_i, and u_i be disturbance. In structural equation modeling (SEM), f_i are linear functions. Also assume that u_i are all independent, as otherwise, we would have latent variables that explain the correlation among u_i's.

- Causal diagrams: structural model can be represented by the causal diagram. Example, consider the following structural model:

$$x = f_X(u_X) \tag{5.75}$$

$$y = f_Y(x, u_Y) \tag{5.76}$$

$$z = f_Z(y, u_Z) \tag{5.77}$$

  It can be written as $X \to Y \to Z$, the random (exogenous) variables: U_X, U_Y, U_Z, are typically omitted. Note that the semantics of the model lies in the missing links, i.e. independence relations, e.g. Z is independent of X, conditional on Y.

- Internventional interpretation: a structural model specifies how a variable in the LHS changes when the variables in the RHS are changed (by external force). Ex. $y = \beta x + \epsilon$, we have: $E(Y|do(x)) = \beta x$, or:

$$\beta = \frac{\partial}{\partial x} E[Y|do(x)] \tag{5.78}$$

- Markovian: the model M is semi-Markovian if the causal diagram $G(M)$ is a DAG; if in addition, the background variables are independent, the model is Markovian. Note that any semi-Markovian model can be converted to a Markovian model by introducting latent variables: e.g. if the disturbance term ϵ_X and ϵ_Y are dependent, then introduce $U \to X$ and $U \to Y$.

- Counterfactuals: what is the probability $Y = y'$ under the treatment $X = x'$, given that the actual situation is $Y = y$ under the treatment $X = x$?

Comparison of structural model and probabilistic graphical model: even though causal Bayesian network can be used as causal model, the structural model approach is preferred because:

- Cyclic graph: with structural model, it is possible to have cyclic relationship, and model the dynamics.

- Intervention: more general interventions such as changing parameters of the deterministic equation can be handled in the structural model.

- Counterfactual: the counterfactual questions can be answered in the structural model. This is due to the "stability" of structural models, where the deterministic relationship does not change. On the other hand, this may not be the case for Bayesian network. Ex. the joint probability $P(x, y) = 1/4 \forall x, y$ for two binary RVs X and Y may correspond to different deterministic equations, and each provides a different answer to the counterfactual problem (page 36).

Procedure of learning ounterfactuals in structural models: suppose we want to learn about Y under $X = x$ given evidence e. Let U be the unobserved variable, then:

- Update the probability $P(u)$ to obtain $P(u|e)$;
- Replace the equations: $X = x$;
- Use the modified model to compute $P(Y = y)$.

5.4.1 Inferring Cusation

Model preference: to infer causality from data, some principles for model preference would be needed; otherwise, one could always come up with arbitrarily complex causal models.

- Minimality (Occam's razor): minimal model that explains the dependency structure in the observation without any additional dependency that is not observed.
- Stability: a stable model is preferred, where the dependency relationship does not change with specific parameters of the model. In general, a simpler model is always a special case of a more complex model, which reduces to the simpler model at special parameter values.

General strategies:

- Bayesian structure learning: a model is evaluated by posterior probability. Simpler models will be preferred because prior distribution penalize complex models.
- Condition independence: (1) determine all conditional independence (CI) relations from observation; (2) search for simplest model that is consistent with the CIs.

Inductive causation algorithm: consists of three main steps:

- Determine CI: for any pair of nodes a and b, search for a set S_{ab} s.t. $a \perp b | S_{ab}$. Then connect an undirected edge for a to b, if no set S_{ab} is found.
- Identifying colliders: if a and b share a common neighbor c, check if $c \in S_{ab}$: if not, then create the arrows: $a \to c \leftarrow b$; otherwise, do nothing (as it could be a fork or chain).
- Orient as many undirected edges as possible: the orientations should not create a new v-structure (colliders) and not create a directed cycle.

Local criteria for causal relations: to determine whether X has a causal influence on Y.

- Potential cause: X has a potential causal influence on Y if X and Y are dependent in every context; and there exists a variable Z and context S s.t. (1) X and Z are indepenedent given S, and (2) Z and Y are dependent given S.
- Genuine cause: see Definition 2.7.2. the idea is: if Z is a potential cause of Y, and the effect of Z can be completely explained by X, i.e. $Z \perp Y | S \cup X$, then X is a genuine cause of Y.

Time and causality:

- Statistical time: given a distribution P, it is defined as an ordering of variables that agree with at least one minimal causal structure consistent with P. Thus P may have multiple statistical time (Markov equivalence class).
- Temporal bias: the physical time should coincide with at least one statistical time.

Comments on condition independence (CI)-based learning:

- Covariates: generally, to resolve $X \to Y$, one will need to introduce additional covariates to test more specific models. Ex. the model $X \to Z \to Y$ is testable, as it entails the CI: $X \perp Y|Z$.
- Limitations: there are often unobserved variables, which may make CI invalid. Ex. in the model $X \to Z \to Y$, if there is an observed variable U, s.t. $U \to X$ and $U \to Y$, then the CI: $X \perp Y|Z$ does not hold. While the CI: $X \perp Y|(Z, U)$ holds, it cannot be tested, as U is not observed. To overcome this limitation, one would need to examine consequences of a model other than CI (which may be too restrictive).

5.4.2 Identification of causal effects

Ref: [Pearl, Causality, 2000, Chapter 3]

Intervention: the fundamental solution to the causality problem is intervention.

- Concept: given a structural model M, intervention amounts to the use of external forces to change the causal relations encoded by the model. Most commonly, this is to set one or more variables to some specified values. Thus intervention $do(x_i)$ means: change the value of x_i, and this results in a new model, M_{x_i} where (1) the equation $x_i = f_i(\pi_i, u_i)$ is replaced by $x_i = x$; (2) any variable x_i in other equations will be replaced by the value x.
- Interpretation of intervention with structural models: suppose a system can be represented by a model M without intervention. With intervention, the external force has to be incorporated into the model (the external force remains constant in the situation with no intervention, thus is not needed and omitted by the model M). Intervention can thus be represented by an augmented graph, where the external force of $do(x)$ is explicitly represented as a node F_x: x will be given by $f(x)$ when $F_x = \text{idle}$; x will be equal to the value given by F_x, when it is not idle.
- Intervention and randomization: In the smoking example, to estimate the effect of smoking on diseases, one should subject the patients to treatment (smoking), and compare the difference of diseases. Doing this, the two groups (smoking and not) are random wrt. genotypes, thus any difference in diseases must be attributable to the difference of treatment. This is like in physical experiments, where everything has to be controlled equal with only treatment differs; the population level, this means, the case and control groups must be randomized with everything else except treatment.

Causal effect estimation and causal model selection:

- These are generally two different problems: the technique of causal effect estimation is insufficient for selecting causal models. Ex. to establish the model $Q \to X \to Y$ (direct influence of X on Y), one needs to consider the conditional independence: $Q \perp Y|X$ (this is not held in the alternative model $Q \to Y \to X$; while causal effect estimatation of X on Y will not distinguish two models.
- For special problems: causal effect estimation may be used to choose between alternative models, most importantly test if any edge is necessary. Ex. given a model $Q \to X \to Y, Q \to Y$, test if the direct effect of X on Y is 0.
- The CI test can be formulated as a problem of estimating causal effect:

Defining causal effects:

1. The causal effect of X on Y can be defined in terms of the probability of Y under the model M_x, written as:
$$P(Y = y|do(x)) = P(Y = y|M_x) \tag{5.79}$$
Thus the effect can be defined as, e.g.
$$E(Y|do(x')) - E(Y|do(x)) \tag{5.80}$$
The causal effect reflects adjusting/controlling for confounding variables.

2. Estimating causal effects: $P(Y = y|M_x)$ can be computed in terms of probability distribution in the original model M. Some examples (suppose X is smoking, Y is disease, Z is genotype and W is some lifestyle that may be affected by smoking):

   - $X \to W \to Y$: marginalization of W (average over the intermediate cause):

   $$P(Y = y|do(x_0)) = \sum_w p(w|x_0)p(y|w) \tag{5.81}$$

   - $X \to Y \leftarrow Z$ and $Z \to X$: adjusting for confounding variable Z:

   $$P(Y = y|do(x_0)) = \sum_z p(y|x_0, z)p(z) \tag{5.82}$$

Causal effects in the presence of unobserved variables:

1. Why it is possible: it is not necessary to include all variables in analysis, e.g some variables that do not directly affect X and Y are already randomized. Consider another case, $X \to U \to Y$, then the information of effect of U has already been included in the conditional distribution $P(y|x)$, in fact:

   $$P(y|\hat{x}) = \sum_u P(u|x)P(y|u) = P(y|x) \tag{5.83}$$

2. Adjusting for direct causes: let π_i be the set of direct causes of X_i, and Y be any set of variables disjoint of $X_i \cap \pi_i$, then:

   $$P(y|\hat{x}_i') = \sum_{\pi_i} P(\pi_i)P(y|x_i', \pi_i) \tag{5.84}$$

   Proof: the joint distribution of M_{x_i} differs M by only the term $P(x_i'|\pi_i)$, and express this conditional probability in terms of $P(\pi_i)$ and the probability where π_i is in the conditional part.

3. Back-door criterion: a set of variables Z is sufficent (or admisible) for adjustment if:

   - No element of Z is a descendant of X: no need to worry about these variables as their effects are already included in X.
   - The elements of Z block all "back-door" paths from X to Y, namely all paths that end with an arrow pointing to X: these variables (pointing to X) are responsible for creating spurious association (e.g. genotype in the smoking example), thus should be measured.

   Given a sufficent set Z, we have:

   $$P(y|\hat{(x)}) = \sum_z P(z)P(y|x, z) \tag{5.85}$$

   Proof: consider the augmented graph with the new node F_x. Let P' be its joint distribution, then we have:

   $$P(y|\hat{x}) = \sum_z P'(z|F_x)P'(y|z, x, F_x) \tag{5.86}$$

   The first term is equal to $P(z)$ as Z is non-descendants of X (the causes of X should not be affected by the intervention). And for the second term, we have: $Y \perp F_x|X, Z$, from the d-seperation of Y and F_x by (X, Z) (Z blocks all back paths, and X blocks all other paths from Y to F_x), thus it is equal to $P(y|x, z)$.

4. Front-door criterion: a set of variables Z satisfy front-door criteria if:

   - Z intercepts all directed paths from X to Y;

- There is no back-door path from X to Z;
- All back-door paths from Z to Y are blocked by X.

Intuitively, Z is the children of X that could influence Y, and there is no other variable that influence Z. We have:

$$P(y|\hat{x}) = \sum_z P(z|x) \sum_{x'} P(x')P(y|x'z) \tag{5.87}$$

Thus it is possible to estimate causal effect of X to Y, even if covariates are "post-treatment" variables (those that can be affected by X).

Identifiability:

- In the presence of latent variables, a causal effect may not be identifiable. Ex. in a simplest model, $U \to X, U \to Y, X \to Y$, the effect of X on Y is not identiable, as any such effect may also be due to the unobserved U.
- Examples of identifiable causal diagrams: those specified by the back-door and front-door criteria are identifiable. There are other cases as well. Note that if there is no back-door path to X, then the effect must be identifiable.
- Examples of non-identiable causal diagrams: often involving cofounding variables on X (i.e. some unobserved variable U influences both X and some other variable).

Application of causal effect estimation to structural learning:

- Structural learning: when the causal model is unknown, we can propose a model, and test if the predicted causal effects from the model are consistent with this model. For a variable X, which should affect Y according to the model, if there is no causal effect, estimated from data, (i.e. intenvention of X has no effect on Y), then the proposed model is wrong.
- Exmaple: suppose we want to test if smoking (X) causes disease (Y), we could add a covariate Z, the amount of tar deposited in the lung, and test the model $X \to Z \to Y$ (using CI). However, with unobserved variable U, s.t. $U \to X$, $U \to Y$, CI testing does not work. But we could estimate the causal effect of X on Y, even in the presence of U, with the help of Z (front-door criterion).

Extensions:

- Cyclic causal diagrams: need a general definition of d-separation.
- General intervention: e.g. intervention changes the functional form or parameters in the deterministic equation.

5.4.3 Linear Structural Model (SEM)

Ref: [Pearl, Causality, Chapter 5, 2000]

Problems of parameteric structural model: in particular, the linear model, or structural equation modeling (SEM):

- Model testing: check the assumptions of the model. The general idea is: calculating the covariance structure from the model and compare those with the sample covariance structure in the data.
- Parameter identification: the parameters of the determistic model. In traditional SEM: MLE; in structural theory: through causal effects (see below).

Global vs local parameter fitting and model testing:

- Global approach: parameter estimation through MLE; then model testing is based on these parameters: calculate the partial correlation coefficients according to these parameters (theoretical values) and compare with the observed values in data.
- Local approach: parameter estimation through local causal effects (below), and the conditional independence only depends on the model structure (not parameters), which can be directly verified.
- Comparison: global approach requires MLE of the entire model, however, some paramaters may not be identifiable, and thus the estimated parameters may be unstable. Local approach is taken by the structural theory.

d-separation and partial correlation coefficient (PCC):

- Partial correlation and regression coefficient: given variables X, Y and Z, the partial variance, covariance and correlation coefficient are defined as the term conditioned on Z: $\sigma^2_{X|Z}$, $\sigma^2_{XY|Z}$ and $\rho_{XY|Z}$ (note that these are well-defined because they are not dependent on the value of Z). The partial regression coefficient is given by:
$$r_{YX|Z} = \rho_{YX|Z}\frac{\sigma_{Y|Z}}{\sigma_{X|Z}} \tag{5.88}$$
It is equal to the coefficient of X in the linear regression of Y on X and Z, i.e. $r_{YX|Z} = \beta_X$ in the regression: $Y = \beta_0 + \beta_X X + \beta_Z Z$. More generally, the coefficient of X in a linear regression:
$$Y = aX + b_1 Z_1 + \cdots + b_p Z_p \tag{5.89}$$
is given by $a = r_{YX|Z_1 Z_2 \cdots Z_p}$.
- Markovian models: a partial correlation coefficient $\rho_{XY|Z} = 0$ iff X and Y are d-separated by Z.
- Semi-Markovian and general models: for a linear model which may include cycles and bidirected arcs (dependent errors), the PCC $\rho_{XY|Z} = 0$ if X and Y are d-separated by Z (where a bidrected arc $i \leftrightarrow j$ is interpreted as a latent common parent $i \leftarrow L \rightarrow j$).

Testing conditional independences (causal assumptions of the model):

- Local testing: test the CI implied by the structure model with PCC, estimatable from the data.
- Graphical basis: not all PCCs need to be tested, as some would imply the other ones. In a DAG model, let Z_{ij} be any set of nodes that d-separates i from j for a nonadjacent pair i and j. Then the set of pairs with PCC $= 0$, one element per nonadjacent pair, constitutes a basis for the set of all zero PCCs entailed by the model (i.e. sufficent to verify all zero PCCs).
- Model equivalence: for Markovian models, observational equivalence (i.e. covariance equivalence) is equivalent to the same CI relations (i.e. the same skeleton plus v-structure). For semi-Markovian models, the CI relations are necessary but not sufficient for observational equivalence, i.e. it is possible that the same CI may imply different covariance equivalence.
- Model equivalence limits our ability of testing models: a model can only be tested/verified up to its equivalence class.

Parameter identification in linear models: in general semi-Markovian models. The basic strategy is to relate the causal parameters (path coefficients) to the observable partial correlation (or regression) coefficients.

- Direct effects: to determine α, the path coefficient associated with $X \rightarrow Y$. Let G_α be the diagram when $X \rightarrow Y$ is removed from G, then $\alpha = r_{YX|Z}$ for a set of variables Z if: (1) Z contains no descendent of Y; and (2) X and Y are d-separated by Z in G_α. The intuition is: X may have both direct, defined by α, and indirect effects on Y. If Z blocks all indirect effects (Z should not contain descendant of Y - the post-treatment variable), then any effect of X on Y (partial regression conditional on Z) must be due to α.

- Back-door criterion: the total effect of X on Y is given by $r_{YX|Z}$ for a set of variables Z if: (1) no member of Z is a descendant of X; and (2) X and Y are d-separated by Z in $G_{\underline{X}}$ formed by deleting all arrows emanating from X. The total effect is defined by the sum of the products of path coefficients along all directed paths from X to Y. Ex. $X \xrightarrow{\alpha} Y, X \xrightarrow{\beta} Z \xrightarrow{\gamma} Y$, the total effect of X on Y is: $\alpha + \beta\gamma$.

- Instrumental variables: in some cases where a parameter may not be identifiable, introducing instrumental variables may make it identifiable. Ex. in $X \xrightarrow{\alpha} Y, X \leftrightarrow Y$, α is not identifiable, however, adding $Z \xrightarrow{\beta} X$, we have: $\beta = r_{XZ}$ and $\alpha\beta = r_{YZ}$.

- Example: $Z \xrightarrow{\beta} X \xrightarrow{\alpha} Y, Z \leftrightarrow Y$, we have: $\alpha = r_{YX|Z}, \beta = r_{XZ}$.

- Example: $X \xrightarrow{\alpha} W \xrightarrow{\beta} Y, X \xrightarrow{\delta} Z \xrightarrow{\gamma} Y, X \leftrightarrow Z, W \leftrightarrow Y$, we have: (1) the direct effect of X on W: X and W are d-separated in G_α, thus $\alpha = r_{WX}$; (2) the total effect of X on Y: X and Y are d-separated by Z in $G_{\underline{X}}$, thus $\alpha\beta = r_{YX|Z}$; (3) the direct effect of Z on Y: they are d-separated by X in G_γ, thus $\gamma = r_{YZ|X}$.

Application of SEM on linear regression: in general, in the regression problem, the predictors $X_1, \cdots, X_p$ are not independent, so it's possible to model the dependence of X_j's and Y.

- Example: $X_1 \to Y, X_2 \to Y$, we have $\beta_1 = r_{YX_1}, \beta_2 = r_{YX_2}$, note that since X_1 and X_2 are independent, we don't need conditioning here. If in addition, we have $X_1 \leftrightarrow X_2$, then conditioning on other explanatory variables is necessary: $\beta_1 = r_{YX_1|X_2}, \beta_2 = r_{YX_2|X_1}$.

- Correlated features: even features are correlated, the model is still identifiable: β_1 from the correlation of X_1 and Y in the stratum of X_2; and similar for β_2. However, correlation reduces the variablity within the stratum (thus parameter estimation is less stble); in the extreme case, X_1 and X_2 are perfectly correlated, then there is no variation of X_1 in the stratum of X_2, thus not possible to estimate β_1.

Identification of parameters in nonparameteric models:

- Limitations of parameter identification: only from the probility distribution. Thus it would be impossible to identify two differenent models if they lead to the same distribution.

- In general, the non-parameter models (functions) have many-to-one relations with probability distributions, thus not identifiable from observational data. However, different models have different interventional properties.

5.4.4 Counterfactuals and Applications

Ref: [Pearl, 2009]

Applications of counterfactual analysis:

- Direct vs total effect: in some cases, we are only interested in direct causal effect. Ex. whether gender (X) directly influences hiring (Y). Let Z be the qualification, then the direct effect of X on Y, as opposed to that mediated by Z, can be defined as, the controlled direct effect:

$$\text{CDE} := E(Y|do(x'), do(z)) - E(Y|do(x), do(z)) \tag{5.90}$$

  Or in counterfactual notation:

$$\text{CDE} := E(Y_{x'z}) - E(Y_{xz}) \tag{5.91}$$

  Thus CDE controls the mediating variable, and the difference must be due to X. In case CDE depends on the value of Z, we should consider the "natural direct effect", where Z is set the counterfactual at given values of X:

$$\text{DE}_{x,x'}(Y) := E(Y_{x',Z_x}) - E(Y_x) \tag{5.92}$$

Under some assumptions, it can be shown that it is the weighted average of CDE:

$$\mathrm{DE}_{x,x'}(Y) = \sum_z P(z|do(x)) \left[E(Y|do(x', z)) - E(Y|do(x, z))\right] \tag{5.93}$$

- Probability of causation: we are interested in the question, given $X = x$ and $Y = y$, what is the probability that Y would be different had $X = x'$. Ex. in the smoking example, this means, for a patient with smoking and disease, what is the chance that he would not have the disease had he not smoke. This would be a measure of causation of disease due to smoking. The probability of causation or necessarity is defined as:

$$PN(x, y) = P(Y_{x'} = y'|X = x, Y = y) \tag{5.94}$$

5.4.5 Comparison with Other Approaches

Potential outcome framework by Neyman-Rubin:

- Missing data approach: given a causal model, to assess the effect of X on Y, introduce random variables Y_x, which is the value of Y had the treatment be x (even if the actual treatment may be different). Given an observation $(X = x, Y = y)$, we have:

$$Y_x = y \tag{5.95}$$

  i.e. the counteractual is equal to the actual observation. Thus, counterfactuals are treated as missing data (half of them are observed, assuming X is binary), and the usual statistical apporach can be applied on the joint distribution $P*$ defined on all variables plus counterfactuals.

- Solving the counterfactuals: the casuality assumptions involved in specifying the causal model will be needed. Ex. if a set of covariates satisfy CI: $X \perp Y_x|Z$, then we have:

$$P * (Y_x = y) = \sum_x P(y|x, z)P(z) \tag{5.96}$$

  which is similar to the equation under the structural theory. The intuition: we should control for confounding variable Z; if given Z, X and Y_x are independent (e.g. given genotype, smoking and smoking suspectibility are independent), then we could estimate the causal effect.

- Difficulties with potential outcome framework: the causality assumptions are not explicitly formulated, thus it may be difficult to verify the probabilistic properties of the distribution involving counterfactuals (i.e. a calculus of counterfactuals, in addition to probability distribution, is missing in this framework).

5.5 Structural Equation Modeling

1. Introduction to Structural Equation Modeling (SEM)

   Reference: [Bollen, Structural Equations with Latent Variables, Chapter 1, 2]

   Modeling strategy of SEM:

   - Latent variable model: the basic SEM strategy is that the true causal relations in the model are encoded by latent variables. Normally, define ξ the n-dim. exogeneous variables (not explained by the variables within the model), and η m-dim. the endogenous variables. Then η can be written as linear models of ξ and η:

$$\eta = B\eta + \Gamma\xi + \zeta \tag{5.97}$$

where B is $m \times m$ matrix, and Γ is $m \times n$ matrix, ζ is the unexplained random error of η (independent of the latent variables). Ex. a model of one exogeneous variable and two endogenous variables:

$$\eta_1 = \gamma_{11}\xi_1 + \zeta_1 \tag{5.98}$$

$$\eta_2 = \beta_{21}\eta_1 + \gamma_{21}\xi_1 + \zeta_2 \tag{5.99}$$

The error assumption: ζ_i is (1) homoscedastic: same variance across multiple observations; (2) non-autocorrelated: independent across multiple observations.

- Measurement error model: the observations are measurements or proxies of the latent variables. Let x be the q-dim. measurement of the exogenous variables and y be the p-dim. measurements of the endogenous variables:

$$x = \Lambda_x \xi + \delta \tag{5.100}$$

$$y = \Lambda_y \eta + \epsilon \tag{5.101}$$

Note that: x and ξ (or y and η) are not necessarily in the same scale, thus there are additional coefficients, encoded by Λ_x.

- Notations: normally assume that all the variables in the model have zero mean. Thus covariance matrices in the latent variable model:

$$\Phi = \mathrm{E}\left(\xi\xi^T\right) \tag{5.102}$$

is the covariance matrix of the endogenous variable, and

$$\Psi = \mathrm{E}\left(\zeta\zeta^T\right) \tag{5.103}$$

is the covarance matrix of the random error of the endogenous variables. For the measurement model,

$$\Theta_\epsilon = \mathrm{E}\left(\epsilon\epsilon^T\right) \qquad \Theta_\delta = \mathrm{E}\left(\delta\delta^T\right) \tag{5.104}$$

are the covariance matrices of ϵ and δ respectively.

SEM representation:

- Path diagram: (Table 2.6) square box for observed variables, circles for latent variables, single-headed arrows for causal relations, disturbrance or errors in unenclosed variables, two-headed arrows for associations.
- Reciprocal or feedback relations: allowed in SEM. Represented by two single-headed arrows.

Problems of SEM:

- Identification: the basic strategy of SEM estimation is the MOM type of estimator. A SEM implies certain covariance/correlation among all observed variables, let it be $\Sigma(\theta)$, a function of θ, where θ represents all model parameters - this is the population covariance. The sample covariance meanwhile can be determined from data. Thus we should have:

$$\Sigma = \Sigma(\theta) \tag{5.105}$$

In general, if the number of covariance terms is greater than the number of parameters, we have overdetermination; and if smaller, we have underdetermination. Examples:

  - Overdetermination: suppose we have a model of p variables, now adding one more variable (with one or two more parameters) linked to one node. The increase of the number of covariance terms is p, and this leads to overdetermination.
  - Underdetermination: given a model of two observables ($X \to Y$), we could estimate the effect of X on Y by sample $\mathrm{Cov}\left(X, Y\right)$. Adding one more latent variable, $X \to Z \to Y$, not possible to determine the parameters related to Z. Effectively, $\mathrm{Cov}\left(X, Y\right)$ can result from the direct effect or indirect effect through Z.

- Estimation: Equation 5.105 is the basic equation for estimation (MOM). However, the system may be overdetermined, so we need to find θ s.t. $\Sigma(\theta)$ is closest to Σ. This can be done via maximum likelihood or other measures of distance between the two matrices. The inference challenges:
  - Testing some parameters or effects: e.g. through LRT.
  - Assessing goodness-of-fit of the model: assessing how good the causal model is. However, in general, we cannot prove the causality even if we have good model fitting (could be other models that fit as well).
- Effects: the total effect of one variable on another is the sum of direct and indirect effects (product of multiple terms). Ex. given a model:

$$\xi_1 \xrightarrow{\gamma_{21}} \eta_2 \qquad \xi_1 \xrightarrow{\gamma_{11}} \eta_1 \qquad \eta_1 \xrightarrow{\beta_{21}} \eta_2 \tag{5.106}$$

  The total effect of ξ_1 on η_2 is thus: $\gamma_{21} + \gamma_{11}\beta_{21}$.

2. SEM with observed variables

   Reference: [Bollen, Structural Equations with Latent Variables, Chapter 4]

   Model specification:

   - Model: no latent variables, let x be q-dim. exogenous variables (or explanatory variables) and y be p-dim. endogenous variables, we have:

$$y = By + \Gamma x + \zeta \tag{5.107}$$

     where B is $p \times p$ matrix, Γ is $p \times q$ matrix, and ζ is the p-dim. error vector. The standard assumption is that errors are uncorrelated with x. We denote $\Psi = \mathrm{E}(\zeta\zeta^T)$ the covariance matrix of the errors, and $\Phi = \mathrm{E}(xx^T)$ the covariance matrix of the explanatory variables.
   - Recursive and nonrecurive models:
     - Recursive models: contain no reciprocal relation or feedback loops. For these models, B is lower triangular matrix (or can be rearranged s.t. it is lower triangular), and the covariance matrix of the errors, Ψ, is diagonal. However, the covariance matrix of x, Φ, can be nondiagonal.
     - Nonrecurive models: reciprocal relation or feedback loops.
   - Implied covariance matrix: assuming the mean of each variable is 0, we have:

$$\Sigma_{yy}(\theta) = \mathrm{E}(yy^T) = (I - B)^{-1}(\Gamma\Phi\Gamma^T + \Psi)(I - B)^{-T} \tag{5.108}$$

     where $-T$ denotes the transpose of the inverse. And similarly,

$$\Sigma_{xx}(\theta) = \Phi \qquad \Sigma_{xy}(\theta) = \Phi\Gamma^T(I - B)^{-T} \tag{5.109}$$

   Model identification:

   - t-rule: for the model to be identified, the number of parameters (t) must be less than the number of free terms in the covariance matrix, so the necessary condition is:

$$t \leq \frac{1}{2}(p + q)(p + q + 1) \tag{5.110}$$

     The condition is not sufficient because it is possible that some parameters remain unidentified even though overall the number of parameters is small.
   - Null B rule: a sufficient condition of model identification is that $B = 0$, i.e. the endogenous variables are only caused by exogenous variables. Remark: this is simply the case of multivariate regression.

- Recursive rule: if the model is recursive, i.e. B is lower triangular and Ψ is diagonal, then the model is identified.
- Order and other conditions: when the model is nonrecursive, a necessary condition for an equation to be identified is that the number of variables excluded from that equation be at least $p-1$.

Estimation: important when the model is overidentified.

- MLE: the joint distribution follows multivariate normal distribution (with mean 0), so the likelihood is a function of $\Sigma(\theta)$. The log-likelihood function according to MVN:

$$\log L(\theta) = -\frac{N(p+q)}{2}\log(2\pi) - \frac{N}{2}\log|\Sigma(\theta)| - \frac{N}{2}\text{tr}\left(S\Sigma^{-1}(\theta)\right) \tag{5.111}$$

  where S is the sample covariance matrix. The log-likelihood at the sample covariance matrix $\Sigma = S$ is given by (replacing Σ with S):

$$\log L(S) = -\frac{N(p+q)}{2}\log(2\pi) - \frac{N}{2}\log|S| - \frac{N}{2}(p+q) \tag{5.112}$$

  From this, we obtain the likelihood ratio test statistic:

$$-2[\log L(\theta) - \log L(S)] = N \cdot F_{\text{ML}}(\theta) \tag{5.113}$$

  where:

$$F_{\text{ML}}(\theta) = \log|\Sigma(\theta)| + \text{tr}\left(S\Sigma^{-1}(\theta)\right) - \log|S| - (p+q) \tag{5.114}$$

  MLE is equivalent to minimizing the function $F_{\text{ML}}(\theta)$ above (similar to squared error). The confidence interval can be obtained from the asymptotic covariance matrix of $\hat{\theta}$.
  - Remark: in the book, the LRT statistic is $(N-1)\cdot F_{\text{ML}}(\theta)$.
- Least square: unweighted least square method that minimizes the function:

$$F_{\text{ULS}} = \frac{1}{2}\text{tr}\left[(S-\Sigma(\theta))^2\right] \tag{5.115}$$

  And similarly one can define weighted least square. These estimators are intuitive, however, they are not scale invariant, thus the values change with any change of scale.

Other issues:

- Causal interpretation of coefficients & comparison of explanatory variables: the causal interpretation is, when X changes by a unit, the change of Y, i.e.

$$\beta = \frac{\Delta Y}{\Delta X} \tag{5.116}$$

  To compare different explanatory variables, however, we need to make them in the same scale. First, elasticity is defined using the percent change of X and Y:

$$\text{Elasticity} := \frac{\Delta Y/Y}{\Delta X/X} = \beta\frac{X}{Y} \tag{5.117}$$

  Normally, it is evaluated at $\bar{X}$ and $\bar{Y}$. However, when X and Y are standarized, i.e. zero mean, elasticity is not well-defined. The alternative is to measure the change using the standard deviation as units:

$$\text{Standarized Coefficient} := \frac{\Delta Y/\sigma_Y}{\Delta X/\sigma_x} = \beta\frac{\sigma_X}{\sigma_Y} \tag{5.118}$$

  Note however, in practice, β, σ_X and σ_Y needs to be replaced by their sample versions.

- Interaction terms: if there is an interaction, say between X_1 and X_2, we simply introduce a new variable $X_3 = X_1 X_2$, and the rest is similar. However, X_3 is no longer normally distributed (the conditional distribution of Y still normal).

3. Confirmatory factor analysis

   Reference: [Bollen, Structural Equations with Latent Variables, Chapter 6,7]

   Measurement model:

   - Modeling strategy: when we have abstract concepts that cannot be directly measured, we treat them as latent variables (e.g. anxiety, terroism), and provide operational definition, i.e. (multiple) proxy variables that are determined by these concepts. We specify the relation between measurement and the latent variables, as the measurement error model.
   - Lessons for developing measurement models: example, we have latent factors for democracy at 1960 and 1965, and some observables dependent on democracy. Some modeling lessons:
     - Abstraction: the key concepts, in this case, the level of democracy.
     - Causal constraints: e.g. a variable can only influence the variables in a later time point. In this example, the factor at 1965 could not affect observables at 1960.
     - Additional relations: e.g. in 1960 and 1965, the measurement model is the same (i.e. the same coefficients of how the observations depend on the latent factor).
   - Exploratory and confirmatory factor analysis: factor analysis is to use a smaller number of (latent) factors to explain the correlation between observed variables. Exploratory factor analysis (EFA) mainly for determining the number of factors. However, statistical methods of EFA has limited power (better to use causal knowledge).

   Confirmatory factor analysis (CFA) model:

   - Model specification: suppose we have n-dim. latent factors (exogenous variables) ξ, and n-dim. observed variables x, our model:
     $$x = \Lambda_x \xi + \delta \tag{5.119}$$
     where δ is the measurement error. We generally assume that $\mathrm{E}(\delta) = 0$ and $\mathrm{E}(\xi\delta) = 0$.
   - Definitions: factor loading - the value of Λ_{ij}, factor complexity - the number of latent factors that influence an observed variable.
   - Implied covariance matrix: it is given by:
     $$\Sigma(\theta) = \mathrm{E}(xx^T) = \Lambda_x \Phi \Lambda_x^T + \Theta_\delta \tag{5.120}$$
     where Φ is the covariance matrix of the latent factors, and Θ_δ is the covariance matrix of the measurement error terms. We consider a special case: one factor, multiple proxy variables (indicators):
     $$x_i = \lambda_i \xi + \delta_i, \qquad i = 1, 2, \cdots, q \tag{5.121}$$
     Then we have the covariance terms:
     $$\mathrm{Var}(x_i) = \lambda_i^2 \phi + \mathrm{Var}(\delta_i) \tag{5.122}$$
     where ϕ is the variance of ξ. And:
     $$\mathrm{Cov}(x_i, x_j) = \lambda_i \lambda_j \phi \tag{5.123}$$

   Identification of CFA model:

- Scale of latent factors: for any laten factor, ξ_i, the scale needs to be specified, otherwise, it is obviously not identified. Typically, let $\lambda_i = 1$ for one of the indicator variable of ξ_i (instead of the variance of any observations).
- One factor - multiple indicator case: suppose there are q indicators, then there are $\frac{1}{2}q(q+1)$ terms in the sample covariance matrix. The number of free parameters is: 2 per indicator (one for λ and one for the variance of δ term), and one parameter for ϕ (the variance of ξ), but we need to subtract 1 (the coefficient of one indicator is equal to 1), so the total number of parameters is $2q$. The model is identified if $q \geq 3$.
- t-rule: let t be the number of parameters, then a necessary condition of model identification is $t < \frac{1}{2}q(q+1)$.
- Three-indicator rule: as discussed before, if a factor has three or more indicators and Θ_δ diagonal, then it is identified.
- Two-indicator rule: suppose the load complexity is exactly one (i.e. each observed variable depends only one latent factor), and there are at least two indicators per factor, then the model is identified if each row of Φ has at least one non-zero off-diagonal element and Θ_δ is diagonal.
  - Idea: if there are only two indicators, but the factors are correlated, then one can borrow information from indicators of correlated factors (s.t. the effective number of indicators is higher).
  - Example: we have the following model:

    $$x_1 \overset{1}{\leftarrow} \xi_1 \xrightarrow{\lambda_2} x_2 \qquad x_3 \overset{1}{\leftarrow} \xi_2 \xrightarrow{\lambda_2} x_4 \qquad \xi_1 \leftrightarrow \xi_2 \tag{5.124}$$

    where the last part specifies correlation between the two latent factors. The model is identified, in particular, we have, the covariance bdtween the two latent factors:

    $$\phi_{12} = \text{Cov}\,(x_1, x_3) \tag{5.125}$$

- Local identification using information matrix: θ is identified at some point if and only if the inverse of the information matrix exists at that point.

Estimation:

- Maximum likelihood: Similar to the case of SEM with observed variables (Equation 5.114), we minimize the function:

  $$F_{\text{ML}}(\theta) = \log|\Sigma(\theta)| + \text{tr}\,(S\Sigma^{-1}(\theta)) - \log|S| - q \tag{5.126}$$

- Improper solutions: the parameter values that are impossible in the population, e.g. negative covariance. This may be caused by several factors:
  - The true values may be close to the boundary, and because of sample fluctuations, the estimated value may appear to be improper.
  - Outliers: because of the assumptions made (multivariate normality), an outlier that violates these assmptions may create improper estimates.
  - Fundamental problem in the model specification.

Model evaluation: overall fit

- Problem: once we formulate an SEM model, we need to evaluate how good the model fits the data (overall model fit), or test specific components/parameters of the model. For the latter, for example:
  - Effect of latent factor on observed variable: test if $\lambda_{ij} = 0$ for some factor i and observed variable j.

- Correlation/covariance between latent factors: e.g. in the two-indicator model (Equation 5.124), test if $\phi_{12} = 0$.

- Likelihood ratio test: we are comparing two models: H_0 a restricted model using MLE from the SEM model, and H_1 the full model using the sample covariance matrix. We use Equation 5.126 and the relation between LRT and F_{ML}:

$$-2\log(L_0 - L_1) = (N-1) \cdot F_{\text{ML}} = (N-1) \cdot \left[\log|\hat{\Sigma}| + \text{tr}\,(S\hat{\Sigma}^{-1}) - \log|S| - q\right] \tag{5.127}$$

  where $\hat{\Sigma}$ is computed from the MLE of θ. It follows χ^2 test with the d.f. $\frac{1}{2}q(q+1) - t$.

- Residuals: the quality of the fit can be judged by how close $\Sigma(\hat{\theta})$ is to the sample covariance S. So a simple statistic is the absolute value of mean (or median) of the elements of the residual matrix $S - \hat{\Sigma}$. May be corrected via: (1) correlation residuals; (2) correct for sample size.

- Incremental fit measure: similar to the LRT, instead of comparing a model with the full model using S, we compare a maintained model (m) with a very restrictive baseline model (b). The percent reduction of the error (F function) is a measure of how much gain is produced by using the model m:

$$\Delta_1 = \frac{F_b - F_m}{F_b} \tag{5.128}$$

  However, the measure does not control for:

  - Degree of freedom: clearly, a complex model would have higher percent reduction.
  - Sample size: while both F_m and F_b may decline with large N, their rate may be different. In particular, F_b may decline more slowly as N increases (a poor model may not benefit much as sample size increases).

  See the text for various normalizations.

Model comparison: specific components

- Likelihood ratio test: suppose we compare a restricted model (r) and unrestricted model (u). Example, for testing the effect size, the restricted model may have some λ term equal to 0. The test statistic:

$$-2\left[\log L(\hat{\theta}_r) - \log L(\hat{\theta}_u)\right] = (N-1)(F_r - F_u) \tag{5.129}$$

- Other large-sample tests can also be used, including Score test and Wald test.

- Example: two-indicator model, Equation 5.124. We want to test $H_0 : \phi_{12} = 0$. The challenge is that the model H_0 is not identified (only two indicators, without correlation).

4. Error-in-variable (EIV) model

   Reference: [Fuller, Measurement Error Models], [Casella, Statistical Inference, Chapter 12], [Total Least Squares and Errors-in-Variables Modeling: Bridging the Gap between Statistics, Computational Mathematics and Engineering, Van Huffel, 2004]

   EIV model overview:

   - EIV model: our observations are (x_i, y_i), x_i are measurements of the true variables ξ_i, with Gaussian errors. In addition, y_i is related to the latent variable by a linear model. So we have:

$$x_i = \xi_i + u_i \qquad u_i \sim N(0, \sigma_u^2) \tag{5.130}$$

$$y_i = \beta_1 \xi_i + \beta_0 + e_i \qquad e_i \sim N(0, \sigma_e^2) \tag{5.131}$$

     This model is different from the ordinary regression model by the measurement error of explanatory variables. We can write the model in a regression form by eliminating ξ_i:

$$y_i = \beta_1(x_i - u_i) + \beta_0 + e_i = \beta_1 x_i + \beta_0 + (e_i - u_i\beta_1) \tag{5.132}$$

This is not standard regression, however, as x_i is correlated with the error term (covariance equal to $-\beta_1\sigma_u^2$).

- Functional and structural model: if we view ξ_i (the true explanatory variables) as unknown constants, the model is known as a functional relationship; if we view ξ_i as random variables and independent of errors, the model is a structural relationship. In particular, if:

$$\xi_i \sim N(\mu_\xi, \sigma_\xi^2) \tag{5.133}$$

then it is a structural model.

- Relationship between the two models: [Casella]
  - Consistent estimators in the functional model are also consistent in the structural model.
  - If a parameter is identifiable in the functional model, then it is identifiable in the structural model.

  The intuition is that the functional model is a special case of the structural model: if a parameter is identifiable in the functional model, then we could do some iterative scheme to identify the parameter in the structural model (at each step assuming ξ_i are known).

Model simplification and identification:

- Identification of the functional model: let $\theta = (\beta_0, \beta_1, \xi_1, \cdots, \xi_n, \sigma_u^2, \sigma_e^2)$, the likelihood function is:

$$L(\theta|\mathbf{x}, \mathbf{y}) = \frac{1}{(2\pi\sigma_u\sigma_e)^n} \exp\left[-\sum_i \frac{(x_i - \xi_i)^2}{2\sigma_u^2}\right] \exp\left[-\sum_i \frac{(y_i - \beta_1\xi_i - \beta_0)^2}{2\sigma_e^2}\right] \tag{5.134}$$

  The problem is: it does not have a finite maximum. To see this, take $\xi_i = x_i$, and let $\sigma_u^2 \to 0$. Thus we need additional constraints on the parameters, most commonly, this is $\sigma_e^2/\sigma_u^2 = \lambda$.

- Structural model and bivariate normal distribution: we could show that with the structural model, the distribution of (x_i, y_i) follows independent bivariate normal distribution with mean:

$$\mathrm{E}(X) = \mu_\xi \qquad \mathrm{E}(Y) = \beta_0 + \beta_1\mu_\xi \tag{5.135}$$

  and covariance matrix:

$$\mathrm{Var}(X) = \sigma_\xi^2 + \sigma_u^2 \qquad \mathrm{Var}(Y) = \beta_1^2\sigma_\xi^2 + \sigma_e^2 \qquad \mathrm{Cov}(X, Y) = \beta_1\sigma_\xi^2 \tag{5.136}$$

- Identification of the structural model: our model has six parameters $(\mu_\xi, \sigma_\xi^2, \sigma_u^2, \beta_0, \beta_1, \sigma_e^2)$, while the bivariate normal distribution has only five parameters. Suppose we perform MOM parameter estimation, we will have five equations (sample mean and variance) for 6 parameters, thus not all parameters are identified.
  - Some parameters may still be identified, e.g. $\mu_\xi = \bar{X}$, regardless of other parameters.
  - Intuition: in the above five equations, suppose we have any value of one parameter, e.g. β_1, we could solve the others. Intuitively, we could fit the model in different ways: strong correlation between the variables (large β_1) but large measurement error; or weak correlation (small β_1) but small measurement error.
- Identifiability conditions: the most commonly used model assumes that $\lambda = \sigma_e^2/\sigma_u^2$ is known. Other identifiable cases include: the measurement error, σ_u^2 is known.

Model with known measurement error:

- MOM estimation: we consider the structural model. Suppose the variance of the measurement error, σ_ξ is known. Let the sample means be $\bar{X}$ and $\bar{Y}$, and the same variance/covariance be

S_{XX}, S_{XY} and S_{YY}. Since (X, Y) is normally distributed, the same mean and variance coverger to the population mean and variance. Solving the five equations, we have:

$$\hat{\beta}_1 = (S_{XX} - \sigma_u^2)^{-1} S_{XY} \tag{5.137}$$

$$(\hat{\sigma}_\xi^2, \hat{\sigma}_e^2) = (S_{XX} - \sigma_u^2, S_{YY} - \hat{\beta}_1 S_{XY}) \tag{5.138}$$

$$(\hat{\mu}_\xi, \hat{\beta}_0) = (\bar{X}, \bar{Y} - \hat{\beta}_1 \bar{X}) \tag{5.139}$$

- Sampling distribution of the estimators: the basic strategy is (1) use Delta Method to express the estimators as linear functions of sample mean and sample covariance; (2) the limiting distribution of sample covariance is given by (extended) CLT. The last step is to replace the true parameter values in the limiting distribution with the MLE (to form the consistent estimator of the limiting distribution).
  - First, we introduce the variables v_i:

    $$v_i = y_i - \beta_1 x_i - \beta_0 = e_i - \beta_1 u_i, i = 1, \cdots, n \tag{5.140}$$

    v_i is similar to the residuals in ordinary linear regression. Clearly, $\mathrm{E}(v_i) = 0$. We could then define the population and sample covariance involving v_i:

    $$\sigma_{Xv} = \mathrm{Cov}(x_i, v_i) = -\beta_1 \mathrm{Var}(u_i) = -\beta_1 \sigma_u^2 = \sigma_{uv} \tag{5.141}$$

    $$S_{Xv} = \frac{1}{n-1} \sum_i (x_i - \bar{x}) v_i \tag{5.142}$$

  - Expansion of estimators: Taylor expansion of $\hat{\beta}_1$ as a function of S_{XY} and S_{XX}, near the true values σ_{XY} and σ_{XX}:

    $$\hat{\beta}_1 \approx \frac{\sigma_{XY}}{\sigma_{XX} - \sigma_u^2} + \frac{S_{XY} - \sigma_{XY}}{\sigma_{XX} - \sigma_u^2} - \frac{\sigma_{XY}(S_{XX} - \sigma_{XX})}{(\sigma_{XX} - \sigma_u^2)^2} \tag{5.143}$$

    Simplify the equation using $\sigma_{XX} - \sigma_u^2 = \sigma_\xi^2$, $\sigma_{XY} = \beta_1 \sigma_\xi^2$, and the equations of S_{Xv} and σ_{Xv}, we have:

    $$\hat{\beta}_1 \approx \beta_1 + \frac{1}{\sigma_\xi^2}(S_{Xv} - \sigma_{Xv}) \tag{5.144}$$

    And similarly, we could have:

    $$\hat{\beta}_0 \approx \beta_0 - (\hat{\beta}_1 - \beta_1)\mu_\xi \tag{5.145}$$

  - Limiting distribution: we use the limiting distribution of $S_{Xv} - \sigma_{Xv}$ from CLT, and this allows us to obtain the limiting distribution of $\hat{\beta}_1$ and $\hat{\beta}_0$.
- Theorem: the vector $\sqrt{n}[\hat{\beta}_1 - \beta_1, \hat{\beta}_0 - \beta_0]$ converges in distribution to a normal random vector with zero mean and covariance matrix:

  $$\Gamma = \begin{bmatrix} \mu_\xi^2 \sigma_\xi^{-4}(\sigma_{XX}\sigma_{vv} + \sigma_{Xv}^2) + \sigma_{vv} & -\mu_\xi^2 \sigma_\xi^{-4}(\sigma_{XX}\sigma_{vv} + \sigma_{Xv}^2) \\ -\mu_\xi^2 \sigma_\xi^{-4}(\sigma_{XX}\sigma_{vv} + \sigma_{Xv}^2) & \sigma_\xi^{-4}(\sigma_{XX}\sigma_{vv} + \sigma_{Xv}^2) \end{bmatrix} \tag{5.146}$$

  For the consistent estimator of Γ, denoted as $\hat{V}(\hat{\beta}_0, \hat{\beta}_1)$, see [Fuller, Theorem 1.2.1].
- Testing β_1: a t-test of β_1 is given by:

  $$t = (\hat{\beta}_1 - \beta_1)/\sqrt{\hat{V}(\hat{\beta}_1)} \tag{5.147}$$

- Remark: an alternative strategy is to use the asymptotic distribution of MLE (the MOM estimator is also MLE). However, the log-likelihood function is a complex function of the parameters, and its second derivartive wrt. the parameters are even more complex.

- Estimating the latent variables: suppose we have the parameter values. We consider two cases:
  - Functional model: treating ξ_i as a constant, then we could view x_i and y_i as linear function of ξ_i:
    $$x_i = \xi_i \cdot 1 + u_i \tag{5.148}$$
    $$y_i = \xi_i \cdot \beta_1 + e_i \tag{5.149}$$
    Treating this as a linear model, where ξ_i is the coefficient.
  - Structural model: the joint distribution of ξ_i, x_i, y_i follows MVN distribution. The problem is thus inferring the conditional distribution of one compoment given the other components in a MVN.

Functional model with known variance ratio:

- Least square estimator: the expectation of the LS estimator:
  $$\mathrm{E}\left(\hat{\beta}_1^{LS}\right) = \mathrm{E}\left[\frac{\mathrm{Cov}\left(X,Y\right)}{\mathrm{Var}\left(X\right)}\right] = \beta_1 \frac{\sigma_\xi^2}{\sigma_\xi^2 + \sigma_u^2} \tag{5.150}$$
  Clearly the LS estimator is biased, and the ratio:
  $$\kappa = \frac{\sigma_\xi^2}{\sigma_\xi^2 + \sigma_u^2} \tag{5.151}$$
  is called the reliability ratio. It is below 1 because of the measurement error of X. It is similar to the heritability in the genetic context.
- Orthogonal regression for functional model: under the assumption that $\lambda = 1$, the likelihood:
  $$L(\beta_1, \beta_0, \sigma_\delta^2, \xi_1, \cdots, \xi_n) \propto \sigma_\delta^{-2n} \exp\left\{-\frac{1}{2\sigma_\delta^2}\sum_i \left[(x_i - \xi_i)^2 + (y_i - \beta_1\xi_i - \beta_0)^2\right]\right\} \tag{5.152}$$
  At any value of σ_δ^2 (it is independent of other parameters), we are minimizing the sum of square of orthogonal distances from data points to thel line. This is the orthogonal regression.

Structural model with known variance ratio:

- Bayesian inference for structural model [The Bayesian Estimation of a Linear Functional Relationship, Lindley & El-Sayyad, JRSSB, 1968]: assume the prior distribution $\xi_i \sim N(0, \tau)$, the likelihood of the i-th observation is:
  $$p(x_i, y_i|\beta, \sigma_\xi^2, \sigma_\epsilon^2, \tau) = \int p(\xi_i|\tau) p(x_i|\xi_i, \sigma_\xi^2) p(y_i|\xi_i, \beta, \sigma_\epsilon^2) d\xi_i \tag{5.153}$$
  All the three distributions are normal, and we have (x_i, y_i) follows bivariate normal distribution with zero mean, and:
  $$\mathrm{Var}\left(x_i\right) = \tau + \sigma_\xi^2 \qquad \mathrm{Var}\left(y_i\right) = \beta_1^2\tau + \sigma_\epsilon^2 \qquad \mathrm{Cov}\left(x_i, y_i\right) = \beta_1\tau \tag{5.154}$$
  The posterior distribution and its approximation can be found at [Lindley68].

Chapter 6

Advanced Statistics

6.1 Gaussian Process

Gaussian process in regression: [Murphy, Section 15.1-15.2]

- Motivation: when learning a function f, regularize f s.t. if x_i and x_j are close, f_i and f_j should be close. Directly regularize a function is of course difficult, but we can consider the data points of f at $x_1, \cdots, x_N$. To predict is then roughly interpolate using given data points.
- Model: Figure 15.1, a PGM, let f_i be $f(x_i)$, and y_i be observed response at x_i. Our model is: $f_i|x_i$ is given by a Gaussian graphical model, that encourages similarity of f_i's when x_i's are close. And $y_i|f_i \sim N(f_i, \sigma_y^2)$, the noise. Formally, we write $p(f|X) = N(\mu, K)$ where μ is the mean function, and K is the covariance, given by the kernel, $\kappa(,\cdot,)$.
- The choice of mean function: commonly use $\mu(X) = 0$, then the model simply smooths the function f. If we use $\mu(X)$ as a linear function of X, then this is similar to linear regression, where the errors are correlated, with the error structure given by K.
- Inference and prediction: prediction is given by

$$p(y_*|y, X, x_*) = \int p(y_*|f, x_*)p(f|X, y)df \tag{6.1}$$

  We can show that the posterior mean of f_* is:

$$\bar{f_*} = \sum_{i=1}^{N} \alpha_i \kappa(x_i, x_*) \qquad \alpha = K_y^{-1} y \tag{6.2}$$

  where $K_y = K + \sigma_y^2 I$. So this is similar to nearest neighbor estimates, where the weight of a training data point i, depends on $\kappa(x_i, x_*)$.
- Impact of hyperparameters: e.g. squared exponential (SE) kernel. It has two parameters, l controls the horizontal scale (smoothness of function) and σ_f^2 controls the vertical scale of the function. See Figure 15.3.
- Estimation of kernel parameters: EB estimation. It is possible to margianlize f to compute $p(y|X) = \int p(y|f)p(f|X)df$. To do parameter estimations, use gradient based methods. Trade-offs between l and σ_f can lead to local optimum, see Figure 15.5: smooth function, but large prediction error; or wiggly function with small prediction error.

6.2 Spatial Statistics

Overview of spatial statistics: use disease count data as an example [Wakefield, Disease mapping and spatial regression with count data, Biostatistics, 2007]]

- Disease mapping problem: estimate the RR of each area.
- Spatial regression problem: identify the potential risk factors of disease.
- Why need a different approach: e.g. with disease mapping problem, the variance of the estimator is high for low density regions.
- General ideas: need to model the variation (spatial in this case) across regions, as well as, the dependency (correlation) between neighbors. The two are generally independent.

Nonspatial model for disease mapping [Wakefield, 2007]

- Poisson model with Gamma random effects: suppose the count in region i is Y_i, and E_i be the expected number (given). The difference of Y_i over E_i is RR, and it is due to fixed effects (known covariates x_i) and random effects (variation across regions not explained by fixed effects). Let μ_i be the fixed effect term and θ_i be the random effect, then:

$$Y_i \sim \text{Pois}(E_i \mu_i \theta_i) \tag{6.3}$$

  where $\mu_i = \mu(x_i, \beta)$ and

$$\theta_i \sim \text{Ga}(\alpha, \alpha) \tag{6.4}$$

  This prior is chosen s.t. the prior mean is 1. The marginal distribution of Y_i is Negative Binomial:

$$\text{E}\, Y_i = E_i \mu_i \qquad \text{Var}\, Y_i = \text{E}\, Y_i (1 + \text{E}\, Y_i / \alpha) \tag{6.5}$$

  Inference can be done with EB, which estimates the MLE of β and α. Then θ_i can be infered, as a weighted combination of Y_i and the prior.

- Poisson model with Gamma random effects (slightly different):

$$Y_i \sim \text{Pois}(E_i \theta_i) \qquad \theta_i \sim \text{Ga}(\mu_i \alpha, \alpha) \tag{6.6}$$

  where $\mu_i = f(x_i, \beta)$ is the fixed effect.

- Poisson model with lognormal random effects:

$$Y_i \sim \text{Pois}(E_i \mu_i e^{V_i}) \qquad V_i \sim N(0, \sigma_v^2) \tag{6.7}$$

  where V_i are area-specific random effects that capture the unexplained log RR in area i.

Spatial model for disease mapping: incorporate spatial dependence [Wakefield, 2007]

- Joint model: let μ_i be the fixed effect term and U_i, V_i be random effects (U_i non-spatial and V_i spatial), we have

$$Y_i \sim \text{Pois}(E_i \mu_i e^{U_i + V_i}) \tag{6.8}$$

  The fixed effect term has two components: one due to known covarates, the other due to large-scale spatial trend (S_i be the spatial location):

$$\log \mu_i = f(x_i, \beta) + g(S_i, \gamma) \tag{6.9}$$

  The non-spatial random effects: $V_i \sim N(0, \sigma_v^2)$. The spatial random effects, the vector U, can be modeled as a MVN where correlation depends on the distance d_{ij}:

$$\text{Var}\, U_i = \sigma_u^2 \qquad \text{corr}(U_i, U_j) = \rho^{d_{ij}} \tag{6.10}$$

  where ρ is a parameter that determines the extend of correlation.

- Conditional model: Intrinsic conditional autoregressive (ICAR) prior, under this model, the random effect U_i depends on its neighbors ∂_i:

$$U_i|U_j, j \in \partial_i \sim N(\bar{U}_i, \omega_u^2/m_i) \tag{6.11}$$

where m_i is the number of neighbors and $\bar{U}_i$ is the mean of the spatial random effects of neighbors.

Bayesian Multiscale Models for Poisson Processes [Kolaczyk, JASA, 1999]

- Model: number of events X_i in the i-th interval, with $X_i \sim \text{Pois}(\Lambda_i)$. Our goal is to estimate spatially smooth Λ_i. To do that, we partition the data into smaller intervals. At the top level, the total count is X_{00}, where the first index is for level, and the second for position. The distribution:

$$X_{00} \sim \text{Pois}(\Lambda_{00}) \tag{6.12}$$

Next, we consider two halves of the total interval, let the counts be X_{10} and X_{11}, respectively. Let $R_{10} = \Lambda_{10}/\Lambda_{00}$, the conditional distribution:

$$X_{10}|X_{00} \sim \text{Bin}(X_{00}, R_{10}) \tag{6.13}$$

and so on. So the total likelihood of all data can be written as the product of conditional distributions, and the model is parameterized by Λ_{00}, and R_{jk}'s. More formally, we have:

$$P(X|\Lambda) = P(X_{00}|\Lambda_{00}) \prod_j \prod_k P(X_{j+1,2k}|X_{jk}, R_{jk}) \tag{6.14}$$

- Prior of R_{jk}: the intuition of R_{jk} is that, to be spatially smooth, most of the time it should be 1/2. So we have this prior:

$$R_{jk} \sim \gamma_{jk} 0.5 + (1 - \gamma_{jk}) B_{jk} \tag{6.15}$$

where

$$\gamma_{jk} \sim \text{Ber}(p_j) \qquad B_{jk} \sim \text{Beta}(a_j, a_j) \tag{6.16}$$

- Model interpretation: When p_j is large, most of the time, R_{jk} is 0.5, thus we have equal rates. So p_j measures the spatial homogeneity. a_j on the other hand controls B_{jk}, corresponding to the "magnitude of effect": if a_j is large, then B_{jk} is close to 1/2; if a_j is small, then B_{jk} has large variation. In the case of peak detection, p_j controls the number and width of peaks, and a_j controls the variability of the magnitude of peaks. Ex. broader peaks captured by small p_j at higher level (more spatial heterogeneity) while shorter peaks represented by small p_j at lower level.

- Remark: the model favors smoothness, but it cannot learn/enforce certain "shapes" of peaks, e.g. it cannot capture the notion that in peak detection, we generally have peaks above the background, but not below. Even if a particular shape occur repeatedly, the model wouldn't capture that.

SMASH: multi-scale (multi-seq) Poisson model[Tom Xing, Sep, 2016]

- Poisson Model: let p_j be the probability of j-th interval in the multi-scale Poisson model [Kolaczyk 1999] (the conditional probability of binomial), under H_0, $p_j = 0.5$. In ASH, we model:

$$\log \frac{p_j}{1 - p_j} = \alpha_j + \beta_j x \tag{6.17}$$

where α captures spatial smoothness and β_j the effect of treatment x. Both α and β are defined at many scales, and we shrink α_j towards 0 at each scale. This model can be used for both treatment and control (differed by x). Note: does not have an explicit model to shrink more at higher spatial resolution, rather, estimate the parameters using EB.

- Use multi-seq to detect difference between samples: the idea is that we have a linear model of α_j, with treatment condition as a covariate. LRT to test if the coefficient of the covariate is 0.

- Remark: does treatment changes the overall shape, e.g. number and width of peaks or specific peaks? In the model, β_j is defined at every location j. So any a given scale, we may have log-OR equals to 0 in the background, but $= \beta_j$ in treatment. The parameter β_j would reflect the difference at the specific location j.

- Remark: the peak locations (consider the case of smoothing only, no covariate) may not match the scales we have defined. Ex. a peak may have 3/4 in the first interval and 1/4 in the second interval defined in the multi-scale model. How do we capture this? Or how do we make the results translational invariant? Idea: try all possible rotations of data. Naive algorithm $O(n^2)$ time; with smarter strategy $O(n \log n)$.

- Joint analysis of DNase-seq and ChIP-seq data: let D be DNase data and C be ChIP-seq data. We infer $P(D|C)$, the expected read count at each position of a TFBS (TF footprint).

Smoothing via Adaptive Shrinkage (smash): denoising Poisson and heteroskedastic Gaussian signals [Xing and Stephens, 2017]

- Normal model with known σ: suppose we have $Y = \mu + \epsilon$, where μ is mean of Y (spatially smooth) and $\epsilon \sim N(0, D)$, where D is the diagnoal matrix with entries $\sigma_1^2, \cdots, \sigma_T^2$. We transform the data using Discrete Wavelet Transform by multiplying a matrix W:

$$WY = W\mu + W\epsilon \tag{6.18}$$

  which we write as: $\tilde{Y} = \tilde{\mu} + \tilde{\epsilon}$, where $\tilde{\epsilon} \sim N(0, WDW^T)$. Note that $\tilde{\mu}$ now represent the coefficients of wavelet functions, and are assumed to be sparse. For simplicity, we assume $\tilde{Y}_j$'s are independent, and use ASH prior for $\tilde{\mu}$, and this can be fit with ASH.

- Additional assumptions: average results over all T rotations of data. Apply ASH to each level of wavelet coefficients.

- Normal model with known μ but unknown σ: we consider $Z_t^2 = (Y_t - \mu_t)^2$, then estimating σ is now a mean estimation problem.

- Normal model with unknown μ, σ: iterate, estimate μ assuming σ known; and estimate σ assuming μ known.

- Including covariates in the normal model: we simply consider the residual, and fit the normal model. Could also do this iteratively: from initial fit of spatially smooth model, estimate the residual again, and refit.

- Poisson model: the difference with the standard model is, we parameterize with $\alpha_j = \log p_j/(1 - p_j)$, which follows ASH prior at each level.

SMASH-GEN: extensions of SMASH [`https://dongyuexie.github.io/smash-gen/index.html`, 2018]

- Claim: if $X \sim \text{Pois}(\lambda)$, then we can approximate it as: $Y = \log(X)$, and $Y \sim N(\mu, \sigma^2)$ where $\mu = \log(\lambda)$ and $\sigma = 1/\lambda$. The problem of estimating λ can be then reduced to estimating μ under a normal distribution.

- Proof: The log-likelihood function of Poisson distribution is:

$$l(\mu) = \log P(x|\mu) = x \log \lambda - \lambda = x\mu - e^{\mu} \tag{6.19}$$

We can use Taylor expansion near μ_0 to approximate the LL:

$$l(\mu) \approx l(\mu_0) + l'(\mu_0)(\mu - \mu_0) + \frac{l''(\mu_0)}{2}(\mu - \mu_0)^2 \tag{6.20}$$

The derivatives are given by:

$$l(\mu_0) = x\mu_0 - e^{\mu_0} \qquad l'(\mu_0) = x - e^{\mu_0} \qquad l''(\mu_0) = -e^{\mu_0} \tag{6.21}$$

The MLE of μ is $\log(x)$, so we choose $\mu_0 = \log(x)$. Plug in this, we have:

$$l(\mu) \approx x\log(x) - x - \frac{x}{2}(\mu - \log(x))^2 \tag{6.22}$$

This is the log-likelihood of normal RV with mean $\log(x) = \mu$ and variance $1/x = 1/\lambda$.

- Intuitions of the algorithm: our analysis suggests that we can focus on fitting $\log(X_i)$, which is roughly normal. We can consider the Taylor approximation of $\log(X_i/\lambda_i)$:

$$\log\frac{X_i}{\lambda_i} = \log\left(1 + \frac{X_i - \lambda_i}{\lambda_i}\right) = \frac{X_i - \lambda_i}{\lambda_i} - \frac{1}{2}\left(\frac{X_i - \lambda_i}{\lambda_i}\right)^2 + \cdots \tag{6.23}$$

  This leads to the approximation:

$$\log(X_i) \approx \log\lambda_i + \frac{X_i - \lambda_i}{\lambda_i} \equiv Y_i. \tag{6.24}$$

  Now we can write $Y_i = \mu_i + \epsilon_i$, where μ_i is spatially smoothed mean, and ϵ_i the heteroscedastic variance: $1/\lambda_i$ (easy to prove). Y_i can be thought of as the 'normal component" of Poisson data. We will develop iterative algorithm to fit Y_i.

- Algorithm: start with some initial estimate of λ_i, we fit spatially smoothed model of $Y_i = \mu_i + \epsilon_i$ using SMASH (wavelet smoothing). Then we have new estimate of μ_i, and we update our definition of $Y_i = \mu_i + (X_i - \lambda_i)/\lambda_i$, and do wavelet smoothing again. Repeat this process until convergence.

- Remark:

  - The algorithm focuses on fitting Y_i instead of $\log(X_i)$. Intuitively, only Y_i represents the normal part of data, and can be approximated by normal wavelet method. The extra deviation of $\log(X_i)$ from Y_i is introduced by Poisson likelihood, and cannot be captured by normal approximation.
  - Behavior of the algorithm: suppose we plot Y_i against i, with the curve $\log(\lambda_i)$. Initially, the fitted results looks spiky because $\log\lambda_i$ are not smooth (which occurs in the definition of Y_i). The deviation of Y_i from $\log(\lambda_i)$ is roughly normal. As the algorithm proceeds, $\log(\lambda_i)$ curve now becomes smooth. The errors are still roughly normal with similar magnitude as before.

- Bayesian interpretation of the iterative algorithm: we should expand near the posterior mean of μ, so intuitively, we should use the spatially smoothed estimate. This gives the iterative procedure above.

- Does $X_i = 0$ create a problem? Y_i is still well-defined at $X_i = 0$. Intuitively, the normal approximation of Poisson log-likelihood remain valid even when λ is small (close to 0).

- Including covariates in the Poisson model: suppose the effects of covariates can be accounted for by a constant term t_i for data point i. Our model is then: $X_i \sim \text{Pois}(t_i\lambda_i)$. We can now write the approximation as:

$$\log(X_i) \approx \log t_i + \log\lambda_i + \frac{X_i - \lambda_i t_i}{\lambda_i t_i} \equiv Y_i. \tag{6.25}$$

  So we fit the model $Y_i = \log t_i + \mu_i + \epsilon_i$, where ϵ_i has variance $1/(\lambda_i t_i)$.

- Nugget effect: we define Y_i as:

$$Y_i = \log \lambda_i + \frac{X_i - \lambda_i}{\lambda_i} + N(0, \sigma^2) \tag{6.26}$$

  where the last term is the Nugget effect. This is important for RNA-seq data, large base level variations that are not captured by spatial effects.

- Remark: how does the model behave in the case where we have hotspots? Initially, suppose our λ_i is flat, then when we fit Y_i, in the hotspot region, Y_i is now large (because of large X_i). The model has to use a large λ_i (mean term) to accommodate, so in next run, it fits larger values of λ_i while striving for smoothness, which is achieved by wavelet (the coefficient of the basis function corresponding to the hotspot range would be non-zero).

- **Lesson**: to estimate parameters, we could transform the data (variable substitution), and work on the new data, whose distribution may be simpler. Ex. we approximate a distribution by normal using appropriate variable substitution.

6.2.1 Bayesian Disease Mapping: Hierarchical Modeling in Spatial Epidemiology

Section 5.1-5.2: Model of disease

- The case event (Poisson process) model: let $\lambda(s)$ be the continuous density at location s, defined over a region T. We have events at $s_1, \cdots, s_m$. Our likelihood is:

$$L(s_1, \cdots, s_m|\Psi) = \prod_{i=1}^{m} \lambda(s_i|\Psi) \exp(-\Lambda_T) \qquad \text{where } \Lambda_T = \int_T \lambda(u|\Psi)du \tag{6.27}$$

  The proof can be done by discretization of space: the Poisson probability for grids with events is proportional to $\lambda(s_i)\exp(-\lambda(s_i))$, and the probability for grids without events is proportion to $\exp(-\lambda(s))$.

- The conditional logistic model: if we have case-control data, with density $\lambda_1(s|\Psi)$ and $\lambda_0(s|\Psi)$. We consider the conditional distribution, and this leads to the Bernoulli distribution for $y_i \sim \text{Bern}(p_i)$, where

$$p_i = \frac{\lambda_1(s_i|\Psi_1)}{1 + \lambda_1(s_i|\Psi_1)} \tag{6.28}$$

- Poisson model for count data in small areas: $y_i \sim \text{Pois}(\mu_i)$, with $\mu_i = e_i\theta_i$, where e_i is the expected rate, and θ_i relative risk.

- Model specification: let η_i be the log expected rate under Poisson process model (or log relative risk under Poisson model). Typical model for η_i is: $\eta_i = x_i\beta + z_i$, where x_i are covariates and z_i random effects. Also non-linear models: e.g. dependency on distance, and polynomial function on coordinates.

Section 5.4. Correlated heterogeneity models

- The count model and let θ_i be the RR, we have:

$$\log(\theta_i) = x_i\beta + u_i + v_i \tag{6.29}$$

  where u_i and v_i are correlated and uncorrelated random effects. It is recommended to have both terms. The model is not identifiable, but the sum of u_i and v_i is.

- Conditional autoregressive (CAR) model: improper model (ICAR). The prior of u_i is given by:

$$p(u|r) \propto \frac{1}{r^{m/2}} \exp\left[-\frac{1}{2r}\sum_i \sum_{j\in\delta_i}(u_i - u_j)^2\right] \tag{6.30}$$

  where δ_i is the neighborhood of i, e.g. all neighbors within a distance. This prior would penalize the difference of neighbors. One can also show that this is the MRF with the weight matrix given by $1/r$ times a matrix with 1 in diagonal, and -1 in neighbors, and 0 elsewhere. The prior of v_i is given by: $v_i \sim N(0, \sigma^2)$, and are independent. The inference (MCMC) is made simpler by the conditional distribution of u_i:

$$u_i|u_{-i}, y, x, r, \sigma^2\beta \sim N(\bar{u}_i, r/n_{\delta_i}) \tag{6.31}$$

  where $\bar{u}_i$ is the average of neighbors, and n_{δ_i} is the number of neighbors.

- Proper CAR model: with ICAR model, the conditional expectation of u_i is exactly the same as neighbors. Instead, we can allow the conditional expectation as a regression function of neighbors. So the new model introduce a parameter ϕ: conditional expectation of u_i is ϕ times average of neighbors. See the book for the case with $x_i\beta$.

Section 5.6: model comparison and diagnosis

- Model comparison and evaluation: use posterior predictive loss (PPL). For data point i (test data), suppose $\hat{y}_{ij}$ is the j-th prediction of i from posterior sample, the PPL is defined as the average loss over posterior sample of prediction:

$$PPL_i = \frac{1}{G}\sum_j f(y_i, \hat{y}_{ij}) \tag{6.32}$$

  where $f(\cdot)$ is the loss function. One can inspect PPL over testing data to explore the model goodness-of-fit.

- Assessing spatial correlation (autocorrelation) of residuals: we want to test if there is any remaining spatial structure in the data. The most common auto-correlation statistic is Moran's I:

$$I = e^T W e / e^T e \tag{6.33}$$

  where e is the residual (standardized), and W the 0-1 adjacency matrix (diagonal 0). Note the definition is similar to correlation between two vectors. In the special case where only adajcent regions have $w_{ij} = 1$, and the rest 0, this is the correlation between adjacent regions. We can estimate it by fitting a regression:

$$e_i = a_0 + \rho e_i^* + \epsilon_i \tag{6.34}$$

  where $e_i^* = \sum_{j\neq i} w_{ij} e_j$.

Chapter 6: Disease cluster detection

- Definitions of clusters: (1) hot-spots: could be on single areas; (2) pre-defined groups, and (3) clusters defined with residuals.

- Residuals: under conditional logistic model, let y_i be count (0 or 1) and p_i be the model prediction, then the residual can be defined as:

$$r_i = \frac{y_i - \hat{p_i}}{\sqrt{\hat{p_i}(1-\hat{p_i})}} \tag{6.35}$$

  Under the Poisson count models, suppose our model is $y_i \sim \text{Pois}(e_i\theta_i)$, and we estimate θ_i using the model, the residual defined as:

$$r_i = \frac{y_i - e_i\hat{\theta_i}}{\sqrt{e_i\hat{\theta_i}}} \tag{6.36}$$

The idea of Bayesian residuals: we obtain posterior sample of parameters, and hence $\hat{p}_i$ or $\hat{\theta}_i$, so we have a posterior sample of r_i. This allows us to define $P(r_i > c)$ for some threshold c.

- Cluster models: suppose we believe the data has K clusters in terms of RRs. For each cluster, it generates some RR in its neighborhood. For Poisson count data, $y_i \sim \text{Pois}(e_i\theta_i)$, our model becomes:

$$\log \theta_i = \alpha_0 + \alpha_1 \sum_{j=1}^{K} \phi_j h(C_i - c_j; \tau_h) \tag{6.37}$$

  where ϕ_j is the RR of cluster j (modeled as random effects), and $h(\cdot)$ captures the influence of clusters. C_i are positions of i, and c_j the cluster centroid. Two special cases: $h(\cdot)$ is normal, or is constant when a point is within a neighborhood of c_j and 0 otherwise.

- Partition models and tree models: all data can be partitioned, and within a partition, RRs are constant, and modeled as random effects. The partition model can be extended to tree models.

6.3 Functional Analysis and Networks

Bayesian Inference and Testing of Group Differences in Brain Networks [Durante & Dunson, arXiv, 2015]

- Problem: suppose we have brain connectivity data of individuals and we also have phenotypic variable (e.g. creatitivity) for individuals. Find the difference of brain network between the two groups (high vs. low creativitiy). The connectivity data can be represented as networks of V nodes ($V = 68$), A_i, and the phenotype data y_i.

- Existings methods:

  - Test individual edges: whether the edge presence correlates with y. Limitation: need multiple testing correction, not taken into account the network dependency structure, thus lose power.
  - Summary statistics of graphs, and correlation with phenotype: e.g. degree, other topological measures. Limitation: lose information of specific regions.

- Intuition: the network edges are highly correlated: for some sets of edges (regions), all edges are either 0 or 1 simultaneously in one individuals. This motivates the use of factor analysis: relatively few latent variables explain the covariance pattern of many variables (edge occurrence).

- Model: directly model the network-valued variable A_i, conditioned on y_i. The naive representation of A has $2^{V(V-1)/2}$ variables (number of possible graphs). Propose a mixture of low-rank factorizations: each component of the mixture specifies the edge probability π_l for edge $l = 1, \cdots, V(V-1)/2$. To model the h-th component, $\pi^{(h)}$, assume it is generated from $S^{(h)}$ (logistic regression model), which has a low-rank factorization.

- **Lesson**: whenever we have many random variables, some of which may be correlated, we can use latent variable model (factor analysis).

Bayesian Functional Quantile regression (FQR) [Yusha Liu from Jeff Morris group]

- Problem: MS data, comparison of cancer vs. normal.

- Motivation for quantile regression: two distributions may have similar mean, but differ in other quantiles (e.g. top quantile). Do quantile regression $Y = X\beta^\tau + \epsilon^\tau$ where β is defined as the effect on a certain quantile. Typically we do this for different τ's (instead of running on multiple quantiles in parallel).

- Basic model: model the entire dataset, response at t

$$Y(t) = XB^{\tau}(t) + E^{\tau}(t) \tag{6.38}$$

  where X is covariates (cancer or normal), $B^{\tau}(t)$ is the effect ot treatment on the quantile τ at point t. Error $E^{\tau}(t)$: asympotic Laplacian (AL) distribution. For simplicity, drop τ in the notations.

- Dealing with spatial continuity of effects: the effects of B of a covariate should be spatially correlated. Model $B(t)$ as a wavelet. Let B^*_{ajh} be the coefficient of covariate a, of scale j and specific wavelet h. Use global-local prior of wavelet coefficients to shrink most to 0 and borrow information across the wavelets of the same scale

$$B^*_{ajh} \sim N(0, \lambda^2_{ajh}\psi^2_{aj}) \tag{6.39}$$

  where λ_{ajh} follows g_1 prior (global), e.g. Laplace prior, and ψ_{aj} follows $g_2(\Psi_{aj})$ prior (local).

- Discussion: how to do predictions if X differs between groups of Y only in quantile, but not mean?

- Remark: in the error model, still independent across ts.

- Lesson: quantile regression can capture the effects that only change extreme quantiles but not mean. The challenge is to model error distribution.

- **Lesson**: Global-local prior can allow one to borrow information across groups of variables. Similar to group-Lasso?

- **Lesson**: modeling of functional data, to capture spatial continuous effects using wavelet transform. Write the true effects as sums of wavelets and learn about the wavelet coefficients.

6.4 Misc. Methods

Nonparametric methods:

- What is nonparametric methods? Defined by the lack of parametric models of the underlying process. The result would not depend on the parametric distribution (thus distribution-free), and the test can be applied to some general statement of the population without using parameters, e.g. the trend, the randomness, etc. (nonparametric test).

- Advantages of nonparametric methods: robustness to the underlying distribution, usually very generally and applicable. The computation and null distribution may often be simple as well.

- Comparing tests: robustness is not a main criterion, but power is. However, it may be difficult to compare nonparametric tests because the power would generally depend on the alternative hypothesis, which is not known (the reason why nonparametric test is used in the first place).

1. Modeling extra evidence in ranking and prediction with latent variables

   Problem: some examples of a common/generic problem:

   - Predict regulatory sequences from expression pattern: suppose we want to predict or rank sequences, S_i, that predict the expression of gene G_j, denoted as y_j. The model of how S_i is releated to y_j is available, and now we want to incorporate extra evidence of the sequences, x_i, which could be conservation, distanct to TSS, histone modification, etc.
   - Ranking pages to some query: the goal is to rank pages, P_i, that are relevant to query Q_j. The relevance function is defined according to the content, and now we want to incorporate evidence of pages, such as page importance (link structure), time, etc.

   Model:

- Probabilistic model: take the example of regulatory sequence prediction, let Z_{ij} be the indicator variable of whether S_i regulates G_j. Our model is: for any given gene G_j: first sample Z_{ij} in all S_i's; and for the chosen sequence, generate y_j from S_i. Without extra evidence, the prior probability $P(Z_{ij})$ is uniform; with extra evidence, we model Z_{ij} via logistic regression of x_i, where the regression coefficients would favor sequences that are more conserved, close to TSS, etc.
- Inferring hidden indicator variables: the posterior probability of a sequence:

$$P(Z_{ij} = 1|x_i, y_j) \propto P(Z_{ij} = 1|x_i)P(y_j|Z_{ij} = 1) \tag{6.40}$$

  where the first term is the prior probability and the second the evidence of S_i. Take log. of the equation, and we note that the final score is the sum of the prior evidence and the LL score. If we treat the LL score as one feature of S_i, this is similar to a classification problem based on all features (the log. of prior is approximately a linear function, if prior is logistic regression). The difference here is: inference could be done without any training data.
- Extensions: the distribution of Z_{ij} can further include the properties of G_j or Q_j. Ex. for certain categories of queries, certain features of pages are generally important.

Alternative model based on prior distribution of parameters:

- Modeling prior: let β_{ij} be the influence of S_i on G_j, the extra evidence can then be modeled as the distribution of β_{ij}, which should be higher with extra favorable evidence, e.g. through a linear regression on features representing the extra evidence.
- Comparison with latent variable approach: in the latern variable approach, the assumption is there is only one that has true influence, which needs to be determined; and in prior modeling approach, each could have influence, while the extent of influence may vary. Ex. in finding causal variation in GWAS, the latent variable approach may be more appropriate [Veyrieras & Pritchard, PG, 2008]; while in finding the possible regulators of expression of many genes, the prior modeling approach may be better [Lee & Koller, Lirnet, PG, 2009].

Remark:

- The difficulty of this type of problems is the extra evidence cannot be easily modeled/connected to the data. The idea here is to introduce latent variables (Z_{ij}), and the extra evidence is incorporated via influencing Z_{ij}.
- Latent variable modeling: a general strategy of probabilistic modeling. The usual models such as regression, stochastic models, etc. can be applied to the latent variables. One special case: the labels of data (classes) are missing.

2. Time series analysis

   Autocorrelation and cross-correlation [Modern Applied Statistics with S, Section 14.1]:

   - Aim: assess the correlation of two time series, allowing time-lag between the two.
   - Autocorrelation: we start with the case of correlation within a time series. Suppose we have a series $X(\tau)$, where $\tau = 1, 2, \ldots, n$, we want to see the correlation of this series with $X(\tau + t)$:

$$\gamma_X(t) = \text{Cov}(X(\tau + t), X(\tau)) \tag{6.41}$$

$$\rho_X(t) = \text{Corr}(X(\tau + t), X(\tau)) \tag{6.42}$$

     Assuming the signal is second-order stationarity, i.e. both quantities do not depend on τ, we have the same mean for $X(\tau)$ and $X(\tau + t)$, let it be $\bar{X}$. We have the estimator for $\gamma_X(t)$:

$$c_X(t) = \frac{1}{n}\sum_s (X(s + t) - \bar{X})(X(s) - \bar{X}) \tag{6.43}$$

where s takes the range $1, \ldots, n-t$ if $t > 0$ and $1-t, \ldots, n$ if $t < 0$. We note that, $\mathrm{Var}(X)$ is exactly $c_X(0)$ (auto-covarance), so the estimator of $\rho_X(t)$ is given by:

$$r_X(t) = \frac{c_X(t)}{c_X(0)} \tag{6.44}$$

- Cross-correlation: when we have two time series $X(\tau)$ and $Y(\tau)$ observed on the same interval, we could do similar analysis. The estimators for covarance and correlation are:

$$c_{XY}(t) = \frac{1}{n}\sum_s (X(s+t) - \bar{X})(Y(s) - \bar{Y}) \tag{6.45}$$

$$r_{XY}(t) = \frac{c_{XY}(t)}{\sqrt{c_X(0) \cdot c_Y(0)}} \tag{6.46}$$

- Remark: the implementation in R follows the above definitions. However, the common/mathematical definition is slightly different (which assumes mean is 0), see Wiki.

3. Sequential analysis

   Reference: `http://www.hsph.harvard.edu/betensky/bio276.html`

   Sequential testing ideas: [Wiki, Sequential probability ratio test]

   - Model: suppose we are testing $H_0 : \theta = \theta_0$ vs. $H_1 : \theta = \theta_1$. Let S_n be the log-likelihood ratio at data points from 1 to n. The stopping rule is that: $S_n < a$ or $S_n > b$. To determine the boundary, a and b should be chosen s.t. the type I and II error are satisfied, i.e.

$$P(S_n < a \text{ or } S_n > b|\theta_0) \leq \alpha \quad P(S_n < a \text{ or } S_n > b|\theta_1) \leq \beta \tag{6.47}$$

   Brownian motion in sequential testing:

   - Brownian motion: $W(t)$ follows the properties: (1) $W(0) = 0$. (2) $W(t) - W(s)$ follows normal distribution $N(\mu(t-s), \sigma^2(t-s)$. (3) For any t_i and s_i, $W(t_i) - W(s_i)$ are independent. It is easy to prove that:
     - The joint distribution of $W(t_1), \cdots, W(t_n)$ is Gaussian.
     - $W(t)$ is Gaussian with mean μt and variance $\sigma^2 t$.
     - The covariance between $W(t_i)$ and $W(t_j)$ is $\sigma^2 t_i$ (assuming t_i is small - the shared time between the two random variable). To see this, we write $W(t_j) = W(t_i) + (W(t_j) - W(t_i))$, where the two terms are independent, so the covariance is simply the variance of $W(t_i)$.
   - Sequential testing by Brownian motion: example, suppose we have $X_1, \cdots, X_n$ iid. $N(\mu, 1)$ and we are testing if $\mu = 0$. The test will be based on the partial sum: $S_n = \sum_i X_i$. Then S_n is a random walk (discrete), and the contiuous relaxation leads to Brownian motion.

4. Hypothesis testing of correlations

   Pearson's correlation coefficient [Correlation, Wiki]:

   - Pearson's correlation coefficient: defined for two random variables X and Y, as:

$$\rho_{X,Y} = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y} \tag{6.48}$$

     When X and Y are jointly normal, they are independent if and only if correlation equals zero. But if not normal, this is not true: independent then $\rho = 0$, but $\rho = 0$ does not necessarily mean they are independent. The sample correlation is computed by:

$$r_{xy} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y} \tag{6.49}$$

where $\bar{x}$ and $\bar{y}$ are sample means, and s_x, s_y are sample standard deviations. Thus r_{xy} is also sample covarance divided by the product of sample deviations.

- Interpretation of Pearson's correlation coefficient:
  - Geometric interpretation: if the vectors of samples are standardized (shifted by sample mean), then it is the cosine of the angle between two vectors.
  - Linear regression: do a linear regression of y on x, then the coefficient is rs_y/s_x. Furthermore, the coefficient of determination (the variance explained) is the square of correlation coefficient.
- Application of Pearson's correlation coefficient: it measures the strength of a linear relationship between two variables that are normal. May not work well if the assumptions are not held, for example, Anscombe's quartet. In particular, one outlier may be enough to produce a high correlation coefficient, even though the relationship between the two variables is not linear.

Non-parameteric correlations [Correlation, Wiki]:

- Spearman's correlation coefficient: a special case of the Pearson product-moment coefficient in which two sets of data X_i and Y_i are converted to rankings x_i and y_i before calculating the coefficient. Significance: best by permutation test.
- Mutual information: Formally, the mutual information of two discrete random variables X and Y can be defined as:
$$I(X;Y) = \sum_{x \in X, y \in Y} p(x,y) \log \frac{p(x,y)}{p_1(x)p_2(y)} \tag{6.50}$$
where $p_1(x)$ and $p_2(y)$ are marginal distributions. Normally base 2 is chosen for log function. Mutual information is a measure of dependence in the following sense: $I(X;Y) = 0$ if and only if X and Y are independent random variables.

Hypothesis testing of correlation coefficients:

- Significance of correlation coefficient by Fisher transformation: let N be the sample size, and r be the sample correlation coefficient, define the transformation:
$$z = \frac{1}{2} \log \frac{1+r}{1-r} \tag{6.51}$$
If (X, Y) has a bivariate normal distribution, then z is approximately normally distributed with mean $\frac{1}{2} \log \frac{1+\rho}{1-\rho}$, and standard deviation: $\frac{1}{\sqrt{N-3}}$. This could be used for constructing confidence interval for ρ.
- Difference between two correlation coefficients from independent samples: let r_1 and r_2 be correlation coefficients of two independent samples, first convert them to z_1 and z_2 respectively with Fisher transformation, then the statistic is $z_1 - z_2$, and its standard error is
$$\sigma_{z_1 - z_2} = \sqrt{\frac{1}{N_1 - 3} + \frac{1}{N_2 - 3}} \tag{6.52}$$
- Difference between two correlation coefficients from dependent samples: suppose we want to test if r_{XY} is significantly different from r_{ZY}. Let n be the number of points, compute:
$$t = (r_{XY} - r_{ZY}) \cdot \sqrt{\frac{(n-3)(1+r_{XZ})}{2(1 - r_{XY}^2 - r_{XZ}^2 - r_{ZY}^2 + 2r_{XY}r_{XZ}r_{ZY})}} \tag{6.53}$$
Then t should be t distribution with degree of freedom $n - 3$.
- Reference:

– Google “Confidence Interval on Pearson’s Correlation” and “Confidence Interval, Difference between Independent Correlations”.
– Cohen & Cohen, Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences, 2003, Section 2.8, and 5.1.
– `http://talkstats.com/showthread.php?t=9011` Or, Blalock, H., 1972. Social Statistics. NY: McGraw-Hill. Page 406-7
– Google “How to compare sample correlation coefficients drawn from the same sample”

Chapter 7

Machine Learning

7.1 Introduction to Statistical Machine Learning

Reference: [Bishop, Pattern Recognition and Machine Learning, Chap. 1], [Hastie, Elements of Statistical Learning, Section 2.3], [Murphy, Chapter 1]

Challenges of learning:

- Model over-fitting: complex models produce very small training error, but have poor generalization performance.
  - Example, in polynomial curve fitting, high order of polynomial leads to overfitting. Intuitively, the more flexible polynomials with larger values of M (order) are becoming increasingly tuned to the random noise on the target values.
  - Maximum-likelihood method for parameter estimation suffers from over-fitting.
- Curse of dimensionaility: at high dimension, the neighbors of any point must be sparse, thus it is theoretically difficult to apply local regression/smoothing.

Paradigms of supervised learning:

- Problem: predict response Y from X, given data (x_i, y_i), where $1 \leq i \leq n$, and the dimensionality of X is p.
- Model-based learning: fit a global model that explains the training data, e.g. linear models with least square fitting.
- Instance-based learning: the prediction on x is determined by the neighbors of x in the training data, e.g. k-nearest neighbor method (KNN):

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \tag{7.1}$$

  where $N_k(x)$ is the neighborhood of x defined by k nearest points of x in the training sample.

Extending simple methods for supervised learning:

- Kernel methods that uses weights that decrease smoothly to zero with the distance to the target point, rather than 0/1 as in nearest neighbor method (discrete selection tends to lead to high variance).
- Local regression that fits linear models by locally weighted least squares.
- Basis expansion of the linear methods.

Unsupervised learning: problems and ideas

- Statistical perspective: we are given the output data only, but not input data (alternatively, only data, but not labels). The goal is to estimate the density, $p(x|\theta)$, from the data of x. This is in contrast to the supervised learning problem, where the goal is to estimate $p(y|x,\theta)$. The challenge is when x has a large number of dimensions,and there is no simple parameteric form of the distribution of x, how to estimate the density.
- Clustering: the simplest structure to take advange of is that some data points form clusters. Ex. in the height-weight data, there is a natural structure: two genders (clusters).
- Discovering latent factors: more generally, to impose structure on the data, one assumes that the data is generated from some latent variables, and in the space of the latent variables, the data points fall in a low-dim. space (dim. reduction).
- Graph structure of the variables: the dependence and/or correlation between variables, or causal models.
- Imputation: the goal is to infer the missing data from given data. Ex. collaborative filtering - missing matrix.

Bayesian methods and roughness penalty: penalize the complex models (regularization) [Hastie, Elements of Statistical Learning, Section 2.7-2.8]

- In general, the function f is chosen to minimize:

$$PRSS(f;\lambda) = RSS(f) + \lambda J(f) \tag{7.2}$$

  where $RSS(f)$ is the residum sum of square of f in the training data, and $J(f)$ penalizes functions that change rapidly over small regions. For example, the cubic smoothing spline:

$$J(f) = \int [f''(x)]^2 dx \tag{7.3}$$

- These methods can be cast in a Bayesian framework, where the penalty J correponds to log-prior, and minimizing $PRSS$ amounts to finding the posterior mode.

Kernel methods and local regression: e.g. Nadaraya-Watson kernel regression method predicts y of a point x_0 by averaging y_i's weighted by the distance of x_i to x_0:

$$\hat{f}(x_0) = \frac{\sum_i K_\lambda(x_0, x_i) y_i}{\sum_i K_\lambda(x_0, x_i)} \tag{7.4}$$

where λ controls the size of the neighborhood.

Cross validation [Hastie, Section 7.10]:

- Cross validation: divide the data into K equal parts. For the k-th part, train the model on the rest $K-1$ parts and calculate the prediction error of the fitted model when predicting the k-th part of the data. We combine the K estimates of the prediction error.
- Cross validation for model selection: given a set of models $f(x,\alpha)$ indexed by a tuning parameter α, we have $CV(\alpha)$ as the estimates of the prediction error of the model $f(x,\alpha)$. We should choosen α that minimizes $CV(\alpha)$, and the final model is $f(x,\hat{\alpha})$.

7.1.1 Assessing Estimators and Statistical Decision Theory

Ref: [Hastie, Elements of Statistical Learning, Section 2.4]

Assessing estimators:

- Mean squared error (MSE): suppose W is an estimator of $\tau(\theta)$, where θ stands for model parameter(s). Then W can be assessed by its MSE:

$$\text{MSE(W)} = E(W - \tau(\theta))^2 \tag{7.5}$$

  Note the frequentist interpretation: the expectation is averaged over all possible datasets generated from the probability distribution.

- Unbiased estimator: often look for unbiased estimator of θ, i.e. $E(\hat{\theta}) = \theta$. However, $\tau(\hat{\theta})$ is not necessarily an unbiased estimator of $\tau(\theta)$:

$$E(\tau\theta) \neq \tau(E\hat{\theta}) = \tau(\theta) \tag{7.6}$$

- Prediction problem: to predict the value of a function for a given x_0: $\hat{y} = f(x_0; \hat{\theta})$, one can view this as estimating a function of parameters. Ex. in linear regression problem, estimating $\mathbf{a}^T\theta$, this is prediction for a new data point $\mathbf{x_0} = \mathbf{a}$.

Loss function approach to regression:

- Loss function criterion: the function should be chosen to minimize the expected loss or expected prediction error (EPE) over the joint distribution of (X, Y), $p(x, y)$. For regression problem, the loss function is often chosen as squared error loss: $L(Y, f(X) = (Y - f(X))^2$. This leads to the criterion for choosing f (suppose $\hat{f}$ is a predictor function):

$$\text{EPE}(\hat{f}) = E(Y - \hat{f}(X))^2 = \int (y - \hat{f}(x))^2 p(x, y) dx dy \tag{7.7}$$

  Note that in this equation, $\hat{f}$ is a deterministic function; if it is estimated from data D, then the EPE of $\hat{f}$ also needs to be averaged over D (see bias-variance decomposition below).

- Optimal pointwise predictor for squared loss [Murphy, Section 5.7]: at a given x, one can show that with squared loss, the optimal predictor of x is:

$$\hat{f}(x) = E(Y|X = x) \tag{7.8}$$

  assuming that the true distribution is known. To see this, consider $L(y, a) = (y - a)^2$ where y is observed response and a predicted. The posterior expected loss is given by:

$$\rho(a|x) = \text{E}\left[(y - a)^2|x\right] = \text{E}\left(y^2|x\right) - 2a\text{E}\left(y|x\right) + a^2 \tag{7.9}$$

  The optimal estimator is thus:

$$\frac{\partial \rho(a|x)}{\partial a} = -2\text{E}\left(y|x\right) + 2a = 0 \tag{7.10}$$

  Solving this leads to $\hat{y} = \text{E}\left(y|x\right)$.

- Remark:
  - The difficulty of applying this theorem is: the data may not contain the value of X to be predicted, thus need assumptions about how the information of related points can be used to make inference for an unseen X.
  - Some methods are motivated by direct approximation of conditional mean, e.g. the KNN method.

- Optimal predictor for L_1 (absolute) loss: we have $L(y, a) = |y - a|$. The optimal predictor is the median of $p(y|X)$.

Loss function approach to classification [Murphy, Section 5.7]:

- Zero-one loss function: the loss function is $L(y, a) = 1$ if there is a mis-classification and 0 otherwise. This function can be generalized to include different costs for FPs and FPs. The optimal pointwise prediction is given by maximizing the posterior probability of the class label:

$$\hat{y}(x) = \text{argmax}_g P(g|X = x) \tag{7.11}$$

  Proof: given $X = x$, the problem is then to find the best point approximation of the conditional distribution $P(Y|X = x)$, and this is given by the Theorem of Point Approximation of Discrete RV.

- Surrogate loss function - *logloss*: 0-1 loss function is not smooth, so difficult for optimizers to work. In practice, log-loss function may be preferred. We consider binary logistic regression $y_i \in \{-1, 1\}$. Our *decision function* is defined as the log odds ratio:

$$f(x_i) = \log \frac{P(y_i = 1|x_i)}{P(y_i = -1|x_i)} \tag{7.12}$$

  The optimizer will then minimize the log-loss function:

$$L(y, f(x)) = -\log P(y|x) = \log(1 + e^{-yf(x)}) \tag{7.13}$$

  where y is true label (1 or -1) and $f(x)$ the decision function at input x. In other words, we should choose the parameters of $f(x)$ s.t. it has high probability of predicting/generating observed labels y. The optimal pointwise predictor (population minimizer) is given by:

$$\hat{f}(x_i) = \frac{1}{2} \log \frac{\pi_i}{1 - \pi_i} \tag{7.14}$$

  where $\pi_i = P(y_i = 1|x_i)$.

- See Table 16.1 [Murphy] for a list of loss functions and "population minimizers".

Remarks of loss function:

- Selection of loss/error function: this should depend on the characteristics of problems. The considerations: weighting of different data points (e.g. positive or negative examples may be weighed differently), the sensitivity to outliers (exponential or L_2 error are more sensitive to outliers than L_1 error), etc.

- Loss function approach: a general way of expressing an inference/decision problem. For inference problem: loss may be interpreted as the departure from the truth; for decision problem: loss as the consequence if wrong decision is made. Ex. for a clustering problem where the goal is to learn class label assignment, π, it may be reasonable to define the loss function (use K points to approximate all data points) as:

$$L(\pi) = \sum_k \sum_{x_i \in C_k} (x_i - \mu_k)^2 \tag{7.15}$$

- More applications of loss function approach:
  - Parameter estimation problem: viewed as given $X_1, \cdots, X_n$, predict X_i using the expectation of X (dependent on parameter). Ex. $X_1, \cdots, X_n$ iid. Bernoulli with probability p, define loss function as $(x_i - p)^2$ (prediction for any data point is p), then minimize the loss gives the estimator $\hat{p} = \sum_i x_i/n$.
  - Dimensionality reduction problem: essentially prediction of high dimensional values using low-dimensional projections, thus could define a loss function and perform minimization.

7.1.2 Model selection and Bias-Variance Tradeoff

Ref: [Hastie, Elements of Statistical Learning, Section 2.9; Bishop, Pattern Recognition and Machine Learning, Section 1.5]

Bias-variance decomposition of estimators: suppose W is an estimator of $\tau(\theta)$, the MSE of W can be decomposed as:

$$E(W - \tau(\theta))^2 = [EW - \tau(\theta)]^2 + \text{Var}(W) \tag{7.16}$$

The first term is the square of bias, and the second variance of W.

Bias-variance decomposition of predictor function:

- Pointwise prediction: given the function $\hat{f}$, and a point x, and the optimal predictor function is $h(x) = \text{E}\,(Y|X = x)$, the EPE of $\hat{f}$ is given by the Theorem of Point Approximation, applied to the distribution $Y|x$:

$$\text{EPE}(\hat{f}|x) = \text{E}\,[Y|x - \hat{f}(x)]^2 = [\hat{f}(x) - h(x)]^2 + \int [y - h(x)]^2 p(x, y) dy \tag{7.17}$$

  The second term does not depend on the choice of $\hat{f}$.

- Averaging over data: since $\hat{f}$ is estimated from data (instead of a deterministic function), EPE should also be averaged over possible data D, thus we write $\hat{f}$ as $\hat{f}_D$. We only need to consider the first term, averaging over D:

$$\text{E}_D[\hat{f}_D(x) - h(x)]^2 = [\text{E}_D(\hat{f}_D(x)) - h(x)]^2 + \text{E}_D[\hat{f}_D(x) - \text{E}_D(\hat{f}_D(x))]^2 \tag{7.18}$$

  by applying the bias-variance decomposition of estimator ($\hat{f}_D(x)$ is an estimator of $h(x)$).

- Now we average over x, and obtain: the full decomposition of the EPE [Bishop, Section 3.2]:

$$\text{EPE}(\hat{f}) = (\text{bias})^2 + \text{variance} + \text{noise} \tag{7.19}$$

  where:

$$(\text{bias})^2 = \int [\text{E}_D(\hat{f}_D(x)) - h(x)]^2 p(x) dx \tag{7.20}$$

$$\text{variance} = \int \text{E}_D[\hat{f}_D(x) - \text{E}_D(\hat{f}_D(x))]^2 p(x) dx \tag{7.21}$$

$$\text{noise} = \int [h(x) - y]^2 p(x, y) dx dy \tag{7.22}$$

  The bias term depends on the truth ($h(x)$), and the variance term is determined by the property of the $\hat{f}_D$, and the noise term is the intrinsic noise in the data (not dependent on the choice of $\hat{f}_D$.

Bias-variance tradeoff:

- Simple vs complex models: consider the pointwise prediction $\hat{f}(x_0)$, and we want to analyze its EPE (the case of parameter estimation is similar):
  - Simple models: the expected prediction will be relatively distant from the true values because simpler models just cannot capture the data, thus high bias.
  - Complex models: the models are excessively tuned for the noises in the training data (thus different for each new data point), thus high variance. Another way of seeing this is: complex models have more parameters, with each parameter contributing to the total variance.

- Bias-variance decompoisition provides a way to analyze model performance. When designing an algorithm, analyze how model performance (bias and variance) depends on the algorithm parameters, e.g. the number of parameters, data processing procedure (such as discretization), etc. In general, choose the complexity parameter to balance bias and variance to minimize EPE. Alternatively, we could say, choose simpler models to reduce variance, without sacrificing too much bias.
- Complexity parameters: often through cross validation on training data. Choose the complexity parameter that minimizes the EPE.

Example: suppose data are generated from $Y = f(X) + \epsilon$, with $\text{Var}(\epsilon) = \sigma^2$. For the KNN method, the EPE at x_0 is:

$$\text{EPE}_k(x_0) = E[(Y - \hat{f}_k(x_0))^2 | X = x] = \sigma^2 + [f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f(x_{(l)})]^2 + \frac{\sigma^2}{k} \tag{7.23}$$

where (l) indicates the indices of points in the neighborhood of x_0. Then:

- When k is small (small neighborhood, more irregular function, thus more complex): only points near x_0 are used, thus small bias; but large variance.
- When k is large (large neighborhood, more regular function, thus simpler): distant points of x_0 are used, thus large bias; but small variance.

Example: linear regression

- Model variance: we are interested in the variance of the prediction at a point x:

$$\text{Var}(x^T\hat{\beta}) = \text{Var}(x_1\hat{\beta}_1 + \cdots + x_p\hat{\beta}_p) \tag{7.24}$$

  The variance of $\hat{\beta}$ is given by the normal distribution:

$$\hat{\beta} \sim N(\beta, (X^TX)^{-1}\sigma^2) \tag{7.25}$$

  Note that in this distribution, X^TX is fixed (conditioned on X), but in our analysis of the predictor variance, we should consider the variance of X as well (this is what leads to the fine-tuning of model to specific data). Assume features are normalized and independent, then X^TX, the covariance matrix of features, is diagonal, and thus its inverse is also diagonal:

$$(X^TX)^{-1} = \text{diag}(1/\sigma_1^2, \cdots, 1/\sigma_p^2) \tag{7.26}$$

  Therefore, $\hat{\beta}_j$ are independent, we have:

$$\text{Var}(x^T\hat{\beta}) = \sum_j x_j^2 \text{Var}(\hat{\beta}_j) = \sum_j x_j^2\sigma^2/\sigma_j^2 \tag{7.27}$$

- Model preference: the total variance of the predictor is thus the sum of variance of each feature, thus reducing the number of features (simpler model) would lead to lower variance (Lasso regression). Furthermore, among all features, those with large variance should be preferred (imagine a low variance feature would lead to instability of parameter estimation).

7.1.3 Basis Expansion

Ref: [Hastie, ESL, 2.6.3, 2.8.3; Bishop, 3.1]

Basis functions: e.g. polynormial fitting can be understood as using basis functions $x, x^2, x^3, \ldots$; and trignometric functions are used as basis functions for Fourier analysis.

Basis expansion: many methods involve basis functions, upon which more complex models are constructed. A natural idea would be to extend with more basis functions.

- Simple basis expansion: e.g. in linear regression, replace, x_j with higher order terms such as x_j^2, and $x_i x_j$.
- Adaptive basis expansion: e.g. spline methods for curve fitting, the basis functions are polynomials (and the final function is piecewise polynomial). Or another example, radias basis function:

$$f_\theta(x) = \sum_m \theta_m K_{\lambda_m}(\mu_m, x) \tag{7.28}$$

  where $K_{\lambda_m}(\mu_m, x)$ may be a Guassian kernel. In these cases, the basis functions contain additional parameters that need to be learned from data (thus the basis functions are "adaptive").

Common basis functions:

- Application in linear models:

$$y = \beta_0 + \sum_j \beta_j \phi_j(x) + \epsilon \tag{7.29}$$

  where $\phi_j(x)$ is a basis function. Since the model is still linear to β_j, all the usual results of linear model still apply.
- Splines: piecewise polynomials, where within each region (defined by "knots"), the basis function is a polynomial. A spline is order M if it has continuous derivative of order $M-2$.
- Gaussian basis functions: suppose μ_j is a point in $\mathbb{R}^p$, want to define a function that is large when close to μ_j, and small when distant, and the decay follows a Gaussian function:

$$\phi_j(x) = \exp\left[-\frac{(x-\mu_j)^2}{2s^2}\right] \tag{7.30}$$

- Other basis functions: sigmoidal function, Fourier series and wavelets (similar to Fourier, but localized in both space and frequency).

Remarks: basis expansion and feature expansion - introduce additional features that may be more predictive of response variable.

- Mathematical expansion: e.g. replace linear term with other nonlinear function, either parametric or nonparameteric (e.g. generalize additive model).
- Composite features: defined on basic features, e.g. interaction terms (model combinations), objects in image analysis, topics in text analysis. The features may be latent variables: the model may take this into account, but the idea is similar.
- Strutured predictors: e.g. X is time-series expression data, then it could be expressed as a combination of "basic" profiles (expression in the beginning, in the middle, in the end, etc.), then these "basic" profiles can be used as features. The idea is related to semi-supervised learning, where unsupervised data is used.

7.2 Partition-based Methods and Model Averaging

Ideas and questions [personal notes]

- Questions: how to control the model complexity of boosting methods, the number of weak learners m?

Motivations/ideas [Hastie, ESL]:

- Partition-based methods: a single global model for the entire data space is not realistic in most cases; on the other hand, the instance-based methods model the neighborhood of each data point, and this does not seem necessary. A compromise is then: partition the data space into regions, and each region is fit or dominated by a single local model.
- Model averaging: suppose there are multiple models, each perhaps explaining part of the data, then combining the models will improve the performance. Usually, fit multiple models with different input regions.
- **Remark**: the crucial idea is "heterogeneity", that the laws governing the objects of interest are not uniform, and depend on objects(many local models instead of one global model) . The key step addressing heterogeneity is to recognize the regions within which the laws are uniform, but across which the laws may be different. For instance:
  - In some problems, there are natural partitions, e.g. in human genetics, the populations form natural partitions.
  - Partition according to some features: this leads to the tree-based methods.
  - Partitions are grouping of the objects: model parameters are different across groups, but could be considered as samples from a larger population. This is the hierarchical model approach.
  - Implicity partitions: data points that conform to simple (local) models. The boosting method.
- Connection with *adaptive basis function model* (ABM) [Murphy 16.1]: we would like ot fit the model of the form
$$f(x) = w_0 + \sum_{m=1}^{M} w_m \phi_m(x) \tag{7.31}$$
where $\phi_m(x)$ is a basis function. This basis function can be linear, or a decision tree (certain combination of features). Our idea is to learn $\phi_m(x)$, each explaining part of the data, and combine the results. Geometrically, each $\phi_m(x)$ can be thought of as a partition of data (decision boundaries).

Tree-based methods [Hastie, ESL]:

- Idea: a simple way of partition the input data space is by the values of the predictors. Ex. for any one predictor, the space can be partitioned into two regions, depending on whether its value is greater than a threshold. Multiple predictors can be combined to partition the space into rectangluar regions.
- Tree model: assume within each partition, the value of the response variable is equal. So the model can be written as:
$$\hat{f}(X) = \sum_{m=1}^{M} c_m I(X \in R_m) \tag{7.32}$$
where m is an index of region, and R_m is the m-th region.

Learning regression and classification trees:

- Choosing split variable and point: Intuition: split a variable s.t. in each partition, the labels are common/non-uniform. For any variable to split, its value in partition (purity within each region) can be assessed by SSE in the regression setting (total within-group variance), and measures such as misclassification error or Gini index. Entropy or Gini index are preferred than misclassification error [Murphy, 16.2.2.2].
- Pruning: finding the optimal tree is NP-hard, so usually learn a large tree from data, and prune the tree afterwards. In a regression tree, we define the criterion of the tree as:
$$C(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T| \tag{7.33}$$

where m is a terminal node, $Q_m(T)$ is the average variance within the region defined by m, N_m is the number of data points in R_m, and α penalizes large trees. Under pruning, the best subtree of T that minmizes the above function can be found with a greedy algorithm: the weakest link is successively pruned, and regions collapsed.

Practical issues of learning trees:

- Categorical predictors and multi-way split: in general, multiway splits fragment the data too quickly, leaving insufficient data at the next level down. For binary outcome, one can order a categorical predictor by the fraction of class 1 instances in each group.
- Tree instability: a major problem with trees is their high variance: the effect of an error in the top split is propagated down to all of the splits below it. Bagging method averages many trees may reduce the variance.
- Difficulty in capturing additive structure: it could be captured in the model, but not particularly easy, e.g. if Y is a linear combination of X_1, X_2, X_3, then need to first split at X_1, then at every region in the next level, split by X_2; then at next level, split by X_3 at every region, etc.

Bagging (bootstrap aggregation) and random forest [Bishop, Chapter 14]:

- Idea: reduce the variance by averaging over multiple models - bias will not be changed by averaging, but variance can be reduced.
- Committee prediction: suppose we form M bootstrap datasets (sample N instances from the original data, with replacement), and train one model $\hat{f}_m(x)$ for each dataset, the committee prediction is given by:

$$\hat{f}_{\text{COM}}(x) = \frac{1}{M} \sum_m \hat{f}_m(x) \tag{7.34}$$

- Variance reduction: if the errors of each of $\hat{f}_m(x)$ is uncorrelated, then the expected error is $1/M$ of the error of any single model. However, since the models and thus errors are highly correlated, the actual reduction is usually much smaller.
- Random forest [Murphy, 16.2.5]: to reduce correlations, use random subsets of data, and random subsets of variables. Also Bayesian approach, Bayesian adaptive regression tree (BART).

Motivation of Boosting [Bishop, Chapter 14]:

- Tree-based methods perform "hard" partition of the data points, and enforce one model per region. A more flexible way is to learn multiple models, where each model dominates some data points.
- The partitions of data points are formed from data points (thus not stric rules such as cubic regions), with smooth boundaries (which correspond to the decision boundaries of the weak learners).

Boosting [Murphy, 16.4]

- Overview: our goal is to fit ABF: $f(x) = w_0 + \sum_m w_m \phi_m(x)$ where each basis function $\phi_m(x)$ is a "weak learner". Boosting is a greedy algorithm for fitting this model. Boosting is among the best off-the-shelf program for classification. It has the advantage of resistance to overfitting. The most commonly used weak learner is a shallow CART, where $\phi_m(x) = I(x \in R_m)$, where R_m specifies the decision boundary.
- Forward stagewise additive modeling: our goal is to solve the problem:

$$\min_f \sum_{i=1}^{N} L(y_i, f(x_i)) \tag{7.35}$$

We have different f given different loss functions (see Table 16.1). Ex. for L_2 loss, the optimal f, assuming the distribution $P(y|X)$ is given is:

$$f^*(x) = \mathrm{E}\,(y|X) \tag{7.36}$$

Finding optimal f is hard, so we tackle it sequentially: similar to numerical optimization, we use gradient methods to find optimum of a function. Our function at step m is:

$$f_m(x) = f_{m-1}(x) + \beta_m \phi(x; \gamma_m) \tag{7.37}$$

Suppose we have already known $f_{m-1}(x)$ (the solution from the previous step), we choose β_m, γ_m to minimize the loss function:

$$(\beta_m, \gamma_m) = \mathrm{argmin}_{\beta,\gamma} \sum_i L(y_i, f_m(x_i)) \tag{7.38}$$

We continue this for a certain number of iterations or decide by model selection criteria.

- L_2 boosting: the loss function at step m for a sample i is:

$$L(y_i, f_{m-1}(x_i) + \beta_m \phi_m(x_i; \gamma_m) = (r_{im} - \beta_m \phi_m(x_i; \gamma_m)) \tag{7.39}$$

  where $r_{im} = y_i - f_{m-1}(x_i)$ is the residual at point i. So at each step, the model is trying to fit/explain the residual from the previous step. Intuitively, this is similar to step-wise regression, where we choose one variable a time, and each time fitting a regression model using residuals as response variable.

- AdaBoost: exponential loss function,

$$L_m(\phi) = \sum_i \exp\left[-y_i(f_{m-1}(x_i) + \beta\phi(x_i))\right] = \sum_i w_{i,m} \exp\left(-\beta y_i \phi(x_i)\right) \tag{7.40}$$

  where the weights are given by: $w_{i,m} = \exp(-y_i f_{m-1}(x_i))$ is the fit of the function at the previous step with observed value. Intuitively, if the prediction is already good at i, it will have low weight in next step. LogitBoost: similar, but use log-loss function.

- Gradient boosting: a generic version that works on any loss function. Our goal is to minimize the function f over a given loss function $L(f)$. We imagine the optimization is done over parameters, $f(x_1), \cdots, f(x_N)$. Let g_m be the gradient at step m:

$$g_{im} = \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \tag{7.41}$$

  The function (vector of N points) is then updated by: $f_m = f_{m-1} = \rho_m g_m$ where ρ_m is step size, which is chosen by minimizing the loss along the direction of g_m (line search). This update is called *functional gradient descent*. This function only optimizes N data points, but we can generalize to optimize a weak learner at each step. Intuitively, we optimizes f by following its gradient along the direction of weak learners: the difference, $f_m - f_{m-1}$, is constrained to have the form of weak learners.

- Intuitions of how boosting works: At step m, we learn the m-th weak learner and its weight w_m. This is done by fitting the weak learner to the "residual" from previous steps. For binary classification problem, where residuals are not defined, we can also view this as data "weighting": for data points fit well by previous steps, their weights are low. Two specific examples of boosting:

  - Linear model with variable selection (L_2 boosting): each weak learner is a linear model with a single variable. At each stage, we choose a variable that explains the "residuals" from previous stages. The residual of a data point can be thought of its weight: when residual is 0, it has low weight.

– Weak learner is a shallow decision tree (AdaBoost): e.g. Figure 16.10, two variables x_1, x_2, and weak learners are decision boundaries (or intervals for each variable). Initially, very crude decision boundary. Later, boosting will try to separate the points in the boundaries (of earlier steps), based on which data points are not classified well - leading to more refined boundaries. This typically involves the use of different features in later stages.

Why we don't need to re-update parameters iteratively? In other words, when we fit later data points, we are updating the function parameters, which may lead to poorer fitting of earlier data points. Intuitively, this is not a problem, as the later stages uses different variables to refine decision boundaries.

- How boosting works from functional gradient perspective: at each step, we are minimizing the loss function by moving along the direction of gradients, but constrained by the fact that the functions need to be sum of weak learners.

AdaBoost algorithm [Bishop, Chapter 14]:

- Intuition: repeat M steps, at each step learn a model $\hat{f}_m$. The instances are weighted differently at each step, so that some models are learned to classify some data points, while the other models learned for other data points. A simple example in 2D: the model f_1 classify according the value of x_1, and f_2 according to x_2, thus the combination of f_1 and f_2 can classify according to more complex boundary.
- Algorithm: at each step m, first train a model by minimizing the weighted error function according to the current weights $w_i^{(m)}, 1 \leq i \leq N$:

$$L_m = \sum_i w_i^{(m)} I(\hat{f}_m(x_i) \neq y_i) \tag{7.42}$$

Then the weights are updated by:

$$w_i^{(m+1)} = w_i^{(m)} \exp\left[\alpha_m I(\hat{f}_m(x_i) \neq y_i)\right] \tag{7.43}$$

where α_m is the measure of how good the classifier $\hat{f}_m$ is (small if the model is poor). The intuition: if $\hat{f}_m$ misclassifies x_i, then it should have a higher weight; if $\hat{f}_m$ is already a good classifier (thus larger α_m), then should put more weight on those examples that it fails. The final function is given by:

$$\hat{f}(x) = \text{sgn}\left(\sum_{m=1}^{M} \alpha_m \hat{f}_m(x)\right) \tag{7.44}$$

- Interpretation: the algorithm minimizes the exponential error function, which is equivalent to find the log-odds ratio at each point x:

$$\hat{f}(x) = \frac{1}{2} \ln \frac{P(y=1|x)}{P(y=-1|x)} \tag{7.45}$$

When all previous α_m and $\hat{f}_m$ are given, it can be shown that minimizing the exponential error function leads to the update formulat of the AdaBoost algorithm. Also note this error function is much more sensitive to the outliers, comparing with cross-entropy, or other common errors.

Mixture of linear regression [Bishop, Chapter 14]:

- Idea: there may be hidden/unmeasured variables Z (e.g. in a population of individuals, geneder may be such unmeasured variable), and it is reasonable to assume that the linear models for groups with different values of Z may be different.

- Model: let the hidden group assignment of a data point x_i be Z_i. The group assignment follows the multinomial distribution π_k for group k, and the linear model for the group k is given by $N(w_k^T x, \sigma^2)$.
- Inference: EM algorithm can be applied, similar to Gaussian mixture model.

Mixture of experts [Bishop, Chapter 14]:

- Idea: instead of hidden groups, assume that the group assignment depends on input, i.e. grouping data points by their input values. The interpretation is: the data point determine which region it belongs to, and with each region, some expert determines the function.
- Model: can be written as:

$$p(y|x) = \sum_{k=1}^{K} \pi_k(x) p_k(y|x) \tag{7.46}$$

  where $\pi_k(x)$ is called the gating function (determine which region, k), and $p_k(y|x)$ is called the expert function for the region k. In simple models, both $\pi_k(x)$ and $p_k(y|x)$ can be modeled with linear functions.

7.3 Kernel and Prototype Methods

Reference: [Murphy, Chapter 14], [Bishop, Chapter 6], [Hastie, Chapter 13]

Motivation: why kernel methods?

- Non-linear decision boundary or regression function: several standard examples for understanding the need of learning based on instances:
  - XOR function: $f(x_1, x_2) = x_1 \text{XOR} x_2$. The decision boundary is clearly not linear.
  - Circles: $y = 1$ if $\|x\| \leq 1$. The decision boundary is circle.
  - Two moons: the positive and negative examples fall into non-convex sets.
  - sinc function: for regression analysis, $f(x) = \sin x / x$.
- Ideas for dealing with non-linearity: (1) feature expansion: e.g. for XOR function, we define features based on logic functions of x_1 and x_2. (2) Learning from instances: e.g. KNN, then for XOR function, there are four clusters in the data, and by predicting the label of a new instance based on its distance to the other instances, the decision can be non-linear. The kernel methods provide a framework to draw inference from examples.
- Example: using product features for circles. The decision boundary of a circle in $\mathbb{R}^2$: $x_1^2 + x_2^2 \leq R^2$, if define features as:

$$(x_1, x_2) \mapsto (z_1 = x_1^2, z_2 = x_2^2, z_3 = x_1 x_2) \tag{7.47}$$

  Then the decision boundary in $\mathbb{R}^3$ becomes linear: $z_1 + z_2 \leq R^2$.
- Use similarity for learning: kernel methods. Define kernel functions (similarlity) between objects, and the predictions of an object x is based on its kernel function $\kappa(x, x')$, where x' is existing data, and y'. Any methods using these functions can be thought as kernel methods.
  - A particularly important example is: prediction of structured objects, e.g. strings and tree. No obvious feature representation is available, and it is more natural to define a kernel on these objects (e.g. the edit distance between two objects).
- Remark: the connection between the two perspectives on dealing with non-linearity. (1) Kernel methods (Mercel kernel): effectively a new feature representation; (2) Prototypes used in kernel methods: represent a particular combination of features, thus could also be viewed as a special kind of feature expansion.

Constructing kernel functions:

- A canonical example for object similarity: suppose we have x and x' as two objects in $\mathbb{R}^D$, their similarlity can be measured by correlation or the angle between the two vectors. First, correlation:

$$r_{x,x'} = \frac{\sum_i (x_i - \bar{x})(x'_i - \bar{x'})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (x'_i - \bar{x'})^2}} = \frac{x \cdot x'}{\|x\|\|x'\|} \tag{7.48}$$

  So the correlation is proportional to the inner prodcut of x and x'. Next, we see that it is exactly $\cos(\theta)$ where θ is the angle between x and x'. In short, the object similarlity can be measured by the inner product, up to a constant.

- Kernel and inner product: To generalize the idea above, we want to define kernels that are effectively inner products. These are called Mercel kernels: a kernel function is Mercel kernel if for any given N inputs, x_i, the Gram matrix, defined by: $K = (\kappa(x_i, x_j))$ is positive definite. We take the Choleksy decomposition of K, or

$$K_{ij} = \phi(x_i)^T \phi(x_j) \tag{7.49}$$

  where $\phi(x_i)$ is some linear function of x_i (based on the decomposition). And we see that K_{ij} is an inner product. More generally, for any Mercel kernel, there exists a feature mapping from x to $\mathbb{R}^D$ s.t.

$$\kappa(x, x') = \langle \phi(x), \phi(x') \rangle \tag{7.50}$$

  The RBF kernels (below) and cosine similarity kernels are all Mercel kernels. Thus Mercel kernels can be understood as inner product in some new feature space.

- Expanding kernels: Equations (6.13) to (6.22) in [Bishop, Chapter 6]. Most importantly, if $k_1(\mathbf{x}, \mathbf{x}')$ is a kernel, then, a polynormal function of $k_1()$ is also a kernel. And if A is a symmetric psd. matrix, then

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T A \mathbf{x}' \tag{7.51}$$

  is also a kernel.

Examples of kernel functions:

- Gaussian (RBF) kernel: the Guassian kernel is defined by:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}(x - x')^T \Sigma^{-1} (x - x')\right) \tag{7.52}$$

  In the special case where Σ is diagonal, this can be written as:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}\sum_{j=1}^{D} \frac{1}{\sigma_j^2}(x_j - x'_j)^2\right) \tag{7.53}$$

  The term σ_j is the characteristic length scale of the j-th dimension. And when σ_j are all equal, we have:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) \tag{7.54}$$

  where σ is called the bandwidth.

- Linear kernel: the simplest kernel is $\kappa(x, x') = x^T x'$, i.e. the feature mapping is $\phi(x) = x$. This is useful when the problem is linearly separable, and there is no need of working on other feature space.

- String kernel: suppose we have a dictionary of strings, and we want to define the kernel of two strings. Let s be any substring, and we could define the string kernel as the number of times s appears in both the input strings x and x', summing over al possible s. Or in other words, the kernel is roughly the number of shared substrings.

Probabilistic kernels:

- Motivation: if we know the process that generates x, then we could use this to define an appropriate kernel. For instance, if x and x' are from a normal distribution, then the appropriate scale for their distance/kernel is the standard deviation. The general idea is that the kernel function between two objects reflects how likely the two are generated from the same distribution (process).

- Probability product kernel: if x_1 and x_2 are close, then they should add similar information to the underlying distribution. From the Bayesian perspective, the posterior distributions $p(x|x_1)$ and $p(x|x_2)$ should be similar. In practice, we often assume $p(x|x_1)$ is close to $p(x|\hat{\theta}(x_1))$ where $\hat{\theta}(x_1)$ is the MLE of the parameter given x_1, and the same for $p(x|x_2)$. The similarlity of the two distributions can be defined via the kernel:

$$\kappa(x_1, x_2) = \int p(x|x_1)^\rho p(x|x_2)^\rho dx \tag{7.55}$$

  where $\rho > 0$. It can be seen, for example, that the kernel is maximum when the two distributions are equal (Cauchy-Scharwz Inequality, the norm is equal to 1). Examples:

  - RBF kernel: when the data is form normal distribution, $p(x|\theta) = N(\mu, \sigma^2 I)$, and $\rho = 1$, we have the RBF kernel.
  - The data can be viewed as generated from a mixture model, then the kernel can be defined according to the probabilisty that two objects are sampled from the same component:

$$k(\mathbf{x}, \mathbf{x}') = \sum_i p(\mathbf{x}|i)p(\mathbf{x}'|i)p(i) \tag{7.56}$$

  - For two sequences, the kernel can be defined assuming they are generated from the same HMM, following the same path:

$$k(\mathbf{x}, \mathbf{x}') = \sum_{\mathbf{z}} p(\mathbf{x}|\mathbf{z})p(\mathbf{x}'|\mathbf{z})p(\mathbf{z}) \tag{7.57}$$

    where $\mathbf{z}$ is the hidden path.

- Fisher kernel: if x and x' are close, then to say that they add the same information, or suggesting the same values of θ, is equivalent to saying the log-likelihood function $\log p(x|\theta)$ and $\log p(x'|\theta)$ should be similar, near $\hat{\theta}$. Define the gradient of the log-likelihood or the score vector,

$$g(x) = \nabla_\theta \log p(x|\theta) \tag{7.58}$$

  Then $g(x)$ should be close to $g(x')$ evaluated at $\hat{\theta}$. This is defined by their generalized inner product:

$$\kappa(x, x') = g(x)^T F^{-1} g(x') \tag{7.59}$$

  where F is the Fisher information matrix evaluated at $\hat{\theta}$.

Motivation of prototype methods:

- Idea: use prototypes to represent the data points in both positive and negative classes, then classification of one instance can be performed by looking at the nearest prototype(s) of this instance. The advantage: Highly unstructured, not dependent on statistical assumptions, can be very effective and often among the best in real data problems. Particularly when the decision boundary is highly irregular.

- Implementations: (1) KNN and local methods: use the nearest neighbors/prototypes to make predictions, weighted by distance; (2) kernel representation: the distance of an instance to all protypes is a new representation of the instance.
- Remark: this is similar to the method of fitting a curve with a piecewise smooth function. The key problem is to select a set of "pivotal" points to cover the data space. These prototypes can simply be all training points (e.g. KNN), or some points to be learned from data (e.g. RBF method).

K-means based methods:

- K-means: cluster data points with K-means algorithm, on both positive and negative classes. And classify any instance to the nearest prototype.
- Gaussian mixture: a "soft" version of clustering.

Learning vector quantization (LVQ):

- Idea: K-means clusters are found for each class separately. But we want to prototypes that may discriminate different classes.
- Algorithm: suppose we have processes all training instances up to x_i, now with the new training instance x_i, we first find the nearest prototype. If the class label of the prototype is the same as y_i, then we move the prototype a bit closer to x_i:

$$m_j(k) \leftarrow m_j(k) + \epsilon(x_i - m_j(k)) \tag{7.60}$$

  where k is the class label, and j the index of the prototype. Otherwise, it will be moved a bit away from x_i.

KNN algorithm:

- Algorithm: no need to fit the model, for any instance to predict, x_0, find its K nearest neighbors in the training data, and classify x_0 according to majority vote.
- The algorithm can be very effective in a large number of applications. It is particularly good when the decision boundary is very irregular. Most methods (even with K-means selection of prototypes) would depend on statistical assumption e.g. natural grouping of data points, which may not be true; in contrast, KNN does not make such assumptions, and the number of prototypes is not fixed a priori.

Asymptotic performance of KNN:

- Model: we consider a data point to be classified, x, and analzye the average loss. The class lable of x is inherently probabilistic as the probability distribution of different classes overlap at x. This uncertainty can be expressed as $p_k(x)$, which is the conditional probability for class x, assume the true models are known.
- Optimal method (Bayes error: posterior probability of class label): let k^* be the dominant class, then the Bayes error is $1 - p_{k^*}(x)$.
- 1-NN: in the asymptotic case, make the correct prediction only if the correct class label of the nearest training point is correct (for the same reason that the class label of the training point is also inherently probabilistic), thus the error is (weighted by the probability of the true class label of x):

$$\text{1NN error} = \sum_{k=1}^{K} p_k(x)(1 - p_k(x)) \geq 1 - p_{k^*}(x) \tag{7.61}$$

  For $K = 2$: the error rate is $2p_{k^*}(x)(1 - p_{k^*}(x)) \leq 2(1 - p_{k^*}(x))$.

Kernel machines:

- Kernalized feature vector: suppose we represent a data point x by its kernel function wrt. K prototypes, called "kernelized feature vector":

$$\phi(x) = [\kappa(x, \mu_1), \cdots, \kappa(x, \mu_K)] \tag{7.62}$$

  where μ_k is the k-th prototype (could be from a clustering algorithm or could be the data points themselves, see below). Then the usual GLM can be applied to the kernelized features - kernel machines. If the RBF kernel is used, this is called an RBF network.

- Why the kernel machines could solve the non-linearity problem? Example: in XOR function, the four prototypes represent four different logic functions (combinations) of the features x_1 and x_2. With these new features, the function is linear.
  - Example: application of RBF network in 1D fitting of $y = f(x)$ that is highly non-linear (Figure 14.3 of Murphy). The prototypes are a subset of points (x_i, y_i): clearly the relationship between these points are not necessarily linear.
  - Bandwidth: low bandwidth leads to very wiggy functions: e.g. for an x, if it is not in the close neighborhood of any prototypes, then its value y is predicted to be 0 (low-bias, high variance). High bandwidth leads to lower variance, but higher bias.

An example of kernel machine: we illsutrate how to make inference and predictions using kernel machines.

- Consider a simple linear model in the kernelized feature space:

$$y = \sum_{j=1}^{K} \beta_j \kappa(x, \mu_j) + \epsilon \tag{7.63}$$

  where K is the number of prototypes, μ_j is the j-th prototype, and ϵ the error term (no need of change).

- Interpretation of β_j: imagine μ_j are well-separated, then $\kappa(\mu_j, \mu_k) \approx 0$ if $j \neq k$. Let y_j be the response variable at μ_j, then: $y_j \approx \beta_j \kappa(\mu_j, \mu_j)$. From this, we see that, $\beta_j \approx y_j / \kappa(\mu_j, \mu_j)$. For RBF kernel, we have $\kappa(\mu_j, \mu_j) = 1$, thus $\beta_j \approx y_j$. So under this special case, we have:

$$y \approx \sum_{j=1}^{K} y_j \kappa(x, \mu_j) \tag{7.64}$$

  Thus the value of y at a point x is simply the weighted average of y_j, with weights determined by the distance of x to μ_j.

- Parameter estimation and prediction: we define the kernelized feature vector for the data point x_i as, $z_i = [\kappa(x_i, \mu_1), \cdots, \kappa(x_i, \mu_K)]$, then apply the usual linear model on z_i, we have:

$$\hat{\beta} = (Z^T Z)^{-1} Z^T y \tag{7.65}$$

  where Z is the $n \times K$ matrix of z_i's. To predict the value of y at a point x^*, we form z^* first, and then our prediction $\hat{f}(x^*) = z^* \hat{\beta} = z^* (Z^T Z)^{-1} Z^T y$.

- Special case of $K = n$: each x_i is a protoptype, then the design matrix is simply $n \times n$ matrix $K = [\kappa(x_i, x_j)]$. When we use the inner product (instead of kernel), it is also $K = XX^T$, the Gram matrix. For symmetric kernels, we have:

$$\hat{\beta} = (K^T K)^{-1} K^T y = K^{-1} y \tag{7.66}$$

And prediction $\hat{f}(x^*) = [\kappa(x^*, x_1), \cdots, \kappa(x^*, x_n)]K^{-1}y$. To see what this means, imagine K is diagonal (the n data points are well-separated), then it's easy to show that:

$$\hat{f}(x^*) = \sum_i \frac{\kappa(x^*, x_i)}{\kappa(x_i, x_i)} y_i \tag{7.67}$$

This is effectively the kernel-weighted average that we'll discuss later, and corresponds to our intuition described above.

Sparse vector machines:

- In the high-dim. case, there is no good way of choosing a small number of prototypes. However, choosing a large number of prototypes makes the inference difficult: D is close to n (the number of features is close to or equal to the number of data points). A special case is every x_i is a prototype:

$$\phi(x) = [\kappa(x, \mu_1), \cdots, \kappa(x, \mu_N)] \tag{7.68}$$

  Then $D = N$. The only way to solve this problem is to use the sparsity-promoting priors for the coefficients in the model. This is called a "sparse vector machine".

- Types of sparse vector machines: L_1 regularization vector machine (L1VM), L2VM, relevance vector machine (RVM) and SVM. Except L2VM, the others are sparse, so there are a small set of positive and negative prototypes among all training examples: prediction of an instance is based on how close it is to these prototypes. These are support vectors in SVM.

- **Question**: Sparsity is formulated in terms of the kernelized feature vectors. How should one formuate the model in terms of the original features?

Kernel trick:

- Claim: we do not have to explicitly model the kernel machines using the kernelized feature vectors. Instead, to make prediction at a new example, we only need to formulate the algorithm in terms of inner product between data points, and between data point and new example. Then we simply replace the inner products with kernels. This is the "kernel trick".

- Kernel trick: if an algorithm can be formulated in terms of inner product $\langle \mathbf{x}, \mathbf{x}' \rangle$, then we could define a feature mapping $\mathbf{x} \mapsto \phi(\mathbf{x})$, and the inner product in the new feature space is a kernel:

$$k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle \tag{7.69}$$

  In this notation, $k(\mathbf{x}, \mathbf{x}')$ is a general measure of similarity bewteen two instances. Then all the computation can be carried out with this kernel function, instead of the explicit feature mapping $\phi()$. The kernel function should be Mercel kernel so that it can be viewed as inner product.

- Computational advantage: the feature mapping into a very high (or even infinite) dim. space is now computationally possible, as long as the kernel function is well-defined and computable. Also, the feature mapping may not need to be explicitly defined, and one only needs to specify the ideas of similarity using kernel function, then the same algorithm can be still applied.

- To prove this claim, we consider the linear model example. We try to write the predicted value of y at example x using innner product. Following Equation 7.74, this can be written as:

$$\hat{f}(x^*) = x^* w = x^* X^T (K^{-1} y) \tag{7.70}$$

  Clearly, if we replace inner product in this equation with kernels, we obtain the results discussed before with explicit kernelized feature representation.

Kernelized distance-based methods:

- Kernelalized KNN classification: to kernelize KNN algorithm, we write the distance as inner products:

$$\|x - x'\|^2 = \langle x, x\rangle + \langle x', x'\rangle - 2\langle x, x'\rangle \tag{7.71}$$

- Kernelalized K-medoids clustering: in the algorithm, one key step is to choose a data member in a cluster, so that the total distance of this member to all other members is minimum. To kernelize it, we replace the distance with kernels.

Kernelized ridge regression:

- Primal problem: minimze the error function:

$$J(\mathbf{w}) = (y - Xw)^T(y - Xw) + \lambda\|w\|^2 \tag{7.72}$$

  And the new prediction: $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. The solution is given by:

$$w = (X^TX + \lambda I_D)^{-1}X^Ty \tag{7.73}$$

  Note that w is written in terms of the covariance between explanatory variables (X^TX), and the covariance between explanatory and response variables.

- Dual problem: we can rewrite $\mathbf{w}$ in terms of the inner product of data vectors. Using the matrix inversion Lemma:

$$w = X^T(XX^T + \lambda I_N)^{-1}y \tag{7.74}$$

  where $K = XX^T$ is the Gram matrix (inner product between any two instances). Now we still have X^T, however, we note that once we compute the prediction w^Tx, we will move this term into an inner product as well. Specifically let $\alpha = (K + \lambda I_N)^{-1}y$ as dual variables, then we have:

$$\hat{f}(x) = w^Tx = (X^T\alpha)x = \sum_{i=1}^{N}\alpha_i\kappa(x, x_i) \tag{7.75}$$

- Analysis: behavior of kernelized ridge regression. We consider the case where the bandwidth of the kernel is very small, thus K is approximately diagonal. And we ignore the regularization term, so $\alpha \approx K^{-1}y$ with:

$$\alpha_i \approx \frac{y_i}{\kappa(x_i, x_i)} \tag{7.76}$$

  Then the predicted value of y is given by:

$$\hat{f}(x) = \sum_{i=1}^{N}\alpha_i\kappa(x, x_i) \approx \sum_{i=1}^{N} y_i\frac{\kappa(x, x_i)}{\kappa(x_i, x_i)} \tag{7.77}$$

  Thus $\hat{f}(x)$ is the average of y_i, weighted by the kernel function $\kappa(x, x_i)$ (similar to local regression).

- Remark: the idea of dual representation is very general: an algorithm/inference procedure can be stated in terms of how variables are related, but can also be stated in terms of the relation between data vectors. The advantage of the dual representation is that when D is large, and n is relatively small, the computational cost in the dual representation is much lower.

7.3.1 Smoothing Kernels and Local Methods

Reference: [Murphy, Chapter 14], [Hastie, Chapter 6]

Kernel smoothing methods:

- Motivation: two related problems:
  - Estimating probability density function: given data points x_i, find $p(x)$ for each x.
  - Estimating a function: given (x_i, y_i), find a function $y = f(x)$.

  Our goal is, for both cases, find smooth functions/density.
- Idea of local methods: probability distribution and functions (to be learned) are generally smooth, i.e. the probability densities at close points are close, and if $x \approx x'$, it's likely that $f(x) \approx f(x')$, etc. This smoothness property can be exploited to learn functions and probability distributions locally.
- Kernel smoothing: to implement the local methods for a point of interest x_0, express the desired quantity as the sum of contributions of many examples, and weigh the examples according to how close they are to x_0, defined by a kernel function, $K_\lambda(x, x_0)$.

Smoothing kernels:

- Smoothing kernel: weighting of points depends on their distance to some reference point, usually 0. It is a function of one argument which satisfies the properties:

$$\int \kappa(x)dx = 1 \qquad \int x\kappa(x)dx = 0 \qquad \int x^2\kappa(x)dx > 0 \tag{7.78}$$

  This will gurantee that the weights sum to 1, and by itself will not introduce bias. For vector input, and non-zero reference point (x_0), we could define a smoothing kernel as:

$$K_h(x, x_0) = \kappa_h(\|x - x_0\|) \tag{7.79}$$

- Gaussian kernel: the simplest case is the standard normal pdf:

$$\kappa(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \tag{7.80}$$

  In general, we introduce a bandwidth parameter h: $\kappa_h(x) = \frac{1}{h}\kappa(\frac{x}{h})$. In the multivariate case, we have:

$$\kappa_h(x) = \frac{1}{h^D(2\pi)^{D/2}} \prod_{j=1}^{D} \exp\left(-\frac{1}{2h^2}x_j^2\right) \tag{7.81}$$

- Other common smoothing kernels: Epanechnikov kernel (bounded on [-1,1]), tri-cube kernel, boxcar kernel (uniform distribution on [-1,1]), etc.

Kernel density estimation (KDE): the goal is to estimate the PDF of a RV at x_0, $p(x_0)$ from an iid sample $x_1, \cdots, x_N$.

- Method: consider a local neighborhood of x_0 with width λ, $N(x_0)$, the simple local estimate would be:

$$\hat{p}(x_0) = \frac{1}{N\lambda} \sum_i I(x_i \in N(x_0)) \tag{7.82}$$

  where $I(\cdot)$ is the indicator function. This estimate is "bumpy", to smooth the estimate, replace the step function with a kernel function, we have:

$$\hat{p}(x_0) = \frac{1}{N\lambda} \sum_i K_\lambda(x_i, x_0) \tag{7.83}$$

  The kernel function is chosen s.t. it is large when close to x_0, and small when distant.

- Interpretation: the PDF can be approximated as a Gaussian mixture model, with each centroid at the data point x_i. Assume the mixture component has equal σ, and mixture weight $1/N$. The total probability density at a point x_0 is the sum of density from every component (weighted by $1/N$).
- Density estimation and classification: any density estimation methods can be used for classification - estimate density separately for each class, and then do classification using Bayesian theorem. The same idea can be applied for regression, where density estimation would include the joint distribution of (x, y), see below and [Bishop, section 6.3].

Density estimation under other perspectives: [personal notes]

- Bayesian nonparametric density estimation: suppose we discretize the PDF into K intervals, then our problem is to estimate a multinomial distribution $(p_1, \cdots, p_K)$ from the data $(x_1, \cdots, x_K)$ where x_k is the number of points falling in the k-th interval. The MLE of p_k is simply x_k/N. The Bayesian inference puts a prior on p_k s.t. it is close to p_{k-1}. For instance, we could have p_k from a stochastic process, where $\mathrm{E}(p_k) = p_{k-1}$.
- Sparse model for density estimation: following the same discretization scheme, we want to estimate p_k through penalized log-likelihood. For example:

$$l(p|x) = \sum_{k=1}^{K} x_k \log p_k - \lambda \sum_k (p_k - p_{k-1})^2 \tag{7.84}$$

  This is similar to fused Lasso in regression setting.
- Direct regularization of density function: in general, we could work directly on the density function. For instance, using the penalized log-likelihood, we have:

$$l(f) = \sum_{i=1}^{N} \log f(x_i) - \lambda \int \|f'(x)\|^2 dx \tag{7.85}$$

Local likelihood method:

- Idea: instead of creating a global model, for any point to study, x_0, create a local model around x_0. Then we assume all data points are generated according to this likelihood model, and estimate the parameters accordingly with weighting of data points. This could be used for a prediction problem, or more generally learning a likelihood model.
- Local likelihood method: let $\theta(x_0)$ be the parameter of the local model, maximize the local likelihood function:

$$l(\theta(x_0)) = \sum_{i=1}^{N} K_\lambda(x_0, x_i) l(x_i; \theta(x_0)) \tag{7.86}$$

  where $l(x_i, \theta)$ is the likelihood of a data point. This is simply the usual log-likelihood function with distance weighting. Specifically, we assume a function near x_0 parameterized by θ, and the data x_i are generated from this local function. This function is smooth: local, linear, polynomial, etc.
- Density estimation using local likelihood: we create intervals, and let y_k be the number of data points in interval k, and we estimate the density near x_0. Assume density is constant near x_0, $p_0 = p(x_0)$. Our model is $y_k \sim \mathrm{Bin}(N, p_0)$ where N is the sample size. The local log-likelihood is:

$$l(p_0) = \sum_k K_\lambda(y_k, x_0)[y_k \log p_0 + (N - y_k) \log(1 - p_0)] \tag{7.87}$$

  Maximizing this function, and use the fact that $\sum_k K_\lambda(y_k, x_0) = \sum_i K_\lambda(x_i, x_0)$, we can obtain the KDE.

Local prediction: the optimal predictor of x_0 is given by $E(Y|X = x_0)$ from statistical decision theory.

- Naive method: the k-nearest neighbor (KNN) average of the point x_0:

$$\hat{f}(x_0) = \text{Ave}(y_i|x_i \in N_k(x)) \tag{7.88}$$

  The function is distinuous, and not optimal (as all points within a region have equal weight).

- Nadaraya-Watson kernel weight average:

$$\hat{f}(x_0) = \frac{\sum_i K_\lambda(x_0, x_i) y_i}{\sum_i K_\lambda(x_0, x_i)} \tag{7.89}$$

  The kernel is usually chosen s.t. only points within a window (metric window, as it is defined by x, not by the rank) can contribute.

- Remark: the local prediction method can be viewed as estimating $E(Y|X = x_0)$ using the estimation of local density at x_0:

$$f(x_0) = E[Y|X = x_0] = \frac{\sum_y y p(x_0, y)}{p(x = x_0)} \tag{7.90}$$

  Plug-in the density estimate (as a mixture of N Gaussian distributions) and we obtain the result.

- Remark: this could also be viewed as an application of local likelihood method. Assuming the local model at x_0 is $y_i \sim N(\theta(x_0), \sigma^2)$, where $\theta(x_0)$ is the expected value of y at x_0 (to be estimated). The likelihood of this model at a point (x_i, y_i) is:

$$l(\theta(x_0)|x_i, y_i) = \log P(y_i|x_i, \theta(x_0)) = -\frac{1}{2\sigma^2}(y_i - \theta(x_0))^2 \tag{7.91}$$

  The local likelihood model is:

$$l(\theta(x_0)) = -\frac{1}{2\sigma^2} \sum_i K_\lambda(x_0, x_i)(y_i - \theta(x_0))^2 \tag{7.92}$$

  Maximize this function by taking derivative of $\theta(x_0)$, and we obtain Equation 7.89.

Local regression:

- Idea: the kernel weight average method still assumes a constant near x_0 to be estimated. More generally, fit a local model at x_0. The model should minimize the error, where the errors in regions distant to x_0 will be discounted.

- Local linear regression: at x_0, fit the local linear model, $y = \alpha(x_0) + \beta(x_0)x$:

$$\min_{\alpha(x_0), \beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \left[y_i - \alpha(x_0) - \beta(x_0) x_i\right]^2 \tag{7.93}$$

  The solution is similar to the result of linear regression, except that X^T will be replaced by $X^T W(x_0)$, where $W(x_0)$ is the $N \times N$ diagonal matrix with i-th diagonal element $K_\lambda(x_0, x_i)$. The result is a linear combination of y_i:

$$\hat{f}(x_0) = \sum_i l_i(x_0) y_i \tag{7.94}$$

- Varying coefficient model: more generally, this is the varying coefficient model, or structured regression [Hastie, 6.4.2]. Suppose among all features, we can divide them into two groups, x and z. The response variable y depends on x in a linear model, but on z in a very non-linear fashion. Then we could formulate this as a model, where the coefficients of y on x is a local model of z. Specifically, we solve the regression problem for each z_0:

$$\min_{\alpha(z_0),\beta(z_0)} \sum_{i=1}^{N} K_\lambda(z_0, z_i) \left[y_i - \alpha(z_0) - \beta(z_0)x_i\right]^2 \tag{7.95}$$

  An example of this is the application to time-series data, where z_i is the time point. We learn a linear model for each time point, but with the constraint that the models of the nearest time points should have similar coefficients.

- Bayesian perspective: we are trying to estimate a local model, however, the estimator using only local data points is a poor estimator (very few data points, thus high variance), so it's better to use other neighbors as priors. Similar to Bayesian density estimation, we could formulate a hierarchical model, where $\beta(z)$ follows random effects, with its prior favoring similar values of β for close z.

Selecting the width of kernel λ in kernel smoothing methods:

- Kernel width: defined by a parameter λ. For the Gaussian, it is the standard deviation; for KNN, it is the radius spanned by the nearest k members; for tri-cube kernel, the radius of the support region.

- Bias-variance tradeoff: small λ, a small number of points are used to estimate $\hat{f}(x_0)$, thus large variance, but small bias as each of $f(x_i)$ should be close to $f(x_0)$. Large λ is opposite: small variance, but large bias.

- Dependency on density: the variance (determined by the number of points near x_0) is inverstly proportional to the local density. Thus, at regions of low density, it may be better to use larger λ. More generally, an adaptive neighborhood can be used:

$$K_\lambda(x_0, x) = D\left(\frac{x - x_0}{h_\lambda(x_0)}\right) \tag{7.96}$$

7.4 Unsupervised Learning

1. Overview of unsupervised learning

   Challenges of unsupervised learning [Murphy, Chapter 1]:

   - Goal: discover the "structure" from the data, in a general sense. This could be: find a simple representation of data, categorize the data, etc. so that when new instances are presented, one could find similar objects in "memory", categorize the new instances, etc.
   - Statistical perspective: the goal is to learn a density function $p(x|\theta)$, as opposed to learning a conditional density $p(y|x, \theta)$ in the classification problem.
   - Cluster analysis: discover the clusters (objects belong to the same category) in the data. Ex. from a set of images, group them into categories such as animals, fruits, furnitures, etc.
   - Discover latent factors: the data may be best explained through the action of certain latent factors. For example,the variability of a set of images of the same object but under different background, light conditions, etc. can be explained by a small number of latent factors such as lighting, pose, etc.
   - Graph (dependency) structure: the multiple variables in the data are related to each other in a certain way.

- Imputation (filling the missing data): an application of the unsupervised learning is that given part of the data (object), learn the missing part. Ex. image inpainting, collaborative filtering (we know some movies a user like, and to predict the rest).

2. Cluster analysis [Hastie, Chapter 14]

   Distance/similarity metrics:

   - Distance vs. similarity: often distance or dissimilarity is used, instead of similarity, but this would depend on the clustering algorithms. K-means and hierarchical clustering use distance measures, while spectral clustering uses similarity measures. To convert distance to similarity (e.g. to apply spectral clustering), one can use the Gaussian kernel, $a = \exp(-d^2/2\sigma^2)$.
   - The choic of distance metric strongly depends on the problems of hand, this is similar to the choice of error/loss function. Considerations may include: the sensitivity to large difference of features (if so, then sensitive to outliers), the asymptotic behavior (e.g. the distance may approach a constant when the difference is large), the weighting of features, etc.
   - Common distance metrics for feature vectors: (1) for ordinal variables represented by M continguous integers: replace the value by $(i - \frac{1}{2})/M$, and treat as quantitative variables; (2) for categorical variables: suppose there are M categories, then $M \times M$ matrix for every possible pair.
   - Feature weighting: let w_j be the weight of feature j when computing distance, then the average distance of all pairs:
   $$\bar{D} = \sum_{j=1}^{p} w_j \bar{d}_j \tag{7.97}$$
   where $\bar{d}_j$ is the average distance of the j-th feature, it is also equal to $2 \times \text{var}_j$. Thus the relative importance of each feature is proportional to its variance over the dataset (intuitively clear as large variance features should make the data more separable). Note that under common standardization, the variance of all feature is equal to 1, and this may not be good.
   - Distance/similarity metrices for general objects, e.g. sequences: (1) edit distance, this includes variants that considers the alignment of two objects (e.g. cross-correlation for time-series data); (2) feature representation of objects: e.g. Fourier transform of time-series data, and use the coefficients as the features; (3) hidden dimensions: e.g. divergence time between two sequences, spatial distance betwen two objects.
   - Properties of distance measure: generally need nonnegativity, and $d(x, x) = 0$, symmetry. Triangle inequality may be desired, but not always satisfied.
   - Symmetric distance: if a distance measure is not symmetric, one can often transform it as a symmetric measure as: $[d(x, y) + d(y, x)]/2$.
   - Reference: [Liao, Clustering of time series data - a survey, Pattern Recognition, 2005].

   Probablity-based distance/similarity measures:

   - KL divergence: suppose θ_x and θ_y are the models that generate the two objects x and y, respectively, then the distance between x and y can be defined as $KL(\theta_x||\theta_y)$. Note that KL divergence has a likelihood interpretation: the divergence bewteent two distributions P and Q is the likelihood of generating data of P using the distribution Q, when the data size approaches infinity.
   - Likelihood based measure: for instance, for two sequences, x and y, suppose the model of x is θ_x (MLE), then the similarity between x and j can be defined as the normalized log likelihood:
   $$l_{xy} = \frac{1}{\text{length}(y)} \log P(y|\theta_x) \tag{7.98}$$

To make it symmetric, we simply have: $d_{xy} = (l_{xy} + l_{yx})/2$. Transformation may be desired to make this a distance, in particular, to use the difference of log likelihood as the distance, e.g.:

$$d_{xy}^{BP} = \frac{1}{2}\left(\frac{l_{xy} - l_{xx}}{l_{xx}} + \frac{l_{yx} - l_{yy}}{l_{yy}}\right) \tag{7.99}$$

See [Garcia-Garcia, A new distance measure for model-based sequence clustering, IEEE Pattern Analysis and Machine Learning].

- Hypothesis testing: this is to test the hypothesis, H_0: the objects x and y are from the same distribution, vs. H_A: they are from different distributions. Suppose we form the LRT of the hypothesis, λ, then the distance can be defined as λ or the CDF function at λ (effectively normalize λ s.t. it is between 0 and 1). See [Kumar, Clustering Seasonality Patterns in the Presence of Errors, KDD02], [Bagnall, A likelihood ratio distance measure for the similarity between the fourier transform of time series].

Combinatorial algorithms for clustering: one major paradigm for clustering.

- Idea: find cluster assignment s.t. the total distance within clusters is minimum, or the distance between clusters is maximum. These two objectives are equal: suppose the total distance is T, and $W(C)$ is the within-cluster distance and $B(C)$ the between-cluster distance, then:

$$T = W(C) + B(C) \tag{7.100}$$

- Search strategy: often greedy algorithm that converge to local optima.

K-means and K-mediods algorithm:

- K-means: alternate two steps: (1) given the cluster assignment, find the means of the clusters; (2) given the cluster means, find the best cluster assignment.
- K-mediods: when the data cannot be treated as Gaussian variables, cluster means are not well-defined, thus for each cluster, use one cluster member instead. The same algorithm can be applied, except that the cluster means are replaced by the cluster centers.
- Choosing K: plot the reduction of total within-cluster distance as a function of K. The reduction should be large when K is less than the natural/ideal K, but slow down when K is bigger than the natural value. So look for the "kink" in the plot. The idea can be formulated by the gap statistic: the reduction of distance comparing with the reduction under the uniform (no cluster-structure) data.
- Application: image compression/vector quantization. No need to represent every pixel (8-bit, for 256 colors) in an image: there are a much smaller number of basic patterns (e.g. all-black, all-white), thus apply K-means to cluster all basic blocks and only record the cluster index of each block.

3. Spectral clustering

Overview of spectral clustering:

- Global vs. local structure: the common clustering algorithms, are based on the global structure of the data. E.g. the k-means algorithm is based on the distance of a point to the center of a cluster. The local structure cannot be captured. For instance, suppose the cluster consists of points in a circle, then the points are mutually close, but not necessarily close to any center.
- Non-convex sets: A particular example of the problem is that the clusters may not for convex-sets, which are generally not captured by k-means, k-medoid, etc.

- Idea of spectral clustering: find clusters s.t. the points with a cluster are mutually close, and the points across clusters are distant. The difference with algorithms such as k-means: the distances of close neighbors are easier defined (e.g. use ϵ-neighborhood graph, see below) than the distance of a point to the center.
- Reference: [Luxburg, A Tutorial on Spectral Clustering, 2007]

Similarity graph:

- ϵ-neighborhood graph: connect all points whose pairwise distances are less than ϵ, and the rest are not directly linked. Usually used as unweighted graph.
- k-nearest neighbor graph: for each point, connect to its k nearest neighbors. However, this is asymmetric. To have an undirect (symmetric) graph, one can link i and j if i is the among the top k nearest neighbors of j, and vice versa.
- Fully connected graph: for any two points, link the two with the weight defined as a function of the distance between two points. One commonly use the Gaussian similarity function:

$$s(x_i, x_j) = \exp\left[-\|x_i - x_j\|^2/(2\sigma^2)\right] \tag{7.101}$$

Unnormalized graph Laplacian:

- Definitions: the weighted adjacency matrix $W = (w_{ij}) \geq 0, 1 \leq i, j \leq n$, if i and j are linked by an edge, and it is 0 otherwise. The degree matrix is a diagonal matrix $D = (d_i)$, where $d_i = \sum_j w_{ij}$. We define the weights between two sets A and B as:

$$W(A, B) = \sum_{i \in A, j \in B} w_{ij} \tag{7.102}$$

  The volume of a set A is defined as: $\text{vol}(A) = \sum_{i \in A} d_i$. The unnormalized graph Laplacian is defined as:

$$L = D - W \tag{7.103}$$

- Graph Laplacian and smoothness of the graph: the primary motivation of defining a graph Laplacian is that it is related to the smoothness measure of a graph. Suppose we have labels of the nodes in the graph, f_i (e.g. the cluster membership). The "smoothness" of f_i, i.e. how abruptly f_i changes in neighbors (in general, we want highly-weighted neighbors to have simliar labels), can be defined as the sum of $(f_i - f_j)^2$, weighted by the edge weight, over all edges. This smoothness penalty is related to the graph Laplacian by:

$$f^T L f = \frac{1}{2} \sum_{i,j} w_{ij}(f_i - f_j)^2 \tag{7.104}$$

  for any $f \in \mathbb{R}^n$.

- Eigenvalues and eigenvectors of graph Laplacian: L has the following properties:
  - L is symmetric and postive semi-definite. The proof follows from Equation 7.104: $f^T L f \geq 0$ for any f.
  - L has n non-negative eigenvalues with $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, and the eigenvector corresponding to $\lambda_1 = 0$ is the unit vector $\mathbf{1}$. This is easy to prove by checking that $L\mathbf{1} = 0\mathbf{1}$.
- Graphs with multiple connected components: for a graph with k connected components, $A_1, \cdots, A_k$, the multiplicity of the eigenvalue 0 is equal to k, and the corresponding eigenvectors are $\mathbf{1}_{A_1}, \cdots, \mathbf{1}_{A_k}$ respectively, where $\mathbf{1}_A$ represents a vector whose i-th compoment is equal to 1 if $i \in A$ and 0 otherwise.

Proof: first, when $k = 1$, i.e. the graph is connected, let f be the eigenvector of 0, then $Lf = 0$ and thus:

$$f^T Lf = 0 = \frac{1}{2}\sum_{i,j} w_{ij}(f_i - f_j)^2 \tag{7.105}$$

Thus every term in the RHS is 0. If i and j are linked, then $w_{ij} > 0$, thus $f_i = f_j$. From this, we see that f must be proportional to the vector $\mathbf{1}$.
For arbitrary k, we could order the vertices s.t. the matrix W has a block form, and so with L, with k blocks, $L_1, \cdots, L_k$ (and all other terms in L are equal to 0). We apply the previous result to each of L_k.

Normalized graph Laplacian:

- Two normalized graph Laplacian: if we normalize the label f of a graph, as: $\tilde{f}_i = f_i/\sqrt{d_i}$, or $\tilde{f} = D^{-\frac{1}{2}} f$, then the smoothness penalty is the quandratic from of the matrix, $L_{\text{sym}} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$:

$$f^T L_{\text{sym}} f = \frac{1}{2}\sum_{i,j} w_{ij}\left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}}\right)^2 \tag{7.106}$$

  A second normalized graph laplacian is defined as (used in random walk):

$$L_{\text{rw}} = D^{-1}L = I - D^{-1}W \tag{7.107}$$

- Eigenvalues and eigenvectors of symmetric graph Laplacian: the matrix L_{sym} is symmetric and p.s.d, and have n non-negative eigenvalues with $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, and the eigenvector corresponding to $\lambda_1 = 0$ is $D^{\frac{1}{2}}\mathbf{1}$.
  Proof: check that:

$$L_{\text{sym}} D^{\frac{1}{2}}\mathbf{1} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} D^{\frac{1}{2}}\mathbf{1} = D^{-\frac{1}{2}} L\mathbf{1} = 0 \tag{7.108}$$

- Eigenvalues and eigenvectors of random walk graph Laplacian:
  - λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ is an eigenvalue of L_{sym} with eigenvector $w = D^{\frac{1}{2}}u$.
  - λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ and u solves the generalized eigen-problem $Lu = \lambda Du$.
  - The matrix L_{rw} is symmetric and p.s.d, and have n non-negative eigenvalues with $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, and the eigenvector corresponding to $\lambda_1 = 0$ is $\mathbf{1}$.
- Graphs with multiple connected components: for a graph with k connected components, $A_1, \cdots, A_k$, the multiplicity of the eigenvalue 0 of both L_{sym} and L_{rw} is equal to k. For L_{rw}, the corresponding eigenvectors are $\mathbf{1}_{A_1}, \cdots, \mathbf{1}_{A_k}$ respectively. For L_{sym}, the corresponding eigenvectors are $D^{\frac{1}{2}}\mathbf{1}_{A_1}, \cdots, D^{\frac{1}{2}}\mathbf{1}_{A_k}$ respectively.

Spectral clustering idea:

- Simple case: a graph with K connected components. The graph Laplacian has K eigenvectors with $u_k = \mathbf{1}_k, 1 \leq k \leq K$. We could form the matrix $U = (u_1, \cdots, u_K)$. Clearly, U stores the membership indices: for the k-th component (cluster), only those x_i's in this cluster has $U_{ik} = 1$. Thus we could view the i-th row of U as a K-dimensional representation of x_i ($u_1, \cdots, u_k$ form an orthogonal basis).
- General case: perturbation of the simple case, where there may be (weak) edges between components. The k-th eigenvector u_k would not be exactly equal to $\mathbf{1}_k$, but will be close. u_k stores the projection of all x_i's on the k-th dimension, and it will be dominated by the k-th cluster (with small contributions from other clusters weakly connected to k).

- Algorithmic idea: to partition a graph into K clusters, we compute the K eigenvectors corresponding to the K smallest eigenvalues, and form the matrix U. Then the i-th row is a representation of x_i in the K-dim. eigenspace. We then use usually clustering algorithm, e.g. K-means, on the new representation.
- Three graph Laplacian and three spectral clustering algorithms: all three Laplacian, L, L_{sym} and L_{rw} encodes the information of connected components, and cluster structure (through perturbation analysis), so we can use the eigenvectors corresponding to any of these Laplacian, leading to three Spectral Clustering algorithms. However, for L_{sym}, since the norm of the eigenvectors of connected components are not constant, we need to normalize the row of the matrix U.

Graph cut:

- Motivation: partition the graph into k clusters s.t. the weights (similarity) within components are high and the weights across components are low. Since the sum of all weights is constant, we will only need to focus on the weights across the components. Given a graph partition, $A_1, \cdots, A_k$, we define the cut as:
$$\text{cut}(A_1, \cdots, A_k) = \frac{1}{2}\sum_{i=1}^{k} W(A_i, \bar{A}_i) \tag{7.109}$$
- Problem of unnormalized graph cut: two ways of seeing the problem:
  - Trivial solution: to minimize $f^T L f$ without any additional constraint, the trivial solution is $f = 0$.
  - Number of edges between components: Consider the case of $k = 2$, clearly, if $|A| = 1$, there are $n-1$ (upper bound) edges between A and $\bar{A}$; at $|A| = 2$, there are $2(n-2)$ edges between A and $\bar{A}$; and so on. The number of edges is not constant.

  So we will need to either reformulat the objective function (a normalized version), or impose additional constraints.
- Normalization: we could define the cut as the average weight between two components:
$$\text{RatioCut}(A_1, \cdots, A_k) = \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{|A_i|} \tag{7.110}$$
  We could also define the normalized cut as:
$$\text{Ncut}(A_1, \cdots, A_k) = \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\text{vol}(A_i)} \tag{7.111}$$

RatioCut at $k = 2$:

- Objective function: we have the optimization problem:
$$\min_{A \subset V} \text{RatioCut}(A, \bar{A}) \tag{7.112}$$
  We define a membership representation: $f_i = a > 0$ if $v_i \in A$, and $f_i = -b < -$ if $v_i \in \bar{A}$. Clearly, finding A is equivalent to finding f s.t. some constraint.
- Express the RatioCut as a function of f: we have:
$$f^T L f = \frac{1}{2}\sum_{i,j} w_{ij}(f_i - f_j)^2 = (a+b)^2 \text{cut}(A, \bar{A}) \tag{7.113}$$
  If we choose $a = \sqrt{|\bar{A}|/|A|}$ and $b = \sqrt{|A|/|\bar{A}|}$, then we have:
$$f^T L f = |V| \text{RatioCut}(A, \bar{A}) \tag{7.114}$$

The vector f is subject to the constraint that $\sum_i f_i = 1$, i.e. $f \perp \mathbf{1}$, and $\|f\|^2 = n$. Our problem is thus to minimze $f^T L f$ subject to the constraints of f (discrete).

- Continuous relaxiation: the problem is NP-hard because f can only take two values. We perform the continuous relaxation, and this leads to the problem:

$$\min_{f \in \mathbb{R}^n} f^T L f \text{ subject to } f \perp \mathbf{1}, \|f\| = \sqrt{n} \tag{7.115}$$

  We could understand the contraints as: (1) The labels should be balanced, i.e. $\sum_i f_i = 0$; (2) The scale of the labels should be equal to 1 (a constant), otherwise, we could scale f_i to be arbitrarily small. This means the mean of f_i norm should be 1, or $\|f\| = \sqrt{n}$.

- Solving the optimization problem: Using the Rayleigh quotient, it is to see that the solution is the eigenvector corresponding to the second smallest eigenvalue of L (recall that the smallest eigenvalue of L is 0 with eigenvector $\mathbf{1}$). Once we have f, we simply cluster all points by f (1-D clustering).

RatioCut for arbitrary k:

- Idea: express the RatioCut at the general cut, as the sum of $\text{cut}(A_k, \bar{A}_k), 1 \le k \le K$. Each of the K terms is related to the graph Laplacian. If we define a membership matrix $H = (h_{ik}), 1 \le i \le n, 1 \le k \le K$, where $h_{ik} = 1/\sqrt{|A_k|}$ if $v_i \in A_k$, and 0 otherwise, then we have:

$$\text{RatioCut}(A_1, \cdots, A_K) = \sum_{k=1}^{K} h_k^T L h_k = \text{tr}\,(H^T L H) \tag{7.116}$$

  The matrix H is subject to the constraint $H^T H = I$.

- Algorithm: do the relaxation, and solve the trace minimization problem. The resulting algorithm: find the top K eigenvectors (corresponding to K smallest eigenvalues) of L, and use the K-means algorithm to cluster the rows of the matrix of the K eigenvectors.

Ncut algorithm:

- Idea for $k = 2$: similar to RatioCut, define the cluster membership function f (in the case of $K = 2$), s.t. the quadratic form $f^T L f$ corresponds to the Ncut. Specifically, we are solving the problem:

$$\min_{f \in \mathbb{R}^n} f^T L f \text{ subject to } Df \perp \mathbf{1}, f^T D f = \text{vol}(V) \tag{7.117}$$

  This is the problem of Generalized Rayleigh Quotient, and it can be solved by the transformation, $g = D^{1/2} f$.

- Intuition of the optimization problem: similar to the RatioCut case, we could understand the contraints as: (1) The labels should be balanced with weighting, i.e. $\sum_i d_i f_i = 0$; (2) The weighted average of the norm of f_i should be equal to 1, or $\sum_i d_i f_i^2 = \sum_i d_i$.

- Algorithm in the general case: find the first K eigenvectors of the matrix L_{rw}, or the first K generalized eigenvectors of $Lu = \lambda D u$. Then cluster the n rows of the matrix consisting of the K eigenvectors.

Random walk interpretation of Ncut:

- Transition between clusters: suppose G is connected and non-bipartite, suppose X_0 is the stationary distribution, we define $P(B|A) = P(X_1 \in B | X_0 \in A)$, then we have:

$$\text{Ncut}(A, \bar{A}) = P(A|\bar{A}) + P(\bar{A}|A) \tag{7.118}$$

Proof: for any set A and B, we have:

$$P(A,B) = \sum_{i\in A, j\in B} P(X_0 = i, X_1 = j) = \sum_{i\in A, j\in B} \pi_i q_{ij} = \frac{1}{\text{vol}(V)} \sum_{i\in A, j\in B} w_{ij} \tag{7.119}$$

We could easily obtain $P(A)$, and thus the conditional probability $P(B|A)$. Plug in the terms for $P(A|\bar{A})$ and $P(\bar{A}|A)$, respectively.

- Random walk interpretation: minizing Ncut is equivalent to finding a partition of the graph s.t. the transitions between the two clusters have low probabilities.

How spectral clustering works?

- Community/clique structure: if there is a clique structure in the graph, then it is costly to break the elements of the clique into multiple components (large inter-cluster weights). Thus the algorithm will try to put any clique-like structure into the same cluster.
- Boundary at low-density regions: if a region has low density, then the cost of split the points in the region into multiple clusters is relatively low. Thus the boundary of the clusters from the spectral clustering algorithm tends to be in the low-density regions: for many problems, if the clusters are separable, then these should match the true cluster boundaries.

Considerations of spectral clustering:

- Lesson: in general, a statistical learning method makes some assumptions of the data, and if these assumptions do not hold, the performance of the method may be poor. It is important to: (1) understand the implications of these assumptions: what kind of solutions will be "favored" by the methods; (2) examine these assumptions in a problem, e.g. by examining some plausible special cases.
- Overlapped clusters: if clusters highly overlap, then the assumption that cluster boundaries are in the low-density regions does not hold.
- Different cluster sizes: the RatioCut and Ncut algorithms all make some implicit assumptions of the cluster sizes: e.g. RatioCut tends to favor equal cluster sizes. When the real cluster sizes are very different, this may create a problem.
- Different cluster densities: this would imply that the degrees of nodes within clusters are very different across clusters. This is related to the issue of how to define the similarity graph: e.g. a fixed K-NN graph may not be a good choice.
- Which graph Laplacian? L_{rw} is generally preferred, it minimizes the inter-cluster similarity, and also maximizes the intra-cluster similarity. It has a random-walk interpretation and better consistency properties than L_{sym}.

Relation to Laplace operator:

- The smoothness function: describe how variable f is across the graph. This is similar to the problem of describing how a function varies in a region. Let $w_{ij} = 1/d_{ij}^2$, then:

$$w_{ij}(f_i - f_j)^2 = \left(\frac{f_i - f_j}{d_{ij}}\right)^2 \tag{7.120}$$

is like a difference quotient. Then to minimize the function $\sum_{ij} w_{ij}(f_i - f_j)^2$ is like to minimize $\int_V \|\nabla f\|^2 dV$, and the solution satisfies the Laplace's Equation $\Delta f = 0$.

Questions of Spectral Clustering:

- How to choose the number of clusters? And how the eigenvalues can be used?

7.5 Manifold Learning

Visualizing Data Using t-SNE [Laurens van der Maaten, Google Talks], An illustrated introduction to the t-SNE algorithm, `https://www.oreilly.com/learning/an-illustrated-introduction-to-the-t-sne-algorithm`

- Embedding problem: represent high-D. data points in a low-D, such that the distance in high-D is preserved. The challenge is what distance metric to be preserved. For non-linear data, generally manifold distance is more preferred.
- Example: images of 10 digits. Why PCA does not work? PCA maximizes variance, it cares about large distance. Ex. two very different digits 0 and 1, should be separated in PC space (maximum variance requirement). But it does not force close points in high-D to be close in low-D.
- Idea of t-SNE: let x_i be original data, and y_i be low-dim. embedding. We define p_{ij} as distance of x_i and x_j in the original space, and q_{ij} in the low-D space. The idea is to place y_is s.t. p_{ij} are similar to q_{ij}; more precisely, if p_{ij}s are small, q_{ij} should also be small.
- High-dim. distance, we use Gaussian kernel p_{ij}. Because density varies at different data points, so use conditional Gaussian condition, $p_{j|i} \propto \exp(-\|x_i - x_j\|^2/2\sigma_i^2)$, where σ_i varies with i. Similarly define $p_{i|j}$, and the average of the two is p_{ij}.
- Low-dim. distance: similar idea, but use t-distribution. The idea is to use a long tail distribution: since we mainly care about preserving local structure. As long as two points close in high-D space close are also close in low-D, that's fine.
- Minimizing KL divergence between P and Q:

$$KL(P||Q) = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{7.121}$$

  The effect of this is: when p_{ij} is large, but q_{ij} small, then large penalty (we wrongly place distant points in the original space close to each other in the manifold). But when p_{ij} is small, and q_{ij} large, small penalty.
- Gradient interpretation of the algorithm (Physical analogy): N-body problem, the gradient of the objective function represents the force acting on an object. The solution is the equilbrium of the system.
- Computational efficiency: group data points that are close, and represent them by their centroid. Do this recursively (tree).
- Applications of t-SNE to the digit example: 10 clusters, with each representing a digit. The distance between clusters are not meaningful in t-SNE.
- More applications: use deep learning to extract features, then visualize with t-SNE.
- **Lesson**: to make inference, a different class of methods focus on inferring the underlying parameters/processes that "preserves" the structure of data, in terms of pairwise distance, instead of generating data itself.

7.5.1 Semi-supervised learning

Reference: [Zhu & Goldberg, Introduction to Semi-supervised Learning, 2009]

Why semi-supervised learning is possible?

- Better decision boundary: e.g. in a 1D case, given a small number of labeled instances, there are many possible boundaries to separate the positive and negative instances. Suppose we add the unlabeled data, and choose the boundary between two clusters, then the decision boundary is close to the real boundary.
- Label propagation: by propagating labels to nearest instances (most confident), we add new data points to training data, and this may help (self-training is a simple way of using this idea).
- Caveats: the unlabeled data may not always help, e.g. when the clusters highly overlap.

Inductive and transductive learning:

- Inductive learning: predict the labels on future test data.
- Transductive learning goal: predict the labels on the unlabeled instances in the training sample.

Self-training:

- Notation: (X_l, Y_l) are labeled data and X_u are unlabeled data, want to learn the function $f : X \to Y$.
- Algorithm:
  - Train f from (X_l, Y_l);
  - Predict on $x \in X_u$;
  - Add $(x, f(x))$ to the labeled data;
  - Repeat.

  An example is the propgating 1-nearest neighbor (1-NN) algorithm, which adds the nearest neighbor of each labeled instance in the unlabeled data at each step.
- Variations of the basic algorithm:
  - Add a few most confident $(x, f(x))$ to labeled data
  - Add all $(x, f(x))$ to labeled data
  - Add all $(x, f(x))$ to labeled data, weighting each by confidence
- Remark: the self-training algorithm makes the assumption that the high-confience predictions tend to be correct. However, when the algorithm makes a mistake at the beginning, the error will be amplified. The algorithm is sensitive to outliers, e.g. the ones that are close to both clusters, and easy to misclassifer.

Mixture model approach:

- The likelihood: suppose we have the labeled data, $(x_i, y_i), 1 \leq i \leq l$, and unlabeled data, $x_i, l+1 \leq i \leq l+u$. The log-likelihood is:

$$\log P(D|\theta) = \sum_{i=1}^{l} \log p(y_i|\theta) p(x_i|y_i, \theta) + \sum_{i=l+1}^{l+u} p(x_i|\theta) \tag{7.122}$$

  We then perform a mixture-model estimation plus the labeled data. EM algorithm can be similarly used. The intuition (for GMM): at each step, we update the mean of each cluster using all labeled data in that cluster plus all predicted data in that cluster weighted by the posterior probabilities.

- Extension: typically, we have a weighted log-likelihood so that the labeled data contribute more:

$$\sum_{i=1}^{l} \log p(y_i|\theta)p(x_i|y_i,\theta) + \lambda \sum_{i=l+1}^{l+u} p(x_i|\theta) \qquad 0 \le \lambda \le 1 \tag{7.123}$$

  As $\lambda \to 0$, the problem is reduced to supervised learning using only the labeled data.

Cluster-and-label approach:

- Idea: if two data points belong to the same cluster, then they are likely to share the same label.
- Algorithms: the basic framework is: cluster data into multiple clusters, then apply a classifier in each cluster using the labeled instances in that cluster. Some options for the classifier:
  - Majority vote: run clustering first on X_l and X_u, then label a data point in X_u by the majority of points in that cluster.
  - Cluster kernel: (used with discriminative learning framework) the kernel function of two data points depends on how often they are clustered together, supposing a clustering algorithm, e.g. K-means, is run multiple times (or on subset of data via sampling). [Weston & Noble, Bioinfo, 2005]

Graph-based method: MinCut algorithm:

- Model: let $f_i \in \{-1, +1\}$ be the predicted label of the i-th node, our goal is to minimize the cut:

$$\min_{f:f_i\in\{-1,+1\}} \sum_{i,j} w_{ij}(f_i - f_j)^2 \text{ subject to } f_i = y_i, 1 \le i \le l \tag{7.124}$$

- Algorithm: there exists efficient algorithm for solving the MinCut problem above (max. flow?).
- Remark: a number of problems with this version of MinCut, the multiplicity of solutions, lack of normalization, etc.

Gaussian random field (GRF) method:

- Reference: [Zhu & Lafferty, Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions, ICML, 2003]
- GRF model: suppose we have labels f, which satisfy $f_i = y_i$ for labeled data. We want to choose f on the unlabeled data s.t. the unlabeled points that are nearby in the graph have similar labels. This is accompolished by the energy function:

$$E(f) = \frac{1}{2} \sum_{i,j} w_{ij}(f_i - f_j)^2 \tag{7.125}$$

  The probability distribution of f thus follows the Boltzman distribution, $P(f) = \frac{1}{Z} \exp[-\beta E(f)]$. Instead of sampling from $P(f)$, we find f that minimizes $E(f)$.
  Note: in the Ising model, the energy function has the form $w_{ij} f_i f_j$, and this is different from GRF model.

- Solving the GRF model: we write the optimization problem in matrix form:

$$\min E(f) = f^T L f \text{ subject to } f_l = y_l \tag{7.126}$$

  where f_l and y_l represent the vector of f and y on the labeled data. We use the Lagrange multiplier method, define:

$$\Omega(f, \lambda) = f^T L f + \lambda(f_l - y_l) \tag{7.127}$$

Take the derivative of $\Omega(f, \lambda)$:

$$\frac{\partial \Omega(f,\lambda)}{\partial f} = 2Lf + \begin{bmatrix} \lambda \\ 0 \end{bmatrix} = 0 \tag{7.128}$$

We have the solution: $Lf = 0$ on unlabeled data, and $f_l = y_l$ on the labeled data. To have a closed-form solution of f, we solve $Lf = 0$ or $Wf = Df$ on the unlabeled data:

$$\begin{bmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{bmatrix} \begin{bmatrix} f_l \\ f_u \end{bmatrix} = \begin{bmatrix} D_l f_l \\ D_u f_u \end{bmatrix} \tag{7.129}$$

The solution: $f_u = (D_{uu} - W_{uu})^{-1} W_{ul} f_l$.

- Harmonic function analogy: this is similar to the problem of minimizing the Dirichlet energy of f on a region, and the solution is given by the Laplace's Equation, $\Delta f = 0$, i.e. the solution f is a harmonic function.

- Interpretation of the solution: expand $Lf = 0$ on the unlabeled data, we have, for any point j, its label f_j is the weighted average of the labels of its neighbors:

$$f_j = \frac{1}{d_j} \sum_i w_{ij} f_i \tag{7.130}$$

We could also write this as:

$$f = Qf \tag{7.131}$$

where $Q = D^{-1}W$ is the transition probability matrix. This leads to a fixed point iteration algorithm: starting with any random assignment, at each step, update the label of a node with the weighted average of its neighbors. To prove the convergence, we use the fact that the eigenvalues of Q is in $[-1, 1]$ (Perron-Frobenius Theorem).

- Normalization by class mass normalization (CMN): the problem with the GRF model is that it tends to produce severely unbalanced classification (the same reason that spectral clustering without normalization tends to produce very unbalanced clusters). To address this problem, we choose the decision threshold to reflect the prior ratio of positive and negative classes. Let q and $1 - q$ be the postive and negative class priors, and define the mass of class 1 as $\sum_i f_u(i)$, and the mass of class 0 as $\sum_i (1 - f_u(i))$. We classify a node i as 1 iff:

$$q \frac{f_u(i)}{\sum_i f_u(i)} > (1 - q) \frac{1 - f_u(i)}{\sum_i (1 - f_u(i))} \tag{7.132}$$

- Incorporating external classifier (label prior): suppose we have a prior of unlabeled data, h_u, e.g. from an external classifier. The h_u is similar to vertex potentials in the random field. We could incorporate h_u s.t. a node with high h_u is likely to be positive. To do this, we add an auxiliary node for each unlabeled i with label h_i, and add an edge between i and its auxilary node with weight η (and split the weights of the rest of neighbors).

Relation of GRF model to random walk and electric resistance:

- Relation to random walk: suppose our label is 0 or 1, and we define a random walk on G with the labeled instances as sinks. The label f_i is the probability that a particle, starting from the node i, hits a labeled node with label 1. To see this, let Q be the transition probability matrix, $q_{ij} = w_{ij}/d_i$, then the "sinking probability" of a node j is given by: (moving one step further)

$$f_j = \sum_i q_{ji} f_i = \sum_i \frac{w_{ji}}{d_j} f_i = \frac{1}{d_j} \sum_i w_{ij} f_i \tag{7.133}$$

- Relation to electric resistance network: suppose the labeled positive nodes are linked to a hihg-voltage point (say 1), and the negative nodes to the ground. Let w_{ij} be the conductance of the edge (i, j). Then f_i is the voltage of the i-node. To see this, we apply Kirchoff's Law on the node j:

$$\sum_i (f_i - f_j) w_{ij} = 0 \Rightarrow \sum_i w_{ij} f_i = d_j f_j \tag{7.134}$$

Issues/extensions of the basic GRF model:

- Allowing errors in the labeled data: obviously, the labels may not be perfect, to allow some disagreement with the given labels, we solve the problem:

$$\min_f \sum_i (f_i - y_i)^2 + \lambda f^T L f \tag{7.135}$$

- Balancing clusters: the objective function of the basic GRF algorithm is not normalized, thus it has the same problem as the unnormalized graph cut for spectral clustering. In particular, the classes (clusters) may not be balanced. A post-processing by CMN might alleviate the problem, but it's more preferred to change the objective function for normalization.

- Scaling of labels: similar to spectral clustering, the basic objective function favor small f_i's. The labeled instances may alleviate the problem, but cannot completely solve it, e.g. when the unlabeled nodes dominate and the clusters are not well separable.

Learning with Local and Global Consistency [Zhou & Scholkopf, NIPS, 2003]:

- Using normalized graph Laplacian: we want the label f_i to have roughly the same norm 1 (we are doing contiuous relaxation, and in the discrete version, f_i should be either 1 or -1), or in other words, the weighted average of f_i to be close to 1. The problem is in the weighted average, the nodes with high degree dominate, thus we divide f_i by $\sqrt{d_i}$, so that the contribution of each node to the weighted average is about the same (the weight is $\sqrt{d_i}$). And we know that:

$$f^T L_{\text{sym}} f = \frac{1}{2} \sum_{i,j} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 \tag{7.136}$$

  where $L_{\text{sym}} = D^{-1/2} L D^{-1/2}$.

- Model: we have a regulazation framework, (1) the fitting constraint: f_i should be close to known labels y_i (0 for unlabeled nodes); (2) the smoothness constraint: the labels of the neighbors should be close (with weighting). So we have the objective function:

$$Q(f) = \frac{1}{2} \sum_{i,j} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 + \mu \sum_i (f_i - y_i)^2 \tag{7.137}$$

  where $\mu > 0$ is a regularization parameter. Written in the matrix form:

$$Q(f) = f^T L_{\text{sym}} f + \mu (f - y)^T (f - y) \tag{7.138}$$

- Optimization: first we write $L_{\text{sym}} = D^{-1/2}(D - w)D^{-1/2} = I - S$, where $S = D^{-1/2} W D^{-1/2}$. It is given by:

$$S_{ij} = \frac{w_{ij}}{\sqrt{d_i d_j}} \tag{7.139}$$

Thus it is simply a normalized version of W, but unlike $Q = D^{-1}W$, the stochastic matrix, the normalization is symmetric wrt. rows and columns. The derivative of $Q(f)$:

$$\frac{\partial Q}{\partial f} = 2[(I - S)f + \mu(f - y)] = 0 \tag{7.140}$$

Solving the equation above, let $\alpha = 1/(1 + \mu)$:

$$f = \alpha S f + (1 - \alpha)y \tag{7.141}$$

The closed-form solution is: $f = (1 - \alpha)(I - \alpha S)^{-1}y$.

- Iterative algorithm: is more efficient in solving the linear equation, especially when the graph is sparse.
  - Background: to solve a linear equation $Ax = b$, we could use an iterative algorithm that converges to the correct solution. In general, suppose we have a recurrence equation: $x_n = Ax_{n-1} + b$, when the absolute value of the largest eigenvalue (absolute value) of A is less than 1, then the algorithm converges to $x^* = Ax^* + b$.
  - Iterative algorithm: to solve our problem, we define the recurrence:

$$f(t + 1) = \alpha S f(t) + (1 - \alpha)y \tag{7.142}$$

    Since $0 \leq \alpha < 1$ and the eigenvalues of S are in $[-1, 1]$ (S is similar to the stochastic matrix), the algorithm converges.

- Intepretation of the iterative algorithm: the algorithm performs label propagation: the lable of the node i in $t + 1$ is the weighted average of (1) the labels of its neighbors in t, and (2) its initial label (label bias). Note that the label propagation from the neighbors: is now weighted by s_{ij} instead of w_{ij} (symmetric).
- Remark: the graph does not have self-loops, i.e. $w_{ii} = 0$.

Questions of GRF model:

- The difference between different normalization methods (CMN vs. normalized graph Laplacian)?
- Setting the initial label bias y, and how it affects the results? How to control for class balancing?

7.6 Multi-Modal Machine Learning

A Survey of Multi-View Representation Learning [Li and Zhang, IEEE, 2019]

- Problem: learn common representations using paired data, e.g. document in two languages, image-text/caption.
- Multi-view representation alignment: we project both views in a common space, using $f(X)$ and $g(Y)$. The projections of two views from the same pair should be close. Ex. Cross-modal Factor Analysis minimizes the distance of two projections:

$$\min_{W_x, W_y} \sum_i (x_i^T W_x - y_i^T W_y)^2 + r_x(W_x) + r_y(W_y) \tag{7.143}$$

  where $r_x(W_x)$ and $r_y(W_y)$ are regularization terms.

- CCA: let X and Y be paired random vectors. After projection, we want projected X's and Y's to be correlated (smallest angle). Find vectors w_X and w_Y s.t. $\rho = \text{corr}(w_x^T X, w_y^T Y)$ is maximized. In the matrix form (averaging over all paired samples), let X and Y be data matrix, we want:

$$\max \text{corr}(w_x^T X, w_y^T Y) \propto w_x^T C_{XY} w_y \tag{7.144}$$

  where C_{XY} is the covariance matrix.

- Deep CCA: MLP applied to X and Y, the objective is to maximize the correlation of the hidden states.

- Multi-view representation fusion: learn a common representation to relate the two views. Graphical model approach: $p(x, y, z)$, and the representation is given by $p(z|x, y)$. Ex. probabilistic collective matrix factorization (PCMF), let u_i be common hidden variable, and x_i, y_i be two views, the model:

$$x_i \sim N(V_X u_i, \sigma_X^2 I) \qquad y_i \sim N(V_Y u_i, \sigma_Y^2 I) \tag{7.145}$$

- Multi-modal LDA: image and text data. For image data, N regions, and text data, M words. The image data at region n, r_n depends on the latent variable z_n (topic at region n), following MVN. For the word at position m, w_m: first needs to sample which region is belongs to y_m, and then use the topic at this region to sample words.

- Multi-modal DBM: use a common hidden layer as the common representations.

- Bimodal autoencoder: e.g. audio, video. Let $\hat{x} = f_\theta(x)$ be the output of the audio autoencoder, and $\hat{y} = g_{\theta'}(y)$ be the output of the video autoencoder. The two autoencoder share a common bottleneck layer. Choose parameters to minimize the reconstruction error for both x and y.

Multimodal learning with Deep Boltzmann Machines [Srivastava, JMLR, 2014]

- Motivation: we have paired image and text data (tags of images). The goal is to learn a model that can predict text of images and images from text.

- Multinomial output of RBMs: Replicated Softmax Model (Figure 2). To model words at a document: for each word, its multinomial probability, $P(v_i = 1|h)$, is a softmax function of hidden variables h. And the weights are repeated over all words.

- Multimodal DBM of joint image-text data (Figure 3): we have DBM for both image and text data (two hidden layers each). In addition, a hidden layer $h^{(3)}$ that is connected to $h^{(2)}$ of both image and text DBM. Intuition: need the hidden layer to connect the hidden representations of image and text. Ex. a 'nature' theme in the $h^{(3)}$ would correlate to words such as "river", "mountain", "tree" in text DBM, and to relevant image features in image DBM.

- Applications: (1) Learning joint representations: $P(h^{(3)}|v^m, v^t)$. (2) Impute missing modalities: $P(v^t|v^m)$, see Figures 5 and 6.

7.7 Misc. Topics in Machine Learning

1. Feature Selection

   Reference: [Guyon, JMLR, 2003]

   Motivations: choose variables to:

   - Improving the prediction performance;
   - Providing faster and more cost-effective predictors;

- Providing a better understanding of the underlying process that generated the data: the most important featurs.

Variable ranking:

- Correlation: correlation of varaibles and responses. Most commonly Pearson's correlation. In the case of classification, Pearson's correlation is similar to t-test. By the same token, other two-sample tests can be used for classification: Mann Whitney test, etc.
- Single variable classifiers: choose a threshold for the variable. The performance is measured by error rate, or some other criteria defined via FP rate and FN rate, e.g. break-even point (the hit rate for a threshold value for FP rate = FN rate), AUC of ROC.
- Information theoretical criteria: mutual information. Beause it needs to know the distributions of the variable and class label, it is most often applied to discrete cases. If continues variables, may discretize or use kernel density estimates.

Feature relevance and redundancy:

- Redundant variables can help each other: e.g. (Fig. 1) a variable Z that is a linear combination of variables X and Y (thus redundant), can increase the information gain (by projecting data in a differentn dimension, making them more dispersed in two classes).
- A useless variable, when used with other ones, can be useful: e.g. XOR function, each of the features is completely useless.

Variable subset selection:

- Types: wrapper method (utilize the learning machine of interest as a black box to score subsets of variable according to their predictive power); embedded method (perform variable selection in the process of training and are usually specific to given learning machines), e.g. decision tree method such as CART.
- Search strategies: forward selection, backward elimiation.
- Objective functions for evaluating variable subsets: goodness-of-fit, and regularization term.

Feature construction:

- Clustering: of features by their class labels.
- Matrix factorization: e.g. SVD - form a set of features that are linear combinations of the original variables.

Significant features/features subsets:

- Variable ranking: fake variables, e.g. random values, or permutation test (permutate the feature vectors).
- Model selection (choose variable subsets): validation (by partitioning training data into training and validating sets), or cross-validation.

2. Bioinformatics

   Factor regression model in gene expression data:

   - Model: suppose we are solving a regression problem $y = f(x)$, we assume all the observations (including predictors and response) are determined by a (smaller) set of latent variables. Then we have a joint linear model of (X, Y) as functions of Z (latent variables).

- Sparse factor model in genomics [High-dimenstional sparse factor modeling: applications in gene expression genomics, JASA, 2008]: suppose for the i-th sample, we have k latent variables, λ_i, representing the hidden pathway/module/regulator activities. Then expression of any gene g in the i-th sample is thus:

$$\mathrm{E}\left(x_{g,i}\right) = \mu_g + \alpha_g^T \lambda_i \tag{7.146}$$

  where μ_g is the intercept of the gene g, and α_g is the effect of λ on the gene. And the class of the i-th sample is similar:

$$\mathrm{E}\left(y_i\right) = \mu_y + \alpha_y^T \lambda_i \tag{7.147}$$

  This is a joint linear model of p genes and class label on the k latent varaibles. For the model to be identifiable, need to add constraints, including sparse factor loading (each latent factor is associated with only a small set of observed variables).

Context-specific independence mixture modeling for positional weight matrices [ISMB, 2006]:

- Idea: to learn a mixture of PWMs from sequence data, if assume each component of the mixture model has its own PWM, overparameterization. One may assume that different PWMs may share distributions in certain columns. This would reduce the model complexity.
- Model: suppose there are K PWMs to learn, and there are p columns. We assume a structure G represents the sharing of columns among the K PWMs: i.e. at position j, G_j is the partition of K into groups, where each group has the same distribution at the position j. We define the prior distribution on G that favors simpler models (more sharing)

$$P(G) = \prod_j \alpha^{z_j} \tag{7.148}$$

  where z_j is the number of groups in the partition G_j and $\alpha < 1$ is a hyperparameter.
- Inference: learn the model $M = (G, \theta_M)$ through an iterative algorithm, update/sample G at each cycle according to the posterior distribution, and the parameter then can be estimated through EM.
- Remark: the CSI idea is that one may define a strcuture to represent the commonality/sharing among distributions, and a prior on the structure that favor simpler ones. For example, this could be applied to a dynamic model, where coefficients may change over time $\beta(t)$, but tend to stay the same. The associated structure can be represented by a binary matrix M, where M_{jt} denotes if β_j changes at time t (if change, then resample β_j using prior).

3. Network models

   Supervised random walks: Predicting and Recommending Links in Social Networks [Backstrom & Leskovec, WSDM11]:

- Motivation: link prediction in social networks. Depends on both network topology (closeness) and attributes of nodes/edges. Ex. if two people share a lot of friends, then the two are also likely to be friends (network topology); if the two work in the same company, the chance they are friends is increased (attributes).
- Random walk with restart model: let a_{uv} be the weigth of the edge (u, v), then Q'_{uv} is the normalized transition probability from the node u to v:

$$Q'_{uv} = \frac{a_{uv}}{\sum_w a_{uw}} \tag{7.149}$$

  And $Q'_{uv} = 0$ if (u, v) is not linked. To keep the random walk around the starting node, we introduce the restart probability α, i.e. the probability of jumping back to the seed node s at each step. Thus the true transition probability is:

$$Q_{uv} = (1 - \alpha) Q'_{uv} + \alpha \mathbf{1}(v = s) \tag{7.150}$$

- Incorporating attributes: let Ψ_{uv} be the node and edge attributes of (u, v). We assume the edge weight is a function of these attributes, $a_{uv} = f_w(\Psi_{uv})$. The goal is to estimate w s.t. (given a source node s), the stationary probability $p_d > p_l$, where d is some node that are linked to s and l is the node not linked to s (training data).
  - Edge type: an important attributes. For an edge (u, v), define its edge type as $(0, 1)$, $(1, 1)$, etc., depending on the distance of u and v to the source node s. It was found that the edge type is an important attribute.
- Optimization: the objective function is a function of w, taking the above criterion into account, with regularization term (s.t. few attributes have non-zero weight in f). This can be achieved by any method using the gradient, where the gradient can be computed in an iteriative fashion (similar to PageRank).

4. NLP

   Named entity recognition: [ACL, 2011]

   - Background: in general to recognize mentionings with entities, use the discourse, background knowledge or string similarity.
   - Idea: a graphical model of strings (mentions), define distance between string, and define potentials: affinity between close strings, and repulsion between distant strings. The method essentially do clustering on strings.

   Unsupervised predicate extraction from documents [ACL, 2011]:

   - Idea: a generative model of predicate sentences consisting of (1) arguments; (2) syntactic realization.

5. Computer Vision

   Image cosegmentation: [Distributed Cosegmentation via Submodular Optimization on Anisotropic Diffusion, Gunhee Lee et al., ICCV, 2011]

   - Background: image segmentation - extract salient features/divide an image into areas with salient features. Image cosegmentation - from multiple images, segment them simutaneously s.t. the features in different image align with each other (e.g. the same kind of objects).
   - Application of heat diffusion models in image analysis: when putting heat source in some space, heat will diffused according to the heat equation, and the space with high thermal conductivity will tend to have similar temperatures. This is analogous to image analysis: if we map similarity between adajcent pixels to thermal conductivity, and the clustering of pixels (i.e. similar pixels belonging to the same clusters) to the temperature distribution, then we have regions with high similarity should be put into the same cluster.
   - Remark: this is potentially much faster than MRF, which fails to take into account the spatial structure (each pixel is treated independently).

   Hierarchical model for image classification:

   - Reference: A Bayesian Hierarchical Model for Learning Natural Scene Categories [ICCV, 2005]; and some recent papers
   - Idea: to recognize an object in an image, we may be able to recognize the common feature of a class of objects; and identifying the class would help identify individual objects (the class provides prior for individual objects). This can be captured by a hierarchical model.
   - Example: a class of objects: animal, vehicle; and objects within animal class: horse, cow, etc.

6. Maximum margin methods

   Graphical model learning by max margin and max entropy [Jun Zhu's work, 10/17/2011]:

   - Background:
     - Use graphical model for prediction (multi-class labels), the structure of input-ouput variables.
     - Models: CRF, maximum margin Markovian network (M3N).
     - Challenges of graphical models: sparseness, prior, allow latent variables, stationarity, etc.
   - Maximum entropy discrimination (MED) learning [Jaakkola, 1999]
     - Classical predictions: minimize loss function plus some penalty for regularization, e.g. logistic regression (squared error) and SVM (hinge loss)
     - Comparison of likelihood and max-margin methods: likelihood method is easy to allow latent variables, etc.; max-margin is free of probabilistic models.
     - Motivation: combine the advantages of the two, e.g. averaging the parameters in the max-margin methods (Bayesian perspective), mixture of SVM models, etc.
     - Maximum entropy discrimination (MED) learning (Jaakkola, 1999): prediction of label y is the predicted value averaged over the parameters w. The learning of parameters is the minimization of KL divergence between the prior of the parameters and posterior (?) of parameters, subject to some contraints.
   - Structured MaxEnt Discrimination: extend the MED learning to graphical models.
     - Under some specific conditions: the loss function (linear function) and the linear slack variables, and uniform prior of parameters, then Structured MaxEnt reduces to M3N.
     - Extension to models with latent variables.
     - Application to supervised learning with unlabelled data: traditionally, LDA + SVM. The new method would guide the LDA process towards better discrimination.
   - Infinity SVM: infinite number of mixture components of SVM models.

7. Multi-class problems

   Hierarchical binary classifiers: [Koller]

   - Idea: suppose we want to classify objects into K classes, say 1 to 6. There is structure in the K classes, e.g. classes 1 and 2 are more related; 5 and 6 are more related, etc. To exploit this structure, we could first build a binary classifier that classify classess 1,2 (positive), 5,6 (negative) and 3,4 (ignore); then repeat this process (another binary classifer) on objects in classes 1,2,3,4 and on objects in class11429es 3,4,5,6; and so on.
   - Objective function: at each level, learn a labeling of classes (positive, negative and ignore) while simulatenously perform binary classification.
   - Remark: each level is independent, thus the errors in the top level would be propagated into lower levels.

Chapter 8

Artificial Neuron Networks

Reference:

- Neural Networks for Machine Learning [Hinton, Coursera, 2016]
- Stanford Computer Vision class, `http://cs231n.stanford.edu`.
- Goodfellow, Deep Learning, 2016

Chapter 1: Introduction [Goodfellow et al]

- Deep learning approach: Create new **representations**. Ex. from pixel representation to geometric shapes. Mathematically, new representation may make the problem much easier, e.g. linearly separable.
- Deep learning approach: Make **abstractions**. The challenge is often the "factors of influences", e.g. age, gender and accent in speech. We create new representations that are invariant of such factors. The new representations are defined in terms of simpler representations: eg. edge in terms of pixels, and corners/contours in terms of edges, and object parts in terms corners and contours.

Deep learning [LeCun & Hinton, Nature, 2016]

- Representation learning: existing classifiers require expert-curated "features". Learning these features itself is what's called **representation learning**. For example, in vision, we have multiple layers of neurons: one layer learns edges, the next layer learns shapes from the previous layer (combination of edges), and the next layer learns objects, and so on.
- Why deep learning can do better than conventional "shallow" classifiers? The challenge of shallow classifiers is that it does not have a representation of high-level features. Ex. in vision, pixels are the basic features, and edges represents combination of these features. In linear classifiers, one would need to encode them as "interaction terms", which cannot be made very complex.
  - Challenge of machine learning: e.g. learn images of wolf vs. a dog breed that is similar to wolf. Without feature extraction, two images of the same wolf can be much more different (e.g. in different positions, illumuniations) than two images of wolf and dog.
  - Non-linearity of NN: as shown in Figure 1, even if the input functions are non-linear (i.e. positive and negative class has a non-linear boundary), the hidden layer can distort the input s.t. they become linearly separable.
  - The limitation of kernel methods: cannot generalize far enough beyond existing training examples (because they are based on similarity with existing examples).
- Pre-training and resurgence of ANN: pre-training is a form of unsupervised learning that learns representation. This is achieved by minimizing "reconstruction error". Autoencoder?

- Convolutional neural networks (ConvNets): designed for array data, e.g. image. Its design: between two layers, not fully connected. It has two features:
  - Local features: e.g. edge is a feature derived from local pixels. In the convolutional layer, we have feature maps, where each feature map is linked only to some feature maps in the previous layer, representing a local motif (the links are called filter-banks).
  - Semantic similarity of features: slight variation of a feature (e.g. shifting position) can be captured by the same neurons. This is achieved by a pooling layer.
- Recurrent neural networks: designed for sequential/time series data. To predict output of the next element, we have neurons for the current input x_t, and neurons representing all past input s_{t-1}. The result is the node s_t. RNN uses all the data to train the weights (if unfolded wrt time, we have many layers correpsonding to each time point, with shared weights).
  - Importance of memory: proposal that we create memory cells to represent the past.
- Future of deep learning: (1) Importance of unsupervised learning. (2) Combination of representation learning and complex reasoning.
- Question: ConvNets, how to represent a feature, e.g. edge, at different positions?

Can we open the black box of AI? [Nature, 2016]

- Deep Dream: e.g. a deep NN trained for recognizing animal faces. Given input of flower, continue to modify the input images to increase the response rates. After some iterations, we see images where animal faces emerging in flowers - halluciation.
- General problem: neural networks are surprisingly easy to fool with images that to people look like random noise.

Lecture 1: Introduction to Deep Learning [Coursera]

- Examples of machine learning tasks: handwritten digit recognition (MNIST database), ImageNet (1000 classes, millions of images), speech recognition.
- How brain works? Each neuron receives multiple inputs (dendrites) and has one output (axon). Synapses: the strength can be changed (adapt). Human brain: 10^{11} neurons and each 10^4 weights.
- Model of neurons (activation function): let x be input and $z = \sum_i w_i x_i$ be the weighted sum of input, and $y = f(x)$ is the activation function. It can be linear, binary, rectified linear, sigmoid or stochastic binary.
- Example of handwritten digit recognition: for each digit, two layered neurons, one for input the other output. The learned weights match the template of the digit. However, cannot capture variations.
- Unsupervised learning: learn internal representation of data. Main applications:
  - Useful for later supervised learning.
  - Low-dim. representation of data. PCA is one very limited example.
  - Denoising data: economic representation of input.
  - Clustering: if one view each cluster as a feature, then clustering is a simple sparse coding, where for each sample, only one feature is non-zero.

8.1 Feedforward Neuron Networks

Lecture 2: Perceptron

- Types of NN: feedforward neuron networks and recurrent neuron networks (RNN). RNN is designed for sequential data: it allows neurons to form circles. RNN is similar to very deep NN, where each layer corresponds to one time point (but the same weights).
- Perceptron: binary threshold neurons, two layers, input-output. Its output is 1 if $w \cdot x + b > 0$, and 0 otherwise. We could transform the perceptron s.t. it has an extra input with constant value 1 (then we remove the bias term).
- Geometric intuition of perceptron: we consider the weight space. For each x_i, suppose $y_i = 1$, then we should have $w \cdot x_i > 0$ (ignoring threshold). This poses *constraint* on w: it must be in one side of the hyperplane defined by $w \cdot x_i = 0$. Each x_i limits the possible locations of w. Ex. for x_1 and x_2, the feasible set may be a cone. The problem is thus to determine the feasible region/solution from all constraints posed by x_i's.
  - Remark: this is basically a linear programming problem. And when we maximize the margin while finding a feasible set, this becomes SVM.
- Perceptron learning algorithm: if we make a mistake with x_i using the current weight w, we add or subtract x_i to w, depending on the mistake. Intuitively, if $y_i = 1$, but our $w \cdot x_i < 0$, we make $w' = w + x_i$, then $w'x_i = w \cdot x_i + x_i \cdot x_i$, and its becoming more positive. Proof idea: we can always choose a solution in the "generous feasible" region (margin at least x_i), then this rule will guarantee that the weight will become closer to the target solution.
  - Remark: the difficult part to understand, how do we know that a new w will not violate previous x_i's.
  - Example 1: a single x_1 with $y_1 = 1$. The rule will keep increasing $w \cdot x_1$ until $w \cdot x_1 > 0$.
  - Example 2: we have two data points x_1 and x_2, repeatedly. Suppose the feasible set is the cone defined by two input vectors, one can see that eventually w will reach the feasible region.

  One can show that the objective function is convex.
- Limitations of perceptron: cannot learn XOR (not linearly separable), and the example of translation wrap-around (which forms a group). Intuition: for each w_j, its value depends on how many times the corresponding x_j is activated in positive vs. negative examples. The perceptron cannot distinguish two patterns if a pixel is equally likely to be activated. Group Invariance Theorem - cannot distinguish transformations of a pattern.
- Questions: is RNN biologically motivated?

Back-propagation [Wiki; personal notes]:

- Derivative of logistic function: suppose we use logistic function as our activation function, we have this simple result:
$$\frac{dy}{dx} = y(1-y) \tag{8.1}$$
- Our NN is $f(x, w)$ where w needs to be learned. The error is $E(w) = \frac{1}{1}\sum_i (f(x_i; w) - t_i)^2$, where t_i is the target output of input x_i. We estimate w via gradient descent. Let w_{ij} be the weight of the link from neuron i to j. For neuron j, let o_j be its output, z_j be its total input, we have:
$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial z_j}\frac{\partial z_j}{\partial w_{ij}} \tag{8.2}$$

The second term is determined by the activation function of neurons:

$$\frac{\partial o_j}{\partial z_j} = o_j(1 - o_j) \tag{8.3}$$

The third term is simply o_i. For the first term, it is simple when j is in the output layer. If not, we note that E is a function of o_u, where u is a neuron in next layer of j, and o_u depends on o_j (this is how o_j may affect output). We use the chain rule to write:

$$\frac{\partial E}{\partial o_j} = \sum_u \frac{\partial E}{\partial o_u} \frac{\partial o_u}{\partial z_u} \frac{\partial z_u}{\partial o_j} \tag{8.4}$$

Plug in the relevant terms:

$$\frac{\partial E}{\partial o_j} = \sum_u \frac{\partial E}{\partial o_u} o_u(1 - o_u) w_{ju} \tag{8.5}$$

This gives the recurrence in terms of $\partial E/\partial o_j$, and we can solve them using dynamic programming. Once we have the gradient, we update w_{ij} using gradient descent: $\delta w_{ij} = -\alpha \frac{\partial E}{\partial w_{ij}}$, where α is the step size.

- Algorithm: at each iteration, we first have the forward pass to obtain o_j for every neuron j; next we use backward pass to obtain the derivatives $\partial E/\partial o_j$, and the derivatives wrt. the weights. Then we update the weights using gradient descent.
- Analysis: why backpropgation is better than simple numerical differentiation? Let m be the number of edges and n be number of neurons. Simple numerical derivative needs $O(mn)$ computations (compute m times, and each time, compute the whole NN). While backpropgation needs $O(n)$ steps for computing $\frac{\partial E}{\partial o_j}$ and $O(m)$ steps for each of the m edges.
- Backpropagation with constraints: sometimes (e.g. in CNN or RNN), we have constraints on parameters, say $w_1 = w_2 = w$. To implement these constraints, we use this as gradient on w:

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial w_1} + \frac{\partial E}{\partial w_2} \tag{8.6}$$

So we do the same backpropagation on w_1, w_2, but use the sum of gradients on all w_i's as the gradient of w.

Lecture 3: Backpropagation [Coursera]

- Iterative optimization: an optimization problem, e.g Least square problem, can often be solved by iterative algorithm (even if other solutions exist). They may be slower, but very easy to generalize.
  - Different versions of iterative algorithm: batch version (using all training examples to update parameters), and online algorithm (update parameters after each example).
  - Pictures of iterative algorithm: gradient descent would move parameters along the direction of gradient, while online algorithm move in a zig-zag fashion.
- Delta-rule for linear neurons: consider input-output NN (single output). Let i be index of example and j be index of feature. Our error $E(w) = \frac{1}{2}\sum_i (t_i - y_i)^2$, where $y_i = \sum_j w_j x_{ij}$. It is easy to show that:

$$\frac{\partial E}{\partial w_j} = -\sum_i x_{ij}(t_i - y_i) \tag{8.7}$$

The update rule is $\delta w_j = -\epsilon \frac{\partial E}{\partial w_j}$. So the update is similar to perceptron learning (proportional to input vector), except that the change is weighted by learning rate ϵ and the difference between t_i and y_i.

- Delta-rule for logistic neurons: let $z_i = \sum_j w_j x_{ij}$ be the total input of the output neuron for example i. We can use chain rule to derive:

$$\frac{\partial E}{\partial w_j} = \sum_i \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial z_i} \frac{\partial z_i}{\partial w_j} = -\sum_i x_{ij} y_i (1 - y_i)(t_i - y_i) \tag{8.8}$$

  So it's similar to delta rule except the extra weight $y_i(1-y_i)$, which is the slope of the logistic function.

- Backpropagation: Intuition of why backprogation works: simple numerical derivative wrt. w_j is equivalent to perturb every weight. Backprogation effectively perturbs the activity of each hidden neuron: we consider $\partial E/\partial y_i$, where y_i is the activity of neuron i. We can compute this derivative for all neurons simultaneously. As the number of hidden neurons is smaller in number than that of weights, the algorithm more efficient. Formally, let y_j be the output of i-th neuron and z_j be its total input, and w_{ij} be the weight of neuron i to j (assuming there is only one training example). Using chain rule:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial z_j} \cdot \frac{\partial z_j}{\partial w_{ij}} \tag{8.9}$$

Automatic differentiation (autodiff) [Wiki] Automatic Differentiation, Explained [Towards Data Science]

- Idea: to compute the derivative of a function $y = f(x)$, we write it as some function g of $z = h(x)$, we can then apply the chain rule to reduce the derivative to some simpler derivatives, which are easier to compute. The rule can be repeatedly applied. More generally, we represent the computation as a graph.

- Note: different from symbolic calculation, which can be very cumbersome. Also different from numerical method (finite difference), which could have large numeric errors.

- Example: see Figure 2, Automatic Differentiation, Explained [Towards Data Science]. Suppose we have $z = y - \max(0, wx+b)$ and we want to compute $\partial z/\partial w$. We build a computation graph: where each leaf node is the "input": w, x, b, y and the output node is z. The internal node represents the intermediate expressions. Next, we compute the partial derivatives of each connection (output vs. input), evaluated at a given input. With these, we can then compute $\partial z/\partial w$ in terms of these connections.

8.2 Convoulutional Neuronal Networks

Lecture 5, Coursera

- Challenges of object recognition in images: the main challenge is that an object might have different appearances, because of:
  - Lighting: affecting pixel intensity.
  - Viewpoint: esp. a problem for 3D object.
  - Deformation/abstraction: e.g. a chair may have different shapes. Another example: tri-angles.

  Other challanges may include: segmentation (other objects in images).
  - Perspective: the appearance of an object is some transformations of the original object. Our problem is to find such "inverse transformation".

- Ideas for addressing object invariance:
  - Normalization by putting a box: the challenge is of course without knowing the object, it's hard to know where to put the box. Idea: try all box positions.

  - Replicate features (ConvNet) s.t. different neurons represent all possible variations of the same underlying feature.

- ConvNet:
  - Convolution layer: each neuron represents a feature in a particular position (possible variation).
  - Pooling layer: pool neurons from convolution layers, this would indicate whether a feature is present.
  - Fully-connected (FC) layer: topmost layers, combine information of all features.

  The problem is that the position information is lost after pooling, which can be important when we recognize composite object (relative positions of multiple basic objects matter). This may be solved by representing features in particular regions (after pooling, a neuron represents if a feature is present in a given region).

- Training of ConvNet: backpropagation with constraint. Ex. to satisfy $w_1 = w_2$, we calculate $\frac{\partial E}{\partial w_1} + \frac{\partial E}{\partial w_2}$ and use it to update both w_1 and w_2.

- Role of prior knowledge: in general, we use prior knowledge in two ways: (1) architecture of the NN; (2) training data. One idea is to enhance training data, by generating variations/transformations from the original training data.

- Example: recognition of "+" in an image, allowing positional variation. Suppose we have a 10 by 10 image, and the sign is 3 by 3. We first divide the image into 3×3 regions. In convolution layer, each neuron is connected to a 3 by 3 region, and is activated when a plus is present. In the pooling layer, one neuron to represent whether the plus sign is present in any of the regions.

- Example: recognition of circle on top of plus signs. Convulution layer: recognize circle and plus signs. Pooling: circle (with region information, i.e. pool only circles within a region) and plus. FC layer: match circle and plus in the same regions.

- Success stories: LeNet for hand-digit recognition and ImageNet. The key tricks of ImageNet include: ReLU, dropout (remove half of the neurons during training), hardware.

- Question: how to model scale invariance? Ex. rectangle, where the lengths of edges can vary.

Stanford lecture notes: ConvNets (Lecture 4)

- Convolution layer: we have filters, which represent features. Each neuron of a filter represents a feature in a particular region (size defined by *receptive field*). Each filter has many neurons, with adjacent neurons defined by *strides*. All filters of a given region are called the *depth column* (representing all features in this region). The parameters of these strides, number of filters, etc. are hyperparameters.
  - Volume analysis: volume of a layer is defined by number of elements in width $\times$ height, and depth (for input: number of colors; for other layers, number of filters).

- Pooling layer: reduce the representation in convolution layers. Ex. a neuron in the pooling layer may represent 4 adjacent regions. MAX is most common.

- Architecture: commonly, convolution, then pool, and fully-connected (FC) as the last layer. Pooling layer may be less important.

- Question: we only defines filters in conv. layer, how do we ensure that we actually learn distinct features? And what are the features?

8.3 Sequential Neural Networks

Lecture 6-7, Coursera

- Linear Dynamic System and HMM.
- RNN model: let y_t be the neuron activity at time t, we have $y_t = f(y_{t-1}; w)$, where f is the activating function and w the weights. It is similar to a feedforward NN except that the weights are constrained to be the same across time.
- Architecture of RNN?
- Training RNN: we can use backprop, but the challenge is the exploding or vanishing gradients. In backprop, the relationship between gradients in successive layers is linear (not logistic), thus the weights can get exponentially large or small over many layers (many time steps).
- Long short-term memory (LSTM): a memory cell with self-weight of 1 (thus in each time step, keep its value). It has multiple gates: Write, Keep and Read.

Lecture 10 [Stanford]

- Problems solved by Recurrent Neuron Networks (RNN): some examples
  - One to many: image caption.
  - Many to one: sentiment of sentence.
  - Many to many: machine translation.

  The problem differs from image recognition problem in that: one needs to have a representation of what has been seen so far (not all at once); and the sequence can be arbitrarily long.
- Design/architecture of RNN: we consider the case where we have $x = \{x_t\}$ as input, and y as output, which could be single (or a few), or a sequence as well. The key elements of designing RNN:
  - Use hidden neurons to represent what has been seen at a point. The hidden neuron should depend on the hidden neuron in the previous time step and the current input.
  - The functions of how output depends on the hidden neurons, and how hidden neurons depend on previous time step and input should be invariant to time.

  We denote x_t, h_t, y_t as input, hidden neuron and output at time t (each a vector), then we have:

$$h_t = f_W(h_{t-1}, x_t) \qquad y_t = g_W(h_t) \tag{8.10}$$

  Consider a specific form:

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \qquad y_t = W_{hy}h_t \tag{8.11}$$

  Ex. predicting next character: the input vocabularoy is 26, and let's say there are 100 hidden neurons. Then the dimensions of W_{xh}, W_{hh} and W_{hy} are 26×100, 100×100 and 100×26 respectively.
- How RNN works- what do the hidden neurons represent? In the character prediction example, hidden neurons generally represent the information of sentence we have seen. Ex. we may have a neuron for the prefix "ho", and two neurons for the words "hors" and "hous": they are activated by the letter "e".
- Training of RNN: how would backpropagation work? We do forward pass through all time steps to obtain h_t and y_t. Then do backward pass through time to obtain gradients. Q: constraint that W is constant over time?

- Connecting CNN and RNN: image captioning. The output of CNN (not the final output, but the layer before) is supplied as the first hidden layer h_0 (time 0) of the RNN. Then we use RNN to generate the sequence of words as caption: we use the current output (sampling) as the input of next time step. Intuitively: the hidden neurons have two functions: representing image information and record what has been produced in the sentence.
- Challenge of RNN: vanishing gradient problem. LSTM.

RNN, LSTM and Transformer [Ben Lai, Group meeting, 2020]. Visualizing A Neural Machine Translation Model [Jay Alammer] `https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2se`

- Motivation of sequential model: suppose we have DNA sequence and ChIP-seq data of whether a TF is bound. Our goal is to predict TF binding (experimental data) from DNA sequences.
- RNN: at each time step, input x_t, and previous hidden state a_{t-1}, and compute the new hidden state a_t, and the output y_t as a function of a_t. All the parameters are shared across time steps. In TFBS example: our hidden state records the motifs read so far, and if we enter a new motif, we update hidden state (in motif) and predict output (TF binding). If we exit a motif, we update hidden state (out of motif now) and predict output (no TF binding).
- Problem of RNN: hard to capture long-term dependency, and vanishing gradient.
- Motivation of LSTM: e.g. CTCF as boundaries of enhancers. To predict TFBS, only the sequence in an enhancer is relevant.
- LSTM: at each step, the hidden state is more complex: it has 4 gates, one gate allows you to forget (reset hidden states). LSTM still slow and ineffective for long sequences.
- Applications in genomics: DanQ, combine CNN and LSTM. The output of CNN: hidden features, are used as inputs of LSTM. Learning deep protein representation (prediction of missing residuals) and applications in protein engineering.
- Seq2seq model (Neural machine translation): encoder and decoder, both RNNs. Input: word embedding. (1) Encoder: input to context vector. (2) Decoder: context vector to output (another language). Note: similar to autoencoder, context is the bottleneck layer.
- Attention: the idea is to focus only on relevant parts of the input sequences. Use all hidden states as input of the decoder (all times steps): however, weigh the time points - hence Attention. Then use the sum of the weighted hidden states as the context vector in the Decoder. At each time step of the Decoder, use different weights, thus focusing on different parts of the input sequence.
- Transformer: see, The Illustrated Transformer, `http://jalammar.github.io/illustrated-transformer/`

8.4 Deep Generative Models

Boltzmann machines [Chapter 20, Deep Learning, Goodfellow et al, 2016]

- Goal: model probability distribution over d-dim. binary vectors. This can be done with MRF:

$$P(x) \propto \exp(-E(x)) \qquad E(x) = -x^T U x - b^T x \tag{8.12}$$

  where $E(x)$ is the energy function.
- Boltzmann machines: when not all variables are observed. Let h be the hidden variables and v be observed ones, or $x = (v, h)$. Then the energy function becomes:

$$E(v, h) = -v^T R v - v^T W h - h^T S h - b^T v - c^T h \tag{8.13}$$

  This allows one to compute $P(v|h)$ and $P(h|v)$.

- Restricted Boltzmann machines (RBMs): a special type of Boltzmann machines where all the v's are in the same layer, and all the h's are in the different layer. Only connections across the layers, but not within a layer - Figure 20.1 (a). The energy function is simpler: $E(v,h) = -b^T v - c^T h - v^T W h$. The conditional distribution has closed form. This enables efficient block Gibbs sampling.

Deep belief Network (DBN) and Deep Boltzmann machines (DBM) [ibid]

- DBN: multiple hidden layers (undirected network), and directed graph from h to v (observed). Figure 20.1 (b). Fall out of fashion now.
- DBM: multiple hidden layers and one observed layer, all undirected network, Figure 20.1 (c). Inference: use VB mean-field approximation.
- Gaussian-Bernoulli RBMs: generalize to real valued data. This can be simply modeled as observed data is a linear function of hidden layers: $P(v|h) = N(v|Wh, \beta^{-1})$, where β is the precision matrix.

Variational auto-encoders (VAE)

- Ref: `https://jaan.io/what-is-variational-autoencoder-vae-tutorial/`. [Tutorial on Variational Autoencoders, arxiv, 2016]. Alan Selewa presentation.
- Background: stochastic gradient descent (SGD). In gradient descent, suppose we want to optimize $Q(w)$, our update rule for parameters:

$$w := w - \eta \nabla Q(w) = w - \eta \sum_i \nabla Q_i(w)/n \tag{8.14}$$

  where $\nabla Q_i(w)$ is the gradient at sample i. When the number of training samples is large, a single iteration of gradient descent is slow. SGD updates w immediately after seeing one training example:

$$w := w - \eta \nabla Q_i(w) \tag{8.15}$$

  So SGD can potentially be much faster. It may osculate between different values, but in practice, they are close to the local (or global) optimum.

- Background: autoencoder [Hinton and Salakhutdinov, Science, 2006]. Figure 3: better classification of digits than PCA.
- Neural nets perspective: we have an encoder network x to latent variable z: denoted as $q_\theta(z|x)$ and decoder network z to x: denoted as $p_\phi(x|z)$. The objective (loss) function for a single example x_i is given by:

$$l_i(\theta, \phi) = -\mathrm{E}_{z \sim q_\theta(z|x_i)}[\log p_\phi(x_i|z)] + \mathrm{KL}(q_\theta(z|x_i)||p(z)) \tag{8.16}$$

  where $p(z)$ is the prior distribution of z. The loss function has two parts: (1) reconstruction loss at x_i: the encoder and decoder network should be chosen s.t. the probability of generating x_i is large. Note that this prob. is averaging over the distribution z given x_i. (2) Regularization: the posterior of z given x_i should be close to the prior.

- Probability model perspective: VAE as a graphical model: $z \sim N(0, I)$ (prior), and $x_i|z \sim p(x|z)$. Why this is a good model? Even with simple normal distribution of z, we can generate very complex distributions. See the example in Tutorial: 2D normal distribution generates data points in a ring.
- Variational inference of VAE: let ϕ be the model parameters. Our goal should be to find ϕ that max. $P(x)$. However, this is difficult. Our intuition is to use $q_\theta(z|x)$ to approximate the posterior $P(z|x)$ by min. KL divergence between the two. We can express the KL divergence in terms of ELBO:

$$\log P(x) = \mathrm{ELBO}(\lambda) + \mathrm{KL}(q_\theta(z|x)||p(z|x)) \tag{8.17}$$

where λ indexes the q distribution and the ELBO function is defined as:

$$\text{ELBO}(\lambda) = \text{E}_{z\sim q_\theta(z|x)}[\log p_\phi(x|z)] - \text{KL}(q_\theta(z|x)||p(z)) \tag{8.18}$$

Since $\log P(x)$ is constant (independent of θ), so min. KL divergence is equivalent to max. ELBO. Since ϕ is not known, we maximize ELBO in terms of both θ and ϕ: this maximizes $P(x)$ while minimizing KL divergence q and $P(z|x)$. The objective function for SGD of a single example is then:

$$\text{ELBO}_i(\theta, \phi) = \text{E}_{z\sim q_\theta(z|x_i)}[\log p_\phi(x_i|z)] - \text{KL}(q_\theta(z|x_i)||p(z)) \tag{8.19}$$

So our objective is to maximize data likelihood (first term), with regularization (second term).

- Form of $q(z|x)$ function: normal distribution, mean and covariance depends on data and θ. This is the encoder network.
- Example: image generation, the pixel (0 or 1) is sampled from Bernoulli distribution. The q function is:

$$p_\theta(x|z) \sim \text{Ber}(\pi_z) \qquad q_\phi(z) \sim N(\mu, \text{diag}(\sigma_1^2, \cdots, \sigma_d^2)) \tag{8.20}$$

  And π_z are sigmoid function of z. For this model, ELBO has simple/closed forms: where the first term matches the binomial likelihood and the second term penalize large μ and σ very different from 1.
- Optimization: stochastic gradient descent, ELBO at each data point i. Problem is that the NN has stochastic nodes z and backpropagation does not work. Reparameterization trick: sample ϵ, and add to z, then do back-propagation.
- Application of VAE: pictures of humans, and learn the latent variables: smile, gender, beard, glass, etc.

Bibliography

Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. 2003.

Kevin Murphy. *Machine Learning: a probabilistic perspective*. 2012.