Machine Learning Submodule
Model Assessment and Selection

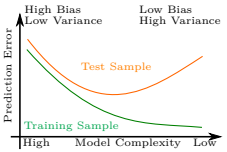
Definition 1.1 Statistical Inference: Is the process of deducing properties of an underlying probability distribution by mere analysis of data.

Definition 1.2
Model Selection:
Is the process of selecting a model f from a given or chosen class of models $\mathcal{F}$

Definition 1.3 Hyperparameter Tuning: Is the process of choosing the hyperparameters θ of a given model $f \in \mathcal{F}$

Definition 1.4 Model Assessment/Evaluation: Is the process of evaluating the performance of a model.

Definition 1.5 Overfitting:
Describes the result of training/fitting a model f to closely to the training data $\mathcal{Z}^{\text{train}}$.
That is, we are producing overly complicated model by fitting the model to the noise of the training set.
Consequences: the model will generalize poorly as the test set $\mathcal{Z}^{\text{test}}$ will not have not the same noise
$\Rightarrow$ big test error.



1.1. Empirical Risk Minimization

2. Generalization Error

Definition 1.6
Generalization/Prediction Error (Risk): Is defined as the expected value of a loss function l of a given predictor m, for data drawn from a distribution $\mathrm{p}_{\mathcal{X},\mathcal{Y}}$.

$$R_{\mathrm{p}}(m) = \mathbb{E}_{(\mathbf{x},y)\sim \mathrm{p}}[l(y;m(\mathbf{x}))] = \int_{\mathcal{D}} \mathrm{p}(\mathbf{x},y) l(y;m(\mathbf{x}))\,\mathrm{d}\mathbf{x}\,\mathrm{d}y$$
$$= \int_{\mathcal{X}}\int_{\mathcal{Y}} \mathrm{p}(\mathbf{x},y) l(y,m(\mathbf{x}))\,\mathrm{d}\mathbf{x}\,\mathrm{d}y$$
$$\overset{??}{=} \int_{\mathcal{X}}\int_{\mathcal{Y}} l(y,m(\mathbf{x}))\mathrm{p}(y|\mathbf{x})\mathrm{p}(\mathbf{x})\,\mathrm{d}\mathbf{x}\,\mathrm{d}y \quad (1.1)$$

Interpretation

Is a measure of how accurately an algorithm is able to predict outcome values for future/unseen/test data.

Definition 1.7 Expected Conditional Risk: If we only know a certain $\mathbf{x}$ but not the distribution of those measurements ($\mathbf{x} \sim \mathrm{p}_{\mathcal{X}}(\mathbf{x})$), we can still calculate the expected risk given/conditioned on the known measurement $\mathbf{x}$:

$$\mathcal{R}_{\mathrm{p}}(m,\mathbf{x}) = \int_{\mathcal{Y}} l(y,m(\mathbf{x}))\mathrm{p}(y|\mathbf{x})\,\mathrm{d}y$$

Corollary 1.1 Note: [def. 1.6] $\Longleftrightarrow$ [def. 1.7]:

$$R_{\mathrm{p}}(m) = \mathbb{E}_{\mathbf{x}\sim\mathrm{p}}[R_{\mathrm{p}}(m,\mathbf{x})] = \int_{\mathcal{X}} \mathrm{p}(\mathbf{x}) R_{\mathrm{p}}(m,\mathbf{x})\,\mathrm{d}\mathbf{x} \quad (1.2)$$

2.1. Expected Risk Minimizer

Definition 1.8 Expected Risk Minimizer (TRM) m^*:
Is the model m that minimizes the total expected risk:

$$m^* \in \underset{m\in\mathcal{C}}{\arg\min}\, \mathcal{R}(m) = \underset{m\in\mathcal{C}}{\arg\min}\, \mathbb{E}_{\mathrm{p}}[l(y;m(\mathbf{x}))] \quad (1.3)$$

3. Empirical Risk

In practice we do neither know the distribution $\mathrm{p}_{\mathcal{X},\mathcal{Y}}(\mathbf{x},y)$, nor $\mathrm{p}_{\mathcal{X}}(\mathbf{x})$ or $\mathrm{p}_{\mathcal{Y}|\mathcal{X}}(y|\mathbf{x})$ (otherwise we would already know the solution).
But: even though we do not know the distribution of $\mathrm{p}_{\mathcal{X},\mathcal{Y}}(\mathbf{x},y)$ we can still sample from it in order to define an empirical risk.

Definition 1.9 Empirical Risk:
Is the the average of a loss function of an estimator h over a finite set of data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ drawn from $\mathrm{p}_{\mathcal{X},\mathcal{Y}}(\mathbf{x},y)$:

$$\hat{\mathcal{R}}_n(m) = \frac{1}{n}\sum_{i=1}^{n} l(m(\mathbf{x}_i), y_i)$$

3.1. Empirical Risk Minimizer

Definition 1.10 Empirical Risk Minimizer (ERM) $\hat{m}$:
Is the model $\widehat{m}$ that minimizes the total empirical risk:

$$\hat{m} \in \underset{m\in\mathcal{C}}{\arg\min}\, \hat{\mathcal{R}}(m) = \underset{m\in\mathcal{C}}{\arg\min}\, n^{-1}\sum_{i=1}^{n} l\left(m(\mathbf{x}_i), y_i\right) \quad (1.4)$$

Questions

1. How far is the true risk $\mathcal{R}(m)$ from the empirical risk $\hat{R}(m)$, for a given m
2. Given a chosen hypothesis class $\mathcal{F}$. How far is the minimizer of the true cost way from the minimizer of the empirical cost

$$m^*(\mathbf{x}) \in \underset{m\in\mathcal{F}}{\arg\min}\, \mathcal{R}(m) \quad \text{vs.} \quad \hat{m}(\mathbf{x}) \in \underset{m\in\mathcal{F}}{\arg\min}\, \hat{\mathcal{R}}(m)$$

We hope that $\lim_{n\to\infty} \hat{\mathcal{R}}_n(m) = \mathcal{R}(m)$.

3.1.1. Squared Loss

Expected Squared Risk

Definition 1.11 Mean Squared Error (MSE):

$$\mathcal{R}(m) = \mathrm{MSE}(x) = \mathbb{E}\left[(\widehat{m}(x) - m(x))^2\right] \quad (1.5)$$

Corollary 1.2 title:

Add proof

$$\mathrm{MSE}(x) = \mathrm{Bias}^2(x) + \mathbb{V}(x) = \left(\mathbb{E}\left[\widehat{m}(x) - m(x)\right]\right)^2 + \mathbb{V}(\widehat{m}(x)) \quad (1.6)$$

Definition 1.12
Integrated Means Squared Error (IMSE)/(MISE):
the integrated MSE or *Mean integrated sqquare error* (MISE) is defined as:

$$\mathrm{IMSE} = \int_x \mathrm{MSE}(x)\,\mathrm{d}x = \int_x \mathbb{E}\left[(\widehat{m}(x) - m(x))^2\right]\mathrm{d}x \quad (1.7)$$

Empirical Squared Risk

Definition 1.13
Mean/Average Squared Prediction Error (MSPE):
the empirical MSE or *Mean/Average Squared Error of Prediction* (MSEP)

$$\hat{\mathcal{R}}_n(m) = \mathrm{ave}_n\,(\widehat{m})^2 = \frac{1}{n}\sum_{i=1}^{n}(\widehat{m}(x_i) - m(x_i))^2 \quad (1.8)$$

Corollary 1.3 [proof 3.2]
MSEP for new observations: Given a new observation x_{new} distributed as:

$$Y_{\text{new}} = m(x_{\text{new}}) + \epsilon \qquad \epsilon \overset{\text{i.e.}}{\sim} \mathcal{N}(0,\sigma^2)$$

then it holds that:

$$\mathrm{MSEP}(x_{\text{new}}) = \mathrm{MSE}(x_{\text{new}}) + \sigma^2 \quad (1.9)$$

Explanation 1.1. *The mean squared error of prediction does not go to zero if $n \to \infty$ as it has an irreducable noise σ.*

Definition 1.14 [example 3.9],[proof 3.1]
Bayes' optimal predictor for the L2-Loss:
Assuming: i.i.d. generated data by $(\mathbf{x}_i, y_i) \sim \mathrm{p}(\mathcal{X},\mathcal{Y})$.
Considering: the least squares risk:

$$R_{\mathrm{p}}(h) = \mathbb{E}_{(\mathbf{x},y)\sim\mathrm{p}}[(y - h(\mathbf{x}))^2]$$

The best hypothesis/predictor h^* minimizing $R(h)$ is given by **conditional mean/expectation** of the data:

$$h^*(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}] \quad (1.10)$$

Cross Validation

Definition 1.15 Cross Validation: Is a model validation/assessment techniques in order to improve the model generalization performance.

Explanation 1.2. *Cross validation helps to increase the model ability to predict out of sample data.*

Definition 1.16 Labeled Data $\mathcal{D}/\mathcal{Z}$:

$$Z = \mathcal{D} := \left\{z_j = (\mathbf{x}_j, \mathbf{y}_j) \;\middle|\; \mathbf{x}_j \in \mathcal{X}, \mathbf{y}_j \in \mathcal{Y}\right\}$$

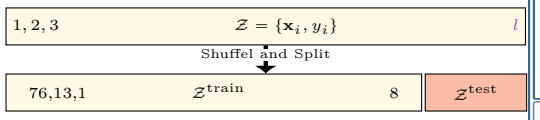
3.2. Training Set

Definition 1.17 Training Set $\mathcal{Z}^{\text{train}} \subset \mathcal{Z}$:
Is a part of the data on which we train our model $\widehat{m}$ in order to reduce the empircal

$$\mathcal{Z}^{\text{train}} = \left\{(\mathbf{x}_1^{\text{train}}, y_1^{\text{train}}), \ldots, (\mathbf{x}_n^{\text{train}}, y_n^{\text{train}})\right\}$$

Definition 1.18 [??]
Training Error $\hat{\mathcal{R}}(\hat{f}, \mathcal{Z}^{\text{train}})$:
is the model that minimizes the empirical risk[def. 1.10] on the training data[def. 1.17]:

$$\widehat{m} \in \underset{\widehat{m}\in\mathcal{F}}{\arg\min}\, \hat{\mathcal{R}}(\widehat{m}, \mathcal{Z}^{\text{train}}) \quad (1.11)$$
$$= \underset{\widehat{m}\in\mathcal{F}}{\arg\min}\, n^{-1} \sum_{(\mathbf{x}_i,y_i)\in\mathcal{Z}^{\text{train}}} l(\widehat{m}(\mathbf{x}_i), y_i)$$



3.3. Testing Set

Definition 1.19 [??]
Test Set $\mathcal{Z}^{\text{test}} \subset \mathcal{Z}$:
Is part of the data that is used in order to test the performance of our model.

$$\mathcal{Z}^{\text{test}} = \left\{(\mathbf{x}_1^{\text{test}}, y_1^{\text{test}}), \ldots, (\mathbf{x}_m^{\text{test}}, y_m^{\text{test}})\right\}$$

Definition 1.20 Test Error $\hat{\mathcal{R}}(\hat{f}, \mathcal{Z}^{\text{test}})$:
Is the error over the test set $\mathcal{Z}^{\text{test}}$ of a predictor $\widehat{m}$ that has been trained on the training set[def. 1.17]:

$$\hat{\mathcal{R}}(f, \mathcal{Z}^{\text{test}}) = m^{-1} \sum_{(\mathbf{x}_i,y_i)\in\mathcal{Z}^{\text{test}}} l(\widehat{m}(\mathbf{x}_i), y_i) \quad (1.12)$$

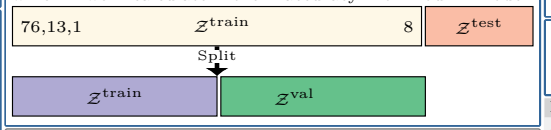
3.4. Validation Set

Definition 1.21 Validation Set $\mathcal{Z}^{\text{val}} \subset \mathcal{Z}^{\text{train}}$:
Is the part of the data that is used in order to select the our model $\widehat{m}$ from a given hypothesis class $\mathcal{F}$.

Explanation 1.3. *We want to select a model $\widehat{m}$ from $\mathcal{F}$ but in order to do so we need to determine the how well it predicts $\Rightarrow$ validation set.*

3.5. Validation Set/Split Once Approach

Definition 1.22 Hold out/Validation Set:
Split the data into a training set on which we train out model $\widehat{m}$ and a validation set on which we calculate the accuracy of our model:



Cons
- We do not use all information/data for training.
- We obtain a high variance estimate depending on the split.

Algorithm 1.1 Validation Set Approach:
Given: set of function classes $\mathcal{F}$ and a loss l
1: train the model on the training set:

$$\hat{m} \in \underset{m\in\mathcal{F}}{\arg\min}\, \hat{\mathcal{R}}\left(m, \mathcal{Z}^{tr}\right) = \underset{m\in\mathcal{F}}{\arg\min}\, \frac{1}{n}\sum_{i=1}^{n} l\left(y_i, m(\mathbf{x}_i)\right)$$

2: Determine the best parameter θ^* by using the validation set:

$$\underline{\hat{\theta}\left(\mathcal{Z}^{val}\right)} \in \underset{\theta:\hat{m}_\theta\in\mathcal{F}_\theta}{\arg\min}\, \hat{R}\left(\hat{m}_\theta\left(\mathcal{Z}^{tr}\right), \mathcal{Z}^{val}\right)$$

3: Use the tests set in order to test the model:

$$\hat{\mathcal{R}}\left(\hat{m}_{\underline{\hat{\theta}(\mathcal{Z}^{val})}}\left(\mathcal{Z}^{tr}\right), \mathcal{Z}^{test}\right)$$

Note: overfitting to the validation set

Tuning the configuration/hyperparameters of the model based on its performance on the validation set can result in overfitting to the validation set, even though your model is never directly trained on it $\Rightarrow$ split the data into a test and training and validation set.

3.6. Leave-One-Out Cross Validation (LOOCV)

Definition 1.23
Leave One Out Cross-Validation (LOOCV):
Train n models on $n-1$ observations and use the left out observations for prediction:

$$\widehat{m}_{n-1}^{-i} \in \underset{m\in\mathcal{F}}{\arg\min}\, \frac{n-1}{n}\sum_{\substack{j=1\\ j\neq i}}^{n} l(y_j, m(x_j)) \qquad \forall i \in \{1,\ldots,n\}$$

$$\hat{\mathcal{R}}^{\text{LOOCV}} = n^{-1}\sum_{i=1}^{n} l\left(y_i, \widehat{m}_{n-1}^{-i}(x_i)\right) \quad (1.13)$$

Pros
- Is basically unbiased estimator, as we use $n-1$ training samples.
- Can have a high variance due to highly correlated training sets, as the only vary in one observation.
- Can be better as $K-$fold cross-validation for small data sets, as small data sets have usually a higher fluctuation $\Rightarrow$ higher variance (as the are more sensitive to any noise/sampling artifacts).

Cons
- computational expensive, only for small data sets possible.
- Variance of the average can be very high due to highly correlated training sets.

3.6.1. LOOCV for Squared Loss and lin. Operator

Theorem 1.1 LOOCV Error for squared loss: For models that can be represented by a linear fitting operator $\mathbf{S}$:

$$\left[\widehat{m}(x_1)\cdot\cdots\cdot\widehat{m}(x_n)\right]^{\intercal} = \mathbf{S}\mathbf{Y} \quad (1.14)$$

it holds for the squared loss that:

$$n^{-1}\sum_{i=1}^{n}\left(y_i - \widehat{m}_{n-1}^{-i}(x_i)\right)^2 = n^{-1}\sum_{i=1}^{n}\left(\frac{y_i - \widehat{m}(x_i)}{1 - \mathbf{S}_{ii}}\right)^2 \quad (1.15)$$

Definition 1.24 Generalized Cross Validation (GCV):

$$\mathrm{GCV} = n^{-1}\sum_{i=1}^{n}\frac{(y_i - \widehat{m}(x_i))^2}{(1 - n^{-1}\,\mathrm{tr}(\mathbf{S}))^2} \quad (1.16)$$

Explanation 1.4. *It holds $\overline{S_{ii}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{S}_{ii} = \frac{1}{n}\,\mathrm{tr}(\mathbf{S})$ thus we can rewirte the mean as the trace, which can efficiently calculated in $\mathcal{O}(n)$.*

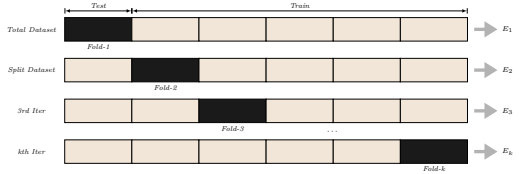
Note

GCV is a misdemeanor as it is an approximation and not a generalization.

3.7. K-Fold Cross Validation

Explanation 1.5 (K-fold Cross-Validation)**.**

1. *use all of the data by splitting the data into K random folds.*
2. *Calculate the training error K times by leaving out the k-th fold, fit the model to the other $K-1$ combined folds (training set) of size* $n \cdot \frac{K-1}{K}$. ($\hat{=}$ validation set)
3. *Do this by choosing each fold $k = 1, \dots, K$ once as validation set and calculate cross-validation error by averaging over them.*



Definition 1.25 K-fold Cross Validation:

$$\mathcal{Z} = \mathcal{Z}_1 \cup \dots \cup \mathcal{Z}_\nu \cup \dots \cup \mathcal{Z}_K \qquad \forall k \in \{1, \dots, K\}$$

$$\widehat{m}^{-\mathcal{Z}_k}_{n-|\mathcal{Z}_k|} \in \arg\min_{m \in \mathcal{F}} \frac{|\mathcal{Z}_k|}{|\mathcal{Z}|} \sum_{i \in \mathcal{Z} \setminus \mathcal{Z}_k} l(y_i, m(x_i)) \tag{1.17}$$

$$\hat{\mathcal{R}}^{\mathrm{CV}} = K^{-1} \sum_{k=1}^{K} |\mathcal{Z}_k|^{-1} \sum_{i \in \mathcal{Z}_k} l\left(y_i, \widehat{m}^{-\mathcal{Z}_k}_{n-|\mathcal{Z}_k|}(x_i)\right) \tag{1.18}$$

Note

A good heuristic for choosing K is 5, or 10 or:

$$k = \min\left(\sqrt{n}, 10\right)$$

Pros

- faster then LOOCV.

Cons

- runs $\approx K$ times slower than traing/test-split, as we need to train the model K times.
- Has higher bias then LOOCV.
  There exits systematic tendency to underfit, as each of the K-fold cross validation models uses only $n \cdot \frac{K-1}{K}$ training samples
  $\Rightarrow$ the estimates of prediction error will typically be more biased (towards simpler models), as the bias increases with a lower number of sampls/d.o.f. (see Rao Cramer).
- Depends on the explicit realization of the K subsets.

3.8. Many Random Divisions

Definition 1.26 Leave d-out CV:
Generalize LOOCV/d-fold CV by considering all possible realization?? of d samples:

$$\mathcal{Z} = \mathcal{Z}_1 \cup \dots \cup \dots \cup \mathcal{Z}_{\binom{n}{d}} \qquad \forall k \in \left\{1, \dots, \binom{n}{d}\right\}$$

$$\widehat{m}^{-\mathcal{Z}_k}_{n-|\mathcal{Z}_k|} \in \arg\min_{m \in \mathcal{F}} \frac{|\mathcal{Z}_k|}{|\mathcal{Z}|} \sum_{i \in \mathcal{Z} \setminus \mathcal{Z}_k} l(y_i, m(x_i)) \tag{1.19}$$

$$\hat{\mathcal{R}}^{\mathrm{CV}} = \binom{n}{d}^{-1} \sum_{k=1}^{\binom{n}{d}} |\mathcal{Z}_k|^{-1} \sum_{i \in \mathcal{Z}_k} l\left(y_i, \widehat{m}^{-\mathcal{Z}_k}_{n-|\mathcal{Z}_k|}(x_i)\right) \tag{1.20}$$

Explanation 1.6. *Is a generalization of LOOCV as it does not depend on the indexing in comparison to classical K-CV.*

Pros

- has often a smaller variance.

A Statistical Perspective

1. Information Theory

1.1. Information Content

Definition 3.1 Information (*Claude Elwood Shannon*):
Information is the resolution of uncertainty.

Amount of Information

The information gained by the realization of a coin tossed n-times should equal to the sum of the information of tossing a coin once n-times:

$$I(\mathrm{p}_0 \cdot \mathrm{p}_1 \cdots \mathrm{p}_n) = I(\mathrm{p}_0) + I(\mathrm{p}_1) + \cdots + I(\mathrm{p}_n)$$

$\Rightarrow$ can use the logarithm to satisfy this

Definition 3.2 Surprise/Self-Information/-Content: Is a measure of the information of a realization x of a random variable $X \sim \mathrm{p}$:

$$I_X(x) = \log\left(\frac{1}{\mathrm{p}(X=x)}\right) = -\log \mathrm{p}(X=x) \tag{3.1}$$

Explanation 3.1 (Definition 3.2).
$I(A)$ measures the number of possibilities for an event A to occur in bits:

$$I(A) = \log_2 (\#\textit{possibilities for } A \textit{ to happen})$$

Corollary 3.1 Units of the Shannon Entropy:
The Shannon entropy can be defined for different logarithms

	log	units
$\triangleq$ units:	Base 2	Bits/Shannons
	Natural	Nats
	Base 10	Dits/Bans

Explanation 3.2. *An uncertain event is much more informative than an expected/certain event:*

$$\textit{surprise/inf. content} = \begin{cases} \textit{big} & \textit{if } \mathrm{p}_X(x) \textit{ unlikely} \\ \textit{small} & \textit{if } \mathrm{p}_X(x) \textit{ likely} \end{cases}$$

1.2. Entropy

Information content deals with a single event. If we want to quantify the amount of uncertainty/information of a probability distribution, we need to take the expectation over the information content[def. 3.2]:

Definition 3.3 Shannon Entropy [example 3.3]:
Is the expected amount of information of a random variable $X \sim \mathrm{p}$:

$$H(\mathrm{p}) = \mathbb{E}_X[I_X(x)] = \mathbb{E}_X\left[\log \frac{1}{\mathrm{p}_X(x)}\right] = -\mathbb{E}_X[\log \mathrm{p}_X(x)]$$
$$= -\sum_{i=1}^{n} \mathrm{p}(x_i) \log \mathrm{p}(x_i) \tag{3.2}$$

Definition 3.4 Differential/Continuous entropy: Is the continuous version of the Shannon entropy[def. 3.3]:

$$H(\mathrm{p}) = \int_{x\sim \mathrm{p}} -f(x) \log f(x)\, \mathrm{d}x \tag{3.3}$$

Notes

- The Shannon entropy is maximized for uniform distributions
- People somtimes write $H(X)$ instead of $H(\mathrm{p})$ with the understanding that p is the distribution of p.

Property 3.1 Non negativity:
Entropy is always non-negative:

$$H(X) \geqslant 0 \quad \text{if } X \text{ is deterministic} \quad H(X) = 0 \tag{3.4}$$

1.2.1. Conditonal Entropy

Proposition 3.1 Conditioned Entropy $H(Y|X=x)$:
Let X and Y be two random variables with a condititional pdf $\mathrm{p}_{X|Y}$. The entropy of Y conditioned on X taking a certain value x is given as:

$$H(Y|X=x) = \mathbb{E}_{Y|X=x}\left[\log \frac{1}{\mathrm{p}_{y|X}(Y|X=x)}\right]$$
$$= -\mathbb{E}_{Y|X=x}\left[\log \mathrm{p}_{Y|X}(y|X=x)\right] \tag{3.5}$$

Definition 3.5 proof 3.4
Conditional Entropy $H(Y|X)$:
Is the amount of information need to determine Y if we arleady know X and is given by averagin $H(Y|X=x)$ over X.

$$H(Y|X) = [\mathbb{E}_X H(Y|X=x)] = -\mathbb{E}_{X,Y}\left[\log \frac{\mathrm{p}(x,y)}{\mathrm{p}(x)}\right] \tag{3.6}$$
$$= \mathbb{E}_{X,Y}\left[\log \frac{\mathrm{p}(x)}{\mathrm{p}(x,y)}\right]$$

Definition 3.6 proof 3.5
Chain Rule for Entropy:

$$H(Y|X) = H(X,Y) - H(X)$$
$$H(X|Y) = H(X,Y) - H(Y) \tag{3.7}$$

Property 3.2 Monotonicity:
Information/conditioning reduces the entropy
$\Rightarrow$ *Information never hurts.*

$$H(X|Y) \geqslant H(X) \tag{3.8}$$

Corollary 3.2 From eq. (3.17):

$$H(X,Y) \leqslant H(X) + H(Y) \tag{3.9}$$

1.3. Cross Entropy

Definition 3.7 Cross Entropy proof 3.3:
Lets say a model follows a true distribution $X \sim \mathrm{p}$ but we model X with a different distribution $X \sim q$. The cross entropy between p and q measure the average amount of information/bits needed to model an outcome $x \sim X \sim \mathrm{p}$ with q:

$$H(\mathrm{p}, q) = \mathbb{E}_{x\sim \mathrm{p}}\left[\log\left(\frac{1}{q(x)}\right)\right] \tag{3.10}$$
$$= -\mathbb{E}_{x\sim \mathrm{p}}[\log q(x)] \tag{3.11}$$
$$= H(\mathrm{p}) + D_{\mathrm{KL}}(\mathrm{p} \parallel q) \tag{3.12}$$

Corollary 3.3 Kullback-Leibler Divergence:
$D_{\mathrm{KL}}(\mathrm{p} \parallel q)$ measures the extra price (bits) we need to pay for using q.

1.4. Kullback-Leibler (KL) divergence

If we want to measure how different two distributions q and p are w.r.t. to the same random variable X, we can define another measure.

Definition 3.8
Kullback–Leibler divergence. examples 3.4 and 3.7
/Relative Entropy from p to q: Given two probability distributions p, q of a random variable X. The Kullback–Leibler divergence is defined to be:

$$D_{\mathrm{KL}}(\mathrm{p} \parallel q) = \mathbb{E}_{x\sim \mathrm{p}}\left[\log \frac{\mathrm{p}(x)}{q(x)}\right] = \mathbb{E}_{x\sim \mathbb{F}}[\log \mathrm{p}(x) - \log q(x)] \tag{3.13}$$

and measures how far away a distribution q is from a another distribution p.

Explanation 3.3.
- p *decides where we put the mass if* $\mathrm{p}(x)$ *is zero we do not care about* $q(x)$.
- $\mathrm{p}(x)/q(x)$ *determines how big the difference between the distributions is.*

Intuition

The KL-divergence helps us to measure just how much information we lose when we choose an approximation.

Property 3.3 Non-Symmetric:

$$D_{\mathrm{KL}}(\mathrm{p} \parallel q) \neq D_{\mathrm{KL}}(q \parallel \mathrm{p}) \qquad \forall \mathrm{p}, q \tag{3.14}$$

Property 3.4:

$$D_{\mathrm{KL}}(\mathrm{p} \parallel q) \geqslant 0 \tag{3.15}$$
$$D_{\mathrm{KL}}(\mathrm{p} \parallel q) = 0 \iff \mathrm{p}(x) = q(x) \forall x \in \mathcal{X} \tag{3.16}$$

Note

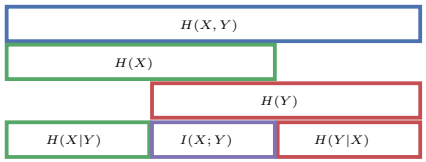
The KL-divergence is not a real distance measure as $\mathrm{KL}(\mathbb{P} \parallel Q) \neq \mathrm{KL}(Q \parallel \mathbb{P})$

Corollary 3.4 Lower Bound on the Cross Entropy: The entropy provides a lower bound on the cross entropy, which follows directly eq. (3.16). from

1.5. Mutual Information

Definition 3.9 **example 3.8**
Mutual Information/Information Gain: Let X and Y be two random variables with a joint probability distribution. The mutal information of X and Y is the reduction in uncertainty in X if we know Y and vice versa.

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{3.17}$$
$$= H(X) + H(Y) - H(X,Y)$$
$$= D_{\mathrm{KL}}\left(\mathrm{p}_{X,Y} \parallel \mathrm{p}_X \mathrm{p}_Y\right)$$



Explanation 3.4 (Definition 3.9).

$$I(X;Y) = \begin{cases} \textit{big} & \textit{if } X \textit{ and } Y \textit{ are highly dependent} \\ 0 & \textit{if } X \textit{ and } Y \textit{ are independent} \end{cases} \tag{3.18}$$

Property 3.5 Symmetry:

$$I(X;Y) = I(Y,X)$$

Property 3.6 Positiveness:

$$I(X;Y) \geqslant 0 \qquad \text{if } X \perp\!\!\!\perp Y \qquad I(X;Y) = 0 \tag{3.19}$$

Property 3.7:

$$I(X;Y) \leqslant H(X) \qquad I(X;Y) \leqslant H(Y) \tag{3.20}$$

Property 3.8 Self-Information:

$$H(X) = I(X;X)$$

Property 3.9 Montone Submodularity: Mutual information is monotone submodular**??**:

$$H(X,z) - H(x) \geqslant H(Y,z) - H(Y) \tag{3.21}$$
$$\overset{\text{[def. 3.6]}}{\iff} \qquad H(z|X) \geqslant H(x|Y) \tag{3.22}$$

2. Proofs

Proof 3.1 Bayes Optimal Predictor[def. 1.14]:

$$\min_h R(h) = \min_h \mathbb{E}_{(\mathbf{x},y)\sim \mathrm{p}}[(y-h(\mathbf{x}))^2]$$
$$\overset{??}{=} \min_h \mathbb{E}_{\mathbf{x}\sim \mathrm{p}_{\mathcal{X}}}\left[\mathbb{E}_{\mathbf{y}\sim \mathrm{p}_{\mathcal{Y}|\mathcal{X}}}\left[(y-h(\mathbf{x}))^2\,|\mathbf{x}\right]\right]$$
$$\overset{\heartsuit}{=} \mathbb{E}_{\mathbf{x}\sim \mathrm{p}_{\mathcal{X}}}\Big[\min_h(\mathbf{x})\underbrace{\mathbb{E}_{\mathbf{y}\sim \mathrm{p}_{\mathcal{Y}|\mathcal{X}}}\left[(y-h(\mathbf{x}))^2\,|\mathbf{x}\right]}_{\mathcal{R}_{\mathrm{p}}(h,\mathbf{x})\ ([\text{def. 1.7}])}\Big]$$

Now lets minimize the conditional executed risk:

$$h^*(\mathbf{x}) = \arg\min_h \mathbb{E}_{\mathbf{y}\sim \mathrm{p}_{\mathcal{Y}|\mathcal{X}}}\left[(y-h(\mathbf{x}))^2\,|\mathbf{x}\right] \tag{3.23}$$
$$0 \overset{!}{=} \frac{\mathrm{d}}{\mathrm{d}h^*}\mathcal{R}_{\mathrm{p}}(h^*,\mathbf{x}) = \frac{\mathrm{d}}{\mathrm{d}h^*}\int (y-h^*)^2 \mathrm{p}(y|x)\,\mathrm{d}y$$
$$= \int \frac{\mathrm{d}}{\mathrm{d}h^*}\left(y-h^*\right)^2 \mathrm{p}(y|x)\,\mathrm{d}y = \int 2(y-h^*)\mathrm{p}(y|x)\,\mathrm{d}y$$
$$= -2h^*\underbrace{\int \mathrm{p}(y|x)\,\mathrm{d}y}_{=1} + 2\underbrace{\int y\mathrm{p}(y|x)\,\mathrm{d}y}_{\mathbb{E}_Y[Y|X=x]}$$

Proof 3.2 Irreducible Error[cor. 1.3]:

$$\begin{aligned}\mathrm{MSEP}(x_n) &= \mathbb{E}\left[\left(Y-\hat{Y}(x_n)\right)^2\right] = \mathbb{E}\left[(Y-\widehat{m}(x_n))^2\right]\\ &= \mathbb{E}\left[(\epsilon + m(x_n) - \widehat{m}(x_n))^2\right]\\ &= \mathbb{E}\left[\epsilon^2\right] + 2\mathbb{E}\left[\epsilon\cdot(m(x_n)-\widehat{m}(x_n))\right]\\ &\quad + \mathbb{E}\left[(\epsilon + m(x_n) - \widehat{m}(x_n))\right]^2\\ &= \mathbb{E}\left[\epsilon^2\right] + 2\mathbb{E}\left[\epsilon\cdot(m(x_n)-\widehat{m}(x_n))\right]\\ &\quad + \mathbb{E}\left[(\epsilon + m(x_n) - \widehat{m}(x_n))\right]^2\\ &= \mathbb{V}\left[\epsilon\right] + \underbrace{2\mathbb{E}\left[\epsilon\right]}_{=0}\cdot\mathbb{E}\left[(m(x_n)-\widehat{m}(x_n))\right]\\ &\quad + \mathbb{E}\left[(\epsilon + m(x_n) - \widehat{m}(x_n))\right]^2\\ &= \mathbb{V}\left[\epsilon\right] + \mathrm{MSE}(x_n)\end{aligned}$$

Proof 3.3: [def. 3.7]

$$\mathbb{E}_{x\sim q}\left[\log\left(\frac{1}{\mathrm{p}(x)}\right)\right] = \mathbb{E}_{x\sim q}\left[\log\left(\frac{q(x)}{\mathrm{p}(x)}\right) + \log\left(\frac{1}{q(x)}\right)\right]$$
$$= H(\mathrm{p}) + D_{\mathrm{KL}}(\mathrm{p}\parallel q)$$

Notes: ♡

Since we can pick $h(\mathbf{x}_i)$ independently from $h(\mathbf{x}_j)$.

Note

$$\mathbb{E}[X]\,\mathbb{E}[Y|X] = \int_{\mathcal{X}} \mathrm{p}_X(x)\,\mathrm{d}x \int_{\mathcal{Y}} \mathrm{p}(y|x)\,\mathrm{d}y$$
$$= \int_{\mathcal{X}}\int_{\mathcal{Y}} \mathrm{p}_X(x)\mathrm{p}(y|x)xy\,\mathrm{d}x\,\mathrm{d}y = \mathbb{E}[X,Y]$$

Proof 3.4: Definition 3.5

$$\begin{aligned}\mathbb{E}_X\left[H(Y|X=x)\right] &= \sum_{x\in\mathcal{X}}\mathrm{p}(x)\sum_{y\in\mathcal{Y}}\mathrm{p}(y|x)\log\mathrm{p}(y|x)\\ &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\mathrm{p}(x)\mathrm{p}(y|x)\log\mathrm{p}(y|x)\\ &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\mathrm{p}(x,y)\log\mathrm{p}(y|x)\\ &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\mathrm{p}(x,y)\log\left(\frac{\mathrm{p}(x,y)}{\mathrm{p}(x)}\right)\end{aligned}$$

Proof 3.5: [def. 3.6] We start from eq. (3.6):

$$\begin{aligned}H(Y|X) &= -\mathbb{E}_{X,Y}\left[\log\frac{\mathrm{p}(x,y)}{\mathrm{p}(x)}\right]\\ &= -\sum_{x,y}\mathrm{p}(x,y)\log\mathrm{p}(x,y) + \sum_x \mathrm{p}(x)\log\frac{1}{\mathrm{p}(X)}\\ &= H(X,Y) - H(X)\end{aligned}$$

Proof 3.6: example 3.4

$$\begin{aligned}\mathrm{KL}(\mathrm{p}||q) &= \mathbb{E}_{\mathrm{p}}\left[\log(\mathrm{p}) - \log(q)\right]\\ &= \mathbb{E}_{\mathrm{p}}\Big[\frac{1}{2}\log\frac{|\Sigma_q|}{|\Sigma_{\mathrm{p}}|} - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{\mathrm{p}})^\intercal\Sigma_{\mathrm{p}}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{\mathrm{p}})\\ &\quad + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_q)^\intercal\Sigma_q^{-1}(\mathbf{x}-\boldsymbol{\mu}_q)\Big]\\ &= \frac{1}{2}\mathbb{E}_{\mathrm{p}}\left[\log\frac{|\Sigma_q|}{|\Sigma_{\mathrm{p}}|}\right] - \frac{1}{2}\mathbb{E}_{\mathrm{p}}\left[(\mathbf{x}-\boldsymbol{\mu}_{\mathrm{p}})^\intercal\Sigma_{\mathrm{p}}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{\mathrm{p}})\right]\\ &\quad + \frac{1}{2}\mathbb{E}_{\mathrm{p}}\left[(\mathbf{x}-\boldsymbol{\mu}_q)^\intercal\Sigma_q^{-1}(\mathbf{x}-\boldsymbol{\mu}_q)\right]\\ &= \frac{1}{2}\log\frac{|\Sigma_q|}{|\Sigma_{\mathrm{p}}|} - \frac{1}{2}\mathbb{E}_{\mathrm{p}}\left[(\mathbf{x}-\boldsymbol{\mu}_{\mathrm{p}})^\intercal\Sigma_{\mathrm{p}}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{\mathrm{p}})\right]\\ &\quad + \frac{1}{2}\mathbb{E}_{\mathrm{p}}\left[(\mathbf{x}-\boldsymbol{\mu}_q)^\intercal\Sigma_q^{-1}(\mathbf{x}-\boldsymbol{\mu}_q)\right]\end{aligned}$$

$$\mathbb{E}_{\mathrm{p}}[a] \overset{\mathrm{tr}(\mathbb{R})=\mathbb{R}}{=} \mathbb{E}_{\mathrm{p}}\left[\mathrm{tr}\left\{(\mathbf{x}-\boldsymbol{\mu}_{\mathrm{p}})^\intercal\Sigma_{\mathrm{p}}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{\mathrm{p}})\right\}\right]$$
$$\overset{??}{=} \mathbb{E}_{\mathrm{p}}\left[\mathrm{tr}\left\{(\mathbf{x}-\boldsymbol{\mu}_{\mathrm{p}})(\mathbf{x}-\boldsymbol{\mu}_{\mathrm{p}})^\intercal\Sigma_{\mathrm{p}}^{-1}\right\}\right]$$
$$= \mathbb{E}_{\mathrm{p}}\left[\mathrm{tr}\left\{\Sigma_{\mathrm{p}}\Sigma_{\mathrm{p}}^{-1}\right\}\right]$$
$$\overset{??}{=} \mathbb{E}_{\mathrm{p}}\left[\mathrm{tr}\left\{\mathbf{I}_d\right\}\right] = \mathbb{E}_{\mathrm{p}}[d] = d$$

$$\mathbb{E}_{\mathrm{p}}[b] \overset{??}{=} (\boldsymbol{\mu}_{\mathrm{p}}-\boldsymbol{\mu}_q)^\intercal\Sigma_q^{-1}(\boldsymbol{\mu}_{\mathrm{p}}-\boldsymbol{\mu}_q) + \mathrm{tr}\left\{\Sigma_q^{-1}\Sigma_{\mathrm{p}}\right\}$$

3. Examples

Example 3.1 : Normal distribution has two population parameters: the mean μ and the variance σ^2.

Example 3.2 Various kind of estimators:

- Best linear unbiased estimator (**BLUE**).
- Minimum-variance mean-unbiased estimator (**MVUE**): minimizes the risk (expected loss) of the squared-error loss-function.
- Minimum mean squared error (**MMSE**).
- Maximum likelihood estimator (**MLE**): is given by the least squares solution (minimum squared error), assuming that the noise is i.i.d. Gaussian with constant variance and will be considered in the next section.

Example 3.3 Entropy of a Gaussian:

$$H(\mathcal{N}(\mu,\Sigma)) = \frac{1}{2}\ln|2\pi\mathrm{e}\Sigma| \overset{??}{=} \frac{1}{2}\ln\left((2\pi\mathrm{e})^d|\Sigma|\right)$$
$$= \frac{d}{2}\ln(2\pi\mathrm{e})^d + \log|\Sigma| \tag{3.24}$$
$$\overset{\Sigma=\mathrm{diag}(\sigma_1^2,\dots,\sigma_d^2)}{=} \frac{1}{2}\ln|2\pi\mathrm{e}| + \frac{1}{2}\sum_{i=1}^d \ln\sigma_i^2$$

Example 3.4 proof 3.6

KL Divergence of Gaussians:

Given two Gaussian distributions:

$\mathrm{p} = \mathcal{N}(\mu_{\mathrm{p}},\Sigma_{\mathrm{p}})$ $\quad q = \mathcal{N}(\mu_q,\Sigma_q)$ it holds

$$D_{\mathrm{KL}}(\mathrm{p}\parallel q) = \frac{\mathrm{tr}\left(\Sigma_q^{-1}\Sigma_{\mathrm{p}}\right) + (\mu_q-\mu_{\mathrm{p}})^\intercal\Sigma_q^{-1}(\mu_q-\mu_{\mathrm{p}}) - d + \ln\left(\frac{|\Sigma_q|}{|\Sigma_{\mathrm{p}}|}\right)}{2}$$

Example 3.5 KL Divergence of Scalar Gaussians:

$\theta \sim q(\theta|\boldsymbol{\lambda}) = \mathcal{N}\left(\mu_q,\sigma_q^2\right)$ $\quad \boldsymbol{\lambda} = [\mu_q \quad \sigma_q]$

$\mathrm{p} = \mathcal{N}\left(\mu_{\mathrm{p}},\sigma_{\mathrm{p}}^2\right)$

$$D_{\mathrm{KL}}(\mathrm{p}\parallel q) = \frac{1}{2}\left(\frac{\sigma_{\mathrm{p}}^2}{\sigma_q^2}(\mu_q-\mu_{\mathrm{p}})^2\sigma_q^{-2} - 1 + \log\left(\frac{\sigma_q^2}{\sigma_{\mathrm{p}}^2}\right)\right)$$

Example 3.6 KL Divergence of Diag. Gaussians:

$\theta \sim q(\theta|\boldsymbol{\lambda}) = \mathcal{N}\left(\boldsymbol{\mu}_q,\mathrm{diag}\left(\sigma_1^2,\dots,\sigma_d^2\right)\right)$ $\quad \boldsymbol{\lambda} = [\mu_{1:d} \quad \sigma_{1:d}]$

$\mathrm{p} = \mathcal{N}\left(\mu_{\mathrm{p}},\mathrm{diag}\left(\sigma_1^2,\dots,\sigma_d^2\right)\right)$

Example 3.7 KL Divergence of Gaussians:

$\mathrm{p} = \mathcal{N}(\mu_{\mathrm{p}},\mathrm{diag}\left(\sigma_1^2,\dots,\sigma_d^2\right))$ $\quad q = \mathcal{N}(\mathbf{0},\mathbf{I})$ it holds

$$D_{\mathrm{KL}}(\mathrm{p}\parallel q) = \frac{1}{2}\sum_{i=1}^d\left(\sigma_i^2+\mu_i^2-1-\ln\sigma_i^2\right)$$

Example 3.8 Gaussian Mutal Information:

Given $X\sim\mathcal{N}(\mu,\Sigma)$ $\quad Y = X+\epsilon$ $\quad \epsilon\sim\mathcal{N}(0,\sigma\mathbf{I})$

$$\begin{aligned}I(X;Y) &= H(Y) - H(Y|X) = H(Y) - H(\epsilon)\\ &\overset{\text{eq. (3.24)}}{=} \frac{1}{2}\ln(2\pi\mathrm{e})^d|\Sigma+\sigma^2\mathbf{I}| - \frac{1}{2}\ln(2\pi\mathrm{e})^d|\sigma^2\mathbf{I}|\\ &= \frac{1}{2}\ln\frac{\cancel{(2\pi\mathrm{e})^d}\,\cancel{\sigma^{2d}}\,|\sigma^{-2}\Sigma+\mathbf{I}|}{\cancel{(2\pi\mathrm{e})^d}\,\cancel{\sigma^{2d}}\,|\mathbf{I}|}\\ &= \frac{1}{2}\ln|\mathbf{I}+\sigma^{-2}\Sigma|\end{aligned}$$

Example 3.9 Bayes Optimal Predictor and MLE[def. 1.14]: Problem: we do not know the real distribution $\mathrm{p}_{\mathcal{Y}|\mathcal{X}}(y|\mathbf{x})$, which we need in order to find the bayes optimal predictor according to eq. (1.10).

Idea:

1. Use artificial data/density estimator $\hat{\mathrm{p}}(\mathcal{Y}|\mathcal{X})$ in order to estimate $\mathbb{E}[\mathcal{Y}|\mathcal{X}=\mathbf{x}]$
2. Predict a test point $\mathbf{x}$ by:

$$\hat{y} = \hat{\mathbb{E}}[\mathcal{Y}|\mathcal{X}=\mathbf{x}] = \int \hat{\mathrm{p}}(y|\mathbf{X}=\mathbf{x})y\,\mathrm{d}y$$

Common approach: $\mathrm{p}(\mathcal{X},\mathcal{Y})$ may be some very complex (non-smooth, ...) distribution ⇒ need to make some assumptions in order to approximate $\mathrm{p}(\mathcal{X},\mathcal{Y})$ by $\hat{\mathrm{p}}(\mathcal{X},\mathcal{Y})$

Idea: choose parametric form $\hat{\mathrm{p}}(Y|\mathbf{X},\theta) = \hat{\mathrm{p}}_\theta(Y|\mathbf{X})$ and then optimize the parameter θ

which results in the so called maximum likelihood estimation section 1.

Supervised Learning

Definition 3.10 Statistical Inference: Goal of Inference

1. What is a good guess of the parameters of my model?
2. How do I quantify my uncertainty in the guess?

$$\mathcal{D} \xrightarrow[\text{Learning method}]{\text{Model Fitting}} \left(\mathcal{X}\xrightarrow{c}\mathcal{Y}\right) \xrightarrow[\text{of data } \mathbf{x} \text{ without label}]{\text{Prediction}} \hat{\mathbf{y}}$$

Recall: goal of supervised learning

Given: training data:

$$\mathcal{D} = \{(\mathbf{x}_1,y_1),\dots,(\mathbf{x}_n,y_n)\}\subseteq\mathcal{X}\times\mathcal{Y}$$

find a hypothesis $h:\mathcal{X}\mapsto\mathcal{Y}$ e.g.

- Linear Regression: $h(\mathbf{x}) = \mathbf{w}^\intercal\mathbf{x}$
- Linear Classification: $h(\mathbf{x}) = \mathrm{sing}(\mathbf{w}^\intercal\mathbf{x})$
- Kernel Regression: $h(\mathbf{x}) = \sum_{i=i}^n \alpha_i\,\mathbf{k}(\mathbf{x}_i,\mathbf{x})$
- Neural Networks (single hidden layer): $h(\mathbf{x}) = \sum_{i=1}^n \mathbf{w}_i'\phi(\mathbf{w}_i^\intercal\mathbf{x})$

s.t. we minimize prediction error/empirical risk [def. 1.10].

Fundamental assumption

The data is generated *i.i.d.* from some unknown probability distribution:

$$(\mathbf{x}_i,y_i)\sim\mathrm{p}_{\mathcal{X},\mathcal{Y}}(\mathbf{x}_i,y_i)$$

Note

The distribution $\mathrm{p}_{\mathcal{X},\mathcal{Y}}$ is dedicated by nature and may be highly complex (not smooth, multimodal,...).

4. Estimators

Definition 3.11 (Sample) Statistic: A statistc is a measuarble function f that assigns a **single** value F to a sample of random variables: $\mathbf{X} = \{X_1,\dots,X_n\}$

$f:\mathbb{R}^n\mapsto\mathbb{R}$ $\quad F = f(X_1,\dots,X_n)$

E.g. F could be the mean, variance,...

Note

The function itself is independent of the sample's distribution; that is, the function can be stated before realization of the data.

Definition 3.12 Statistical/Population Parameter:

Is a parameter defining a family of probabilty distributions see example 3.1

Definition 3.13 (Point) Estimator $\hat{\theta} = \hat{\theta}(\mathbf{X})$:

Given: n-samples $\mathbf{x}_1,\dots,\mathbf{x}_n\sim\mathbf{X}$ an estimator

$$\hat{\theta} = h(\mathbf{x}_1,\dots,\mathbf{x}_n) \tag{3.25}$$

is a statistic/randomn variable used to estimate a true (population) parameter θ[def. 3.12] see also example 3.2.

Note

The other kind of estimators are interval estimators which do not calculate a statistic **but** an interval of plausible values of an unknown population parameter θ.

The most prevalent forms of interval estimation are:

- Confidence intervals (frequentist method).
- Credible intervals (Bayesian method).

Generalized Linear Models (GLMs)

Definition 3.14 Generalized Linear Model (GLM):

$$\mu = \mathbb{E}\left[\mathbf{Y}|\mathbf{X}\right] = g^{-1}\left(\eta\right) \tag{3.26}$$

$$\eta = \sum_{j=0}^{p} \beta_{jm} X_j \tag{3.27}$$

$$g\left(\mathbb{E}\left[\mathbf{Y}|\mathbf{X}\right]\right) = \eta \tag{3.28}$$

Generalized Additive Models (GAMs)

Definition 3.15 Generalized Additive Models (GAMs):

$$sdf \tag{3.29}$$

Regression

Definition 4.1
Explanatory-/Indep.-/Predi.-/Variables/Covariates x:
Are the input variable(s) that we want to relate to the response variable(s)[def. 4.2].

Definition 4.2
Response-/Dependent-/Variable(s) y:
Are the output quantities that we are interested in.

Definition 4.3 Coefficients β: Are the coefficients that we are seeking.

Definition 4.4 Regression: Is the process of finding a possible relationship via some coefficients β between *response-variables* $\mathbf{x}$ and a *predictor-variable*(s) $\mathbf{y}$ up to some error ϵ:

$$\mathbf{y} = f(\mathbf{x}, \beta) + \epsilon \tag{4.1}$$

Note

The term regression comes from the latin term "regressus" and means "to go back" to something. Historically the term was introduced by Galton, who discovered that given an outlier point, further observations will regress back to the mean. In particular he discovered that children of very tall/small people tend to be a smaller/larger.

Definition 4.5 Linear Regression: Refers to regression that is linear w.r.t. to the parameter vector β (but not necessarily the data):

$$\mathbf{y} = \beta^\intercal \phi(\mathbf{x}) + \epsilon \tag{4.2}$$

Linearity

Linearity is w.r.t. the coefficients β_j.
Thus a model with transformed non-linear predictor[def. 4.1] variables is still called *linear*.

Definition 4.6 Residual r:
Let us consider n observations $\{x_i, y_i\}_{i=1}^n$. The residual (error) is the deviation of the observed values from the predicted values:

$$r_i := e_i = \hat{\epsilon}_i = y_i - \hat{y}_i = y_i - \hat{\beta}^\intercal \mathbf{x}_i \qquad i = 1, \dots, n \tag{4.3}$$

Simple (linear) regression (SLR)

Definition 4.7 [example 4.1]
Simple Linear Regression: Is a *linear regression*[def. 4.8] with only one explanatory variable[def. 4.1]:

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i \qquad i = 1, \dots, n \tag{4.4}$$

Multiple (linear) regression (MLR)

Definition 4.8 Multiple Linear Regression:
Is a linear regression model with multiple $\{\beta_j\}_{j=1}^p$ explanatory[def. 4.1] variables:

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} + \epsilon_i \\ &= \beta_0 + \sum_{j=1}^p \beta_j x_{ij} + \epsilon_i = \beta^\intercal x_i + \epsilon_i \end{aligned} \qquad i = 1, \dots, n$$

$$\begin{bmatrix} \\ \mathbf{X} \\ \\ \end{bmatrix} \begin{bmatrix} \\ \beta \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \mathbf{Y} \\ \\ \end{bmatrix} \qquad \mathbf{y} = \mathbf{X}\beta \tag{4.5}$$

Design Matrix:
$\mathbf{X} \in \mathbb{R}^{n,(p+1)}$
$\mathbf{y} \in \mathbb{R}^n$
$\beta \in \mathbb{R}^{p+1}$

add offset inclusion and fix dimension either p+1 or p' or something

Note

Eq. 4.8 is usually an over-determined system of linear equations i.e. we have more observations then predictor variables.

Multiple vs. Multivariate lin. Reg.

Multivariate linear regression is simply linear regression with multiple response variables and thus nothing else but a set of simple linear regression models that have the same types of explanatory variables.

Definition 4.9 [example 4.2]
Simple Linear Quadratic Regression: Is a *linear regression*[def. 4.8] with two explanatory variables[def. 4.1] written as:

$$y_i = \beta_1 + \beta_2 x_i + \beta_3 x_i^2 + \epsilon_i \qquad i = 1, \dots, n \tag{4.6}$$

0.0.1. Existence

Corollary 4.1 Existence:

$$\exists \beta : \begin{array}{rcl} x_{11}\beta_1 + x_{12}\beta_2 + \dots + x_{1p}\beta_p &=& y_1 \\ x_{21}\beta_1 + x_{22}\beta_2 + \dots + x_{2p}\beta_p &=& y_2 \\ \vdots & & \vdots \\ x_{n1}\beta_1 + x_{n2}\beta_2 + \dots + x_{np}\beta_p &=& y_n \end{array} \tag{4.7}$$

$$\iff \mathbf{y} \in \mathfrak{R}(\mathbf{X}) \tag{4.8}$$

1. Linear/Ordinary Least Squares (OLS)

Problem: for an over determined system $n > p$ (usually) $\nexists \mathbf{y} \in \mathfrak{R}(\mathbf{X})$ (in particular given round off errors) s.t. there exists no parameter vector β that solves [def. 4.8].
Idea: try to find the next best solution by minimizing the residual(s)[def. 4.6].

Definition 4.10 Residual Sum of Squares:
Is the sum of residuals[def. 4.6]:

$$\mathrm{RSS}(\beta) := \sum_{i=1}^n e_i^2 = \sum_{i=1}^n \|y_i - \hat{y}_i\|_2^2 \tag{4.9}$$

Definition 4.11 Least Squares Regression lsq(X, y):
Minimizes the residual sum of squares:

$$\hat{\beta} \in \arg\min_\beta \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 = \arg\min_{\mathbf{u} \in \mathfrak{R}(\mathbf{X})} \|\mathbf{y} - \mathbf{u}\|_2^2 \tag{4.10}$$

$$= \arg\min_\beta \|\mathbf{r}\|_2^2 = \sum_{i=1}^n \left(\sum_{j=1}^p x_{ij}\beta_j - y_i \right)^2 = \mathrm{RSS}(\beta)$$

Alternative Formulation

Sometimes people write eq. (4.10) as $\frac{1}{2}\arg\min_\beta \|\mathbf{r}\|_2^2$ which leads to the same solution??.

2. Maximum Likelihood Estimate

Ridge MLE

Proposition 4.1 (Gauss Markov Assumptions)
Assumptions for Linear Regression Model:
1. The $\{\mathbf{x}_i\}_{i=1}^n$ are deterministic and measured without errors.
2. The variance of the error terms is *homoscedastic*??:
$$\mathbb{V}[\epsilon_i] = \sigma^2 < \infty \qquad \forall i \tag{4.11}$$
3. The errors are uncorrelated:
$$\mathrm{Cov}[\epsilon_i, \epsilon_j] = 0 \qquad \forall i \neq j \tag{4.12}$$
4. The errors are jointly normally distributed with mean 0 and constant variance σ^2:
$$\epsilon_i \sim \mathcal{N}(0, \sigma^2) \quad \forall i = 1, \dots, n \iff \epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n) \tag{4.13}$$

Definition 4.12 [proof 4.2]
Simple Linear Regression Log-Likelihood:
Assume: a linear model $\mathbf{y} = \mathbf{X}\beta + \epsilon$
with Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I}\sigma^2)$
With: $\mu = \mathbb{E}_\epsilon[\mathbf{y}] = \mathbb{E}_\epsilon[\mathbf{X}\beta + \epsilon] = \mathbf{X}\beta + 0$
$\mathbb{V}_\epsilon[\mathbf{y}] = \mathbb{V}_\epsilon[\mathbf{X}\beta + \epsilon] = 0 + \mathbb{V}[\epsilon] = \mathbf{I}\sigma^2$
Thus: $\mathbf{Y}|\mathbf{X} \sim \mathcal{N}(\mathbf{X}\beta, \mathbf{I}\sigma^2)$ $Y_i|\mathbf{X} \sim \mathcal{N}(\mathbf{x}_i^\intercal \beta, \sigma^2)$
with: $\theta = (\beta^\intercal \ \ \sigma)^\intercal \in \mathbb{R}^{p+1}$

$$\ln(\mathbf{y}|\mathbf{X}, \theta) \propto -\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \beta^\intercal \mathbf{x}_i)^2 = -\frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{X}\beta\|^2$$

$$\theta^* \in \arg\max_{\theta \in \mathbb{R}^{p+1}} \ln(\mathbf{y}|\mathbf{X}, \theta) = \arg\min_{\theta \in \mathbb{R}^{p+1}} -\ln(\mathbf{y}|\mathbf{X}, \theta) \tag{4.14}$$

2.1. The Normal Equation

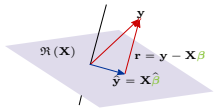
Definition 4.13 [proof 4.4]
The Normal Equations:
Is the equation we need to solve in order to solveeq. (4.10) or equivalently eq. (4.14) and is no longer an over determined system:

$$\begin{bmatrix} \\ \mathbf{X}^\intercal\mathbf{X} \\ \\ \end{bmatrix} \begin{bmatrix} \\ \beta \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \mathbf{X}^\intercal \\ \\ \end{bmatrix} \begin{bmatrix} \\ \mathbf{Y} \\ \\ \end{bmatrix} \qquad \mathbf{X}^\intercal\mathbf{X}\hat{\beta} = \mathbf{X}^\intercal\mathbf{Y} \tag{4.15}$$

$\mathbf{X}^\intercal\mathbf{X} \in \mathbb{R}^{p \times p}$
$\beta \in \mathbb{R}^p$
$\mathbf{X}^\intercal \in \mathbb{R}^{p \times n}$
$\mathbf{Y} \in \mathbb{R}^n$

Geometric Interpretation

Corollary 4.2 [proof 4.5]
Geometric Interpretation:



We want to find $\arg\min_{\beta \in \mathbb{R}^n} \|\mathbf{X}\beta - \mathbf{y}\|_2^2$ which is equal to finding:

$$\arg\min_{\hat{\mathbf{y}} \in \{\mathbf{X}\beta : \beta \in \mathbb{R}^n\} = \mathfrak{R}(\mathbf{X})} \|\hat{\mathbf{y}} - \mathbf{y}\|_2^2$$

but this minimum is equal to the orthogonal projection?? of $\mathbf{y}$ onto $\mathfrak{R}(\mathbf{X})$ i.e. the map:

$$\mathbf{y} \mapsto \hat{\mathbf{y}}$$

is the orthogonal projection of $\mathbf{y}$ onto $\mathfrak{R}(\mathbf{X})$.

Corollary 4.3 Orthogonality of residuals [proof 4.6]:
Corollary 4.2 implies that the residuals are orthogonal w.r.t. to all the column vectors of $\mathbf{X}$:

$$\mathbf{r}^\intercal \mathbf{x}^{(j)} = 0 \qquad \forall j = 1, \dots p \tag{4.16}$$

2.1.1. The Least Squares Solution $\hat{\beta} = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{Y}$

Proposition 4.2 Least Squares Solution:

$$\hat{\beta} = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{y} := \mathbf{X}^\dagger\mathbf{y} \tag{4.17}$$

Note

$\mathbf{X}^\dagger$ is the Moore-Penrose pseudo-inverse of the matrix $\mathbf{X}$.

add MPPI to linear algebra appendix

2.1.2. Solving The Normal Equation

Cholesky Decomposition

Corollary 4.4 Computational Complexity: $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{w} \in \mathbb{R}^d$ with n, the number of observations and d, the number of equations/feautres/dimension of the problem.
Assume: $d \leqslant n$, that is we have an overdetermined system, more equations than unkowns.
1. Compute regular matrix (Matrix Product):
$\mathbf{C} := \mathbf{X}^\intercal\mathbf{X} \triangleq \mathcal{O}(n \cdot d^2)$.
2. Compute the r.h.s. vector (Matrix-Vector):
$\mathbf{c} := \mathbf{X}^\intercal\mathbf{y} \in \mathbb{R}^d \triangleq \mathcal{O}(nd)$.
3. Solve s.p.d. LSE via. Cholesky decomposition:
$\mathbf{C}\mathbf{w} = \mathbf{c} \triangleq \mathcal{O}(d^3)$.
Thus the total cost amounts to $\mathcal{O}(d^3 + nd^2)$.

Note: s.p.d. C and cholesky decomposition

Assume: $\mathbf{X}$ has a trivial kernel $\iff \mathbf{X}^\intercal\mathbf{X}$ is invertible.
1. Symmetric: a transposed matrix times itself is symmetric $\Rightarrow \mathbf{C}$ is symmetric.
2. Posistive definite:
$$\mathbf{w}^\intercal\mathbf{C}\mathbf{w} = \mathbf{w}^\intercal\mathbf{X}^\intercal\mathbf{X}\mathbf{w} = \|\mathbf{X}\mathbf{w}\|^2 > 0 \qquad \forall \mathbf{w} \neq 0$$
has trivial kernel

QR Decomposition

2.1.3. Simple Linear Regression Solution

Definition 4.14 [proof 4.4]
Linear Regression Solution:

$$\hat{\beta} = \underbrace{(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal}_{\mathbf{X}^\dagger}\mathbf{y} \quad \text{with} \quad \begin{array}{l} \Sigma^2 = (\mathbf{X}^\intercal\mathbf{X})^{-1} \\ \mathrm{P} = \mathbf{X}^\intercal\mathbf{y} \end{array} \tag{4.18}$$

Σ^2 : Varianece-Covar. M. P : Inp./Oup. Covariance

Moore-Penrose pseudo-inverse: $\mathbf{X}^\dagger$ with $\mathbf{X}^\dagger\mathbf{X} = \mathbf{I}$ (4.19)

2.1.4. Making Predictions

Definition 4.15 $\mathbf{P}/\mathbf{H} = \mathbf{X}(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal : y \mapsto \hat{y}$
Hat/Projection Matrix:
Is the matrix that projects the y onto the $\hat{y}$:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{X}(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{y} =: \mathbf{P}\mathbf{y} \tag{4.20}$$

Property 4.1 Symmetry: $\mathbf{P}$ is trivially symmetric.

Property 4.2 Idem-potent $\mathbf{P}^2 = \mathbf{P}$: $\mathbf{P}$ is idem-potent i.e. projecting multiple times by $\mathbf{P}$ is the same as projecting once.

Property 4.3 Trace:

$$\begin{aligned} \mathrm{tr}(P) &= \mathrm{tr}(\mathbf{X}(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal) = \mathrm{tr}((\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{X}) \\ &= \mathrm{tr}(I_{p \times p}) = p \end{aligned}$$

Corollary 4.5 $\mathbf{P} : \mathbb{R}^n \mapsto \mathcal{X} \subseteq \mathbb{R}^p$: From these three properties it follows that $\mathbf{P}$ is an orthogonal projection onto a p-dim subspace.

Corollary 4.6 Residual Projection: The residual can be represented in terms ofeq. (4.20):

$$\mathbf{r} = (\mathbf{I} - \mathbf{P})\mathbf{Y} \tag{4.21}$$

it follows that $\mathbf{I} - \mathbf{P}$ is an orthogonal projection onto $(n-p)$-dim subspace $\mathcal{X}^\perp = \mathbb{R}^n \setminus \mathcal{X}$.

Uniqueness

Theorem 4.1 : Let $\mathbf{A} \in \mathbb{R}^{p,p}$, $p \geqslant p$ then it holds that:

$$\mathtt{N}(\mathbf{A}) = \mathtt{N}(\mathbf{A}^\intercal\mathbf{A}) \qquad \mathfrak{R}(\mathbf{A}^\intercal) = \mathfrak{R}(\mathbf{A}^\intercal\mathbf{A}) \tag{4.22}$$

add rest from formulary

Theorem 4.2 Full-Rank Condition F.R.C.:
Equation 4.13 has a unique least squares solution given by:

$$\hat{\beta} = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{Y} \tag{4.23}$$

$$\iff \mathtt{N}(\mathbf{X}) = \{0\} \iff \mathrm{rank}(\mathbf{X}) = p \quad p \geqslant p \tag{4.24}$$

2.2. Moments and Distributions

Property 4.4 Moments of $\hat{\beta}$ [proof 4.7]:

$$\mathbb{E}[\hat{\beta}] = \beta \qquad \mathbb{V}[\hat{\beta}] = \mathrm{Cov}[\hat{\beta}] = \sigma^2(\mathbf{X}^\intercal\mathbf{X})^{-1} \tag{4.25}$$

$$\hat{\beta} \sim \mathcal{N}_p\left(\beta, \sigma^2(\mathbf{X}^\intercal\mathbf{X})^{-1}\right) \tag{4.26}$$

Property 4.5 Moments of $\hat{\mathbf{y}}$ [proof 4.9]:

$$\mathbb{E}[\hat{\mathbf{y}}] = \mathbb{E}[\mathbf{y}] = \mathbf{X}\beta \qquad \mathbb{V}[\hat{\mathbf{y}}] = \mathrm{Cov}[\hat{\mathbf{y}}] = \sigma^2\mathbf{P} \tag{4.27}$$

$$\hat{\mathbf{y}} \sim \mathcal{N}_n\left(\mathbf{X}\beta, \sigma^2\mathbf{P}\right) \tag{4.28}$$

Property 4.6 Moments of r:

$$\mathbb{E}[\mathbf{r}] = 0 \qquad \mathrm{Cov}[\mathbf{r}] = \sigma^2(\mathbf{I} - \mathbf{P}) \tag{4.29}$$

$$\mathbf{r} \sim \mathcal{N}_n\left(\mathbf{0}, \sigma^2(\mathbf{I} - \mathbf{P})\right) \tag{4.30}$$

Property 4.7 Moments of $\hat{\sigma}$:

$$\hat{\sigma}^2 := \frac{1}{n-p}\sum_{i=1}^n r_i^2 \qquad \Longrightarrow \qquad \mathbb{E}[\hat{\sigma}] = \sigma \tag{4.31}$$

$$\hat{\sigma}^2 \sim \frac{\sigma}{n-p}\chi_{n-p}^2 \tag{4.32}$$

Note

The standard deviation σ^2 is given by $\epsilon \sim \mathcal{N}0, \sigma^2$. However we may not know σ^2, thus we can estimate it by using the residuals $\mathbf{r}$.

add proofs and explainations see Greens economics 2003 p48. greens

Proof 4.1 Property 4.7: $\hat{\sigma}^2$ is an unbiased estimator of σ:

2.2.1. The Gaus Markov Theorem

Theorem 4.3 Gauss–Markov theorem [proof 4.10]**:** The BLUE of the β coefficients, of a linear regression model, satisfying the **Gauss-Markov assumptions** is given by the ordinary least squares (OLS) estimator, provided it exists (is invertible).

$$\mathbb{V}\left[\hat{\beta}\right] \leqslant \mathbb{V}\left[\tilde{\beta}\right] \quad \text{with} \quad \begin{array}{l} \hat{\beta} = (\mathbf{X}^\intercal \mathbf{X})^{-1} \mathbf{X}^\intercal y = \mathbf{C}\mathbf{y} \\ \tilde{\beta} \text{ any lin. unb. est. for } \beta \end{array} \tag{4.33}$$

add delete next section

3. MLE with linear Model & Gaussian Noise

3.1. MLE for conditional linear Gaussians

Questions: what is $\mathbb{P}(Y|X)$ if we assume a relationship of the form: We can use the MLE to estimate the parameters $\theta\,\mathbb{R}^k$ of a model/distribution h s.t.

$$\mathbf{y} \approx h(\mathbf{X};\theta) \qquad \Longleftrightarrow \qquad \mathbf{y} = h(\mathbf{X};\theta) + \epsilon$$

$\mathbf{X}$: set of explicative variables. ϵ: noise/error term.

Lemma 4.1 : The conditional distribution D of Y given $\mathbf{X}$ is equivilant to the unconditional distribution of the noise ϵ:

$$\mathbb{P}(Y|\mathbf{X}) \sim D \qquad \Longleftrightarrow \qquad \epsilon \sim D$$

Example: Conditional linear Gaussian

Assume: a linear model $h(\mathbf{x}) = \mathbf{w}^\intercal\mathbf{x}$

and Gaussian noise $\epsilon \sim \mathcal{N}(0,\sigma^2)$

With $\mathbb{E}[\epsilon] = 0$ and $y_i = \mathbf{w}^\intercal\mathbf{x} + \epsilon$, as well as **??** it follows:

$$y \sim \hat{\mathbb{p}}(Y = y|\mathbf{X} = \mathbf{x},\theta) \sim \mathcal{N}(\mu = h(\mathbf{x}),\sigma^2)$$

with: $\theta = (\mathbf{w}^\intercal \;\; \sigma)^\intercal \in \mathbb{R}^{n+1}$

Hence Y is distributed as a linear transformation of the $\mathbf{X}$ variable plus some Gaussian noise ϵ: $y_i \sim \mathcal{N}(\mathbf{w}^\intercal\mathbf{x}_i,\sigma^2) \Rightarrow$ Conditional linear Gaussian.

if we consider an i.i.d. sample $\{y_i,\mathbf{x}_i\}_{i=1}^n$, the corresponding conditional (log-)likelihood is defined to be:

$$\begin{aligned}\mathcal{L}_n(Y|\mathbf{X},\theta) &= \hat{\mathbb{p}}(y_1,\dots,y_n|\mathbf{x}_1,\dots,\mathbf{x}_n,\theta)\\ &\overset{\text{i.i.d.}}{\underset{??}{=}} \prod_{i=1}^n \hat{\mathbb{p}}_{Y|\mathbf{X}}(y_i|\mathbf{x}_i,\theta) = \prod_{i=1}^n \mathcal{N}(\mathbf{w}^\intercal\mathbf{x}_i,\sigma^2)\\ &= \prod_{i=1}^n \frac{1}{\sqrt{\sigma^2 2\pi}}\exp\left(-\frac{(y_i-\mathbf{w}^\intercal\mathbf{x}_i)^2}{2\sigma^2}\right)\\ &= \left(\sigma^2 2\pi\right)^{-\frac{n}{2}}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^n (y_i-\mathbf{w}^\intercal\mathbf{x}_i)^2\right)\end{aligned}$$

$$\mathrm{l}_n(Y|\mathbf{X},\theta) = -\frac{n}{2}\ln\sigma^2 - \frac{n}{2}\ln 2\pi - \frac{1}{2\sigma^2}\sum_{i=1}^n (y_i-\mathbf{w}^\intercal\mathbf{x}_i)^2$$

$$\theta^* = \underset{\mathbf{w}\in\mathbb{R}^d,\sigma^2\in\mathbb{R}_+}{\arg\max}\;\; \mathrm{l}_n(Y|\mathbf{X},\theta)$$

$$\frac{\partial \mathrm{l}_n(Y|\mathbf{X},\theta)}{\partial\theta} = \begin{pmatrix}\frac{\partial \mathrm{l}_n(Y|\mathbf{X},\theta)}{\partial w_1}\\ \vdots \\ \frac{\partial \mathrm{l}_n(Y|\mathbf{X},\theta)}{\partial w_d}\\ \frac{\partial \mathrm{l}_n(Y|\mathbf{X},\theta)}{\partial \sigma^2}\end{pmatrix} \overset{!}{=} \begin{pmatrix}\mathbf{0}_d\\ 0\end{pmatrix}$$

$$\begin{aligned}\frac{\partial \mathrm{l}_n(Y|\mathbf{X},\theta)}{\partial\mathbf{w}} &= \frac{1}{\sigma^2}\sum_{i=1}^n \mathbf{x}_i\left(y_i-\mathbf{w}^\intercal\mathbf{x}_i\right) = \mathbf{0}\in\mathbb{R}^d\\ &= \left(\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\intercal\right)\mathbf{w} = \sum_{i=1}^n \mathbf{x}_i y_i\end{aligned}$$

$$\frac{\partial \mathrm{l}_n(Y|\mathbf{X},\theta)}{\partial\sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^n (y_i-\mathbf{w}^\intercal\mathbf{x}_i)^2 = 0$$

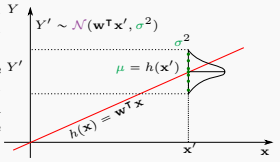
$$\theta^* = \begin{pmatrix}\mathbf{w}^*\\ \sigma^2_*\end{pmatrix} = \begin{pmatrix}\left(\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\intercal\right)^{-1}\left(\sum_{i=1}^n \mathbf{x}_i y_i\right)\\ \frac{1}{n}\sum_{i=1}^n\left(y_i-\mathbf{w}_*^\intercal\mathbf{x}_i\right)^2\end{pmatrix} \tag{4.34}$$

Note

- The mean μ of the normal distribution follows from:
$$\mathbb{E}\left[\mathbf{w}^\intercal\mathbf{x}_i + \epsilon_i\right] = \mathbb{E}[\underbrace{\mathbf{w}^\intercal\mathbf{x}_i}_{\text{const}}] + \underbrace{\mathbb{E}[\epsilon_i]}_{=0} = \mathbf{w}^\intercal\mathbf{x}_i$$
- The noise ϵ must have zero mean, otherwise it wouldn't be randomn anymore.
- The optimal function $h^*(\mathbf{x})$ determines the mean μ.
- We can also minimize:
$$\theta^* = \underset{\theta}{\arg\max}\;\hat{\mathbb{p}}(Y|\mathbf{X},\theta) = \underset{\theta}{\arg\min} -\hat{\mathbb{p}}(Y|\mathbf{X},\theta)$$

The Liklihood does not explicitly care about the distribution of the $\mathbf{x}$ **but** for a given $\mathbf{x}'$ it carse about modeling the distribution of the chosen model $f_{\mathcal{Y}|\mathcal{X}}(y|\mathbf{x}',\theta)$ around the linear function $h(\mathbf{x}') = \mathbf{w}^\intercal\mathbf{x}'$

$\Rightarrow y'$ is an sample from the distribution/model.



3.2. Conditional MLE ≙ Least Squares

Assuming that the noise is i.i.d. Gaussian with *constant* variance σ, that is $\theta = \left(\mathbf{w} \;\; \sigma^2\right)^\intercal$

and considering the negative log likelihood in order to minimize $\arg\max\alpha = -\arg\min\alpha$:

$$-\mathrm{l}_n(\mathbf{w}) = -\prod_{i=1}^n \ln\mathcal{N}(\mathbf{w}^\intercal\mathbf{x}_i,\sigma^2) = \frac{n}{2}\ln(2\pi\sigma^2) + \sum_{i=1}^n \frac{(y_i-\mathbf{w}^\intercal\mathbf{x}_i)^2}{2\sigma^2}$$

$$\arg\max_{\mathbf{w}} \mathrm{l}_n(\mathbf{w}) \iff \arg\min_{\mathbf{w}} -\mathrm{l}_n(\mathbf{w})$$

$$\arg\min_{\mathbf{w}} \frac{1}{\sigma^2}\sum_{i=1}^n (y_i-\mathbf{w}^\intercal\mathbf{x}_i)^2 = \arg\min_{\mathbf{w}} \sum_{i=1}^n (y_i-\mathbf{w}^\intercal\mathbf{x}_i)^2 \tag{4.35}$$

Thus Least squares regression equals Conditional MLE with a linear model + Gaussian noise.

Maximizing Liklihood $\iff$ Minimizing least squares

Corollary 4.7 : The Maximum Likelihood Estimate (MLE) for i.i.d. Gaussian noise (and general models) is given by the squared loss/Least squares solution, assuming that the variance is constant.

Heuristics for ??

Consider a sample $\{y_1,\dots,y_n\} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu,\sigma^2)$

$$\frac{\partial \mathrm{l}_n(y|\mathbf{x},\theta)}{\partial\mu} = \frac{1}{\sigma^2}\sum_{i=1}^n (y_i-\mu) \overset{!}{=} 0$$

$$\frac{\partial \mathrm{l}_n(y|\mathbf{x},\theta)}{\partial\sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^n (y_i-\mu)^2 \overset{!}{=} 0$$

$$\theta^* = \begin{pmatrix}\mu^*\\ \sigma^2_*\end{pmatrix} = \begin{pmatrix}\frac{1}{n}\sum_{i=1}^n y_i\\ \frac{1}{n}\sum_{i=1}^n (y_i-\overline{y}_i)^2\end{pmatrix} \tag{4.36}$$

So, the optimal MLE correspond to the empirical mean and the variance.

Note

$$\frac{\partial\mathbf{w}^\intercal\mathbf{x}}{\partial\mathbf{w}} = \frac{\partial\mathbf{x}^\intercal\mathbf{w}}{\partial\mathbf{w}} = \mathbf{x}$$

3.3. MLE for general conditional Gaussians

Suppose we do not just want to fit linear functions but a gerneal class of models $Hsp := \{h : \mathcal{X} \mapsto \mathbb{R}\}$ e.g. neural networks, kernel functions,...

Given: data $\mathcal{D} = \{(\mathbf{x}_1,y_1),\dots,(\mathbf{x}_n,y_n)\}$ The MLE for general models h and i.i.d. Gaussian noise:

$$h \sim \hat{\mathbb{p}}_{Y|\mathbf{X}}(Y = y|\mathbf{X} = \mathbf{x},\theta) = \mathcal{N}(y|h^*(\mathbf{x}),\sigma^2)$$

Is given by the least squares solution:

$$h^* = \underset{h\in\mathcal{H}}{\arg\min}\sum_{i=1}^n (y_i - h(\mathbf{x}_i))^2$$

E.g. for linear models $\mathcal{H} = \{h(\mathbf{x}) = \mathbf{w}^\intercal\mathbf{x}$ with parameter $\mathbf{w}\}$

Other distributions

If we use other distributions instead of Guassian noise, we obtain other loss functions e.g. L1-Norm for **Poission Distribution**.

$\Rightarrow$ if we know somthing about the distribution of the data we know which loss fucntion we should chose.

Ridge Max Prior

Prior

Assume: prior $\mathbb{P}(\beta|\Sigma)$ on the model parameter β is gaussian as well and depends on the hyperparameter ([def. 6.7]) Σ (≙ co-variance matrix):

$$\begin{aligned}\beta &\sim \mathrm{p}^{\text{Ridge}}(\beta|\Sigma) = \mathcal{N}(\beta|0,\Sigma)\\ &\overset{??}{=} (2\pi)^{-\frac{d+1}{2}}\det(\Sigma)^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\beta^\intercal\Sigma^{-1}\beta\right)\end{aligned}$$

$$\mathrm{l}_n(\beta|\Sigma) = -\frac{1}{2}\ln\det(\Sigma)^{-1} - \frac{d+1}{2}\ln 2\pi - \frac{1}{2}\beta\Sigma^{-1}\beta \tag{4.37}$$

eventually add change of variabels formula and check if factor n is missing

Max Prior

$$\begin{aligned}\beta^* &\in \underset{\beta\in\mathbb{R}^{d+1}}{\arg\max}\,\mathrm{l}_n(\beta|\Sigma)\\ &= \underset{\beta\in\mathbb{R}^{d+1}}{\arg\max} -\frac{1}{2}\ln\det(\Sigma)^{-1} - \frac{d+1}{2}\ln 2\pi - \frac{1}{2}\beta\Sigma^{-1}\beta\end{aligned}$$

$$0 \overset{!}{=} \frac{\partial}{\partial\beta^*}\mathrm{l}_n(\beta^*|\Sigma) = -\frac{\partial}{\partial\beta^*}\beta^*\Sigma^{-1}\beta^* \overset{\text{eq. (4.46)}}{=} -2\Sigma^{-1}\beta^*$$

$$\beta^* \in \underset{\beta\in\mathbb{R}^{d+1}}{\arg\max}\log\mathrm{p}(\beta|\Sigma) = \underset{\beta\in\mathbb{R}^{d+1}}{\arg\min} -\mathrm{l}_n(\beta|\Sigma) = 2\Sigma^{-1}\beta^* \tag{4.38}$$

Log-MAP

$$\begin{aligned}\beta^* &\in \underset{\beta\in\mathbb{R}^{d+1}}{\arg\max}\,\mathbb{P}(\beta|\mathbf{X},\mathbf{y})\\ &= \underset{\beta\in\mathbb{R}^{d+1}}{\arg\min} -\log\overbrace{\mathbb{P}(\beta|\Sigma)}^{\text{eq. (4.38)}} - \log\overbrace{\mathbb{P}(\mathbf{X},\mathbf{y}|\beta)}^{??}\\ &= \Sigma^{-1}\beta^* - \frac{1}{\sigma^2}\mathbf{X}^\intercal\mathbf{y} + \frac{1}{\sigma^2}\mathbf{X}^\intercal\mathbf{X}\beta^* = 0\\ &\iff (\Sigma^{-1} + \mathbf{X}^\intercal\mathbf{X}\sigma^{-2})\beta^* = \sigma^{-2}\mathbf{X}^\intercal\mathbf{y}\\ &\quad (\sigma^2\Sigma^{-1} + \mathbf{X}^\intercal\mathbf{X})\hat{\beta} = \mathbf{X}^\intercal\mathbf{y}\\ &\quad \hat{\beta}^{\text{MAP}} = \left(\sigma^2\Sigma^{-1} + \mathbf{X}^\intercal\mathbf{X}\right)^{-1}\mathbf{X}^\intercal\mathbf{y}\end{aligned}$$

Definition 4.16 Ridge MAP: For ridge regression we assume that the noise of the prior is uncorrelated/diagonal i.e.

$$\Sigma^{-1} = \mathbf{I}\sigma^{-2} \quad \textbf{and let} \quad \Lambda := \sigma^2\Sigma^{-1} = \mathbf{I}\frac{\sigma^2}{\sigma^2} \tag{4.39}$$

which leads to:

$$\hat{\beta}^{\text{MAP}} = (\Lambda + \mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{y} \quad \text{with} \quad \Lambda = \mathbf{I}\lambda = \mathbf{I}\frac{\sigma^2}{\sigma^2} \tag{4.40}$$

Definition 4.17 Regularization: Regularization is the process of introducing additional information/bias in order to solve an ill-posed problem or to prevent overfitting.
(It is not feature selection)

Definition 4.18 Tikhonov regularization: Commonly used method of regularization of ill-posed problems.

$$\|\mathbf{X}\beta - \mathbf{y}\|^2 + \|\Gamma\beta\|^2 \tag{4.41}$$

Γ: Tikhonov matrix in many cases, this matrix is chosen as $\Gamma = \alpha\mathbf{I}$ giving preference to solutions with smaller norms; this is known as **Ridge/L2 regularization**.

Gaussian Prior/Liklihood MAP inference

$$\begin{aligned}\hat{\beta}^{\text{Ridge}} &= \underset{\beta}{\arg\min}\Big\{\underbrace{(\mathbf{y}-\mathbf{X}\beta)^\intercal(\mathbf{y}-\mathbf{X}\beta)}_{\text{data term}} + \underbrace{\beta^\intercal\Lambda\beta}_{\text{regularizer/penalty}}\Big\}\\ &= \underset{\beta}{\arg\min}\left\{\|\mathbf{y}-\mathbf{X}\beta\|^2 + \beta^\intercal\Lambda\beta\right\}\\ &\overset{\text{eq. (4.39)}}{=} \underset{\beta}{\arg\min}\left\{\|\mathbf{y}-\mathbf{X}\beta\|^2 + \lambda\|\beta\|^2\right\}\\ &= \underset{\beta}{\arg\min}\left\{\|\mathbf{y}-\mathbf{X}\beta\|^2 + \lambda\sum_{i=1}^d \beta_i^2\right\}\end{aligned}$$

- $\|\mathbf{y}-\mathbf{X}\beta\|^2$ is forced to be small so that we find a weight vector β that matches the data as close as possible:
$$y_i = \beta_i\mathbf{x}_i + \epsilon_i \qquad \text{s.t.} \qquad \sum_{i=1}^n \epsilon_i \text{ small}$$
In other words we want to fit the data well.
- $\beta^\intercal\Lambda\beta \overset{\text{ridge}}{=} \lambda\|\beta\|^2$ says chose a model with a small magnitude $\|\beta\|^2$.
Thus the smaller λ the bigger can the data faith fullness term be $\|\mathbf{y}-\mathbf{X}\beta\|^2$.

Note

The intercept β_0 in the regularizer term has to be left out. Penalization of the intercept would make the procedure depend on the origin chosen for y.
Thus we actually have (for data with non-zero mean):

$$\beta^* = \underset{\beta\in\mathbb{R}^d}{\arg\min}\left\{\|\mathbf{y}-(\mathbf{X}\beta+\beta_0)\|^2 + +\lambda\sum_{i=1}^d \beta_i^2\right\}$$

Note: SVD

Using SVD one can show that ridge regression shrinks first the eigenvectors with minimum explanatory variance.
Hence L2/Ridge regression can be used to estimate the predictor importance and penalize predictors that are not important (have small explanatory variance).

Note: no feature selection

The coefficients in a ridge will go to zero as λ increases but will no become zero (as long as $\lambda \neq \infty$)!
They are fit in a restricted fashion controlled by the shrinkage penalty λ.

$$\text{dofs}(\lambda) = \begin{cases} d & \text{if } \lambda = 0 \text{ (no regularization)}\\ \to 0 & \text{if } \lambda \to \infty\end{cases} \tag{4.42}$$

$\Rightarrow$ Ridge cannot be used for variable selection since it retains all the predictors

Balance of $\lambda = \frac{\sigma^2}{\sigma_\beta^2}$ controls the tradeoff between simplicity and data faith fullness because:

① $\lambda \xrightarrow[\sigma\downarrow]{\sigma\uparrow} \infty$: $\|\beta\|^2$ must be minimized:
- $\sigma\uparrow$: model does not need to match data so perfectly as we have more noise in our data/observations $\Longleftrightarrow$ bigger errors (recall $\epsilon \sim \mathcal{N}(0, \mathbf{I}\sigma^2)$).
- $\sigma_\beta\downarrow$: prior has smaller variance, thus our prior knowledge of the model is pretty exact/important (recall $\beta \sim \mathcal{N}(\beta|0, \mathbf{I}\sigma_\beta)$)

② $\lambda \xrightarrow[\sigma\uparrow]{\sigma\downarrow} 0$: $\|\mathbf{y} - \mathbf{X}\beta\|^2$ must be minimized: model must match data perfectly
- $\sigma\downarrow$: model does need to match perfectly, our observation/data has small variance/is well defined $\Longleftrightarrow$ do not allow big errors (recall $\epsilon \sim \mathcal{N}(0, \mathbf{I}\sigma^2)$).
- $\sigma_\beta\uparrow$: our knowledge about the model is pretty vague (recall $\beta \sim \mathcal{N}(\beta|0, \mathbf{I}\sigma_\beta)$)

Note
- Often $\Lambda^{-1} = \mathbb{1} \in \mathbb{R}^{d+1\times d+1}$
- Λ is symmetric and diagonal.
- $(d+1)$ dimension as we included offset into β.

Heuristic Map Inference

A really large weight vector β will result in amplifying noise/larger variance/fluctuations $\triangleq$ overfitting.
This is because the complexity of the estimate increases with the magnitude of the parameter as it becomes easier to fit complex noise.

Ill-posed problem/Invertability and Ridge

Another advantage of Ridge regression is that, even if $\mathbf{X}^\intercal\mathbf{X}$ in eq. (4.40) is not invertible/regular/has not full rank.
Then $(\mathbf{X}^\intercal\mathbf{X} + \mathbf{\Lambda})$ will still be invertible/well posed.
This was the original reason for L2/Ridge Regression.

MAP $\triangleq$ Ridge

$$\arg\max_{\mathbf{w}} \mathbb{P}(\mathbf{w}|\mathbf{x}, y) = \arg\min_{\mathbf{w}} \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{n} (y_i - \mathbf{w}^\intercal\mathbf{x}_i)^2$$

MAP with a linear model and Gaussian noise equals classical ridge regression **??**.

$$\underbrace{\arg\min_{\mathbf{w}} \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{n} (y_i - \mathbf{w}^\intercal\mathbf{x}_i)^2}_{\text{Ridge Regression}} \equiv \underbrace{\arg\max_{\mathbf{w}} \mathbb{P}(\mathbf{w}) \prod_{i=1}^{n} \mathbb{P}(y_i|\mathbf{x}_i, \mathbf{w})}_{\text{MAP}}$$

Thus if we know our data β, σ we can chose λ statistically and do not need cross-validation.

Generalization

Regularized estimation can often be understood as MAP inference:

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{n} l(\mathbf{w}^\intercal\mathbf{x}_i; \mathbf{x}_i, y_i) + C(\mathbf{w}) =$$
$$= \arg\max_{\mathbf{w}} \prod_{i=1}^{n} \mathbb{P}(\mathbf{w})\mathbb{P}(y_i|\mathbf{x}_i, \mathbf{w}) = \arg\max_{\mathbf{w}} \mathbb{P}(\mathbf{w}|\text{data})$$

with
$$C(\mathbf{w}) = -\log\mathbb{P}(\mathbf{w})$$
$$l(\mathbf{w}^\intercal\mathbf{x}_i; \mathbf{x}_i, y_i) = -\log\mathbb{P}(y_i|\mathbf{x}_i, \mathbf{w})$$

Priors

3.4. Laplace Prior≙Lasso/L1-regularization

Intro

Question: what if $d \gg n$ e.g.
- bag of words with d =nb. of words$\gg n$nb. of documents.
- Genome analysis d =nb. of genes$\gg n$ patients.

Problem: we have more unkowns/parameters than observations $\Rightarrow$ no unique solution. **e.g.**: Trying to fit 1 data point with polynomimal of degree 12.

Question: can we somehow still find a good solution if
$n = \mathcal{O}(\ln d)$ $\Longleftrightarrow$ exp. more dim. than observations

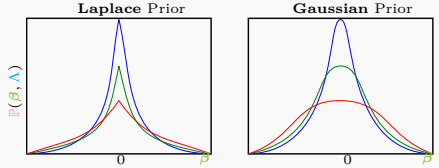
Idea: If most of the dimensions are irrelevant for the problem, then we can find a good (sparse) solution $\triangleq$ feature selection/dimensionality reduction.

Given: Laplacian model prior $\beta \sim \mathrm{p}(\beta|\Lambda)$:

$$\mathbb{P}^{\text{Lasso}}(\beta|\Lambda) \overset{??}{=} \frac{\Lambda}{2} e^{(-\Lambda|\beta|)} = \prod_{j=1}^{d} \frac{\lambda_j}{2} e^{-\lambda_j|\beta_j|}$$

With $\Lambda^{-1} := \Sigma$ hyperparameter/covariance matrix

This leads to a L1 regularized model:



Thus: laplace priors gives sparesness, higher liklihood to get value at $\beta = 0$.

$$-\ln\mathbb{P}(\beta|\Lambda) = \sum_{j=1}^{d} \lambda_j|\beta_j| - d\ln\frac{\lambda_j}{2} \tag{4.43}$$

Laplacian MAP Prior Inference

$$\beta^* = \arg\min_{\beta\in\mathbb{R}^d} \left\{\|\mathbf{y} - (\mathbf{X}\beta + \beta_0)\|^2 + \lambda\|\beta\|_1\right\}$$
$$= \arg\min_{\beta\in\mathbb{R}^d} \left\{\|\mathbf{y} - (\mathbf{X}\beta + \beta_0)\|^2 + \lambda \sum_{i=1}^{d} |\beta_i|\right\} \tag{4.44}$$

$|\beta|_i$ does not change β_i while β_i^2 becomes very small for values$\in (0, 1)$ thus when minimizing the L2 error $\|betac\|^2 \to 0$ **but** not β_i while for L1 regularization will actually have to set β_i values to zero for large enough λ.

Advantage

Combines advantages of Ridge regression (convex function/optimization) and L0-regression (sparse and easy to interpret solution).

Difference L1& L2 penalties

Typically ridge or L2 penalties are much better for minimizing prediction error rather than L1 penalties. The reason for this is that when two predictors are highly correlated, L1 regularizer will simply pick one of the two predictors. In contrast, the L2 regularizer will keep both of them and jointly shrink the corresponding coefficients a little bit. Thus, while the L1 penalty can certainly reduce overfitting, you may also experience a loss in predictive power.

Notes

The unconstrained convex (see **??**) optimization problem eq. (4.44) is not differentiable at $\beta_i = 0$ and **thus** has no closed form solution as the L2 problem $\Rightarrow$ quadratic programming.

3.5. Sparsness Priors/L0-regularization

$$-\ln\mathbb{P}(\beta|s) = s\sum_{j=1}^{d} \mathbb{1}_{\beta_j\neq 0} = s\sum_{j=1}^{d} \cdot \begin{cases} 1 & \text{if } \beta_j \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.45}$$

$\Rightarrow$ measure for the number of possible non-zero dimesnions/-parameters in β.

Advantage
- Leads always to sparse solution.
- Indicates/Explains model well as we only get a few non-zero parameters that determine/characterize the model.

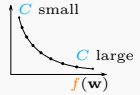
Drawback

Non-convex, non-differentiable problem $\Rightarrow$ computationally difficult combinatorics.

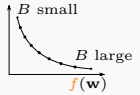
Scalarization vs. Constrained Optimization

Their are two equivilant ways of trading:
- $g(\beta) = \|\mathbf{y} - \mathbf{X}\beta\|^2$: the data term and
- $f(\beta)$: the Regularizer.

Scalariztion $\min_{\mathbf{w}} f(\beta) + Cg(\beta)$



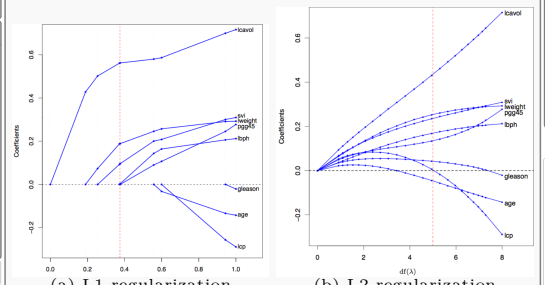
Constraint opt. $\min_{\mathbf{w}} g(\beta)$ s.t. $f(\mathbf{w}) \leqslant B$



Note

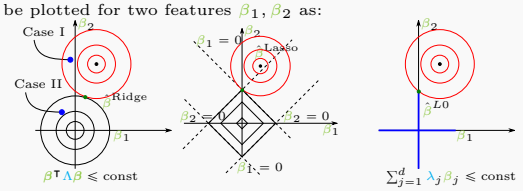
Scalarization and **constrained optimzization** gives the same curves $\Longleftrightarrow$ f, g are both convex functions.
This is not necessarily for the same values of C and B **but** their exisits always a relationship $C = u(B)$ s.t. this is true.

Comparison of priors



(a) L1-regularization (b) L2-regularization

The constraint formulation of the optimization problems can be plotted for two features β_1, β_2 as:



- **Ridge Regression/L2-regression**: if the leasts squares error solution satisfies the constraint, we are fine (Case II), otherwise we do violated the constraint $\beta_1^2 + \beta_2^2 \leqslant$ const (Case I).
- **Lasso/L1-regression**: Here the constraint equals $|\beta_1| + |\beta| \leqslant$ const and leads to polyhedron. Most of the time we obtain a sparse solution $\triangleq$ corner, due to the fact that corner regions increas much faster in volume, as the mixed regions (sparseness increases with number of dimensions).
- **Sparsness prior/L0-regression**: Leads to a super spiky geometry $\Rightarrow$ always leads to a sparse solution.

Liklihoods

3.6. Student's-t likelihood loss function

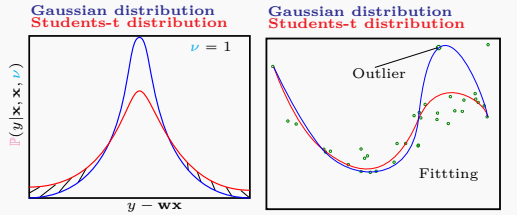
Students-t Distribution:

$$f(y|\mathbf{x}, \mathbf{w}, \nu, \sigma^2) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\nu\sigma^2\Gamma\left(\frac{\nu}{2}\right)}} \left(1 + \frac{(y - \mathbf{w}^\intercal\mathbf{x})^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}}$$

ν: determines speed of decay.

Problem L2/squared loss functions lead to estimates that are sensitive to outliers, that is because something that is far away, from the expected value, will be increased/influences the model very much.

- For Gaussian noise: outliers are very unlikly and thus will have a big influence on the model.
- For Students-t noise: noise, outliers are not as unlikly as for Gaussian noise and thus will not have that much of an influence on the model.



Speed of Decay: $\mathbb{P}(|y - \mathbf{w}^\intercal\mathbf{x}| > t)$ probability of having a outlier/derivation of larger than t, for linear regression.

Students-t $\mathbb{P}(|y - \mathbf{w}^\intercal\mathbf{x}| > t) = \mathcal{O}(t^{-\alpha})$ $\alpha > 0$
(Polynomial decay)

Gaussian $\mathbb{P}(|y - \mathbf{w}^\intercal\mathbf{x}| > t) = \mathcal{O}(\exp^{-\alpha t})$ $\alpha > 0$
(Exponential decay)

$\Rightarrow$ Students-t distribution decays less fast then the Gaussian distribution and **thus** has heavier tails/tailmasses and does not get so easily influenced by noise.
Thus if we know that our model contains outliers/noise, we should use student's t distribution.

4. Proofs

Proof 4.2 4.12: From eq. (4.12) it follows that the response variables are uncorrelated given the explanatory variables $\mathrm{Cov}[Y_i, Y_j|\mathbf{X}] = 0$. Hence we have i.i.d. samples with a corresponding conditional (log-)likelihood given by:

$$\mathscr{L}_n(\mathbf{y}|\mathbf{X}, \theta) \overset{\text{i.i.d.}}{=} \prod_{i=1}^{n} \mathrm{p}(\mathbf{x}_i, y_i|\theta) = \prod_{i=1}^{n} \mathcal{N}(\beta^\intercal\mathbf{x}_i, \sigma^2)$$
$$= \prod_{i=1}^{n} \frac{1}{\sqrt{\sigma^2 2\pi}} \exp\left(-\frac{(y_i - \beta^\intercal\mathbf{x}_i)^2}{2\sigma^2}\right)$$
$$= \left(\sigma^2 2\pi\right)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n} (y_i - \beta^\intercal\mathbf{x}_i)^2\right)$$

$$\mathrm{l}_n(\mathbf{y}|\mathbf{X}, \theta) = -\frac{n}{2}\ln\sigma^2 - \frac{n}{2}\ln 2\pi - \frac{1}{2\sigma^2}\sum_{i=1}^{n} (y_i - \beta^\intercal\mathbf{x}_i)^2$$

Proof 4.3 Definition 4.14:

$$\beta^* \in \arg\min_{\beta\in\mathbb{R}^p} -\mathrm{l}_n(\mathbf{y}|\mathbf{X}, \theta)$$
$$= \arg\min_{\beta\in\mathbb{R}^p} \frac{1}{2\sigma^2}\sum_{i=1}^{n} (y_i - \beta^\intercal\mathbf{x}_i)^2$$
$$= \arg\min_{\beta\in\mathbb{R}^p} \frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\beta)^\intercal(\mathbf{y} - \mathbf{X}\beta)$$
$$= \arg\min_{\beta\in\mathbb{R}^p} (\mathbf{y} - \mathbf{X}\beta)^\intercal(\mathbf{y} - \mathbf{X}\beta)$$
$$\overset{\star}{\Longleftrightarrow} \left(-2\mathbf{y}^\intercal\mathbf{X} + 2\mathbf{X}^\intercal\mathbf{X}\beta^*\right) \overset{!}{=} 0$$
$$\Rightarrow \mathbf{X}^\intercal\mathbf{X}\beta^* = \mathbf{X}^\intercal\mathbf{y}$$

add to proof and linear algrba section

Note: ★

$$\begin{aligned}&(\mathbf{y}-\mathbf{X}\beta)^\intercal(\mathbf{y}-\mathbf{X}\beta)\\&=\mathbf{y}^\intercal\mathbf{y}-\mathbf{y}^\intercal\mathbf{X}\beta+(\mathbf{X}\beta)^\intercal\mathbf{y}-(\mathbf{X}\beta)^\intercal(\mathbf{X}\beta)\\&=\mathbf{y}^\intercal\mathbf{y}-2\mathbf{y}^\intercal\mathbf{X}\beta+\beta^\intercal\mathbf{X}^\intercal(\mathbf{X}\beta)\end{aligned}$$

$$\frac{\partial}{\partial\mathbf{x}}\mathbf{M}\mathbf{x}=\mathbf{M}\quad\text{and}\quad\frac{\partial}{\partial\mathbf{x}}\mathbf{x}^\intercal\mathbf{M}\mathbf{x}=(\mathbf{M}+\mathbf{M}^\intercal)\mathbf{x}\tag{4.46}$$

If we let $\mathbf{M}=\mathbf{X}^\intercal\mathbf{X}$ then it follows:

$$\frac{\partial}{\partial\beta}\beta^\intercal\mathbf{X}^\intercal(\mathbf{X}\beta)=(\mathbf{X}^\intercal\mathbf{X}+(\mathbf{X}^\intercal\mathbf{X})^\intercal)\beta=2\mathbf{X}^\intercal\mathbf{X}\beta$$

Thus

$$0=\frac{\partial}{\partial\beta}(\mathbf{y}-\mathbf{X}\beta)^\intercal(\mathbf{y}-\mathbf{X}\beta)=2\mathbf{X}^\intercal(\mathbf{X}\beta-\mathbf{y})\tag{4.47}$$

combine proofs

Proof 4.4: [def. 4.13]

$$\begin{aligned}\text{lsq}(\mathbf{X},\mathbf{y})&=(\mathbf{y}-\mathbf{X}\beta)^\intercal(\mathbf{y}-\mathbf{X}\beta)\\&=\mathbf{y}^\intercal\mathbf{y}-\mathbf{y}^\intercal\mathbf{X}\beta+(\mathbf{X}\beta)^\intercal\mathbf{y}-(\mathbf{X}\beta)^\intercal(\mathbf{X}\beta)\\&=\mathbf{y}^\intercal\mathbf{y}-2\mathbf{y}^\intercal\mathbf{X}\beta+\beta^\intercal\mathbf{X}^\intercal(\mathbf{X}\beta)\end{aligned}$$

$$0=\frac{\partial}{\partial\beta}\text{lsq}(\mathbf{X},\mathbf{y})=2\mathbf{X}^\intercal\mathbf{X}\beta-2\mathbf{X}^\intercal\mathbf{y}=2\mathbf{X}^\intercal(\mathbf{X}\beta-\mathbf{y})$$

Note

$$\frac{\partial}{\partial\beta}\beta^\intercal\mathbf{X}^\intercal(\mathbf{X}\beta)\overset{??}{=}(\mathbf{X}^\intercal\mathbf{X}+(\mathbf{X}^\intercal\mathbf{X})^\intercal)\beta=2\mathbf{X}^\intercal\mathbf{X}\beta$$

Proof 4.5: Corollary 4.2

$$\begin{aligned}&(\mathbf{X}\beta-\mathbf{y})&&\perp\mathfrak{R}(\mathbf{X})\\\iff&(\mathbf{X}\beta)^\intercal(\mathbf{X}\beta-\mathbf{y})=\mathbf{0}&&\forall\beta\in\mathbb{R}^m\\\iff&\mathbf{X}^\intercal(\mathbf{X}\beta-\mathbf{y})=\mathbf{0}\end{aligned}$$

where $\mathbf{X}=\{\mathbf{x}_{:,1},\dots,\mathbf{x}_{:,m}\}$ is the "basis" of the Range space:

$$(\mathbf{X}\beta-\mathbf{y})^\intercal\mathbf{x}_{:,j}=\mathbf{0}\qquad\forall j=1,\dots,m$$

Proof 4.6 Corollary 4.3: From [def. 4.13] it follows:

$$\begin{aligned}&\mathbf{X}^\intercal\mathbf{Y}=\mathbf{X}^\intercal\mathbf{X}\hat{\beta}=\hat{\beta}^\intercal\mathbf{X}^\intercal\mathbf{X}=(\mathbf{X}\hat{\beta})^\intercal\mathbf{X}\\\iff\quad&(\mathbf{Y}-\mathbf{X}\hat{\beta})\mathbf{X}=\mathbf{r}^\intercal\mathbf{X}=0\end{aligned}$$

Proof 4.7 Property 4.4: $\hat{\beta}$ an unbiased estimator of β:

$$\begin{aligned}\hat{\beta}&=(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal y=(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal(\mathbf{X}\beta+\epsilon)\\&=\beta+(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\epsilon\\\mathbb{E}_\epsilon[\hat{\beta}]&=\mathbb{E}[\beta]+\mathbb{E}[(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\epsilon]\\&=\beta+(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\underbrace{\mathbb{E}[\epsilon]}_{=0}=\beta\end{aligned}$$

Proof 4.8 Property 4.4: Covariance $\sigma^2(\mathbf{X}^\intercal\mathbf{X})^{-1}$:

$$\begin{aligned}\text{Cov}\left[\hat{\beta}\right]&=\overbrace{\text{Cov}\left[\beta\right]}^{=0}+\overbrace{\text{Cov}\left[(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\epsilon\right]}^{:=\mathbb{V}[\alpha\epsilon]}=\mathbb{E}[(\alpha\epsilon)^2]-\overbrace{\mathbb{E}[\alpha\epsilon]^2}^{=0}\\&=\mathbb{E}[(\alpha\epsilon)^\intercal(\alpha\epsilon)]=\mathbb{E}\left[(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\epsilon\epsilon^\intercal\mathbf{X}(\mathbf{X}^\intercal\mathbf{X})\right]\\&=(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbb{E}\left[\epsilon\epsilon^\intercal\right]\mathbf{X}(\mathbf{X}^\intercal\mathbf{X})\\&=\sigma^2(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{X}(\mathbf{X}^\intercal\mathbf{X})=\sigma^2(\mathbf{X}^\intercal\mathbf{X})^{-1}\end{aligned}$$

Proof 4.9 Property 4.5: $\hat{\mathbf{y}}$ an unbiased estimator of $\mathbf{y}$:

$$\mathbb{E}_\epsilon[\hat{\mathbf{y}}]=\mathbb{E}[\mathbf{X}\hat{\beta}+\epsilon]=\mathbf{X}\mathbb{E}[\hat{\beta}]+0\overset{\text{eq. (4.25)}}{=}\mathbf{X}\beta=\mathbb{E}[\mathbf{y}]$$

Proof 4.10 Theorem 4.3: $\hat{\beta}$ is a linear operator w.r.t. to $\mathbf{y}$:

$$\begin{aligned}\hat{\beta}&=(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal y=:\mathbf{C}\mathbf{y}=\mathbf{C}(\mathbf{X}\beta)\\&=\beta+(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\epsilon=:\tilde{C}\epsilon+\beta\end{aligned}$$

5. Examples

Example 4.1 Simple Linear Regression:

$$p=2\qquad\mathbf{X}=\begin{pmatrix}1&x_1\\1&x_2\\\vdots&\vdots\\1&x_n\end{pmatrix}\qquad\beta=\begin{pmatrix}\beta_1\\\beta_2\end{pmatrix}$$

Example 4.2 Simple Linear Quadratic Regression:

$$p=3\qquad\mathbf{X}=\begin{pmatrix}1&x_1&x_1^2\\1&x_2&x_2^2\\\vdots&\vdots&\vdots\\1&x_n&x_n^2\end{pmatrix}\qquad\beta=\begin{pmatrix}\beta_1\\\beta_2\\\beta_2\end{pmatrix}$$

Classification

6. Intro

Definition 4.19 Training Data $\mathcal{D}$:

$$\mathcal{D} := \left\{(\mathbf{x}_i, y_i) \;\middle|\; \mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d, y_i \in \mathcal{Y} := \{c_1, \dots, c_K\}\right\}$$

Definition 4.20 Classifier c:

Is a mapping that maps the features into classes:

$$c : \mathcal{X} \to \mathcal{Y} \tag{4.48}$$

Definition 4.21 Dichotomy:

Given a set $\mathcal{S} = \{s_1, \dots, s_N\}$ a dichotomy is partition of the set $\mathcal{S}$ into two subsets A, A^{C} that satisfy:

- *Collectively/jointly exhaustiveness*:

$$S = A \cup A^{\mathrm{C}} \tag{4.49}$$

- *Mutual exclusivity*:

$$s \in A \implies s \notin A^{\mathrm{C}} \qquad \forall s \in S \tag{4.50}$$

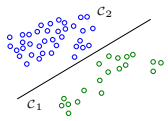
Explanation 4.1. *Nothing can belong simultaneously to both parts A and A^c.*

Types of Classification

Definition 4.22 Binaray Classification:

Is a classification problem where the labels are binary:

$$\mathcal{Y} = \{c_1, c_2\} = \{-1, 1\} \tag{4.51}$$



multiclass

Types of Categorical Data

Definition 4.23 Nominal/Categorical Data:

Is data where variables belong to a finite set of classes $\{c_1, \dots, c_K\}$ that do not have any ordering.

Definition 4.24 Ordinal Data:

Is data where variables belong to a finite discrete set of classes $\{c_1, \dots, c_K\}$ that are ordered/do have an ranking between each other i.e. numbers.

Encodings

6.3.1. Ordinal Encoding

Definition 4.25 Ordinal Encdoing:

Each category gets assigned an integer values to introduce an order to the data.

Usage: for ordinal data, where we want to preserve order.

Cons

- models such as neural networks output a continues value, thus we are in fact treating a mulit-class classification problem as regression problem.

6.3.2. One Hot Encoding

Definition 4.26 One-hot encoding/representation:

Is the representation/encoding of the K categories $\{c_1, \dots, c_K\}\}$ by a *sparse vectors*?? with one non-zero entry, where the index j of the non-zero entry indicates the class c_j:

$$\mathbb{B}^n = \left\{\mathbf{y} \in \{0,1\}^n : \mathbf{y}^\intercal \mathbf{y} = \sum_{i=1}^{n} \mathbf{y} = 1\right\}$$

s.t. $\mathbf{y}_i = \mathbf{e}_j \iff \mathbf{y}_i = c_j$

Usage: for data where we do not want any order.

MNIST

I.e. for digit recognition we should treat our numbers as a set we do care that a 9 is classified as 9 but do not care that it comes after an .

6.3.3. Soft vs. Hard Labels

Definition 4.27 Hard Labels/Targets: Are observations $y \in \mathcal{Y}$ that are consider as true observations. We can encode them using a one hot encoding[def. 4.26]:

$$y = c_k \implies y = \mathbf{e}_k \tag{4.52}$$

Definition 4.28 Soft Labels/Targets: Are observations $y \in \mathcal{Y}$ that are consider as noisy observations or probabilities p. We can encode them using a probabilistic vector??:

$$y = [\mathrm{p}_1 \cdots\cdots \mathrm{p}_K]^\intercal \tag{4.53}$$

Corollary 4.8 Hard labels as special case: If we consider hard targets[def. 4.27] as events with probability one then we can think of them as a special case of the soft labels.

7. Binary Classification $\{-1, 1\}$

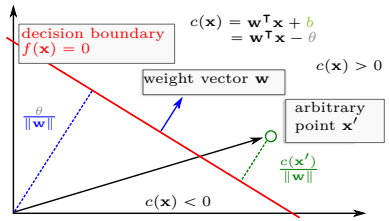
7.1. Linear Classification

Definition 4.29 Linear Dichotomy:

Definition 4.30 Linear Classifier: A linear classifier is a classifier c that assigns labels $\hat{y}$ to samples $\mathbf{x}_i$ using a *linear decision boundary/hyperplane*??:

$$\hat{y} = c(\mathbf{x}_i) = \begin{cases} c_1 \in \mathcal{H}^+ & \text{if } \mathbf{w}^\intercal \mathbf{x} > \theta \\ c_2 \in \mathcal{H}^- & \text{if } \mathbf{w}^\intercal \mathbf{x} < \theta \end{cases} \tag{4.54}$$

Explanation 4.2 (Definition 4.30).



- The $b \in \mathbb{R}$ corresponds to the offset of the decision surface from the origin, otherwise the decision surface would have to pass through the origin.
- $\mathbf{w} \in \mathbb{R}^d$ is the normal unit vector of the decision surface. Its components $\{w_j\}_{j=1}^d$ correspond to the importance of each feature/dimension.

Explanation 4.3 (Threshold θ vs. Bias b). *The offset is called bias if it is considered as part of the classifier $\mathbf{w}^\intercal \mathbf{x} + b$ and as threshold if it is considered to be part of the hyperplane $\theta = -b$, but its just a matter of definition.*

Definition 4.31 (Normalized) Classification Criterion:

$$\tilde{\mathbf{w}}^\intercal \mathbf{x} = \mathbf{w}^\intercal \mathbf{x} y > 0 \qquad \forall (\mathbf{x}, y) \in \mathcal{D} \tag{4.55}$$

Definition 4.32 Linear Separable Data set: A data set is *linearly separable* if there exists a separating hyperplane $\mathcal{H}$ s.t. each label can be assigned correctly:

$$\hat{y} := c(\mathbf{x}) = y \qquad \forall (\mathbf{x}, \mathbf{y}) \in \mathcal{D} \tag{4.56}$$

7.1.1. Normalization

Proposition 4.3 Including the Offset: In order to simplify notation the offset is usually included into the parameter vector:

$$\mathbf{w} \leftarrow \begin{pmatrix} \mathbf{w} \\ b \end{pmatrix} \qquad \mathbf{x} \leftarrow \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

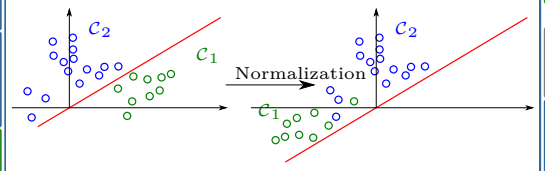
$$\implies \mathbf{w}^\intercal \mathbf{x} = \begin{pmatrix} \mathbf{w}^\intercal & b \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} = \mathbf{w}^\intercal \mathbf{x} + b$$

Proposition 4.4 Uniform Classification Criterion: In order to avoid the case distinction in the classification criterion of eq. (4.54) we may transform the input samples by:

$$\tilde{\mathbf{x}} = \begin{cases} \mathbf{x} & \text{if } \mathbf{w}^\intercal \mathbf{x} > \theta \\ -\mathbf{x} & \text{if } \mathbf{w}^\intercal \mathbf{x} < \theta \end{cases} \tag{4.57}$$

Explanation 4.4 (proposition 4.4).

We transform the input s.t. the separating hyper-plane puts all labels on the same "positive" side $\mathbf{w}^\intercal \mathbf{x} > 0$.



Corollary 4.9 : How can we achieve this in practice?

If $\mathcal{Y} = \{-1, 1\}$ then we can simply multiply with the label y_i:

$$\left.\begin{aligned} \mathbf{w}^\intercal \mathbf{x} > 0 \quad \forall y = +1 \\ \mathbf{w}^\intercal \mathbf{x} < 0 \quad \forall y = -1 \end{aligned}\right\} \iff \mathbf{w}^\intercal \mathbf{x} \cdot \hat{y} > 0 \quad \forall y$$

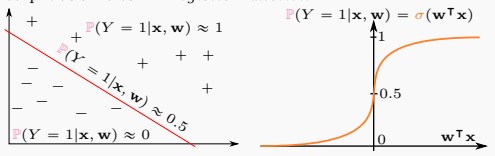
8. Logistic Regression $\mathrm{Bern}(y; \sigma(\mathbf{w}^\intercal \mathbf{x}, \sigma^2))$

Idea: in order to classify dichotomies[def. 4.21] we use a distribution that maps probabilities to a binary values 0/1

$\Rightarrow$ *Bernoulli Distribution*??.

Problem: we need to convert/translate distance $\mathbf{w}^\intercal \mathbf{x}$ into probability in order to use a bernoulli distribution.

Idea: use a sigmoidal function to convert distances $\mathbf{z} := \mathbf{w}^\intercal \mathbf{x}$ into probabilities $\Rightarrow$ *Logistic Function*[def. 4.33].



8.1. Logistic Function

Definition 4.33 Sigmoid/Logistic Function:

$$\sigma(z) = \frac{1}{1 + \mathrm{e}^{-z}} = \frac{1}{1 + \mathrm{e}^{\text{neg. dist. from deci. boundary}}} \tag{4.58}$$

Explanation 4.5 (Sigmoid/Logistic Function).

$$\sigma(z) = \begin{cases} 0 & -z \text{ large} \\ 1 & \text{if} \quad z \text{ large} \\ 0.5 & z = 0 \end{cases}$$

8.2. Logistic Regression

Definition 4.34 Logistic Regression:

models the likelihood of the output y as a Bernoulli Distribution?? $y \sim \mathrm{Bern}(\mathrm{p})$, where the probability p is given by the Sigmoid function[def. 4.33] of a linear regression:

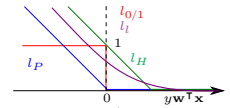
$$\mathrm{p}(y|\mathbf{x}, \mathbf{w}) = \mathrm{Bern}\left(\sigma(\mathbf{w}^\intercal \mathbf{x})\right) = \begin{cases} \frac{1}{1+\mathrm{e}^{-\mathbf{w}^\intercal \mathbf{x}}} & \text{if } y = +1 \\ 1 - \frac{1}{1+\mathrm{e}^{-\mathbf{w}^\intercal \mathbf{x}}} & \text{if } y = -1 \end{cases}$$

$$\overset{??}{=} \overset{4.11}{\frac{1}{1 + \exp(-y \cdot \mathbf{w}^\intercal \mathbf{x})}} = \sigma\left(-y \cdot \mathbf{w}^\intercal \mathbf{x}\right) \tag{4.59}$$

8.2.1. Maximum Likelihood Estimate

Definition 4.35 Logistic Loss l_l proof 4.12:

Is the objective we want to minimize when performing mle[def. 6.3] for a logistic regression likelihood and incurs higher cost for samples closer to the decision boundary:



$$l_l(\mathbf{w}; \mathbf{x}, y) := \log\left(1 + \exp(-y \cdot \mathbf{w}^\intercal \mathbf{x})\right) \tag{4.60}$$

$$\propto \log(1 + \mathrm{e}^z) = \begin{cases} z & \text{for large } z \\ 0 & \text{for small } z \end{cases}$$

Corollary 4.10 MLE for Logistic Regression:

$$\mathrm{l}_n(\mathbf{w}) = \sum_{i=1}^{n} l_l = \sum_{i=1}^{n} \log\left(1 + \exp(-y_i \cdot \mathbf{w}^\intercal \mathbf{x}_i)\right) \tag{4.61}$$

Stochastic Gradient Descent

The logistic loss l_l is a convex function. Thus we can use convex optimization techniques s.a. SGD in order to minimize the objective [cor. 4.10].

Definition 4.36 proof 4.13

Logistic Loss Gradient $\nabla_{\mathbf{w}} l_l(\mathbf{w})$:

$$\nabla_{\mathbf{w}} l_l(\mathbf{w}) = \mathbb{P}(Y = -y|\mathbf{x}, \mathbf{w}) \cdot (-y\mathbf{x}) = \frac{1}{1 + \exp(y\mathbf{w}^\intercal \mathbf{x})} \cdot (-y\mathbf{x}) \tag{4.62}$$

Explanation 4.6.

$$\nabla_{\mathbf{w}} l_l(\mathbf{w}) = \mathbb{P}(Y = -y|\mathbf{x}, \mathbf{w}) \cdot (-y\mathbf{x}) \propto \nabla_{\mathbf{w}} l_H(\mathbf{w})$$

The logistic loss l_l is equal to the hinge loss l_h but weighted by the probability of beeing in the wrong class $\mathbb{P}(Y = -1|\mathbf{x}, \mathbf{w})$. Thus the more likely we are in the wrong class the bigger the step we take:

$$\mathbb{P}(Y = -y|\hat{y} = \mathbf{w}^\intercal \mathbf{x}) = \begin{cases} \uparrow & \textit{take big step} \\ \downarrow & \textit{take small step} \end{cases}$$

Algorithm 4.1 Vanilla SGD for Logistic Regression:

Initalize: $\mathbf{w}$

1: **for** $1, 2 \dots, T$ **do**
2: Pick $(\mathbf{x}, y)$ unif. at randomn from data $\mathcal{D}$
3: $\mathbb{P}(Y = -y|\mathbf{x}, \mathbf{w}) = \frac{1}{(1+\exp(-y \cdot \mathbf{w}^\intercal \mathbf{x}))} = \sigma(y \cdot \mathbf{w}^\intercal \mathbf{x})$
$\triangleright$ compute prob. of misclassif. with cur. model
4: $\mathbf{w} = \mathbf{w} + \eta_t y \mathbf{x} \sigma(y \cdot \mathbf{w}^\intercal \mathbf{x})$
5: **end for**

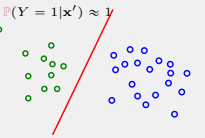
Making Predictions

Given an optimal parameter vector $\hat{\mathbf{w}}$ found by algorithm 4.1 we can predict the output of a new label by eq. (4.59):

$$\mathbb{P}(y|\mathbf{x}, \hat{\mathbf{w}}) = \frac{1}{1 + \exp\left(-y\hat{\mathbf{w}}^\intercal \mathbf{x}\right)} \tag{4.63}$$

Drawback

Logistic regression, does not tell us anything about the liklihood $\mathrm{p}(\mathbf{x})$ of a point, thus it will not be able to detect outliers, as it will assign a very high probability to all correctly classfied points, far from the decsion boundary.



8.2.2. Maximum a-Posteriori Estimates

adapt

8.3. Logistic regression and regularization

Adding Priors to Logistic Liklihood

- **L2 (Gaussian prior)**:

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{n} \log\left(1 + \exp(-y_i \mathbf{w}^\intercal \mathbf{x}_i)\right) + \lambda \|\mathbf{w}\|_2^2$$

- **L1 (Laplace prior)**:

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{n} \log\left(1 + \exp(-y_i \mathbf{w}^\intercal \mathbf{x}_i)\right) + \lambda \|\mathbf{w}\|_1$$

- **Generalized**:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} \log\left(1 + \exp(-y_i \mathbf{w}^\intercal \mathbf{x}_i)\right) + \lambda C(\mathbf{w})$$
$$= \arg\max_{\mathbf{w}} \mathbb{P}(\mathbf{w}|\mathbf{X}, Y)$$

8.4. SGD for L2-gregularized logistic regression

Initalize: $\mathbf{w}$

1: **for** $1, 2 \ldots, T$ **do**

2: Pick $(\mathbf{x}, y)$ unif. at randomn from data $\mathcal{D}$

3: $\hat{\mathbb{P}}(Y = -y|\mathbf{x}, \mathbf{w}) = \frac{1}{(1+\exp(-y \cdot \mathbf{w}^\intercal \mathbf{x}))}$

$\triangleright$ compute prob. of misclassif. with cur. model

4: $\mathbf{w} = \mathbf{w}(1 - 2\lambda\eta_t) + \eta_t y \mathbf{x} \hat{\mathbb{P}}(Y = -y|\mathbf{x}, \mathbf{w})$

5: **end for**

Thus: $\mathbf{w}$ is pulled/shrunken towards zero, depending on the regularization parameter $\lambda > 0$

9. Proofs

Proof 4.11: [def. 4.34] We need to only proof the second expression, as the first one is fulfilled anyway:

$$1 - \frac{1}{1+\mathrm{e}^z} = \frac{1+\mathrm{e}^z}{1+\mathrm{e}^z} - \frac{1}{1+\mathrm{e}^z} = \frac{\mathrm{e}^z + 1 - 1}{1+\mathrm{e}^z} = \frac{\mathrm{e}^z}{\mathrm{e}^z+1}$$
$$= \frac{1}{1+\mathrm{e}^{-z}}$$

Proof 4.12: [def. 4.35]

$$\mathrm{l}_n(\mathbf{w}) = \arg\max_{\mathbf{w}} \mathrm{p}(y_{1:n}|\mathbf{x}_{1:n}, \mathbf{w}) = \arg\min_{\mathbf{w}} -\log \mathrm{p}(Y|\mathbf{X}, \mathbf{w})$$
$$\overset{\text{i.i.d.}}{=} \arg\min_{\mathbf{w}} \sum_{i=1}^{n} -\log \mathrm{p}(y_i|\mathbf{x}_i, \mathbf{w})$$
$$\overset{\text{eq. (4.59)}}{=} -\log \frac{1}{1+\exp(-y_i \cdot \mathbf{w}^\intercal \mathbf{x}_i)}$$
$$= \log\left(1 + \exp(-y_i \cdot \mathbf{w}^\intercal \mathbf{x}_i)\right) =: l_l(\mathbf{w})$$

Proof 4.13: [def. 4.36]

$$\nabla_{\mathbf{w}} l_l(\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}} \log\left(1 + \exp(-y \cdot \mathbf{w}^\intercal \mathbf{x})\right)$$
$$\overset{\text{C.R.}}{=} \frac{1}{(1+\exp(-y \cdot \mathbf{w}^\intercal \mathbf{x}))} \frac{\partial}{\partial \mathbf{w}} \left(1 + \exp(-y \cdot \mathbf{w}^\intercal \mathbf{x})\right)$$
$$\overset{\text{C.R.}}{=} \frac{1}{(1+\exp(-y \cdot \mathbf{w}^\intercal \mathbf{x}))} \exp(-y \cdot \mathbf{w}^\intercal \mathbf{x}) \cdot (-y\mathbf{x})$$
$$= \frac{e^{-z} \cdot (-yx)}{(1+e^{-z})} = \frac{-yx}{e^z(1+e^{-z})} = \frac{-yx}{(e^z + e^{-z+z})}$$
$$= \frac{1}{\exp(y \cdot \mathbf{w}^\intercal \mathbf{x}) + 1} \cdot (-yx)$$
$$\overset{\text{eq. (4.59)}}{=} \hat{\mathbb{P}}(Y = -y|\mathbf{x}, \mathbf{w}) \cdot (-y\mathbf{x})$$

Generalized Linear Models (GLMs)

1. Generalized Additive Models (GAMs)

Definition 5.1 $g_{\mathbf{add}} : \mathbb{R}^p \mapsto \mathbb{R}$

Generalized Additive Models (GAM):
Are generalized linear model where the response variable depends linearly on unknown smooth functions g_j s.t.:

$$g_{\text{add}}(\mathbf{x}) = \mu + \sum_{j=1}^{p} g_j(x_j) \qquad \begin{array}{l} g_j : \mathbb{R} \mapsto \mathbb{R} \\ \mathbb{E}\left[g_j(x_j)\right] = 0 \end{array} \qquad \forall j \in \{1, \dots, p\} \tag{5.1}$$

Pros
- Does not suffer from the curse of dimensionality.

Cons
- does not allow for interaction terms such as $g_{j,k}(x_j, x_k)$.

1.1. Backfitting

add some point check again Deviance and R squared adj from R output

Model Parameter Estimation

1. Maximum Likelihood Estimation

1.1. Likelihood Function

Is a method for estimating the parameters θ of a model that agree best with observed data $\{x_1, \dots, x_n\}$. **Let**: $\theta = (\theta_1 \ \dots \ \theta_k)^\intercal \in \Theta \mathbb{R}^k$ vector of unknown model parameters.
Consider: a probability density/mass function $f_X(\mathbf{x}; \theta)$

Definition 6.1 Likelihood Function $\mathscr{L}_n : \Theta \times \mathbb{R}^n \mapsto \mathbb{R}_+$:
Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ be a random sample of i.i.d. data points drawn from an unknown probability distribution $\mathbf{x}_i \sim p_{\mathcal{X}}$.
The likelihood function gives the likelihood/probability of the *joint probability* of the data $\{x_1, \dots, x_n\}$ *given* a fixed set of model parameters θ:

$$\mathscr{L}_n(\theta|\mathbf{X}) = \mathscr{L}_n(\theta; \mathbf{X}) = f(\mathbf{X}|\theta) = f(\mathbf{X}; \theta) \tag{6.1}$$

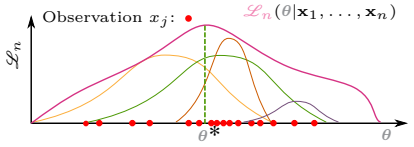


Figure 5: Possible Likelihood function in pink.
Overlayed: possible candidate functions for Gaussian model explaining the observations.

Likelihood function is not a pdf

The likelihood function by default not a probability density function and may not even be differentiable. However if it is, then it may be normalized to one.

Corollary 6.1 i.i.d. data: If the n-data points of our sample are i.i.d. then the likelihood function can be decomposed into a product of n-terms:

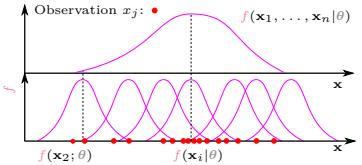


Figure 6: Bottom: probability distributions of the different data points $\mathbf{x}_i$ given a fixed θ for a Gaussian distribution
Top: joint probability distribution of the i.i.d. data points $\{\mathbf{x}_i\}_{i=1}^n$ given a fixed θ

$$f(\mathbf{x}_1, \dots, \mathbf{x}_n|\theta) \overset{\text{i.i.d.}}{=} \prod_{i=1}^n f(\mathbf{x}_i|\theta)$$

Notation

- The probability density $f(\mathbf{X}|\theta)$ is considered for a fixed θ and thus as a function of the samples.
- The likelihood function on the other hand is considered as a function over parameter values θ for a fixed sample $\{\mathbf{x}_i\}_{i=1}^n$ and thus written as $\mathscr{L}_n(\theta|\mathbf{X})$.
- Often the colon symbol **;** is written instead of the *is given* symbol | in order to indicate that θ resp. $\mathbf{X}$ is a parameter and not a randomn variable.

1.2. Maximum Likelihood Estimation (MLE)

Let $f_\theta(\mathbf{x})$ be the probability of an i.i.d. sample $\mathbf{x}$ for a given model.
Goal: find θ of a given model that maximizes the joint probability/likelihood of the observed data $\{x_1, \dots, x_n\}$? $\iff$ maximum likelihood estimator θ^*.

Definition 6.2 Log Likelihood Function $\mathrm{l}_n : \Theta \times \mathbb{R}^n \mapsto \mathbb{R}$:

$$\mathrm{l}_n(\theta|\mathbf{X}) = \log \mathscr{L}_n(\theta|\mathbf{X}) = \log f(\mathbf{X}|\theta) \tag{6.2}$$

Corollary 6.2 i.i.d. data: Differentiating the product of n-Terms with the help of the chain rule leads often to complex terms. As a result one usually prefers maximizing the log (especially for exponential terms), as it does not change the *argmax*–**??**:

$$\log f(\mathbf{x}_1, \dots, \mathbf{x}_n|\theta) \overset{\text{i.i.d.}}{=} \log\left(\prod_{i=1}^n f(\mathbf{x}_i|\theta)\right) = \sum_{i=1}^n \log f(\mathbf{x}_i|\theta)$$

Definition 6.3 Maximum Likelihood Estimator θ^*:
Is the estimator $\theta^* \in \Theta$ that maximizes the likelihood of the model/predictor:

$$\theta^* = \arg\max_{\theta \in \Theta} \mathscr{L}_n(\theta; \mathbf{x}) \quad \text{or} \quad \theta^* = \arg\max_{\theta \in \Theta} \mathrm{l}_n(\theta; \mathbf{x}) \tag{6.3}$$

1.3. Maximization vs. Minimization

For optimization problems we minimize by convention.
The logarithm is a concave function?? $\cap$, thus if we calculate the extremal point we will obtain a maximum.
If we want to calculate a mimimum instead (i.e. in order to be compatible with some computer algorithm) we can convert the function into a convex function?? $\cup$ by multiplying it by minus one and consider it as a loss function instead of a likelihood.

Definition 6.4 Negative Log-likelihood $-\mathrm{l}_n(\theta|\mathbf{X})$:

$$\theta^* = \arg\max_{\theta \in \Theta} \mathrm{l}_n(\theta|\mathbf{X}) = \arg\min_{\theta \in \Theta} -\mathrm{l}_n(\theta|\mathbf{X}) \tag{6.4}$$

1.4. Conditional Maximum Likelihood Estimation

Maximum likelihood estimation can also be used for conditional distributions.
Assume the labels y_i are drawn i.i.d. from a unknown true conditional probability distribution $f_{Y|X}$ and we are given a data set $\mathbf{Z} = \left\{(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}\right\}_{i=1}^n$.
Now we want to find *the parameters* $\theta = (\theta_1 \ \dots \ \theta_k)^\intercal \in \Theta \mathbb{R}^k$ of a hypothesis $\hat{f}_{Y|X}$ that agree best with the given data $\mathcal{Z}$.

Note

For simplicity we omit the hat $\hat{}$ of our model $\hat{f}_{Y|X}$ and simply assume that our data is generated by some data generating probability distribution.

Definition 6.5 Conditional (log) likelihood function:
Models the liklihood of a model with parameters θ given the data $\mathbf{Z} = \{\mathbf{x}_i, y_i\}_{i=1}^n$

$$\mathscr{L}_n(\theta|Y, \mathbf{X}) = \mathscr{L}_n(\theta; Y, \mathbf{X}) = f(Y|\mathbf{X}, \theta) = f(Y|\mathbf{X}; \theta)$$

2. Maximum a posteriori estimation (MAP)

Idea

We have seen (**??**), that trading/increasing a bit of bias can lead to a big reduction of variance of the generalization error. We also know that the least squares MLE is unbiased (**??**).
Thus the question arises if we can introduce a bit of bias into the MLE in turn of decreasing the variance?
$\Rightarrow$ use Bayes rule (**??**) to introduce a bias into our model via. a Prior distribution.

2.1. Prior Distribution

Definition 6.6 Prior (Distribution) $\pi(\theta) = p(\theta)$:
Assumes: that the model parameters θ are no longer constant **but** random variables distributed according to a prior distribution that models some prior belief/bias that we have about the model:

$$\theta \sim \pi(\theta) = p(\theta) \tag{6.5}$$

Notes

In this section we use the terms model parameters θ and model as synonymous, as the model is fully described by its population parameters ([def. 3.12]) θ.

Corollary 6.3 The prior is independent of the data: The prior $p(\theta)$ models a prior belief/bias and is thus independent of the data $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$:

$$p(\theta|\mathbf{X}) = p(\theta) \tag{6.6}$$

Definition 6.7 Hyperparameters $p_{\boldsymbol{\lambda}}(\theta)$:
In most cases the prior distribution are parameterized that is the pdf $\pi(\theta|\boldsymbol{\lambda})$ depends on a set of parameters $\boldsymbol{\lambda}$.
The parameters of the prior distribution, are called hyperparameters and are supplied due to believe/prior knowledge (and do not depend on the data) see example 6.1

2.2. Posterior Distribution

Definition 6.8 Posterior Distribution $p(\theta|\textbf{data})$:
The posterior distribution $p(\theta|\text{data})$ is a probability distribution that describes the relationship of a unknown *parameter* θ a posterior/after observing evidence of a random quantity $\mathbf{Z}$ that is in a relation with θ:

$$p(\theta|\textbf{data}) = p(\theta|\mathbf{Z}) \tag{6.7}$$

Definition 6.9
Posterior Distribution and Bayes Theorem:
Using Bayes **??** we can write the posterior distribution as a product of the *likelihood*[def. 6.1] weighted with our *prior*[def. 6.6] and normalized by the *evidence* $\mathbf{Z} = \{\mathbf{X}, \mathbf{y}\}$ s.t. we obtain a real probability distribution:

$$p(\theta|\textbf{data}) = p(\theta|\mathbf{Z}) = \frac{p(\mathbf{Z}|\theta) \cdot p_{\boldsymbol{\lambda}}(\theta)}{p(\mathbf{Z})} \tag{6.8}$$

$$\text{Posterior} = \frac{\text{Liklihood} \cdot \text{Prior}}{\text{Normalization}} \tag{6.9}$$

$$\boxed{p(\theta|\mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y}|\theta, \mathbf{X}) \cdot p_{\boldsymbol{\lambda}}(\theta)}{p(\mathbf{y}|\mathbf{X})}} \tag{6.10}$$

see proof **??** 6.1

2.2.1. Maximization –MAP

We do not care about the full posterior probability distribution as in Bayesian Inference (section 4). We only want to find a point estimator **??** θ^* that maximizes the posterior distribution.

2.2.2. Maximization

Definition 6.10
Maximum a-Posteriori Estimates (MAP):
Is model/parameters θ that maximize the posterior probability distribution:

$$\boxed{\theta^*_{\text{MAP}} = \arg\max_\theta \mathbb{P}(\theta|\mathbf{X}, \mathbf{y})} \tag{6.11}$$

Log-MAP estimator:

$$\begin{aligned}\theta^* &= \arg\max_\theta \{p(\theta|\mathbf{X}, \mathbf{y})\} \qquad (6.12)\\ &= \arg\max_\theta \left\{\frac{p(\mathbf{y}|\mathbf{X}, \theta) \cdot p_{\boldsymbol{\lambda}}(\theta)}{p(\mathbf{y}|\mathbf{X})}\right\}\\ &\overset{??}{\propto} \arg\max_\theta \left\{p(\mathbf{y}|\theta, \mathbf{X}) \cdot p_{\boldsymbol{\lambda}}(\theta)\right\}\end{aligned}$$

Corollary 6.4 Negative Log MAP:

$$\begin{aligned}\theta^* &= \arg\max_\theta \{p(\theta|\mathbf{X}, \mathbf{y})\} \qquad (6.13)\\ &= \arg\min_\theta -\log \overbrace{p(\theta)}^{\text{Prior}} - \log \overbrace{p(\mathbf{y}|\theta, \mathbf{X})}^{\text{Likelihood}} + \underbrace{\log p(\mathbf{y}|\mathbf{X})}_{\text{not depending on } \theta}\end{aligned}$$

3. Proofs

Proof 6.1: 6.10:

$$\begin{aligned}p(\mathbf{X}, \mathbf{y}, \theta) &= \begin{cases}p(\theta|\mathbf{X}, \mathbf{y})p(\mathbf{X}, \mathbf{y})\\ p(\mathbf{y}|\mathbf{X}, \theta)p(\mathbf{X}, \theta)\end{cases}\\ p(\theta|\mathbf{X}, \mathbf{y})p(\mathbf{X}, \mathbf{y}) &= p(\theta|\mathbf{X}, \mathbf{y})p(\mathbf{y}|\mathbf{X})p(\mathbf{X})\\ p(\mathbf{y}|\mathbf{X}, \theta)p(\mathbf{X}, \theta) &= p(\mathbf{y}|\mathbf{X}, \theta)p(\theta)p(\mathbf{X})\\ &= p(\mathbf{y}|\mathbf{X}, \theta)p(\theta|\mathbf{X})p(\mathbf{X})\\ &\overset{\text{eq. (6.6)}}{=} p(\mathbf{y}|\mathbf{X}, \theta)p(\theta)p(\mathbf{X})\\ \Rightarrow p(\theta|\mathbf{X}, \mathbf{y}) &= \frac{p(\mathbf{y}|\mathbf{X}, \theta)p(\theta)\cancel{p(\mathbf{X})}}{p(\mathbf{y}|\mathbf{X})\cancel{p(\mathbf{X})}}\end{aligned}$$

Note

This can also be derived by using the normal Bayes rule but additionally condition everything on $\mathbf{X}$ (where the prior is independent on $\mathbf{X}$)

4. Examples

Example 6.1 Hyperparameters Gaussian Prior:

$$f_{\boldsymbol{\lambda}}(\theta) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\theta - \mu)^2}{2\sigma^2}\right)$$

with the hyperparameter $\boldsymbol{\lambda} = (\mu \ \ \sigma^2)^\intercal$.

Bayesian Inference/Modeling

Definition 6.11 Bayesian Inference: So far we only really looked at point estimators/estimates[def. 3.13].
But what if we are interested not only into the most likely value but also want to have a notion of the uncertainty of our prediction? Bayesian inference refers to statistical inference[def. 3.10], where uncertainty in inferences is quantified using probability. Thus we usually obtain a distribution over our parameters and not a single point estimates
⇒ can deduce statistical properties of parameters from their distributions.

Definition 6.12 $\mathrm{p}(\mathbf{w}|\mathbf{y},\mathbf{X})/\mathrm{p}(\mathbf{w}|\mathcal{D})$
Posterior Probability Distribution:
1. Specify the prior $\mathrm{p}_\lambda(\mathbf{w})$
2. Specify the likelihood $\mathrm{p}(\mathbf{y}|\mathbf{w},\mathbf{X})/\mathrm{p}(\mathcal{D}|\mathbf{w})$
3. Calculate the evidence $\mathrm{p}(\mathbf{y}|\mathbf{X})/\mathrm{p}(\mathcal{D})$
4. Calculate the posterior distribution $\mathbb{P}(\mathbf{w}|\mathbf{y},\mathbf{X})/\mathrm{p}(\mathbf{w}|\mathcal{D})$

$$\mathrm{p}(\mathbf{w}|\mathbf{y},\mathbf{X}) = \frac{\mathrm{p}(\mathbf{y}|\mathbf{w},\mathbf{X})\cdot \mathrm{p}_\lambda(\mathbf{w})}{\mathrm{p}(\mathbf{y}|\mathbf{X})} = \frac{\text{Liklihood}\cdot\text{Prior}}{\text{Normalization}}$$

Definition 6.13 $\mathrm{p}(\mathbf{y}|\mathbf{X})/\mathrm{p}(\mathcal{D})$
Marginal Likelihood [see proof 10.2]:
is the normalization constant that makes sure that the posterior distribution[def. 6.12] is an true probability distribution:

$$\mathrm{p}(\mathbf{y}|\mathbf{X}) = \int \mathrm{p}(\mathbf{y}|\mathbf{w},\mathbf{X})\cdot \mathrm{p}_\lambda(\mathbf{w})\,\mathrm{d}\mathbf{w} = \int \text{Likelihood}\cdot\text{Prior}\,\mathrm{d}\mathbf{w} \tag{6.14}$$

Note
It is called marginal likelihood as we marginalize over $\mathbf{w}$.

Definition 6.14 Posterior Marginal Distribution: Is the posterior distribution of single elements of our thought after parameter vector:

$$\mathrm{p}(w_i|\mathbf{y},\mathbf{X}) = \int \mathrm{p}(\mathbf{y}|\mathbf{w},\mathbf{X})\,\mathrm{d}\mathbf{w}_{-i} \quad i = 1,\dots \dim(\mathbf{w}) \tag{6.15}$$

Definition 6.15 $\mathrm{p}(\mathbf{f}_*|\mathbf{x}_*,\mathbf{X},\mathbf{y})/\mathrm{p}(\mathbf{f}_*|\mathbf{y})$ [see proof 10.1]
Posterior Predictive Distribution:
is the distribution of a real process $\mathbf{f}$ (i.e. $f(x) = \mathbf{x}^\intercal\mathbf{w}$) given:
- new observation(s) $\mathbf{x}_*$
- the posterior distribution[def. 6.12] of the observed data $\mathcal{D} = \{\mathbf{X},\mathbf{y}\}$
- The likelihood of a real process $\mathbf{f}_*$

$$\mathrm{p}(\mathbf{f}_*|\mathbf{x}_*,\mathbf{X},\mathbf{y}) = \int \mathrm{p}(\mathbf{f}_*|\mathbf{x}_*,\mathbf{w})\cdot\mathrm{p}(\mathbf{w}|\mathbf{X},\mathbf{y})\,\mathrm{d}\mathbf{w} \tag{6.16}$$

it is calculated by weighting the likelihood[def. 6.1] of the new observation $\mathbf{x}_*$ with the posterior of the observed data and averaging over all parameter values $\mathbf{w}$.
⇒ obtain a distribution not depending on $\mathbf{w}$.

Note f vs. y
- Usually $\mathbf{f}$ denotes the model i.e.:
  $\mathbf{f}(\mathbf{x}) = \mathbf{x}^\intercal\mathbf{w}$ or $\mathbf{f}(\mathbf{x}) = \phi(\mathbf{x})^\intercal\mathbf{w}$
  and $\mathbf{y}$ the model plus the noise $\mathbf{y} = \mathbf{f}(\mathbf{x}) + \epsilon$.
- Sometime people also write only: $\mathrm{p}(\mathbf{y}_*|\mathbf{x}_*,\mathbf{X},\mathbf{y})$

5. Types of Uncertainty

Definition 6.16 Epistemic/Systematic Uncertainty:
Is the uncertainty that is due to things that one could in principle know but does not i.e. only having a finite sub sample of the data. The epistemic noise will decrease the more data we have.

Definition 6.17 Aleatoric/Statistical Uncertainty:
Is the uncertainty of an underlying random process/model. The aleatroic uncertainty stems from the fact that we are create random process models. If we run our *trained* model multiple times with *the same* input $\mathbf{X}$ data we will end up with different outcomes $\hat{y}$.
The aleatoric noise is *irreducible* as it is an underlying part of probabilistic models.

Bayesian Filtering

Definition 7.1
Recursive Bayesian Estimation/Filtering: Is a technique for estimating the an unknown probability distribution recursively over time by a measurement-[def. 7.3] and a process-model[def. 7.2] using Bayesian inference[def. 6.11].

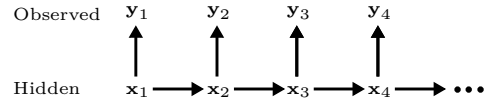


Figure 7: This problem corresponds to a *hidden Markov model (HMM)*[def. 14.1]

$$\mathbf{x}_t = \begin{pmatrix} x_{t,1} & \cdots & x_{t,n}\end{pmatrix} \qquad \mathbf{y}_t = \begin{pmatrix} y_{t,1} & \cdots & y_{t,m}\end{pmatrix}$$

Note
Comes from the idea that spam can be filtered out by the probability of certain words.

Definition 7.2 $\mathbf{x}_{t+1} \sim \mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-1})$
Process/Motion/Dynamic Model: is a model q of how our system state $\mathbf{x}_t$ evolves and is usually fraught with some uncertainty.

Corollary 7.1 Markov Property $\mathbf{x}_t \perp\!\!\!\perp \mathbf{x}_{1:t-2}|\mathbf{x}_{t-1}$**:** The process models[def. 7.2] is Markovian?? i.e. the current state depends only on the previous state:

$$\mathrm{p}(\mathbf{x}_t|\mathbf{x}_{1:t-1}) = \mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{7.1}$$

Definition 7.3 $\mathbf{y}_t \sim \mathrm{p}(\mathbf{y}_t|\mathbf{x}_t)$
Measurement/Sensor–Model/Likelihood: is a model h that maps observations/sensor measurements of our model $\mathbf{y}_t$ to the model state $\mathbf{x}_t$

Corollary 7.2 $\mathbf{y}_t \perp\!\!\!\perp \mathbf{y}_{1:t-1}\mathbf{x}_{1:t-1}|\mathbf{x}_t$
Conditional Independent Measurements: The measurements $\mathbf{y}_t$ are conditionally independent of the previous observations $\mathbf{y}_{1:t-1}$ given the current state $\mathbf{x}_t$:

$$\mathrm{p}(\mathbf{y}_t|\mathbf{y}_{1:t-1},\mathbf{x}_t) = \mathrm{p}(\mathbf{y}_t|\mathbf{x}_t) \tag{7.2}$$

Goal
We want to combine the process model[def. 7.2] and the measurement model[def. 7.3] in a recursive way to obtain a good estimate of our model state:

$$\left.\begin{matrix}\mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-1})\\ \mathrm{p}(\mathbf{y}_t|\mathbf{x}_t)\end{matrix}\right\} \mathrm{p}(\mathbf{x}_t|y_{1:t}) \xrightarrow[\text{recursion rule}]{\mathrm{p}(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})} \mathrm{p}(\mathbf{x}_{t+1}|y_{1:t+1})$$

Definition 7.4 Chapman-Kolmogorov eq. $\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1})$
Prior Update/Prediction Step [proof 10.3]:

$$\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int \mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-1})\mathrm{p}(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\,\mathrm{d}\mathbf{x}_{t-1} \tag{7.3}$$

Prior Distribution:

$$\mathrm{p}(\mathbf{x}_0|\mathbf{y}_{0-1}) = \mathrm{p}(\mathbf{x}_0) = \mathrm{p}_0 \tag{7.4}$$

Definition 7.5 $\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t})$
Posterior Distribution/Update Step [proof 10.4]:

$$\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{Z_t}\mathrm{p}(\mathbf{y}_t|\mathbf{x}_t)\,\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \tag{7.5}$$

Definition 7.6 Normalization [see proof 10.5]:

$$Z_t = \mathrm{p}(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \int \mathrm{p}(\mathbf{y}_t|\mathbf{x}_t)\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1})\,\mathrm{d}\mathbf{x}_t \tag{7.6}$$

Algorithm 7.1 Optimal Bayesian Filtering:
1: **Input:** $\mathrm{p}(\mathbf{x}_0)$
2: **while** Stopping Criterion not full-filed **do**
3: **Prediction Step:**

$$\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{Z_t}\mathrm{p}(\mathbf{y}_t|\mathbf{x}_t)\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1})$$

4: **Update Step:**

$$\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int \mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-1})\mathrm{p}(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\,\mathrm{d}\mathbf{x}_{t-1}$$

with:

$$Z_t = \int \mathrm{p}(\mathbf{y}_t|\mathbf{x}_t)\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1})\,\mathrm{d}\mathbf{x}_t$$

5: **end while**

Corollary 7.3 [proof 10.6]
Joint Probability Distribution of (HMM): we can also calculate the joint probability distribution of the (HMM):

$$\mathrm{p}(\mathbf{x}_{1:t},\mathbf{y}_{1:t}) = \mathrm{p}(\mathbf{x}_1)\mathrm{p}(\mathbf{y}_1|\mathbf{x}_1)\prod_{i=2}^{t}\mathrm{p}(\mathbf{x}_i|\mathbf{x}_{i-1})\mathrm{p}(\mathbf{y}_i|\mathbf{x}_i) \tag{7.7}$$

Example 7.1 Types of Bayesian Filtering:
- **Kalman Filter**: assumes a *linear* system, q, h are linear and Gaussian noise $\mathbf{v}$, $\mathbf{w}$.
- **Extended Kalman Filter**: assumes a *non-linear* system, q, h are non-linear and Gaussian noise $\mathbf{v}$, $\mathbf{w}$.
- **Particle Filter**: assumes a *non-linear* system q, h are non-linear and Non-Gaussian noise $\mathbf{v}$, $\mathbf{w}$, especially multi-modal distributions.

1. Kalman Filters

Definition 7.7 Kalman Filter Assumptions: Assumes a *linear*?? process model[def. 7.2], $\mathbf{q}$ with Gaussian model-noise $\mathbf{v}$ and a linear measurement model[def. 7.3] $\mathbf{h}$ with Gaussian process-noise $\mathbf{w}$.

add difference of prediction, filtering and smoothing for posterior marginals

Definition 7.8 Kalman Filter Model:
Process Model (7.8)

$$\boxed{\mathbf{x}^{(k)} = \mathbf{A}[k-1]\mathbf{x}^{(k-1)} + \mathbf{u}^{(k-1)} + \mathbf{v}[k-1]} \quad \text{with}$$

$$\mathbf{x}^{(0)} \sim \mathcal{N}(\mathbf{x}_0, P_0) \quad \text{and} \quad \mathbf{v}^{(k)} \sim \mathcal{N}(0, Q^{(k)})$$

Measuremnt Model (7.9)

$$\boxed{\mathbf{z}^{(k)} = \mathbf{H}^{(k)}\mathbf{x}^{(k)} + \mathbf{w}^{(k-1)}} \quad \text{with} \quad \mathbf{w}^{(k)} \sim \mathcal{N}(0, R^{(k)})$$

and define:

$$\hat{x}_p^{(k)} := \mathbb{E}[\mathbf{x}_p^{(k)}] \quad \text{and} \quad P_p^{(k)} := \mathbb{V}\left[\mathbf{x}_p^{(k)}\right] \tag{7.10}$$

$$\hat{x}_m^{(k)} := \mathbb{E}[\mathbf{x}_m^{(k)}] \quad \text{and} \quad P_m^{(k)} := \mathbb{V}\left[\mathbf{x}_m^{(k)}\right] \tag{7.11}$$

Note
The CRVs $\mathbf{x}_0$, $\{\mathbf{v}(\cdot)\}$, $\{\mathbf{w}(\cdot)\}$ are mutually independent.

add Kalman algorithm (in slides Joesph Form I think)

Gaussian Processes (GP)

1. Gaussian Process Regression

add complexitit On3 due to inverse

1.1. Gaussian Linear Regression

Given

(1) Linear Model with Gaussian Noise:

$$f(\mathbf{x}) = \mathbf{w}^\intercal \mathbf{x} \qquad \epsilon \sim \mathcal{N}\left(0, \sigma_n^2 \mathbf{I}\right) \tag{8.1}$$
$$\mathbf{y} = f(\mathbf{x}) + \epsilon$$

$$\Rightarrow \text{Gaussian Likelihood:} \quad \mathrm{p}\left(\mathbf{y}|\mathbf{X}, \mathbf{w}\right) = \mathcal{N}\left(\mathbf{X}\mathbf{w}, \sigma_n^2 \mathbf{I}\right)$$

(2) Gaussian Prior: $\mathrm{p}(\mathbf{w}) = \mathcal{N}\left(\mathbf{0}, \Sigma_p\right)$

Sought

(1) Posterior Distribution: $\mathrm{p}(\mathbf{w}|\mathbf{y}, \mathbf{X})$

(2) Posterior Predictive Distribution: $\mathrm{p}(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y})$

Definition 8.1 $\mathrm{p}(\mathbf{w}|\mathbf{y}, \mathbf{X}) = \mathcal{N}\left(\bar{\mathbf{w}}, \Sigma_{\mathbf{w}}^{-1}\right)$
Posterior Distribution proof 10.7:

$$\mu_{\mathbf{w}} = \frac{1}{\sigma_n^2} \Sigma_{\mathbf{w}}^{-1} \mathbf{X}\mathbf{y} \qquad \Sigma_{\mathbf{w}} = \frac{1}{\sigma_n^2} \mathbf{X}\mathbf{X}^\intercal + \Sigma_p^{-1}$$

Note

We could also use a prior with non-zero mean $\mathrm{p}(\mathbf{w}) = \mathcal{N}(\mu, \Sigma_p)$ but by convention w.o.l.g. we use zero mean see **??**.

Definition 8.2 $\mathrm{p}(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mu_*, \Sigma_*)$
Posterior Predictive Distribution proof 10.8:

$$\mu_* = \frac{1}{\sigma^2} \mathbf{x}_*^\intercal \Sigma_{\mathbf{w}}^{-1} \mathbf{X}\mathbf{y} \qquad \Sigma_* = \mathbf{x}_*^\intercal \Sigma_{\mathbf{w}}^{-1} \mathbf{x}_* \tag{8.2}$$

1.2. Kernelized Gaussian Linear Regression

Definition 8.3 Posterior Predictive Distribution:

$$\mathrm{p}(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mu_*, \Sigma_*) \tag{8.3}$$
$$\mu_* \tag{8.4}$$

Definition 8.4 Gaussian Process:

2. Model Selection

2.1. Marginal Likelihood

Approximate Inference

Problem

In statistical inference we often want to calculate integrals of probability distributions i.e.

- Expectations

$$\mathbb{E}_{X\sim \mathrm{p}}\left[g(X)\right] = \int g(x)\mathrm{p}(x)\,\mathrm{d}x$$

- Normalization constants:

$$\mathrm{p}(\theta|y) = \frac{1}{Z}\mathrm{p}(\theta, y) = \frac{\mathrm{p}(y|\theta)\mathrm{p}(\theta)}{Z} = \frac{\mathrm{p}(y|\theta)\mathrm{p}(\theta)}{\mathrm{p}(y)}$$

$$Z = \int \mathrm{p}(y|\theta)\mathrm{p}(\theta)\,\mathrm{d}\theta = \int \mathrm{p}(\theta) \prod_{i=1}^{n} \mathrm{p}(y_i|\mathbf{x}_i, \theta)\,\mathrm{d}\theta$$

For non-linear distributions this integrals are in general intractable which may be due to the fact that there exist no analytic form of the distribution we want to integrate or highly dimensional latent spaces that prohibits numerical integration (curse of dimensionality).

Definition 9.1 Approximate Inference: Is the procedure of finding an probability distribution q that approximates a true probability distribution p as well as possible.

1. Variational Inference

Definition 9.2 Bayes Variational Inference: Given an unormalized (posterior) probability distribution:

$$\mathrm{p}(\theta|y) = \frac{1}{Z}\mathrm{p}(\theta, y) \tag{9.1}$$

seeks an *approximate* probability distribution q_λ, that is parameterized by a *variational parameter* λ and approximates $\mathrm{p}(\theta|\mathbf{y})$ well.

Definition 9.3 Variational Family of Distributions Q: a set of probability distributions Q that is parameterized by the same *variational parameter* λ is called a variational familiy.

1.1. Laplace Approximation

Definition 9.4 [example 10.1], [proof 10.9,10.10,10.11]
Laplace Approximation: Tries to approximate a desired probability distribution $\mathrm{p}(\theta|\,\mathcal{D})$ by a Gaussian probability distribution:

$$Q = \{q_\lambda(\theta) = \mathcal{N}(\lambda)\} = \mathcal{N}(\mu, \Sigma)\} \tag{9.2}$$

the distribution is given by:

$$q(\theta) = c \cdot \mathcal{N}(\theta; \lambda_1, \lambda_2) \tag{9.3}$$

with

$$\lambda_1 = \hat{\theta} = \arg\max_\theta \mathrm{p}(\theta|y)$$
$$\lambda_2 = \Sigma = H^{-1}\left(\hat{\theta}\right) = -\nabla\nabla_\theta \log \mathrm{p}(\hat{\theta}|y)$$

Note

The name *Laplace Approximation* comes from its inventor *Pierre-Simon Laplace.*

Corollary 9.1 : Taylor approximation of a function $\mathrm{p}(\theta|y) \in \mathcal{C}^k$ around its mode $\hat{\theta}$ naturally induces a Gaussian approximation. See proofs 10.9,10.10,10.11

1.2. Black Box Stochastic Variational Inference

The most common way of finding q_λ is by minimizing the KL-divergence[def. 3.8] between our approximate distribution q and our true posterior p:

$$q^* \in \arg\min_{q\in Q} \mathrm{KL}\left(q(\theta) \parallel \mathrm{p}(\theta|y)\right) = \arg\min_{\lambda\in\mathbb{R}^d} \mathrm{KL}(q_\lambda(\theta) \parallel \mathrm{p}(\theta|y))$$

Note

Usually we want to minimize $\mathrm{KL}\left(\mathrm{p}(\theta|y) \parallel q(\theta)\right)$ but this is often infeasible s.t. we only minimize $\mathrm{KL}\left(q(\theta) \parallel \mathrm{p}(\theta|y)\right)$

Definition 9.5 ELBO-Optimization Problem [proof 10.12]:

$$q_\lambda^* \in \arg\min_{\{\lambda: q_\lambda \in Q\}} \mathrm{KL}\left(q_\lambda(\theta) \parallel \mathrm{p}(\theta|y)\right)$$
$$= \arg\max_{\{\lambda: q_\lambda \in Q\}} \mathbb{E}_{\theta\sim q_\lambda}\left[\log \mathrm{p}(y, \theta)\right] + H(q_\lambda) \tag{9.4}$$
$$= \arg\max_{\{\lambda: q_\lambda \in Q\}} \mathbb{E}_{\theta\sim q_\lambda}\left[\log \mathrm{p}(y|\theta)\right] - \mathrm{KL}(q_\lambda(\theta) \parallel \mathrm{p}(\theta)) \tag{9.5}$$
$$:= \arg\max_{\{\lambda: q_\lambda \in Q\}} \mathrm{ELBO}(\lambda) \tag{9.6}$$

Attention: Sometimes people write simply p for the posterior and $\mathrm{p}(\cdot)$ for prior.

Explanation 9.1.

- *eq.* (9.4)*:*
  - *prefer uncertain approximations i.e. we maximize* $H(q)$
  - *that jointly make the joint posterior likely*
- *eq.* (9.6)*: Expected likelihood of our posterior over* q *minus a regularization term that makes sure that we are not too far away from the prior.*

1.3. Expected Lower Bound of Evidence (ELBO)

Definition 9.6 example 10.2/proof 10.13
Expected Lower Bound of Evidence (ELBO):
The evidence lower bound is a bound on the log prior:

$$\mathrm{ELBO}\left(q_\lambda\right) \leqslant \log \mathrm{p}(y) \tag{9.7}$$

1.3.1. Maximizing The ELBO

Definition 9.7 Gradient of the ELBO Loss:

$$\nabla_\lambda L(\lambda) = \nabla_\lambda \mathrm{ELBO}(\lambda) \tag{9.8}$$
$$= \nabla_\lambda\left[\mathbb{E}_{\theta\sim q_\lambda}\left[\log \mathrm{p}(y, \theta)\right] + H(q_\lambda)\right]$$
$$= \nabla_\lambda\left[\mathbb{E}_{\theta\sim q_\lambda}\left[\log \mathrm{p}(y|\theta)\right] - \mathrm{KL}(q_\lambda(\theta) \parallel \mathrm{p}(\theta))\right]$$
$$= \nabla_\lambda \mathbb{E}_{\theta\sim q_\lambda}\left[\log \mathrm{p}(y|\theta)\right] - \nabla_\lambda \mathrm{KL}(q_\lambda(\theta) \parallel \mathrm{p}(\theta))$$

Problem

In order to use SGD we need to evaluate the gradient of the loss:

$$\nabla_\lambda \mathbb{E}\left[l(\theta; \mathbf{x})\right] = \mathbb{E}\left[\nabla_{\mathbf{x}\sim \mathrm{p}} l(\theta; \mathbf{x})\right] = \frac{1}{m}\sum_{i=1}^{m} \nabla_{\mathbf{x}\sim \mathrm{p}} l(\theta; \mathbf{x})$$

however in eq. (9.8) only second term can be derived easily. For the first term we cannot move the gradient inside the expectation as the expectations depends on the parameter w.r.t. which we differentiate:

$$\nabla_\lambda \mathbb{E}_{\theta\sim q_\lambda}\left[\log \mathrm{p}(y|\theta)\right] = \frac{\partial}{\partial\lambda}\int q_\lambda \log \mathrm{p}(y|\theta)\,\mathrm{d}\theta$$

Solutions:

- Score Gradients
- Reparameterization Trick: reparameterize a function s.t. it depends on another parameter and reformulate it s.t. it still returns the same value.

1.4. The Reparameterization Trick

Principle 9.1 proof 10.14
Reparameterization Trick: Let ϕ some base distribution from which we can sample and assume there exist an invertible function g s.t. $\theta = g(\epsilon, \lambda)$ then we can write θ in terms of a new distribution parameterized by $\epsilon \sim \phi(\epsilon)$:

$$\theta \sim q(\theta|\lambda) = \phi(\epsilon)\left|\nabla_\epsilon g(\epsilon; \lambda)\right|^{-1} \tag{9.9}$$

we can then write by the law of the unconscious statistician **??**:

$$\mathbb{E}_{\theta\sim q_\lambda}\left[\log \mathrm{p}(y|\theta)\right] = \mathbb{E}_{\epsilon\sim\phi}\left[\log \mathrm{p}(y|g(\epsilon; \lambda))\right] \tag{9.10}$$

$\Rightarrow$ the expectations does not longer depend on λ and we can pull in the gradient!

$$\nabla_\lambda \mathbb{E}_{\theta\sim q_\lambda}\left[\log \mathrm{p}(y|\theta)\right] = \nabla_\epsilon \mathbb{E}_{\theta\sim\phi}\left[\log \mathrm{p}(y|g(\epsilon; \lambda))\right] \tag{9.11}$$
$$= \mathbb{E}_{\epsilon\sim\phi}\left[\nabla_\lambda \log \mathrm{p}(y|g((\epsilon; \lambda)))\right] \tag{9.12}$$

Definition 9.8 example 10.3
Reparameterized ELBO Gradient[def. 9.7]:
By using the reparameterization trick principle 9.1 we can write the gradient of the ELBO as:

$$\nabla_\lambda L(\lambda) = \nabla_\lambda \mathrm{ELBO}(\lambda) \tag{9.13}$$
$$= \nabla_\lambda \mathbb{E}_{\theta\sim q_\lambda}\left[\log \mathrm{p}(y|\theta)\right] - \nabla_\lambda \mathrm{KL}(q_\lambda(\theta) \parallel \mathrm{p}(\theta))$$
$$= \mathbb{E}_{\epsilon\sim\phi}\left[\nabla_\lambda \log \mathrm{p}(y|g((\epsilon; \lambda)))\right] - \nabla_\lambda \mathrm{KL}(q_\lambda(\theta) \parallel \mathrm{p}(\theta))$$

Corollary 9.2 proof 10.3
Reparameterized ELBO for Gaussians:

$$\nabla_\lambda L(\lambda) = \nabla_\lambda \mathrm{ELBO}(\lambda) \tag{9.14}$$
$$= \nabla_\lambda \mathbb{E}_{\theta\sim q_\lambda}\left[\log \mathrm{p}(y|\theta)\right] - \nabla_\lambda \mathrm{KL}(q_\lambda(\theta) \parallel \mathrm{p}(\theta))$$
$$= \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[\nabla_{\mathbf{C},\mu} \log \mathrm{p}(y|\mathbf{C}\epsilon + \mu)\right] - \nabla_{\mathbf{C},\mu}\mathrm{KL}\left(q_{\mathbf{C},\mu} \parallel \mathrm{p}(\theta)\right)$$
$$\approx \frac{n}{m}\sum_{j=1}^{m} \nabla_{\mathbf{C},\mu} \log \mathrm{p}\left(y_{i_j}|\mathbf{C}\epsilon^j + \mu, \mathbf{x}_{i_j}\right) - \nabla_{\mathbf{C},\mu}\mathrm{KL}\left(q_{\mathbf{C},\mu} \parallel \mathrm{p}(\theta)\right)$$

2. Markov Chain Monte Carlos Methods

Definition 9.9
Markov Chain Monte Carlo (MCMC) Methods:

3. Integrated Nested Laplace Approximation

$$\eta_i = \alpha + \sum_{j=1}^{n_f} f^{(j)}\left(\mathbf{u}_{ji}\right) + \sum_{k=1}^{n_\beta} \beta_k z_{ki} + \epsilon_i \tag{9.15}$$

$$\mathrm{p}(\mathbf{x}, \theta)\mathrm{p}(\mathbf{y}) = \mathrm{p}(\mathbf{x}) \tag{9.16}$$

$$\mathrm{p}(\mathbf{x}_i|\mathbf{y}) = \int \mathrm{p}(x_i|\theta, \mathbf{y})\underline{\mathrm{p}(\theta|\mathbf{y})}\,\mathrm{d}\theta$$
$$\rightarrow \tilde{\mathrm{p}}(\mathbf{x}_i|\mathbf{y}) = \int \tilde{\mathrm{p}}(x_i|\theta, \mathbf{y})\tilde{\mathrm{p}}(\theta|\mathbf{y})\,\mathrm{d}\theta$$
$$\mathrm{p}(\theta_j|\mathbf{y}) = \int \mathrm{p}(\theta|\mathbf{y})\,\mathrm{d}\theta_{-j}$$
$$\rightarrow \tilde{\mathrm{p}}(\theta_j|\mathbf{y}) = \int \tilde{\mathrm{p}}(\theta|\mathbf{y})\,\mathrm{d}\theta_{-j}$$

$\mathrm{p}(x_i|\theta, \mathbf{y})$ and $\mathrm{p}(\theta|\mathbf{y})$ are approximated and the *posterior marginal* densities are then calculated using numerical integration:

Note

The numerical integration is possible if θ is small i.e. $m = \dim(\theta) \leqslant 5$.

4. Approximationing $\mathrm{p}(\theta|\mathbf{y})$ and $\mathrm{p}(x_i|\mathbf{y})$

$$\mathrm{p}(\mathbf{x}, \theta, \mathbf{y}) = \mathrm{p}(\mathbf{x}|\theta, \mathbf{y})\mathrm{p}(\theta, \mathbf{y}) = \mathrm{p}(\mathbf{x}|\theta, \mathbf{y})\mathrm{p}(\theta|\mathbf{y})\mathrm{p}(\mathbf{y})$$

$$\Rightarrow \quad \tilde{\mathrm{p}}(\theta|\mathbf{y}) = \frac{\mathrm{p}(\mathbf{x}, \theta, \mathbf{y})}{\tilde{\mathrm{p}}(\mathbf{x}|\theta, \mathbf{y})\mathrm{p}(\mathbf{y})} \propto \left.\frac{\mathrm{p}(\mathbf{x}, \theta, \mathbf{y})}{\tilde{\mathrm{p}}_G(\mathbf{x}|\theta, \mathbf{y})}\right|_{\mathbf{x}=\mathbf{x}^*(\theta)}$$

1. Marginal Posterior of the latent field $\mathrm{p}(\mathbf{x}_i|\mathbf{y})$ are calculated by first approximating $\underline{\mathrm{p}(\theta|\mathbf{y})}$:

$$\underline{\mathrm{p}(\theta|\mathbf{y})_G} = \mathcal{N}\left(x_i; \mu_i(\theta), \sigma_i^2(\theta)\right)$$

and then numerical integration w.r.t. θ:

$$\tilde{\mathrm{p}}(\mathbf{x}_i|\mathbf{y}) = \sum_k \underline{\mathrm{p}_G\left(\theta_k|\mathbf{y}\right)}\tilde{\mathrm{p}}\left(\theta_k|\mathbf{y}\right)\Delta_k$$

Note

$\tilde{\mathrm{p}}(\theta|\mathbf{y})$ is usually quiet different from a Gaussian s.t. the Gaussian approximation alone is not really sufficient.

Bayesian Neural Networks (BNN)

Definition 10.1 Bayesian Neural Networks (BNN):

(1) Model the prior over our weights $\theta = \left[\mathbf{W}^0 \cdots\cdots \mathbf{W}^L\right]$ by a neural network:

$$\theta \sim \mathrm{p}_\lambda(\theta) = \mathbf{F} \quad \text{with} \quad \begin{aligned} \mathbf{F} &= \mathbf{F}^L \circ \cdots \circ \mathbf{F}^1 \\ \mathbf{F}^l &= \varphi \circ \bar{\mathbf{F}}^l = \varphi\left(\mathbf{W}^l \mathbf{x} + b^l\right) \end{aligned}$$

for each weight $w_{k,j}^{(0)}$ of input x_j with weight on the hidden variable $z_k^{(0)}$ with $a_i^0 = \varphi\left\{\mathbf{z}_i^{(0)}\right\}$ it follows:

$$w_{k,j}^{(0)} = \mathrm{p}_w\left(\lambda_{k,j}\right) \overset{\text{i.e.}}{=} \mathcal{N}\left(\mu_{k,j}, \sigma_{k,j}^2\right)$$

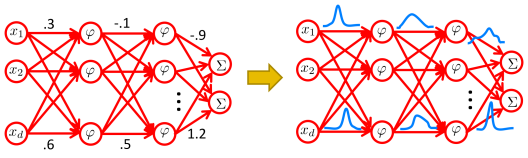


Figure 8

(2) The parameters of likelihood function are modeled by the output of the network:

$$\mathrm{p}(y|F(\theta, \mathbf{X})) \qquad \text{see example 10.4} \qquad (10.1)$$

Note

Recall for normal Bayesian Linear regression we had:

Problem

All the weights of the prior $\mathrm{p}_\lambda(\theta) = \mathbf{F}$ are correlated in some complex way see Figure 8. Thus even if the prior and likelihood are simple, the posterior will be not. $\Rightarrow$ need to approximate the posterior $\mathrm{p}(\theta|\mathbf{y}, \mathbf{X})$ i.e. by fitting a Gaussian distribution to each weight of the posterior neural network.

0.0.1. MAP estimates for BNN

Definition 10.2 BNN MAP Estimate: We need to do a forward pass for each $\mathbf{x}_i$ in order to obtain $\boldsymbol{\mu}(\mathbf{x}_i; \theta)$ and $\sigma\left(\mathbf{x}_i; \theta\right)^2$:

$$\theta^* = \arg\max_\theta \left\{\mathrm{p}(\theta|\mathbf{X}, \mathbf{y})\right\} \overset{\text{eq. (6.13)}}{=} \arg\min_\theta \boldsymbol{\lambda}\|\theta\|_2^2 - \sum_{i=1}^n \left(\frac{1}{2\sigma\left(\mathbf{x}_i; \theta\right)^2}\|y_i - \boldsymbol{\mu}(\mathbf{x}_i; \theta)\|^2 + \frac{1}{2}\log\sigma\left(\mathbf{x}_i; \theta\right)^2\right)$$

Explanation 10.1. *[def. 10.2]*

- $\frac{1}{2}\log\sigma\left(\mathbf{x}_i; \theta\right)^2$: *tries to force neural network to predict small uncertainty*
- $\frac{1}{2\sigma\left(\mathbf{x}_i; \theta\right)^2}\|y_i - \boldsymbol{\mu}(\mathbf{x}_i; \theta)\|^2$: *tries to force neural network to predict accurately* **but** *if this is not possible for certain data points the network can attenuate the loss to a larger variance.*

Definition 10.3 proof 10.15

MAP Gradient of BNN:

$$\theta_{t+1} = \theta_t\left(1 - 2\lambda\eta_t\right) - \eta_t \nabla \sum_{i=1}^n \log \mathrm{p}(y_i|\mathbf{x}_i, \theta) \qquad (10.2)$$

Note

- The gradients of the objective eq. (10.2) can be calculated using auto-differentiation techniques e.g. Pytorch or Tensorflow.
- The BNN MAP estimate fails to predict epistemic uncertainty[def. 6.16] $\iff$ it is overconfident in regions where we haven not even seen any data.
  $\Rightarrow$ need to use Bayesian approach to approximate posterior distribution.

0.1. Variational Inference For BNN

We use the objective eq. (9.14) as loss in order to perform back propagation.

0.2. Making Predictions

Proposition 10.1 Title:

1. Proofs

Proof 10.1: Definition 6.15:

$$\begin{aligned} \mathrm{p}(\mathbf{f}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) &= \frac{\mathrm{p}(\mathbf{f}_*, \mathbf{x}_*, \mathbf{X}, \mathbf{y})}{\mathrm{p}(\mathbf{x}_*, \mathbf{X}, \mathbf{y})} \\ &= \frac{\int \mathrm{p}(\mathbf{f}_*, \mathbf{x}_*, \mathbf{X}, \mathbf{y}, \mathrm{w})\,\mathrm{d}\mathrm{w}}{\mathrm{p}(\mathbf{x}_*, \mathbf{X}, \mathbf{y})} \\ &\overset{??}{=} \frac{\int \mathrm{p}(\mathbf{f}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}, \mathrm{w})\mathrm{p}(\mathbf{x}_*, \mathbf{X}, \mathbf{y}, \mathrm{w})\,\mathrm{d}\mathrm{w}}{\mathrm{p}(\mathbf{x}_*, \mathbf{X}, \mathbf{y})} \\ &\overset{??}{=} \frac{\int \mathrm{p}(\mathbf{f}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}, \mathrm{w})\mathrm{p}(\mathrm{w}|\mathbf{x}_*, \mathbf{X}, \mathbf{y})\cancel{\mathrm{p}(\mathbf{x}_*, \mathbf{X}, \mathbf{y})}\,\mathrm{d}\mathrm{w}}{\cancel{\mathrm{p}(\mathbf{x}_*, \mathbf{X}, \mathbf{y})}} \\ &= \int \mathrm{p}(\mathbf{f}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}, \mathrm{w})\mathrm{p}(\mathrm{w}|\mathbf{x}_*, \mathbf{X}, \mathbf{y})\,\mathrm{d}\mathrm{w} \\ &\overset{\clubsuit}{=} \int \mathrm{p}(\mathbf{f}_*|\mathbf{x}_*, \mathrm{w})\mathrm{p}(\mathrm{w}|\mathbf{X}, \mathbf{y})\,\mathrm{d}\mathrm{w} \end{aligned}$$

Note ♣

- $\mathbf{f}_*$ is independent of $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$ given the fixed parameter w.
- w does only depend on the observed data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$ and not the unseen data $\mathbf{x}_*$.

Proof 10.2: Definition 6.13:

$$\mathrm{p}(\mathbf{y}|\mathbf{X}) = \int \mathrm{p}(\mathbf{y}, \mathrm{w}|\mathbf{X})\,\mathrm{d}\mathrm{w} = \int \mathrm{p}(\mathbf{y}|\mathrm{w}, \mathbf{X})\mathrm{p}(\mathrm{w}|\mathbf{X})\,\mathrm{d}\mathrm{w} \overset{\text{eq. (6.6)}}{=} \int \mathrm{p}(\mathbf{y}|\mathrm{w}, \mathbf{X})\mathrm{p}(\mathrm{w})\,\mathrm{d}\mathrm{w}$$

Proof 10.3: Definition 7.4:

$$\mathrm{p}(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{y}_{1:t_1}) \overset{??}{=} \mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t_1})\mathrm{p}(\mathbf{x}_{t-1}|\mathbf{y}_{1:t_1}) \overset{\text{independ.}}{=} \mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-1})\mathrm{p}(\mathbf{x}_{t-1}|\mathbf{y}_{1:t_1})$$

marginalization/integration over $\mathbf{x}_{t-1}$ gives the desired result.

Proof 10.4: Definition 7.5:

$$\mathrm{p}(\mathbf{x}_t, \mathbf{y}_t|\mathbf{y}_{1:t-1}) \overset{??}{=} \begin{cases} \mathrm{p}(\mathbf{x}_t|\mathbf{y}_t, \mathbf{y}_{1:t-1})\mathrm{p}(\mathbf{y}_t|\mathbf{y}_{1:t-1}) \\ \mathrm{p}(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1})\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \end{cases}$$

$$\mathrm{p}(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1}) \overset{\text{[cor. 7.2]}}{=} \mathrm{p}(\mathbf{y}_t|\mathbf{x}_t)$$

from which follows immediately eq. (7.5).

Proof 10.5: Definition 7.6:

$$\begin{aligned} \mathrm{p}(\mathbf{y}_t|\mathbf{y}_{1:t-1}) &= \int \mathrm{p}(\mathbf{y}_t, \mathbf{x}_t|\mathbf{y}_{1:t-1})\,\mathrm{d}\mathbf{x}_t \\ &= \int \mathrm{p}(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1})\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1})\,\mathrm{d}\mathbf{x}_t \\ &\overset{\text{[cor. 7.2]}}{=} \int \mathrm{p}(\mathbf{y}_t|\mathbf{x}_t)\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1})\,\mathrm{d}\mathbf{x}_t \end{aligned}$$

Proof 10.6: [cor. 7.3]:

$$\begin{aligned} \mathrm{p}(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) &\overset{??}{=} \mathrm{p}(\mathbf{y}_{1:t}|\mathbf{x}_{1:t})\mathrm{p}(\mathbf{x}_{1:t}) \\ &\overset{??}{=} \mathrm{p}(\mathbf{y}_{1:t}|\mathbf{x}_{1:t})\mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-1:0}) \cdots \mathrm{p}(\mathbf{x}_2|\mathbf{x}_1)\mathrm{p}(\mathbf{x}_1) \\ &\overset{\text{eq. (7.1)}}{=} \mathrm{p}(\mathbf{y}_{1:t}|\mathbf{x}_{1:t})\left(\mathrm{p}(\mathbf{x}_1)\prod_{2=1}^t \mathrm{p}(\mathbf{x}_i|\mathbf{x}_{i-1})\right) \\ &\overset{\text{[cor. 7.2]}}{=} \left(\underline{\mathrm{p}(\mathbf{y}_1|\mathbf{x}_1)} \cdots \mathrm{p}(\mathbf{y}_t|\mathbf{x}_t)\right)\left(\mathrm{p}(\mathbf{x}_1)\prod_{2=1}^t \mathrm{p}(\mathbf{x}_i|\mathbf{x}_{i-1})\right) \\ &= \underline{\mathrm{p}(\mathbf{y}_1|\mathbf{x}_1)}\mathrm{p}(\mathbf{x}_1)\prod_{2=1}^t \mathrm{p}(\mathbf{y}_i|\mathbf{x}_i)\mathrm{p}(\mathbf{x}_i|\mathbf{x}_{i-1}) \end{aligned}$$

Proof 10.7: [def. 8.1]

$$\begin{aligned} &\mathrm{p}(\mathrm{w}|\,\mathcal{D}) \propto \mathrm{p}(\mathcal{D}\,|\mathrm{w})\mathrm{p}(\mathrm{w}) \\ &\propto \exp\left(-\frac{1}{2}\frac{1}{\sigma_n^2}(\mathbf{y} - \mathbf{X}\mathrm{w})^\intercal(\mathbf{y} - \mathbf{X}\mathrm{w})\right)\exp\left(-\frac{1}{2}\mathrm{w}^\intercal\Sigma^{-1}\mathrm{w}\right) \\ &\propto \exp\left\{-\frac{1}{2}\frac{1}{\sigma_n^2}\left(\mathbf{y}^\intercal\mathbf{y} - 2\mathrm{w}^\intercal\mathbf{X}^\intercal\mathbf{y} + \mathrm{w}^\intercal\mathbf{X}^\intercal\mathbf{X}\mathrm{w} + \sigma_n^2\mathrm{w}^\intercal\Sigma^{-1}\mathrm{w}\right)\right\} \\ &\propto \exp\left\{-\frac{1}{2}\frac{1}{\sigma_n^2}\left(\mathbf{y}^\intercal\mathbf{y} - \underline{2\mathrm{w}^\intercal\mathbf{X}^\intercal\mathbf{y}} + \mathrm{w}^\intercal\left(\mathbf{X}^\intercal\mathbf{X} + \sigma_n^2\Sigma^{-1}\right)\mathrm{w}\right)\right\} \end{aligned}$$

We know that a Gaussian $\mathcal{N}(\mathrm{w}|\bar{\mathrm{w}}, \Sigma_\mathrm{w}^{-1})$ should look like:

$$\begin{aligned} \mathrm{p}(\mathrm{w}|\,\mathcal{D}) &\propto \exp\left(-\frac{1}{2}(\mathrm{w} - \bar{\mathrm{w}})^\intercal\Sigma_\mathrm{w}(\mathrm{w} - \bar{\mathrm{w}})\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\underbrace{\mathrm{w}^\intercal\Sigma_\mathrm{w}\mathrm{w}}_{\cdots} - \underline{2\mathrm{w}^\intercal\Sigma_\mathrm{w}\bar{\mathrm{w}}} + \bar{\mathrm{w}}^\intercal\Sigma_\mathrm{w}\bar{\mathrm{w}}\right)\right) \end{aligned}$$

Σ_w follows directly $\Sigma_\mathrm{w} = \sigma_n^{-2}\mathbf{X}\mathbf{X}^\intercal + \Sigma_p$

$\bar{\mathrm{w}}$ follows from $\underline{2\mathrm{w}^\intercal\mathbf{X}^\intercal\mathbf{y}} = \underline{2\mathrm{w}^\intercal\Sigma_\mathrm{w}\bar{\mathrm{w}}} \Rightarrow \bar{\mathrm{w}} = \Sigma_\mathrm{w}^{-1}\mathbf{X}^\intercal\mathbf{y}$.

Proof 10.8: [def. 8.2]

Proof 10.9: [def. 9.4] In a Bayesian setting we are usually interested in maximizing the log prior+likelihood:

$$\mathscr{L}_n(\theta) = \log\left(\mathrm{p}(\theta|y)\right) = (\log\text{Prior} + \log\text{Likelihood})$$

we now approximate $\mathscr{L}_n(\theta))$ by a Taylor approximation around its maximum $\hat{\theta}$:

$$\mathscr{L}_n(\theta) = \mathscr{L}_n(\hat{\theta}) + \frac{1}{2}\frac{\partial^2\mathscr{L}_n}{\partial\theta^2}\Big|_{\hat{\theta}}(\theta - \hat{\theta}))^2 + \mathcal{O}\left((\theta - \hat{\theta})^3\right)$$

we can no derive the distribution:

$$\begin{aligned} \mathrm{p}(\theta|y) &\approx \exp(\mathscr{L}_n(\theta)) = \exp(\log\mathrm{p}(\theta|y)) \\ &= \mathrm{p}\left(\hat{\theta}\right)\exp\left(\frac{1}{2}\frac{\partial^2\mathscr{L}_n}{\partial\theta^2}\Big|_{\hat{\theta}}\right) \\ &= \sqrt{2\pi\sigma^2}\,\mathrm{p}\left(\hat{\theta}\right)\mathcal{N}\left(\theta; \hat{\theta}, \sigma\right) \approx \frac{1}{\sqrt{2\pi\sigma^2}}\mathcal{N}\left(\theta; \hat{\theta}, \sigma\right) \end{aligned}$$

Notes

- the derivative of the maximum must be zero by definition
$$\frac{\partial\mathscr{L}_n}{\partial\theta}\Big|_{\hat{\theta}} = 0$$
- we approximate the normalization constant $\frac{1}{Z}$ by $\sqrt{2\pi\sigma^2}\mathrm{p}(\hat{\theta})$.

Proof 10.10: [def. 9.4] 2D:

$$\nabla\mathscr{L}_n(\theta) = \nabla\mathscr{L}_n(\theta_1, \theta_2) = 0$$

$$\mathscr{L}_n(\theta) = \mathscr{L}_n\left(\hat{\theta}\right) + \frac{1}{2}\left(A(\theta_1 - \hat{\theta}_1)^2 + B(\theta_2 - \hat{\theta}_2)^2 + C(\theta_1 - \hat{\theta}_1)(\theta_2 - \hat{\theta}_2)\right)$$

$$\mathscr{L}_n(\theta) = \mathscr{L}_n\left(\hat{\theta}\right) + \left(\theta - \hat{\theta}\right)^\intercal H\left(\hat{\theta}\right)\left(\theta - \hat{\theta}\right) = \mathscr{L}_n\left(\hat{\theta}\right) + \frac{1}{2}Q(\theta)$$

$$A = \frac{\partial^2\mathscr{L}_n}{\partial\theta^2}\Big|_{\hat{\theta}} \qquad B = \frac{\partial^2\mathscr{L}_n}{\partial\theta^2}\Big|_{\hat{\theta}} \qquad C = \frac{\partial^2\mathscr{L}_n}{\partial\theta_1\partial\theta_2}\Big|_{\hat{\theta}}$$

$$H = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \qquad \Sigma = H^{-1}\left(\hat{\theta}\right)$$

Proof 10.11: [def. 9.4] k-dimensional:

$$\mathscr{L}_n(\theta) \approx \mathscr{L}_n\left(\hat{\theta}\right) + \left(\theta - \hat{\theta}\right)^\intercal\nabla\nabla^\intercal\mathscr{L}_n\left(\hat{\theta}\right)\left(\theta - \hat{\theta}\right)$$

$$H(\theta) = \nabla\nabla^\intercal\mathscr{L}_n(\theta) \qquad \Sigma = H^{-1}\left(\hat{\theta}\right)$$

$$\begin{aligned} \mathrm{p}(\theta|y) &= \sqrt{(2\pi)^n\det(\Sigma)}\,\mathrm{p}\left(\hat{\theta}\right)\mathcal{N}\left(\theta; \hat{\theta}, \Sigma\right) \\ &\approx c\frac{1}{\sqrt{(2\pi)^n\det(\Sigma)}}\mathcal{N}\left(\theta; \hat{\theta}, \Sigma\right) \end{aligned}$$

Proof 10.12: [def. 9.5]

$$q^* \in \arg\min_{q \in Q} \mathrm{KL}\left(q(\theta) \parallel p(\theta|y)\right)$$

$$\begin{aligned}
p(\theta|y) &= \tfrac{1}{Z} p(\theta, y) \\
&= \arg\min_q \mathbb{E}_{\theta\sim q}\left[\log \frac{q(\theta)}{\frac{1}{Z} p(\theta, y)}\right] \\
&= \arg\min_q \mathbb{E}_{\theta\sim q}\left[\log q(\theta) - \log \frac{1}{Z} - \log p(\theta, y)\right] \\
&= \arg\min_q \mathbb{E}_{\theta\sim q} - \underbrace{\left[-\log q(\theta)\right]}_{H(q)} + \cancel{\mathbb{E}_{\theta\sim q}\left[\log Z\right]} \\
&\quad - \mathbb{E}_{\theta\sim q}\left[\log p(\theta, y)\right] \\
&= \arg\max_q \mathbb{E}_{\theta\sim q}\left[\log p(\theta, y)\right] + H(q) \\
&= \arg\max_q \mathbb{E}_{\theta\sim q}\left[\log p(\theta|y) + \log p(\theta) - \log q(\theta)\right] \\
&= \arg\max_q \mathbb{E}_{\theta\sim q}\left[\log p(\theta|y)\right] + \mathrm{KL}\left(q(\theta) \parallel p(\theta)\right)
\end{aligned}$$

Proof 10.13: [def. 9.6]

$$\begin{aligned}
\log p(y) &= \log \int p(y, \theta)\, \mathrm{d}\theta = \log \int p(y|\theta) p(\theta)\, \mathrm{d}\theta \\
&= \log \int p(y|\theta) \frac{p(\theta)}{q_\lambda(\theta)} q_\lambda(\theta)\, \mathrm{d}\theta \\
&= \log \mathbb{E}_{\theta\sim q_\lambda}\left[p(y|\theta) \frac{p(\theta)}{q_\lambda(\theta)}\right] \\
\overset{??}{\geqslant} & \mathbb{E}_{\theta\sim q_\lambda}\left[\log\left(p(y|\theta) \frac{p(\theta)}{q_\lambda(\theta)}\right)\right] \\
&= \mathbb{E}_{\theta\sim q_\lambda}\left[\log p(y|\theta) - \log \frac{p(\theta)}{q_\lambda(\theta)}\right] \\
&= \mathbb{E}_{\theta\sim q_\lambda}\left[\log p(y|\theta)\right] - \mathrm{KL}\left(q_\lambda \parallel p(\cdot)\right)
\end{aligned}$$

Proof 10.14: principle 9.1 Let:

$$\begin{aligned}
&\epsilon \sim \phi(\epsilon) && X \sim f_X \\
&\theta = g(\epsilon; \lambda) \quad \text{correspond to} && \mathcal{Y} = \{y | y = g(x), \forall x \in \mathcal{X}\}
\end{aligned}$$

then it follows immediately with **??**:

$$\begin{aligned}
\theta \sim q_\lambda(\theta) = q(\theta|\lambda) &= \frac{f_X\left(g^{-1}(y)\right)}{\left|\frac{\mathrm{d}g}{\mathrm{d}x}\left(g^{-1}(y)\right)\right|} \\
&= \phi(\epsilon) |\nabla_\epsilon g(\epsilon; \lambda)|^{-1}
\end{aligned}$$

$\Rightarrow$ parameterized in terms of ϵ

Proof 10.15: [def. 10.3]

$$\begin{aligned}
\theta_{t+1} &= \theta_t - \eta_t \left(\nabla \log p(\theta) - \nabla \sum_{i=1}^n \log p(y_i|\mathbf{x}_i, \theta)\right) \\
&= \theta_t - \eta_t \left(2\lambda\theta_t - \nabla \sum_{i=1}^n \log p(y_i|\mathbf{x}_i, \theta)\right) \\
&= \theta_t \left(1 - 2\lambda\eta_t\right) - \eta_t \nabla \sum_{i=1}^n \log p(y_i|\mathbf{x}_i, \theta)
\end{aligned}$$

2. Examples

Example 10.1 Laplace Approximation
Logistic Regression Likelihood + Gaussian Prior:

Example 10.2 ELBO Bayesian Logistic Regression:
Suppose:

$$\begin{aligned}
&Q = \text{diag. Gaussians} \quad \Rightarrow \quad \lambda = \begin{bmatrix}\mu_{1:d} & \sigma^2_{1:d}\end{bmatrix} \in \mathbb{R}^{2d} \\
&p(\theta) = \mathcal{N}(0, \mathbf{I})
\end{aligned}$$

Then it follows for the terms of the ELBO:

$$\mathrm{KL}(q_\lambda \parallel p(\theta)) = \frac{1}{2} \sum_{i=1}^d \left(\mu_i^2 + \sigma_i^2 - 1 - \ln \sigma_i^2\right)$$

$$\begin{aligned}
\mathbb{E}_{\theta\sim q_\lambda}\left[p(y|\theta)\right] &= \mathbb{E}_{\theta\sim q_\lambda}\left[\sum_{i=1}^n \log p(y_i|\theta, \mathbf{x}_i)\right] \\
&= \mathbb{E}_{\theta\sim q_\lambda}\left[-\sum_{i=1}^n \log\left(1 + \exp\left(-y_i \theta^\intercal \mathbf{x}_i\right)\right)\right]
\end{aligned}$$

Example 10.3 ELBO Gradient Gaussian: Suppose:

$$\begin{aligned}
&\theta \sim q(\theta|\lambda) = \mathcal{N}(\theta; \boldsymbol{\mu}, \Sigma) \quad \Rightarrow \quad \lambda = \begin{bmatrix}\boldsymbol{\mu} & \Sigma\end{bmatrix} \\
&\boldsymbol{\epsilon} \sim \phi(\epsilon) = \mathcal{N}(\epsilon; \mathbf{0}, \mathbf{I})
\end{aligned}$$

we can reparameterize using principle 9.1 by using:

$$\theta \sim g(\boldsymbol{\epsilon}, \lambda) = \mathbf{C}\boldsymbol{\epsilon} + \boldsymbol{\mu} \quad \text{with} \quad \mathbf{C}: \quad \mathbf{C}\mathbf{C}^\intercal = \Sigma$$

from this it follows: ($\mathbf{C}$ is the Cholesky factor of Σ)

$$g^{-1}(\theta, \lambda) = \epsilon = \mathbf{C}^{-1}(\theta - \boldsymbol{\mu}) \qquad \frac{\partial g(\epsilon; \lambda)}{\partial \boldsymbol{\epsilon}} = C$$

from this it follows:

$$\begin{aligned}
q(\theta|\lambda) &= \frac{\phi(\epsilon)}{\left|\frac{\mathrm{d}g(\epsilon;\theta)}{\mathrm{d}\epsilon}(g^{-1}(\theta))\right|} = \phi(\epsilon)|C|^{-1} \\
&\iff \phi(\epsilon) = q(\theta|\lambda)|C|
\end{aligned}$$

we can then write the reparameterized expectation part of the gradient of the ELBO as:

$$\begin{aligned}
\nabla_\lambda L(\lambda)_1 &= \nabla_\lambda \mathbb{E}_{\epsilon\sim\phi}\left[\log p\left(y|g(\epsilon; \lambda)\right)\right] \\
&= \nabla_{\mathbf{C},\mu} \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[\log p(y|\mathbf{C}\boldsymbol{\epsilon} + \boldsymbol{\mu})\right] \\
&\overset{\text{i.i.d.}}{=} \nabla_{\mathbf{C},\mu} \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[\sum_{i=1}^n \log p(y_i|\mathbf{C}\boldsymbol{\epsilon} + \boldsymbol{\mu}, \mathbf{x}_i)\right] \\
&= \nabla_{\mathbf{C},\mu} \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[n \frac{1}{n} \sum_{i=1}^n \log p(y_i|\mathbf{C}\boldsymbol{\epsilon} + \boldsymbol{\mu}, \mathbf{x}_i)\right] \\
&= \nabla_{\mathbf{C},\mu} n \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[\mathbb{E}_{i\sim\mathcal{U}(\{1,n\})} \log p(y_i|\mathbf{C}\boldsymbol{\epsilon} + \boldsymbol{\mu}, \mathbf{x}_i)\right] \\
&\quad \text{Draw a mini batch } \begin{cases}\epsilon^{(1)}, \ldots, \epsilon^{(m)} \\ j_1, \ldots, j_m \sim \mathcal{U}(\{1, n\})\end{cases} \\
&= n \frac{1}{m} \sum_{j=1}^m \nabla_{\mathbf{C},\mu} \log p\left(y_j|\mathbf{C}\boldsymbol{\epsilon} + \boldsymbol{\mu}, \mathbf{x}_j\right)
\end{aligned}$$

$$\begin{aligned}
\nabla_\lambda L(\lambda) = \nabla_\lambda \mathrm{ELBO}(\lambda) &= \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[\nabla_{\mathbf{C},\mu} \log p(y|\mathbf{C}\boldsymbol{\epsilon} + \boldsymbol{\mu})\right] \\
&\quad - \nabla_{\mathbf{C},\mu}\left(q_{\mathbf{C},\boldsymbol{\mu}} \parallel p(\theta)\right)
\end{aligned}$$

Example 10.4 BNN Likelihood Function Examples:

$$p(y|\mathbf{X}, \theta) = \begin{cases}\mathcal{N}\left(y; \mathbf{F}(\mathbf{X}, \theta), \sigma^2\right) \\ \mathcal{N}\left(y; \mathbf{F}(\mathbf{X}, \theta)_1, \exp \mathbf{F}(\mathbf{X}, \theta)_1\right)\end{cases}$$

Kernels

Given objects we cannot assume that they are vectors/can be represented as vectors in feature space.
Hence it is also not guaranteed that those objects can be added and multiplied by scalars.
Question: then how can we define a more general notion of similarity?

Definition 11.1 Similarity Measure sim(A,B): A similarity measure or similarity function is a real-valued function that quantifies the similarity between two objects.
No single definition of a similarity measure exists but often they are defined in terms of the inverse of distance metrics and they take on large values for similar objects and either zero or a negative value for very dissimilar objects.

Definition 11.2 Dissimilarity Measure disssim(A,B): Is a measure of how dissimilar objects are, rather than how similar they are.
Thus it takes the largest values for objects that are really far apart from another.
Dissimilarities are often chosen as the sqaured norm of two difference vectors:

$$\|\mathbf{x}-\mathbf{y}\|^2 = \mathbf{x}^\intercal\mathbf{x} + \mathbf{y}^\intercal\mathbf{y} - 2\mathbf{x}^\intercal\mathbf{y} \qquad \forall \mathbf{x},\mathbf{y}\in\mathbb{R}^d \tag{11.1}$$
$$\text{dissim}(\mathbf{x},\mathbf{y}) = \text{sim}(\mathbf{x},\mathbf{x}) + \text{sim}(\mathbf{y},\mathbf{y}) - 2\text{dissim}(\mathbf{x},\mathbf{y})$$

Attention

It is better to rely on similarity measures instead of dissimilarity measures. Dissimilarities are often not adequat from a modeling point of view, because for objects that are really dissimilar/far from each other, we usually have the biggest problem to estimate their distance.
E.g. for a bag of words it is easy to determine similar words, but it is hard to estimate which words are most dissimilar. For normed vectors the only information of a dissimilarity defined as in eq. (11.1) becomes $2\mathbf{x}^\intercal\mathbf{y} = 2\text{dissim}(\mathbf{x},\mathbf{y})$

Definition 11.3 Feature Map ϕ: is a mapping $\phi:\mathcal{X}\mapsto\mathcal{V}$ that takes an input $\mathbf{x}\in\mathcal{X}\subseteq\mathbb{R}^d$ and maps it into another feature space $\mathcal{V}\subseteq\mathbb{R}^D$.

Note

Such feature maps can lead to an exponential number of terms i.e. for a polynomial feature map, with monorails of degree up to p and feature vectors of dimension $\mathbf{x}\in\mathbb{R}^d$ we obtain a feature space of size:

$$D = \dim(\mathcal{V}) = \binom{p+d}{d} = \mathcal{O}(d^p) \tag{11.2}$$

when using the polynomial kernel[def. 11.10], this can be reduced to the order d.

Definition 11.4 Kernel k: Let $\mathcal{X}\subseteq\mathbb{R}^d$ be the data space. A map $\mathbf{k}:\mathcal{X}\times\mathcal{X}\mapsto\mathbb{R}$ is called kernel if their exists an inner product space?? called **feature space** $(\mathcal{V},\langle\cdot,\cdot\rangle_\mathcal{V})$ and a map $\phi:\mathcal{X}\mapsto\mathcal{V}$ s.t.

$$\mathbf{k}(\mathbf{x},\mathbf{y}) = \langle\phi(\mathbf{x}),\phi(\mathbf{y})\rangle_\mathcal{V} \qquad \forall\mathbf{x},\mathbf{y}\in\mathcal{X} \tag{11.3}$$

Corollary 11.1 Kernels and similarity: Kernels are defined in terms of inner product spaces and hence the have a notion of similarity between its arguments.

Example

Let $\mathbf{k}(\mathbf{x},\mathbf{y}) := \mathbf{x}^\intercal\mathbf{A}\mathbf{y}$ **thus** the kernel measures the similarity between $\mathbf{x}$ and $\mathbf{y}$ by the inner product $\mathbf{x}^\intercal\mathbf{y}$ weighted by the matrix $\mathbf{A}$.

Corollary 11.2 Kernels and distance: Let $\mathbf{k}(\mathbf{x},\mathbf{y})$ be a measure of similarity between $\mathbf{x}$ and $\mathbf{y}$ then $\mathbf{k}$ induces a dissimilarity/distance between $\mathbf{x}$ and $\mathbf{y}$ defined as the difference betweend the self-similarities $\mathbf{k}(\mathbf{x},\mathbf{x})+\mathbf{k}(\mathbf{y},\mathbf{y})$ and the cross-similarities $\mathbf{k}(\mathbf{x},\mathbf{y})$:

$$\text{dissimilarity}(\mathbf{x},\mathbf{y}) := \mathbf{k}(\mathbf{x},\mathbf{x}) + \mathbf{k}(\mathbf{y},\mathbf{y}) - 2\,\mathbf{k}(\mathbf{x},\mathbf{y})$$

Note

The factor 2 is required to ensure that $d(\mathbf{x},\mathbf{x}) = 0$.

1. The Gram Matrix

Definition 11.5 Kernel (Gram) Matrix:
Given: a mapping $\phi:\mathbb{R}^d\mapsto\mathbb{R}^D$ and a corresponding kernel function $\mathbf{k}:\mathcal{X}\times\mathcal{X}\mapsto\mathbb{R}$ with $\mathcal{X}\subseteq\mathbb{R}^d$.
Let S be any finite subset of data $S=\{\mathbf{x}_1,\dots,\mathbf{x}_n\}\subseteq\mathcal{X}$.
Then the kernel matrix $\mathcal{K}:\in\mathbb{R}^{n\times n}$ is defined by:

$$\mathcal{K} = \phi(\mathbf{X})\phi(\mathbf{X}^\intercal) = (\phi(\mathbf{x}_1),\dots,\phi(\mathbf{x}_n))(\phi(\mathbf{x}_1),\dots,\phi(\mathbf{x}_n))^\intercal$$
$$= \begin{pmatrix}\mathbf{k}(\mathbf{x}_1,\mathbf{x}_1) & \cdots & \mathbf{k}(\mathbf{x}_1,\mathbf{x}_n)\\ \vdots & \ddots & \vdots \\ \mathbf{k}(\mathbf{x}_n,\mathbf{x}_1) & \cdots & \mathbf{k}(\mathbf{x}_n,\mathbf{x}_n)\end{pmatrix} = \begin{pmatrix}\phi(\mathbf{x}_1)^\intercal\phi(\mathbf{x}_1) & \cdots & \phi(\mathbf{x}_1)^\intercal\phi(\mathbf{x}_n)\\ \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_n)^\intercal\phi(\mathbf{x}_1) & \cdots & \phi(\mathbf{x}_n)^\intercal\phi(\mathbf{x}_n)\end{pmatrix}$$
$$\mathcal{K}_{ij} = \mathbf{k}(\mathbf{x}_i,\mathbf{x}_j) = \phi(\mathbf{x}_i)^\intercal\phi(\mathbf{x}_j)$$

Corollary 11.3 $\mathbf{V}\Lambda\mathbf{V}^\intercal$
Kernel Eigenvector Decomposition:
For any symmetric matrix (Gram matrix $\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j)|_{i,j=1}^n$) there exists an eigenvector decomposition:

$$\mathcal{K} = \mathbf{V}\Lambda\mathbf{V}^\intercal \tag{11.4}$$

$\mathbf{V}$: orthogonal matrix of eigenvectors $(\mathbf{v}_{t,i})|_{i=1}^n$
Λ : diagonal matrix of eigenvalues λ_i

Assuming all eigenvalues λ_t are non-negative, we can calculate the mapping:

$$\phi:\mathbf{x}_i\mapsto\left(\sqrt{\lambda_t}\mathbf{v}_{t,i}\right)_{t=1}^n\in\mathbb{R}^n, \qquad i=1,\dots,n \tag{11.5}$$

which allows us to define the Kernel $\mathcal{K}$ as:

$$\phi^\intercal(\mathbf{x}_i)\phi(\mathbf{x}_j) = \sum_{t=1}^n\lambda_t\mathbf{v}_{t,i}\mathbf{v}_{t,j} = (\mathbf{V}\Lambda\mathbf{V}^\intercal)_{i,j} = \mathcal{K}(\mathbf{x}_i,\mathbf{x}_j) \tag{11.6}$$

1.1. Necessary Properties

Property 11.1 Inner Product Space:
$\mathbf{k}$ must be an *inner product* of a suitable space $\mathcal{V}$.

Property 11.2 Symmetry: $\mathbf{k}/\mathcal{K}$ must be symmetric:
$\mathbf{k}(\mathbf{x},\mathbf{y}) = \mathbf{k}(\mathbf{y},\mathbf{x}) = \phi(\mathbf{x})^\intercal\phi(\mathbf{y}) = \phi(\mathbf{y})^\intercal\phi(\mathbf{x}) \qquad \forall\mathbf{x},\mathbf{y}\in\mathcal{X}$

Property 11.3 Non-negative Eigenvalues/p.s.d.s Form:
Let $S=\{\mathbf{x}_1,\dots,\mathbf{x}_n\}$ be an n-set of a *finite* input space $\mathcal{V}$.
A kernel $\mathbf{k}$ must induces a *p.s.d. symmetric* kernel matrix $\mathbf{k}$ for any possible $S\subseteq\mathcal{X}$ see ?? 11.1.
$\iff$ all eigenvalues of the kernel gram matrix $\mathcal{K}$ for *finite* $\mathcal{V}$ must be non-negative ??.

Notes

- The extension to infinite dimensional Hilbert Spaces might also include a non-negative weighting/eigenvalues:
$$\langle\phi(\mathbf{x}),\phi(\mathbf{z})\rangle = \sum_{i=1}^\infty\lambda_i\phi_i(\mathbf{x})\phi_i(\mathbf{z})$$
- In order to be able to use a kernel, we need to verify that the kernel is p.s.d. for all n-vectors $\mathcal{X}=\{\mathbf{x}_1,\dots,\mathbf{x}_n\}$, as well as for future unseen values.

2. Mercers Theorem

Theorem 11.1 Mercers Theorem: Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^n$ and $\mathbf{k}:\mathcal{X}\times\mathcal{X}\mapsto\mathbb{R}$ a kernel function.
Then one can expand $\mathbf{k}$ in a uniformly convergent series of bounded functions ϕ s.t.

$$\mathbf{k}(\mathbf{x},\mathbf{x}') = \sum_{i=1}^\infty\lambda\phi(\mathbf{x})\phi(\mathbf{x}') \tag{11.7}$$

Theorem 11.2 General Mercers Theorem: Let Ω be a compact subset of R^n. Suppose $\mathbf{k}$ is a gernal continuous symmetric function such that the integral operator:

$$T_\mathbf{k}:L_2(\mathbf{X})\mapsto L_2(\mathbf{X}) \qquad (T_\mathbf{k}f)(\cdot) = \int_\Omega\mathbf{k}(\cdot,\mathbf{x})f(\mathbf{x})\,\mathrm{d}\mathbf{x} \tag{11.8}$$

is positve, that is it satisfies:

$$\int_{\Omega\times\Omega}\mathbf{k}(\mathbf{x},\mathbf{z})f(\mathbf{x})f(\mathbf{z})\,\mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{z} > 0 \qquad \forall f\in L_2(\Omega)$$

Then we can expand $\mathbf{k}(\mathbf{x},\mathbf{z})$ in a uniformly convergent series in terms of $T_\mathcal{K}$'s eigen-functions $\phi_j\in L_2(\Omega)$, with $\|\phi_j\|_{L_2}=1$ and positive associated eigenvalues $\lambda_j>0$.

Note

All kernels satisfying mercers condtions describe an inner product in a high dimensional space.
$\Rightarrow$ can replace the inner product by the kernel function.

Check if $\mathbb{R}$ or $\mathbb{R}^n$ as in script

3. The Kernel Trick

Definition 11.6 Kernel Trick: If a kernel has an analytic form we do no longer need to calculate:
- the function mapping $\mathbf{x}\mapsto\phi(\mathbf{x})$ and
- the inner product $\phi(\mathbf{x})^\intercal\phi(\mathbf{y})$

explicitly but simply us the formula for the kernel:

$$\phi(\mathbf{x})^\intercal\phi(\mathbf{x}) = \mathbf{k}(\mathbf{x},\mathbf{y}) \tag{11.9}$$

see examples 11.1 and 11.2

Note

- Possible to operate in any n-dimenional function space, efficiently.
- ϕ not necessary anymore.
- Complexity independent of the functions space.

4. Types of Kernels

4.1. Stationary Kernels

Definition 11.7 Stationary Kernel: A stationary kernel is a kernel that only considers vector differences:

$$\mathbf{k}(\mathbf{x},\mathbf{y}) = \mathbf{k}(\mathbf{x}-\mathbf{y}) \tag{11.10}$$

see example example 11.3

4.2. Isotropic Kernels

Definition 11.8 Isotropic Kernel: A isotropic kernel is a kernel that only considers distance differences:

$$\mathbf{k}(\mathbf{x},\mathbf{y}) = \mathbf{k}(\|\mathbf{x}-\mathbf{y}\|_2) \tag{11.11}$$

Corollary 11.4 :
Isotropic $\rightarrow$ Stationary

5. Important Kernels on $\mathbb{R}^d$

5.1. The Linear Kernel

Definition 11.9 Linear/String Kernel:

$$\mathbf{k}(\mathbf{x},\mathbf{y}) = \mathbf{x}^\intercal\mathbf{y} \tag{11.12}$$

5.2. The Polynomial Kernel

Definition 11.10 Polynomial Kernel:
represents all monomials?? of degree up to m

$$\mathbf{k}(\mathbf{x},\mathbf{y}) = (1+\mathbf{x}^\intercal\mathbf{y})^m \tag{11.13}$$

5.3. The Sigmoid Kernel

Definition 11.11 Sigmoid/tanh Kernel:

$$\mathbf{k}(\mathbf{x},\mathbf{y}) = \tanh\kappa\mathbf{x}^\intercal\mathbf{y} - b \tag{11.14}$$

5.4. The Exponential Kernel

Definition 11.12 Exponential Kernel:
is an continuous kernel that is non-differential $\mathbf{k}\in\mathcal{C}^0$:

$$\mathbf{k}(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|_1}{\theta}\right) \tag{11.15}$$

$\theta\in\mathbb{R}$: corresponds to a threshold.

5.5. The Gaussian Kernel

Definition 11.13 Gaussian/Squared Exp. Kernel/ Radial Basis Functions (RBF):
Is an inifite dimensional smooth kernel $\mathbf{k}\in\mathcal{C}^\infty$ with some usefull properties

$$\mathbf{k}(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\theta^2}\right)\approx\begin{cases}1 & \text{if } \mathbf{x} \text{ and } \mathbf{y} \text{ close}\\ 0 & \text{if } \mathbf{x} \text{ and } \mathbf{y} \text{ far away}\end{cases} \tag{11.16}$$

Explanation 11.1 (Threshold θ). $2\theta\in\mathbb{R}$ *corresponds to a threshold that determines how close input values need to be in order to be considered similar:*

$$\mathbf{k} = \exp\left(-\frac{dist^2}{2\theta^2}\right)\approx\begin{cases}1 \iff sim & \text{if } dist \ll\theta\\ 0 \iff dissim & \text{if } dist\gg\theta\end{cases}$$

or in other words how much we believe in our data i.e. for smaller length scale we do trust our data less and the admitable functions vary much more.

Note

If we chose h small, all data points not close to h will be 0/discared $\iff$ data points are considered as independent.
Length of all vectors in feature space is one $\mathbf{k}(\mathbf{x},\mathbf{x}) = e^0 = 1$.
Thus: Data points in input space are projected onto a high-(infintie-)dimensional sphere in feature space.
Classification: Cutting with hyperplanes through the sphere. **How to chose** h: good heuristics, take median of the distance all points but better is cross validation.

5.6. The Matern Kernel

When looking at actual data/sample paths the smoothness of the Gaussian kernel[def. 11.13] is often a too strong assumption that does not model reality the same holds true for the non-smoothness of the exponential kernel[def. 11.12]. A solution to this dilemma is the Matern kernel.

Definition 11.14 Matern Kernel: is a kernel which allows you to specify the level of smoothness $\mathbf{k}\in\mathcal{C}^{\lceil\nu\rceil}$ by a positive parameter ν:

$$\mathbf{k}(x,y) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\|\mathbf{x}-\mathbf{y}\|_2}{\rho}\right)^\nu\mathcal{K}_\nu\left(\frac{\sqrt{2\nu}\|\mathbf{x}-\mathbf{y}\|_2}{\rho}\right)$$

$\nu,\rho\in\mathbb{R}_+$ ν : Smoothness ρ : Le (11.17)

$\mathcal{K}_\nu$ modified Bessel function of the second kind

6. Kernel Engineering

Often linear and even non-linear simple kernels are not sufficient to solve certain problems, especially for pairwise problems i.e. user & product, exon & intron,....
Composite kernels can be the solution to such problems.

6.1. Closure Properties/Composite Rules

Suppose we have two kernels:

$$\mathbf{k}_1:\mathcal{X}\times\mathcal{X}\mapsto\mathbb{R} \qquad \mathbf{k}_2:\mathcal{X}\times\mathcal{X}\mapsto\mathbb{R}$$

defined on the data space $\mathcal{X}\subseteq\mathbb{R}^d$. Then we may define using Composite Rules:

$$\mathbf{k}(\mathbf{x},\mathbf{x}') = \mathbf{k}_1(\mathbf{x},\mathbf{x}') + \mathbf{k}_2(\mathbf{x},\mathbf{x}') \tag{11.18}$$
$$\mathbf{k}(\mathbf{x},\mathbf{x}') = \mathbf{k}_1(\mathbf{x},\mathbf{x}')\cdot\mathbf{k}_2(\mathbf{x},\mathbf{x}') \tag{11.19}$$
$$\mathbf{k}(\mathbf{x},\mathbf{x}') = \alpha\,\mathbf{k}_1(\mathbf{x},\mathbf{x}') \qquad \alpha\in\mathbb{R}_+ \tag{11.20}$$
$$\mathbf{k}(\mathbf{x},\mathbf{x}') = f(\mathbf{x})f(\mathbf{x}') \tag{11.21}$$
$$\mathbf{k}(\mathbf{x},\mathbf{x}') = \mathbf{k}_3(\phi(\mathbf{x}),\phi(\mathbf{x}')) \tag{11.22}$$
$$\mathbf{k}(\mathbf{x},\mathbf{x}') = p\left(\mathbf{k}(\mathbf{x},\mathbf{x}')\right) \tag{11.23}$$
$$\mathbf{k}(\mathbf{x},\mathbf{x}') = \exp\left(\mathbf{k}(\mathbf{x},\mathbf{x}')\right) \tag{11.24}$$

Where	$f:\mathcal{X}\mapsto\mathbb{R}$	a real valued function
	$\phi:\mathcal{X}\mapsto\mathbb{R}^e$	the explicit mapping
	p	a polynomial with pos. coefficients
	$\mathbf{k}_3$	a Kernel over $\mathbb{R}^e\times\mathbb{R}^e$

Proofs

Proof 11.1: Property 11.3**The kernel matrix is positive-semidefinite**:
Let $\phi:\mathcal{X}\mapsto\mathbb{R}^d$ and $\Phi = [\phi(\mathbf{x}_1)\quad\dots\quad\phi(\mathbf{x}_n)]^\intercal\in\mathbb{R}^{d\times n}$.
Thus: $\mathcal{K} = \Phi^\intercal\Phi\in\mathbb{R}^{n\times n}$.

$$\mathbf{v}^\intercal\mathcal{K}\mathbf{v} = \mathbf{v}^\intercal\Phi^\intercal\Phi\mathbf{v} = (\Phi\mathbf{v})^T\Phi\mathbf{v} = \|\Phi\mathbf{v}\|_2^2 \geqslant 0$$

Examples

Example 11.1 Calculating the Kernel by hand:

Let : $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ $\quad \phi(\mathbf{x}) \mapsto \{x_1^2, x_2^2, \sqrt{2}x_1, x_2\}$

$\phi : \mathbb{R}^{d=2} \mapsto \mathbb{R}^{D=3}$

We can now have a decision boundary in this 3-D feature space $\mathcal{V}$ of ϕ as:

$$\beta_0 + \beta_1 x_1^2 + \beta_2 x_2^2 + \beta_3 \sqrt{2} x_1 x_2 = 0$$

$$\begin{aligned} &\left\langle \phi(\mathbf{x}^{(i)}), \phi(\mathbf{x}^{(j)}) \right\rangle \\ &= \left\langle \left\{ x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}, x_{i2} \right\}, \left\{ x_{j1}^2, x_{j2}^2, \sqrt{2}x_{j1}, x_{j2} \right\} \right\rangle \\ &= x_{i1}^2 x_{j1}^2 + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{i2}x_{j1}x_{j2} \end{aligned}$$

Operation Count:

- $2 \cdot 3$ operations to map $\mathbf{x}_i$ and $\mathbf{x}_j$ into the 3D space $\mathcal{V}$.
- Calculating an inner product of $\langle \phi(\mathbf{x}^{(i)}), \phi(\mathbf{x}^{(j)}) \rangle$ with 3 additional operations.

Example 11.2

Calculating the Kernel using the Kernel Trick:

$$\begin{aligned} \left\langle \phi(\mathbf{x}^{(i)}), \phi(\mathbf{x}^{(j)}) \right\rangle &= \underbrace{\langle \mathbf{x}_i, \mathbf{x}_j \rangle^2}_{:=\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)} = \langle \{x_{i1}, x_{i2}\}, \{x_{i1}, x_{i2}\} \rangle^2 \\ &= \left(x_{i1}x_{i2} + x_{j1}x_{j2} \right)^2 \\ &= x_{i1}^2 x_{j1}^2 + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{i2}x_{j1}x_{j2} \end{aligned}$$

Operation Count:

- 2 multiplicaitons of $\mathbf{x}_{i1}\mathbf{x}_{j1}$ and $\mathbf{x}_{i2}\mathbf{x}_{j2}$.
- 1 operation for taking the square of a scalar.

Conclusion The Kernel trick needed only 3 in comparison to 9 operations.

Example 11.3 Stationary Kernels:

$$\mathbf{k}(\mathbf{x}, \mathbf{y}) = \exp\left(\frac{(\mathbf{x} - \mathbf{y})^\intercal \mathbf{M} (\mathbf{x} - \mathbf{y})}{h^2} \right)$$

is a stationary but not an isotropic kernel.

Time Series

State Space Models

Definition 12.1 State Variables $\mathbf{x}$**:**
Is the smallest set of variables $\{x_1, \dots, x_n\}$ that are fully capable of describing the state of our system which is usually *hidden* and not directly observable.

Definition 12.2 State Space $\mathcal{X}$**:**
Is the n-dimensional space spanned by the state variables??:

$$\mathbf{x} = [x_1 \cdots\cdots x_n]^\intercal \in \mathcal{S} \subseteq \mathbb{R}^n \tag{12.1}$$

Definition 12.3 Input/Control Variables $\mathbf{u} \in \mathcal{A}$**:**
Are a variables $\mathbf{u}$ of the *transition model*[def. 12.5] that influence the propagation of to the state variables $\mathbf{x}$.

Definition 12.4 $\mathbf{y} \in \mathcal{O}$
Output/Measurment Variables/State Observations:
Are a variables $\mathbf{y}$ that are directly related to the state space $\mathbf{x}$ and are usually observable by us.

Definition 12.5 Transition Model f**:**
Describes the transition of the state $\mathbf{x}$ over time.

Definition 12.6 Measurment/Output/Observation Model h**:**
Describes the mapping of the state $\mathbf{x}$ onto the output $\mathbf{y}$.

Definition 12.7 (Discrete) State Space Model:

$$\mathbf{x}^{k+1} = f(t, \mathbf{x}^k, \mathbf{u}^k) \qquad t = 1, \dots, K \tag{12.2}$$

$$\mathbf{y}^k = h(t, \mathbf{x}^k, \mathbf{u}^k) \tag{12.3}$$

Markov Models

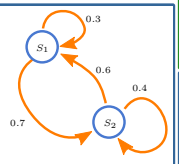
Definition 13.1 States $\mathcal{S} = \{s_1, \dots, s_n\}$**:**
A state s_i encodes all information of the current configuration of a system.

Definition 13.2 Markovian Property/Memorylessness:
Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space with a filtration $(\mathcal{F}_s, s \in I)$, for some index set??; and let $(S, \mathcal{S})$ be a measurable space??.
A $(S, \mathcal{S})$)-valued stochastic process $X = \{X_t : \Omega \to S\}_{t \in I}$ adapted to the filtration is said to possess the Markov property if:

$$\mathbb{P}(X_t \in A | \mathcal{F}_s) = \mathbb{P}(X_t \in A | X_s) \qquad \begin{matrix} \forall A \in \mathcal{S} \\ s, t \in I \quad \text{s.t. } s < t \end{matrix} \tag{13.1}$$

1. Markov Chains

Definition 13.3 Markov Chain:
Is a sequence of random variables $\{X_i\}_{i \in \mathcal{T}}$?? that processes the markovian property[def. 13.2] i.e. each state X_t depend only on the previous state X_{t-1}:



$$\mathbb{P}(X_t = x | X_{t-1} = x_{t-1}, \dots, X_1 = x_1) = \mathbb{P}(X_t = x | X_{t-1} = x_{t-1})$$

Definition 13.4 Initial Distribution $\mathbf{q}_0$**:** Describes the initial distribution of states:

$$\begin{aligned} & q_0(s_i) = \mathbb{P}(X_0 = s_i) && \forall s_i \in S \\ \iff \quad & \mathbf{q}_0 = [q_0(s_1) \quad \cdots \quad q_0(s_n)] && \end{aligned} \tag{13.2}$$

Definition 13.5 Transition Probability $p_{ji}(t)$**:**
is the probability of a random variable X_t in state s_i to transition into state s_j:

$$p_{ij}(t) = \mathbb{P}(X_{t+1} = s_j | X_t = s_i) \qquad \forall s_i, s_j \in S \tag{13.3}$$

Definition 13.6 n^{th} **Transition Probability** $p_{ji}^{(n)}(t)$**:**
denotes the probability of reaching state s_j from state s_i in n steps:

$$p_{ij}^{(n)}(t) = \mathbb{P}(X_{t+n} = s_j | X_t = s_i) \qquad \forall s_i, s_j \in S \tag{13.4}$$

Definition 13.7 Transition Matrix $P(t)$**:**
The transition probabilities eq. (13.4) can be represented by a *row-stochastic matrix*?? $P(t)$ where the i^{th} row represents the transition probabilities for the i^{th} state s_i i.e.

From \ To j		
$\vdots$		
	0.3	0.7
	0.4	0.6

Corollary 13.1 Row stochastic matrices and Graphs:
Row stochastic matrices?? represent graphs where the outgoing edges must sum to one:

$$\sum \delta^+(s_i) = 1 \tag{13.5}$$

1.1. Simulating Markov Chains

Corollary 13.2 proof 13.1
Realization of a Markov Chain:

$$\mathbb{P}(X_0 = x_0, \dots, X_N = x_N) = q_0(x_1) \sum_{n=1}^{N} p_{n-1,n}(t)$$

Algorithm 13.1 Forward Sampling:
Input: $\mathbf{q}(\mathbf{x}_0)$ and P
Output: $\mathbb{P}(X_{0:N})$
Sample $x_0 \sim \mathbb{P}(X_0)$
for $j = 1, \dots, n$ **do**
$x_j \sim \mathbb{P}(X_j | X_{j-1} = x_{j-1})$
end for

1.2. State Distributions

Definition 13.8
Probability Distribution of the States $\mathbf{q}_{n+1}$**:**

$$\begin{aligned} q_{n+1}(s_j) &= \mathbb{P}(X_{n+1} = s_j) && \forall s_i \in S \\ &= \sum_{i=1}^{n} \mathbb{P}(X_n = s_i) \mathbb{P}(X_{n+1} = s_j | X_n = s_i) \\ &= \sum_{i=1}^{n} q_n(s_i) p_{i,j}(t) \end{aligned} \tag{13.6}$$

$$\begin{aligned} \mathbf{q}_{n+1} &= [q_{n+1}(s_1) \quad \cdots \quad q_{n+1}(s_n)] \\ &= \mathbf{q}_n P(t) \\ &= [q_n(s_1) \quad \cdots \quad q_n(s_n)] \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{bmatrix}(t) \end{aligned}$$

Corollary 13.3 [proof 13.2]
Time-homogeneous Markov Transition Probabilities:

$$\mathbf{q}_{n+1} = \mathbf{q}_0 P^{n+1} \tag{13.7}$$

Definition 13.9 Stationary Distribution:
A markov chain has a stationary distribution if it satisfies:

$$\lim_{N \to \infty} q_N(s_i) = \lim_{N \to \infty} \mathbb{P}(X_N = s_i) = \pi_i \qquad \forall s_i \in S$$

$$\lim_{N \to \infty} \mathbf{q}_N = [\pi_1 \quad \cdots \quad \pi_n] \iff \mathbf{q} = \mathbf{q}P(N) \tag{13.8}$$

Corollary 13.4 Existence of Stationary Distributions:
A Markov Chain has a stationary distribution if and only if at least one state is *positive recurrent*!

add matrix version with eigenvector

add recurrent and transient states

1.3. Properties of States

Definition 13.10 Absorbing State/Sink: Is a state s_i that once entered cannot be left anymore:

$$p_{ij}^{(n)}(t) = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases} \tag{13.9}$$

Definition 13.11 Accessible State $s_i \to s_j$**:**
A state s_j is accessible from state s_i iff:

$$\exists n : \quad p_{ij}^{(n)}(t) > 0 \tag{13.10}$$

Definition 13.12 Communicating States $s_i \leftrightarrow s_j$**:**
Two states s_j and s_i are communicating iff:

$$\exists n_1 : \quad p_{ij}^{(n_1)}(t) > 0 \quad \wedge \quad \exists n_2 : \quad p_{ji}^{(n_2)}(t) > 0 \tag{13.11}$$

Definition 13.13 Periodicity of States: A state s_i has period k if any return to state s_i must occur in multiples of k time steps.
In other words k is the *greatest common divisor* of the number of transitions by which state s_i can be reached, starting from itself:

$$k = \gcd\{n > 0 : p_{ii}^{(n)} = \mathbb{P}(X_n = s_i \mid X_0 = s_i) > 0\} \tag{13.12}$$

Definition 13.14 Aperiodic State $k = 1$**:**
Is a state s_i with periodicity[def. 13.13] of one $\Leftrightarrow k = 1$

Corollary 13.5 : A state s_i is aperiodic if there exist two consecutive numbers k and $k+1$ s.t. the chain can be in state s_i at both time steps k and $k+1$.

Corollary 13.6 Absorbing State: An absorbing state is an aperiodic state.

Explanation 13.1 (Defintion 13.14). *Returns to state s_i can occur at irregular times i.e. the state is not predictable.*
In other words we cannot predict if the state will be revisited in multiples of k times.

1.4. Characteristics of Markov Processes/Chains

Definition 13.15
Time-homogeneous/Stationary Markov Chain:
are markov chains[def. 13.3] where the transition probability is independent of time:

$$p_{ji} = \mathbb{P}(X_t = s_j | X_{t-1} = s_i) = \mathbb{P}(X_{t-\tau} = s_j | X_{t-\tau} = s_i) \qquad \forall \tau \in \mathbb{N}_0 \tag{13.13}$$

Corollary 13.7 P
Transition Matrices of Stationary MCs:
Transition matrices of time-homogeneous markov chain are constant/time independent:

$$P(t) = P \tag{13.14}$$

Definition 13.16 Aperiodic Makrov Chain: Is a markov chain where all states are aperiodic:

$$\gcd\{n > 0 : p_{ii}^{(n)} = \mathbb{P}(X_n = s_i \mid X_0 = s_i) > 0\} = 1 \qquad \forall i \in \{1, \dots, n\} \tag{13.15}$$

Definition 13.17 Irreducable Markov Chain: Is a Markov chain that has only *communicating states*[def. 13.12]:

$$s_j \leftrightarrow s_i \qquad \forall i, j \in \{1, \dots n\} \tag{13.16}$$

- $\implies$ no sinks[def. 13.10]
- $\implies$ every state can be reached from every other state

Corollary 13.8 : An *irreducable*[def. 13.17] markov chain is automatically *apperiodic*[def. 13.16] if it has at least one aperiodic state[def. 13.14] $\iff$ *ergodic*[def. 13.18].

Corollary 13.9 : A markov chain is *not-irreducable* if there exist two states with different periods.

Definition 13.18 [example 13.1]
Ergodic Markov Chain: A finite markov chain is ergodic if there exist some number N s.t. any state s_j can be reached from any other state s_i in any number of steps less or equal to a N.
$\Rightarrow$ a markov chains is ergodic if it is:
1. *Irreducable*[def. 13.17]
2. *Aperiodic*[def. 13.16]

Corollary 13.10 Stationary Distribution: An erdodic markov chain has a *unique* stationary distribution[def. 13.9] and converges to it starting from any initial state $q_0(s_i)$

1.5. Types of Markov Chains

	Observable	Unobservable
Uncontrolled	MC[def. 13.3]	HMM[def. 14.1]
Controlled	MDP[def. 15.1]	POMDP[def. 16.1]

1.6. Markov Chain Monte Carlo (MCMC)

2. Proofs

Proof 13.1: [cor. 13.2]

$$\begin{aligned} \mathbb{P}(X_0 = x_0, \dots, X_N = x_N) &= \mathbb{P}(X_0 = x_0) \cdot \\ &\cdot \mathbb{P}(X_1 = x_1 | X_0 = x_0) \cdot \mathbb{P}(X_2 = x_2 | X_1 = x_1, X_0 = x_0) \cdot \\ &\cdots \mathbb{P}(X_N = x_N | X_{N-1} = x_{N-1}, \dots, X_0 = x_0) \end{aligned}$$

and then simply use the Markovian property

Proof 13.2: Corollary 13.3

$$\mathbf{q}_{n+1} = P\mathbf{q}_n = (\mathbf{q}_{n-1}P)P = \mathbf{q}_0 P^{n+1}$$

3. Examples

Example 13.1 Ergodic Markov Chain:

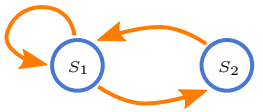
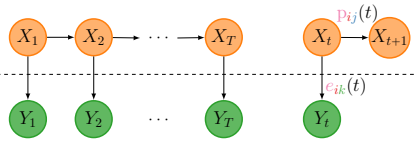


Figure 9: Ergodic for $N = 2$ (can reach s_2 at any $t \leqslant N$ after $N = 2$)

Hidden Markov Model (HMM)

Definition 14.1 $(\mathcal{S}, \mathcal{A}, \mathcal{O}, P, E)$
Hidden Markov Model (HMM):
Is a Markov Chain[def. 13.3] with hidden/latent states S_j that are only partially observable by noisy/indirect observations[def. 14.2]: It is characterized by the 5-tuple of:

1. States[def. 13.1] $\mathcal{S} = \{s_1, \dots, s_n\}$
2. Actions[def. 15.2] $\mathcal{A}/\mathcal{A}_{s_j} = \{a_1, \dots, a_m\}$
3. Observations[def. 14.2] $\mathcal{O}/\mathcal{O}_{s_j} = \{o_1, \dots, o_m\}$
4. Transition Probabilities[def. 13.5] $P(s_i, s_j)$
5. Emission/Output Probabilities[def. 14.3] $e_{ij}(t)$



Definition 14.2 Observations $\mathcal{O} = \{o_1, \dots, o_l\}$**:**
Are indirect or noisy observations that are related to the true states s_j.

Definition 14.3
Emission/ Output Probabilities $e_{ij}(t)$**:**
Given a state $X_t = s_i$ the output probability is the probability of the output random variable Y_t to be in state o_j:

$$e_{ij}(t) = \mathbb{P}(Y_t = o_j | X_t = s_i) \qquad \begin{cases} \forall o_i \in \mathcal{O} \\ \forall s_j \in \mathcal{S} \end{cases} \tag{14.1}$$

Markov Decision Processes (MDP)

Definition 15.1 $(\mathcal{S}, \mathcal{A}, \mathrm{P}_a, R_a)$
Markov Decision Process (MDP): A markov decision process is a *controlled* markov process/chain with an associated reward, where the transition can by steered by an actions. It is characterized by the 4-tuple of:

1. States[def. 13.1] $\mathcal{S} = \{s_1, \dots, s_n\}$
2. Actions[def. 15.2] $\mathcal{A}/\mathcal{A}_{s_j} = \{a_1, \dots, a_m\}$
3. Transition Probabilities[def. 15.3] $\mathrm{p}_a(s_i, s_j)$
4. Rewards[def. 15.4] $r_a(s_i, s_j)$

Definition 15.2
Actions $\mathcal{A}_{s_i} = \{a_1, \dots, a_m\}$:
Is the set of possible actions from which we can choose at each state and may depend on the state s_j itself.

Definition 15.3 Transition Probability $\mathrm{p}_a(s_j, s_i)(t)$:
is the probability of a random variable X_t in state s_i to transition into state s_j and depends also on the current action a:

$$\mathrm{p}_a(s_j, s_i) = \mathrm{p}(s_j|s_i, a) = \mathbb{P}(x_{t+1} = s_j | x_t = s_i, a_t = a) \quad \forall s_i, s_j \in \mathcal{S}, \forall a \in \mathcal{A} \tag{15.1}$$

Definition 15.4 Reward $r_a(s_i, s_j)$:
is a function or probability distribution that measures the immediate reward and may depend on a any subset of (x_{t+1}, x_t, a):

$$(x_{t+1}, x_t, a) \mapsto R_{t+1} \in \mathcal{R} \subset \mathbb{R} \tag{15.2}$$

Markov decision processes require us to plan ahead. This is because the immediate reward[def. 15.4], that we obtain by greedily picking the best action may result in non-optimal local actions.

1. Policies and Values

Definition 15.5
Optimizing Agent/ Decision Making Policy $\pi(s_i)$:
Is a policy on how to choose an action $a \in \mathcal{A}$ based on a objective/value function[def. 15.8] and can be deterministic or randomized:

$$\pi : \mathcal{S} \mapsto \mathcal{A} \quad \text{or} \quad \pi : \mathcal{S} \mapsto \mathbb{P}(\mathcal{A}) \tag{15.3}$$

Definition 15.6 Discounting Factor γ:
Is a factor $\gamma \in [0, 1)$ that signifies that future rewards are less valuable then current rewards.

Explanation 15.1 (Definition 15.6). *The reason for the discounting factor is that we may for example not even survive long enough to obtain future payoffs.*

Definition 15.7 Expected Discounted Value $\mathrm{J}(\pi)$:
Is the *discounted* expected (reward) of the whole markov process:

$$\mathrm{J}(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(X_t, \pi(X_t)) \right] \tag{15.4}$$

Definition 15.8
Value Function $\mathrm{V}^\pi(x)$:
Is the *discounted* expected reward[def. 15.4] of the whole markov process given an inital state $X_0 = x$:

$$\mathrm{V}^\pi(x) = \mathrm{J}(\pi|X_0 = x) \tag{15.5}$$

$$= \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(X_t, \pi(X_t)) \middle| X_0 = x \right] \tag{15.6}$$

$$\tag{15.7}$$

1.1. Calculating the value of V^π

Definition 15.9 [proof 16.1]
Value Iteration:

$$\begin{aligned} \mathrm{V}^\pi(x) &= \mathrm{J}(\pi|X_0 = x) \\ &= \mathbb{E}_{x'|x,\pi(x)} \left[r(x, \pi(x)) + \gamma \mathrm{V}^\pi(x') \right] \\ &= r(x, \pi(x)) + \gamma \mathbb{E}_{x'|x,\pi(x)} \left[\mathrm{V}^\pi(x') \right] \\ &= \boxed{r(x, \pi(x)) + \gamma \sum_{x' \in \mathcal{S}} \mathbb{P}(x'|x, \pi(x)) \mathrm{V}^\pi(x')} \end{aligned} \tag{15.8}$$

We can now write this for all possible initial states as:

$$\mathbf{V}^\pi = \mathbf{r}^\pi + \gamma \mathrm{P}^\pi \mathbf{V}^\pi \iff (\mathbf{I} - \gamma \mathrm{P}^\pi) \mathbf{V}^\pi = \mathbf{r}^\pi \tag{15.9}$$

with:

$$\mathbf{V}^\pi = \begin{bmatrix} \mathrm{V}^\pi(s_1) \\ \cdots \\ \mathrm{V}^\pi(s_n) \end{bmatrix} \qquad \mathbf{r}^\pi = \begin{bmatrix} r^\pi(s_1, \pi(s_1)) \\ \cdots \\ r^\pi(s_n, \pi(s_n)) \end{bmatrix}$$

$$\mathrm{P}^\pi = \begin{bmatrix} \mathbb{P}(s_1|s_1, \pi(s_1)) & \mathbb{P}(s_2|s_1, \pi(s_1)) & \cdots & \mathbb{P}(s_n|s_1, \pi(s_1)) \\ \mathbb{P}(s_1|s_2, \pi(s_2)) & \mathbb{P}(s_2|s_2, \pi(s_2)) & \cdots & \mathbb{P}(s_n|s_2, \pi(s_2)) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{P}(s_1|s_n, \pi(s_n)) & \mathbb{P}(s_2|s_n, \pi(s_n)) & \cdots & \mathbb{P}(s_n|s_n, \pi(s_n)) \end{bmatrix}$$

1.1.1. Direct Mehtods

Corollary 15.1 LU-decomposition $\mathcal{O}(n^3)$:
The linear system from eq. (15.9):

$$(\mathbf{I} - \gamma \mathrm{P}^\pi) \mathbf{V}^\pi = \mathbf{r}^\pi \tag{15.10}$$

can be solved *directly* using Gaussian elimination in polynomial time $\mathcal{O}(n^3)$.

Note – invertebility
If $\gamma < 1$ then $(\mathbf{I} - \gamma \mathrm{P}^\pi)$ is full-rank/invertible as EVs$(\mathrm{P}^\pi) \leqslant 1$.

1.1.2. Fixed Point Iteration

Corollary 15.2 Fixed-Point Iteration $\mathcal{O}(n \cdot |\mathcal{S}|)$:
The linear system from eq. (15.9) can be solve using *fixed-point iteration*?? in at most $\mathcal{O}(n \cdot |\mathcal{S}|)$ (if every state s_i is connected to every other state $s_j \in \mathcal{S}$)

Algorithm 15.1 Fixed Point Iteration:

Input: Inital Guess: $\mathbf{V}_0^\pi \overset{\text{i.e.}}{=} 0$
1: **for** $t = 1, \dots, T$ **do**
2: Use the fixed point method:

$$\mathbf{V}_t^\pi = \phi \mathbf{V}_t^\pi = \mathbf{r}^\pi + \gamma \mathrm{P}^\pi \mathbf{V}_{t-1}^\pi \tag{15.11}$$

3: **end for**

Corollary 15.3
Policy Iterration Contraction [proof 16.2]:
Fixed point iteration of policy iteration is a contraction?? that leads to a fixed point V^π with a rate depending on the discount factor γ.

$$\begin{aligned} \|\mathbf{V}_t^\pi - \mathrm{V}^\pi\| &= \|\phi \mathbf{V}_{t-1}^\pi - \phi \mathrm{V}^\pi\| \\ &\leqslant \gamma \|\mathbf{V}_{t-1}^\pi - \mathrm{V}^\pi\| = \gamma^t \|\mathbf{V}_0^\pi - \mathrm{V}^\pi\| \end{aligned} \tag{15.12}$$

Explanation 15.2.
- $\gamma \downarrow$: *the less we plan ahead/the smaller we choose γ the shorter it takes to converge. But on the other hand we only care greedily about local optima and might miss global optima.*
- $\gamma \uparrow$: *the more we plan ahead/the larger we choose γ the longer it takes to converge but we will explore all possibilities. But for to large γ we will simply keep exploring without sticking to a optimal poin*

Note contraction
For a contraction:
- A unique fixed point exists
- We converge to the fixpoint

1.2. Choosing The Policy

Question how should we choose the π? **Idea** compute $\mathrm{J}(\pi)$ for *every possible* policy:

$$\pi^* = \arg\max \mathrm{J}(\pi) \tag{15.13}$$

Problem this is unfortunately infeasible as there exist $m^n = |\mathcal{A}|^{|\mathcal{S}|}$ policies that we need to calculate the value for.

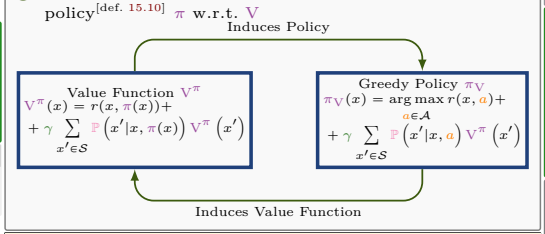
Note
The problem is that $\mathrm{J}/\mathrm{V}^\pi$ depend on π but if we do not know π yet we cannot compute those.

1.2.1. Greedy Policy

Definition 15.10 Greedy Policy:
Assuming we know $\mathrm{V}^{\pi t-1}$ then we could choose a greedy policy:

$$\begin{aligned} a^* &= \pi_t(x) \\ &:= \arg\max_{a \in \mathcal{A}} r(x, a) + \gamma \sum_{x' \in \mathcal{S}} \mathbb{P}(x'|x, a) \mathrm{V}^{\pi t-1}(x') \end{aligned} \tag{15.14}$$

1. Given a policy π however we can calculate a value function V^π
2. Given a value function V we can induce a greedy policy[def. 15.10] π w.r.t. V



Theorem 15.1 Optimality of Policies [Bellman]:
A policy π_V is optimal if and only if it is greedy w.r.t. its induced value function

Definition 15.11 Non-linear Bellman Equation: States that the optimal value is given by the action/policy that maximizes the value function eq. (15.8):

$$\mathrm{V}^*(x) = \max_{a \in \mathcal{A}} \left[r(x, a) + \gamma \sum_{x' \in \mathcal{S}} \mathbb{P}(x'|x, a) \mathrm{V}^*(x') \right] \tag{15.15}$$

$$:= \max_{a \in \mathcal{A}} Q^*(x, a) \tag{15.16}$$

Note
This equation is non-linear due to the max in comparison to eq. (15.8).

1.2.2. Policy Iteration

Algorithm 15.2 Policy Iteration:

Initialize: Random Policy: π
1: **while** Not converged $t = t + 1$ **do**
2: Compute $\mathrm{V}^{\pi t}(x)$

$$\mathrm{V}^{\pi t}(x) = r(x, \pi(x)) + \gamma \sum_{x' \in \mathcal{S}} \mathbb{P}(x'|x, \pi_t(x)) \mathrm{V}^{\pi t}(x')$$

3: Compute greedy policy π_G:

$$\pi_G(x) = \arg\max_{a \in \mathcal{A}} r(x, a) + \gamma \sum_{x' \in \mathcal{S}} \mathbb{P}(x'|x, a) \mathrm{V}^{\pi t}(x')$$

4: Set $\pi_{t+1} \leftarrow \pi_G$
5: **end while**

Algorithm 15.2

Pros
- Monotonically improves $\mathrm{V}^{\pi t} \geqslant \mathrm{V}^{\pi t-1}$
- is guaranteed to converge to an optimal policy/solution π^* in polynomial *#iterations*: $\mathcal{O}\left(\frac{n^2 m}{1-\gamma}\right)$

Cons
- Complexity *per iteration* requires to evaluate the policy V^π which requires us to solve a linear system.

1.2.3. Value Iteration

Definition 15.12 Value to Go $\mathrm{V}_t(x)$:
Is the maximal expected reward if we *start* in state x and have t time steps to go.

Algorithm 15.3 Value Iteration [proof 16.3]:

Initialize: $\mathrm{V}_0(x) = \max_{a \in \mathcal{A}} r(x, a)$
1: **for** $t = 1, \dots, \infty$ **do**
2: Compute:

$$\mathrm{Q}_t(x, a) = r(x, a) + \gamma \sum_{x' \in \mathcal{S}} \mathbb{P}(x'|x, a) \mathrm{V}_{t-1}(x') \quad \forall a \in \mathcal{A}, \forall x \in \mathcal{S}$$

4: for all $x \in \mathcal{S}$ let:

$$\mathrm{V}_t(x) = \max_{a \in \mathcal{A}} \mathrm{Q}_t(x, a)$$

5: **if** $\max_{x \in \mathcal{S}} |\mathrm{V}_t(x) - \mathrm{V}_{t-1}(x)| \leqslant \epsilon$ **then**
6: break
7: **end if**
8: **end for**
9: Choose greedy policy π_{V_t} w.r.t. V_t

Corollary 15.4 [proof 16.4]
Value Iterration Contraction:
Algorithm 15.3 is guaranteed to converge to a ϵ optimal policy:

$$\left\| \left(\mathrm{V}_t - \mathrm{V}^* \right) \right\|_\infty \leqslant \gamma^t \left\| \left(\mathrm{V}_0 - \mathrm{V}^* \right) \right\|_\infty \tag{15.17}$$

$$\implies t \approx \ln \frac{\gamma}{\epsilon} \left\| \mathrm{V}_0 - \mathrm{V}^* \right\|_\infty \quad \text{for} \quad \left\| \left(\mathrm{V}_t - \mathrm{V}^* \right) \right\|_\infty \leqslant \epsilon$$

Algorithm 15.3

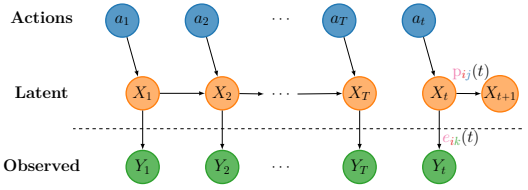
Pros
- Finds ϵ-optimal solution in polynomial *#iterations* $\mathcal{O}(\ln \frac{1}{\epsilon})$[cor. 15.4].
- Complexity *per iteration* requires us to solve a linear system $\mathcal{O}(m \cdot n \cdot s) = \mathcal{O}(|\mathcal{A}| \cdot |\mathcal{S}| \cdot s)$ where s is the number of states we can reach.
  For small s and small m we are roughly linear w.r.t. the states $\mathcal{O}(n) = \mathcal{O}(|\mathcal{S}|)$

Cons
- Only ϵ-optimal solution.

Partially Observable MDP (POMDP)

Definition 16.1 $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathrm{P}_a, E, R_a)$
Partially Observable Markov Decision Process:
A (POMDP) is a markov decision process[def. 15.1] with hidden markov states[def. 14.1]. It is characterized by the 6-tuple of:

1. States[def. 13.1] $\mathcal{S} = \{s_1, \dots, s_n\}$
2. Actions[def. 15.2] $\mathcal{A}/\mathcal{A}_{s_j} = \{a_1, \dots, a_m\}$
3. Observations[def. 14.2] $\mathcal{O}/\mathcal{O}_{s_j} = \{o_1, \dots, o_m\}$
4. Transition Probabilities[def. 15.3] $\mathrm{p}_a(s_i, s_j)$
5. Emission/Output Probabilities[def. 14.3] $e_{ij}(t)$
6. Rewards[def. 15.4] $r_a(s_i, s_j)$



Explanation 16.1.
Now our agent has only some indirect noisy observation of true state.

1. POMDPs as MDPs

POMDPs can be converted into *belief state*?? MDPs[def. 15.1] by introducing a *belief state space* $\mathcal{B}$.

Definition 16.2 History H_t**:**
Is a sequence of actions, observations and rewards:

$$H_t = \{\{a_0, o_0, r_0\}, \ldots, \{a_0, o_0, r_0\}\}$$

Definition 16.3 Belief State Space $\mathcal{B}$**:** Is a $|\mathcal{S}|-1$ dimensional simplex or ($|S|$-dimensional probability vector??) whose elements b are probabilities:

$$\mathcal{B} = \Delta(|\mathcal{S}|) = \left\{ b_t \in [0,1]^{|\mathcal{S}|} \;\middle|\; \sum_{x=1}^{n} b_t(x) = 1 \right\} \tag{16.1}$$

Definition 16.4 Belief State $b_t \in \mathcal{B}$**:** Is a probability distribution over the states $\mathcal{S}$ conditioned on the history H_t[def. 16.2].

1.1. Transition Model

Definition 16.5 [proof 16.5]
POMDP State/Posterior Update:

$$\begin{aligned} b_{t+1}(s_i) &= \mathbb{P}(X_{t+1} = s_i | Y_{t+1} = o_k) \\ &= \frac{1}{Z} \mathbb{P}(Y_{t+1} = o_k | X_{t+1} = s_i, a_t) \\ &\quad \cdot \sum_{s_j \in Pa(s_i)} b_t(s_j) \mathbb{P}(X_{t+1} = s_i | X_t = s_j, a_t) \end{aligned} \tag{16.2}$$

Definition 16.6 Stochastic Observation Model:

$$\mathbb{P}(Y_{t+1} = o_k | b_t, a_t) = \sum_{s_i \in \mathcal{S}} b_t(s_i) \mathbb{P}(Y_{t+1} = o_k | X_t = s_i, a_t) \tag{16.3}$$

1.2. Reward Function

Definition 16.7 POMDP Reward Function:

$$r(b_t, a_t) = \sum_{s_j \in \mathcal{S}} b_t(s_i) r(s_i, a_t) \tag{16.4}$$

Note

For finite horizon T, the set of reachable belief states is finite however exponential in T.

add defintion of simplex to math appendix which is basically this.

2. Proofs

2.1. Markov Decision Processes

Proof 16.1: [def. 15.8]

$$\begin{aligned} \mathrm{V}^{\pi}(x) &= \mathbb{E}_{X_{1:\infty}}\left[\sum_{t=0}^{\infty} \gamma^t r(X_t, \pi(X_t)) \mid X_0 = x\right] \\ &= \mathbb{E}_{\mathbf{X}}\left[\gamma^0 r(X_0, \pi(X_0)) + \sum_{t=1}^{\infty} \gamma^t r(X_t, \pi(X_t)) \mid X_0 = x\right] \\ &\overset{\substack{\gamma^0=1 \\ X_0=x}}{=} r(x, \pi(x)) + \mathbb{E}_{\mathbf{X}}\left[\sum_{t=1}^{\infty} \gamma^t r(X_t, \pi(X_t)) \mid X_0 = x\right] \\ &\overset{\text{re-index}}{=} r(x, \pi(x)) + \mathbb{E}_{\mathbf{X}}\left[\sum_{t=0}^{\infty} \gamma^{t+1} r(X_{t+1}, \pi(X_{t+1})) \mid X_0 = x\right] \\ &= r(x, \pi(x)) + \gamma \mathbb{E}_{\mathbf{X}}\left[\sum_{t=0}^{\infty} \gamma^t r(X_{t+1}, \pi(X_{t+1})) \mid X_0 = x\right] \\ &= r(x, \pi(x)) \\ &\quad + \gamma \mathbb{E}_{X_1}\left[\mathbb{E}_{X_{2:\infty}}\left[\sum_{t=0}^{\infty} \gamma^t r(X_{t+1}, \pi(X_{t+1})) \middle| X_1 = x'\right] \middle| X_0 = x\right] \\ &\overset{??}{=} r(x, \pi(x))) \\ &\quad + \gamma \sum_{x' \in S} \mathbb{P}(x' | x, \pi(x)) \mathbb{E}_{X_{2:\infty}}\left[\sum_{t=0}^{\infty} \gamma^t r(X_{t+1}, \pi(X_{t+1})) \middle| X_1 = x'\right] \\ &\overset{\text{eq. (13.13)}}{=} r(x, \pi(x))) \\ &\quad + \gamma \sum_{x' \in S} \mathbb{P}(x' | x, \pi(x)) \mathbb{E}_{X_{2:\infty}}\left[\sum_{t=0}^{\infty} \gamma^t r(X_t, \pi(X_t)) \middle| X_0 = x'\right] \\ &= r(x, \pi(x)) + \gamma \sum_{x' \in S} \mathbb{P}\left(x' | x, \pi(x)\right) \mathrm{V}^{\pi}\left(x'\right) \end{aligned}$$

Proof 16.2 [cor. 15.3]: Consider $\mathrm{V}, \mathrm{V}' \in \mathbb{R}^n$ and let ϕ:

$$\phi x := r^{\pi} + \gamma \mathrm{P}^{\pi} x \qquad \Longrightarrow \qquad \phi \mathrm{V}^{\pi} = \mathrm{V}^{\pi}$$

then it follows:

$$\begin{aligned} \left\| \phi \mathrm{V} - \phi \mathrm{V}' \right\| &= \left\| \cancel{r^{\pi}} + \gamma \mathrm{P}^{\pi} \mathrm{V} - \cancel{r^{\pi}} - \gamma \mathrm{P}^{\pi} \mathrm{V}' \right\| \\ &= \left\| \gamma \mathrm{P}^{\pi} \left(\mathrm{V} - \mathrm{V}' \right) \right\| \\ &\overset{??}{\leqslant} \gamma \left\| \mathrm{P}^{\pi} \right\| \cdot \left\| \left(\mathrm{V} - \mathrm{V}' \right) \right\| \\ &\overset{\text{i.e. } L_2}{\leqslant} \gamma \cdot 1 \cdot \left\| \left(\mathrm{V} - \mathrm{V}' \right) \right\|_2 \end{aligned}$$

Proof 16.3: algorithm 15.3

$$\begin{aligned} \mathrm{V}_0(x) &= \max_{a \in \mathcal{A}} r(x, a) \\ \mathrm{V}_1(x) &= \max_{a \in \mathcal{A}} r(x, a) + \gamma \sum_{x' \in \mathcal{S}} \mathbb{P}\left(x' | x, a\right) \mathrm{V}_0\left(x'\right) \\ \mathrm{V}_{t+1}(x) &= \max_{a \in \mathcal{A}} r(x, a) + \gamma \sum_{x' \in \mathcal{S}} \mathbb{P}\left(x' | x, a\right) \mathrm{V}_t\left(x'\right) \end{aligned}$$

Proof 16.4: [cor. 15.4] Let $\phi : \mathbb{R}^n \mapsto \mathbb{R}^n$, with:

$$\left(\phi \mathrm{V}^*\right)(x) = Q(x, a) = \max_a \left[r(x, a) + \gamma \sum_{x'} \mathbb{P}\left(x' | x, a\right) \right]$$

Bellman's theorem 15.1 $\qquad \phi \mathrm{V}^* = \mathrm{V}^*$

and consider $\mathrm{V}, \mathrm{V}' \in \mathbb{R}^n$

$$\begin{aligned} \left\| \phi \mathrm{V} - \phi \mathrm{V}' \right\|_{\infty} &= \max_x \left| (\phi \mathrm{V})(x) - (\phi \mathrm{V}')(x) \right| \\ &= \max_x \left| \max_a Q(x, a) - \max_{a'} Q'(x, a') \right| \\ &\overset{??}{\leqslant} \max_x \max_a \left| Q(x, a) - Q'(x, a) \right| \\ &= \max_{x,a} \left| \cancel{r} + \gamma \sum_{x'} \mathbb{P}\left(x' | x, a\right) \mathrm{V}(x') - \cancel{r} - \gamma \sum_{x'} \mathbb{P}\left(x' | x, a\right) \mathrm{V}'(x') \right| \\ &= \gamma \max_{x,a} \left| \sum_{x'} \mathbb{P}\left(x' | x, a\right) \left(\mathrm{V}(x') - \mathrm{V}'(x') \right) \right| \\ &\overset{??}{\leqslant} \gamma \max_{x,a} \overbrace{\left| \sum_{x'} \mathbb{P}\left(x' | x, a\right) \right|}^{\leqslant 1} \cdot \left| \left(\mathrm{V}(x') - \mathrm{V}'(x') \right) \right| \\ &\leqslant \gamma \cdot 1 \cdot \left\| \left(\mathrm{V}(x') - \mathrm{V}'(x') \right) \right\|_{\infty} \end{aligned}$$

Note

For the policy iteration the calculation was easier as the rewards canceled, however here we have the max.

2.2. MDPs

Proof 16.5: Defintion 16.5 Directly by definition 7.5 and its corresponding proof 10.4 with additional action a_t:

$$\begin{aligned} b_{t+1}(s_i) &= \mathbb{P}(X_{t+1} = s_i | y_{t+1}) \\ &= \frac{1}{Z} \mathbb{P}(y_{1:t+1} | s_i) \sum_{j=1} \overbrace{\mathbb{P}(X_t = s_j | y_{1:t})}^{b_t(s_j)} \overbrace{\mathbb{P}(X_{t+1} = s_i | y_{1:t})}^{\mathbb{P}(s_i | s_j)} \end{aligned}$$

Reinforcement Learning

Now we are working with an *unknown* MDP[def. 15.1] meaning that:
1. we do no longer know the transition model[def. 15.3]
2. We do no longer know the reward function
3. We might not even know all the states

However we can observe them when taking steps.

Note
- Reinforcement learning is different than supervised learning as the data is no longer i.i.d. (data depends on previous action).
- Need to do exploration vs exploitation in order to learn policy and reward functions.

Definition 17.1 Agent:
Is the *learner/decision maker* of our *unknown* MDP.

Definition 17.2 Environment: Is the representation of the world in which our agents acts.

Definition 17.3 On-Policy Learning: At any given time the agent has full control which actions to pick.

Definition 17.4 Off-Policy Learning: The agent has to fix a policy in advance based on behavioral observations.

Definition 17.5 Trajectory τ**:**
Is a set of consecutive 3-tuples of states, actions and rewards:

$$\tau = \{s_t, a_t, r_t\} \qquad t = 1, \dots, \tau \tag{17.1}$$

Definition 17.6 Episodic Learning: Is a setting where we generate multiple K-episodes of different trajectories $\left\{\tau^{(k)}\right\}_{i=1}^{K}$ from which the agent can learn.

Explanation 17.1. *For each episode the agent starts in a random state and follows a policy.*

1. Model Based Reinforcement Learning

Proposition 17.1 Model Based RL:
Try to learn the MDP [def. 15.1] by:
1. Estimating
   - the transition probabilities[def. 15.3] $\quad p_a(s_i, s_j)$
   - the reward function[def. 15.4] $\quad r(b_t, a_t)$
2. Optimizing the policy of the estimated MDP

1.1. Estimating Transitions and Rewards

Formula 17.1 Estimating Transitions and Rewards:
Given a data set $D = \{(\mathbf{x}_0, a_0, r_0, \mathbf{x}_1), (\mathbf{x}_1, a_1, r_1, \mathbf{x}_2), \dots\}$ we estimate the transitions and rewards using a categorical distribution**??**:

$$N_{s_i|s_j,a} := \sum_{k=1}^{t} \delta_{\left(X_{k+1}=s_i|X_k=s_j, A_k=a\right)} \tag{17.2}$$

$$N_{s_j,a} := \sum_{k=1}^{t} \delta_{\left(X_k=s_j, A_k=a\right)} \tag{17.3}$$

$$p_a(s_i, s_j) \approx \frac{N_{s_i|s_j,a}}{N_{s_j,a}} \tag{17.4}$$

$$r(s_i, a) \approx \frac{1}{N_{s_i,a}} \sum_{k=1}^{t} \delta_{(X_k=s_i, A_k=a)} r(X_k, A_k) \tag{17.5}$$

1.2. Choosing the next step

How should we choose the action $a \in \mathcal{A}$ in order to balance exploration vs exploitation?

1.3. ϵ_t Greedy Learning

Algorithm 17.1 Epsilon Greedy Learning:
1: **for** $t = 1, \dots, T$ **do**
2: Pick next action

$$a_t = \begin{cases} \arg\max_a Q_t(a) & \text{with probability } \epsilon_t \\ \text{random } a & \text{with probability } 1 - \epsilon_t \end{cases}$$

3: **end for**

Corollary 17.1 Necessary Condition for Convergence:
If the sequence ϵ_t satisfies the *Robbins Monro* (RM) conditions

$$\sum_t \epsilon_t < \infty, \qquad \sum_t \epsilon_t^2 < \infty \qquad \text{(i.e. } \epsilon_t = 1/t\text{)} \tag{17.6}$$

then algorithm 17.1 converges to an optimal policy with probability one.

Add general definition of RM conditions and sequence

Pros
- Simple

Cons
- Clearly sub optimal actions are not eliminated fast enough

1.4. The $R_{\max}$ Algorithm

Algorithm 17.2 [Brafman & Tennenholz '02]
R-max Algorithm:
Initialize every state with:

$$\hat{r}(s_t, a) = R_{\max} \qquad \hat{p}_a(X_{t+1}|X_t = s_i, a) = 1 \tag{17.7}$$

Set min. number Δ of observations for policy update
Compute Policy π_1 of the MDP[def. 15.1] using $(\hat{p}, \hat{r})$: π_t
1: **for** $k = 1, \dots, K$ **do**
2: Choose $a = \pi_t(x_t)$ and observe (s, r)
3: Calculate:

$$N_{\mathbf{x}_t,a} += 1 \qquad r(x_t, a) += r(x_t, a) \tag{17.8}$$

$$N_{\mathbf{x}_{t+1}|\mathbf{x}_t,a} += 1 \tag{17.9}$$

4: **if** $k == \Delta$ **then**
5: Re-calculate (based on eqs. (17.4) and (17.5)):

$$\hat{r}(s_t, a) = R_{\max} \qquad \hat{p}_a(X_{t+1}|X_t = s_i, a) = 1$$

and update the policy $\pi_t = \pi_t(\hat{p}, \hat{r})$
6: **end if**
7: **end for**

Note
Other ways of updating the policy at certains times exist.

Problems

Cons
- Memory: for all $a \in \mathcal{A}$, $\mathbf{x}_{t+1}, \mathbf{x}_t \in \mathcal{X}$ we need to store $\hat{p}_a(x_{t+1}|x_t, a)$ and $\hat{r}(s_t, a)$ which results in $|\mathcal{S}|^2|\mathcal{A}|$ (for dense MDP).
- Computation Time: We need to calculate the π_t using policy (**??** 1.2.2) or value iteration (**??** 1.2.3) $|\mathcal{A}| \cdot |\mathcal{S}|$ whenever we update out policy.

1.4.1. How many transitions do we need?

Proposition 17.2 [proof 17.1]
Number of Samples to bound Reward:

$$\mathbb{P}\left(\hat{r}(s, a) - r(s, a) \leqslant \epsilon\right) \geqslant 1 - \delta \iff n \in \mathcal{O}\left(\frac{{R_{\max}}^2}{\epsilon^2} \log \frac{1}{\delta}\right) \tag{17.10}$$

Theorem 17.1 : Every T timesteps, with high probability, $R_{\max}$ either:
- Obtain near optimal reward, or
- Visits at leas one unkown state-action pair

Theorem 17.2 Performance of R-max: With probability $\delta - 1$, $R_{\max}$ will reach an ϵ-optimal policy in a number of steps that is polynomial in $|\mathcal{X}|, |\mathcal{A}|, T, 1/\epsilon$.

2. Model Free Reinforcement Learning

Proposition 17.3 Model Free RL:
Tries to estimate the value function[def. 15.8] directly in order to act greedily upon it.
- Policy Gradient Methods
- Actor Critic Methods

2.1. Temporal Difference Learning (TD)

Assume we fix a random intial policy π and s.t. we have $\hat{V}_0^{\pi}(s_j)$.
Goal: want to calculate an unknown value function V^{π}.
If the reward and the next states are stochastic variables (R, X) we can calculate the reward usingeq. (15.8):

$$\hat{V}^{\pi}(x_t) = \mathbb{E}_{X_{t+1},R}\left[R + \gamma \hat{V}^{\pi}(X') | X, a\right] \tag{17.11}$$

Now assume we observe a single example

$$(X_{t+1} = s_j, a, r, X_t = s_i)$$

then we can use monte carlos sampling**??** with a single sample to approximate the expectation ineq. (17.11):

$$\hat{V}_{t+1}^{\pi}(s_i) = r + \gamma \hat{V}_t^{\pi}(s_j)$$

Problem: high variance of estimates $\Rightarrow$ average with previous estimate.

Definition 17.7 Temporal Difference (TD) Learning:

$$\hat{V}(x_{t+1}) = (1 - \alpha_t)\hat{V}(x_t) + \alpha_t\left(r + \gamma \hat{V}(x_{t+1})\right) \tag{17.12}$$

Corollary 17.2 Necessary Condition for Convergence:
If the learning rate α_t satisfies the *Robbins Monro* (RM) conditions

$$\sum_t \alpha_t < \infty, \qquad \sum_t \alpha_t^2 < \infty \qquad \text{(i.e. } \alpha_t = 1/t\text{)} \tag{17.13}$$

and all state-action pairs (s_i, a_j) are chosen infinitely often, then we converge to the correct value function:

$$\mathbb{P}\left(\hat{V} \to \hat{V}^{\pi}\right) = 1 \tag{17.14}$$

2.2. Q-Learning

Definition 17.8 Action Value/Q-Function:

$$Q \tag{17.15}$$

2.2.1. Policy Gradients

2.2.2. Actor-Critic Methods

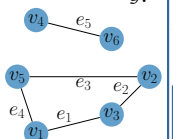
3. Proofs

Proof 17.1: proposition 17.2 using hoeffdings bound**??** with δ and $b - a = R_{\max}$.

Graph Theory

Definition 18.1 Graph $\mathcal{G}$:

A graph $\mathcal{G}$ is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of a finite set of vertices $\mathcal{V}$[def. 18.4] and a multi set?? of edges $\mathcal{E}$[def. 18.8].



Definition 18.2 Order $n = |\mathcal{V}|$:
The order of a graph is the cardinality of its vertix set.

Definition 18.3 Size $m = |\mathcal{E}|$:
The size of a graph is the number of its edges.

Corollary 18.1 n-Graph: Is a graph $\mathcal{G}$[def. 18.1] of order n.

Corollary 18.2 (p, q)-Graph: Is a graph $\mathcal{G}$[def. 18.1] of order p and size q.

Vertices

Definition 18.4 Vertices/Nodes $\mathcal{V}$:
Is a set of entities of a graph connected and related by edges in some way:

Definition 18.5 Neighbourhood $N(v)$:
The neighborhood of a vertix $v_i \in \mathcal{V}$ is the set of all adjacent vertices:

$$N(v_i) = \{v_k \in \mathcal{V} : \exists e_k = \{v_i, v_j\} \in \mathcal{E}, \forall v_j \in \mathcal{E}\} \quad (18.1)$$

Degree Matrix

Definition 18.6 Degree of a Vertix δ:
The degree of a vertix v is the cardinality of the neighborhood[def. 18.5] – the number of adjacent vertices:

$$\deg(v_i) = \delta(v) = |N(v)| = \sum_{\substack{j<i \\ j=1}} \mathbf{A}_{ij} \quad (18.2)$$

Definition 18.7 Degree Matrix $\mathbf{D}$:
Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ its degree matrix is a diagonal matrix $\mathbf{D} \in \mathbb{N}^{n,n}$ defined as:

$$\mathbf{D}_{i,j} := \begin{cases} \deg(v_i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (18.3)$$

Edges

Definition 18.8 Edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$:
Represent some relation between edges[def. 18.4] and are represented by two-element subset sets of the vertices:

$$e_k = \{v_i, v_j\} \in \mathcal{E} \iff v_i \text{ and } v_j \text{ connected} \quad (18.4)$$

Proposition 18.1 Number of Edges:
A graph $\mathcal{G}$ with $n = |\mathcal{V}|$ has between $\left[0, \frac{1}{2}n(n-1)\right]$ edges.

Graph Representations

Adjacency Matrix

Definition 18.9 (unweighted) Adjacency Matrix $\mathbf{A}$:
Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ its *adjacency matrix* is a square matrix $\mathbf{A} \in \mathbb{N}^{n,n}$ defined as:

$$\mathbf{A}_{i,j} := \begin{cases} 1 & \text{if } \exists e(i,j) \\ 0 & \text{otherwise} \end{cases} \quad (18.5)$$

Definition 18.10 weighted Adjacency Matrix $\mathbf{A}$:
Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ its *weighted adjacency matrix* is a square matrix $\mathbf{A} \in \mathbb{R}^{n,n}$ defined as:

$$\mathbf{A}_{i,j} := \begin{cases} \theta_{ij} & \text{if } \exists e(i,j) \\ 0 & \text{otherwise} \end{cases} \quad (18.6)$$

Diagonal Elements

For a graph without self-loops the diagonal elements of the adjacency are all zero.

Adjacency List

defintion

pros and cons list vs matrix

Operations on Graphs

1. Walks

Definition 19.1 Walk: A walk of a graph $\mathcal{G}$ as a sequence of vertices with corresponding edges:

$$W = \{v_k, v_{k+1}\}_k^K \in \mathcal{E} \quad (19.1)$$

Definition 19.2 Length of a Walk K: Is the number of edges of that Walk.

2. Paths

Definition 19.3 Path P: Is a walk of a graph $\mathcal{G}$ where all visited vertics are distinct (no-repetitions).

Attention: Some use the terms walk for paths and simple paths for paths.

3. Cycles

Definition 19.4 Cycle: Is a path[def. 19.3] of a graph $\mathcal{G}$ where the last visited vertix is the one from which we started.

Types of Graphs

1. Subgraph

Definition 20.1 Subgraph $\mathcal{H} \subseteq \mathcal{G}$:
A graph $\mathcal{H} = (U, F)$ is a *subgraph* of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ iff:

$$U \subseteq \mathcal{V} \quad \text{and} \quad F \subseteq \mathcal{E} \quad (20.1)$$

2. Components

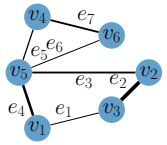
Definition 20.2 Component: A connected component of a graph $\mathcal{G}$ is a *connected*[def. 20.6] subgraph[def. 20.1] of $\mathcal{G}$ that is *maximal by inclusion* – there exist no larger connected containing subgraphs.
The number of components of a graph $\mathcal{G}$ is defined as $c(\mathcal{G})$.

3. Weighted Graph

Definition 20.3 Weighted Graph:

Is a graph $\mathcal{G}$ where edges are associated with a weight:

$$\exists \theta_i := \text{weight}(e_i) \qquad \forall e_i \mathcal{E}$$



4. Spanning Graph

Definition 20.4 Spanning Graph:

Is a subgraph[def. 20.1] $\mathcal{H} = (U, F)$ of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ for which it holds:

$$U = \mathcal{V} \quad \text{and} \quad F \subseteq \mathcal{E} \quad (20.2)$$

4.1. Minimum Spanning Graph

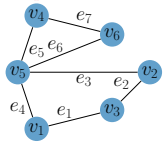
Definition 20.5 Minimum Spanning Graph: Is a spanning graph[def. 20.4] $\mathcal{H} = (U, F)$ of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with minimal weights/distance of the edges.

5. Connected Graphs

Definition 20.6 (Weakly) Connected Graph:

Is a graph $\mathcal{G}$[def. 18.1] where there exists a path between any two vertices:

$$\exists P(v_i, \dots, v_j) \qquad \forall v_i, v_j \in \mathcal{V} \quad (20.3)$$



Corollary 20.1 Strongly Connected Graph: A directed Graph[def. 20.8] is called strongly connected if every nodes is *reachable* from every other node.

Corollary 20.2 Components of Connected Graphs: A connected Graph[def. 20.6] consist of one component $c(\mathcal{G}) = 1$.

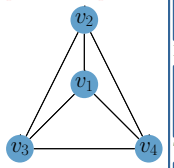
5.1. Fully Connected/Complete Graph

Definition 20.7 Fully Connected/Complete Graph:

Is a connected graph $\mathcal{G}$[def. 20.6] where each node is connected to every other node.

$$\exists e \forall \{v_i, v_j\} \qquad \forall v_i, v_j \in \mathcal{V} \quad (20.4)$$

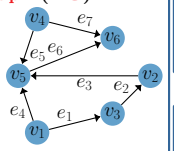
$$|\mathcal{V}| = \frac{1}{2}|\mathcal{V}|(|\mathcal{V}| - 1) \quad (20.5)$$



5.2. Directed Graphs

Definition 20.8 Directed Graph/Digraph (DG):

A directed graph $\mathcal{G}$ is a graph where edges are direct arcs[def. 20.9].



Definition 20.9 Directed Edges/Arcs: Represent some *directional* relationship between edges[def. 18.4] and are represented by *ordered* two-element subset sets of vertices:

$$e_k = \{v_i, v_j\} \in \mathcal{E} \iff v_i \text{ goes to } v_j \quad (20.6)$$

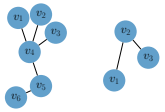
Acyclic Graphs

Definition 21.1 Acyclic Graphs:
Are graphs[def. 18.1] where no cycles[def. 19.4] exist.

Forests

Definition 21.2 Forests:

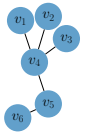
Are acyclic graphs[def. 21.1]:



Trees

Definition 21.3 Trees:

Are acyclic graphs[def. 21.1] that are connected[def. 20.6].



Binary Trees

Definition 21.4 Binary Tree:
Is a tree where each node $v_i \in \mathcal{V}$ has up to two children:

$$\deg(v_i) \leqslant 2 \qquad \forall v_i \in \mathcal{V} \quad (21.1)$$

Definition 21.5 Binary Search Tree (BST):
Is a binary tree[def. 21.5], where the left subtree of a node contains only values smaller than the parent and the right subtree contains only values larger than the parent.

Corollary 21.1 Balanced Binary Search Tree:
Is a tree that ensures $\mathcal{O}(\log n)$ time for finding or inserting a node. It is a tree where the number of left and right descendants is roughly equal.

Red-black tree

AVL trees

Definition 21.6 Complete Binary Trees:
A complete binary tree is a tree in which every node of every level of tree has two children, except the last, to the extent that it has to be filled left to right.

Definition 21.7 Fully Binary Tree: Is a tree where every node has either zero or two children.

Definition 21.8 Perfect Binary Tree: Is a complete binary tree where the last level is also filled, a perfect tree of height n needs to have 2^{n-1} nodes.

Binary Max/Min-Heaps

Definition 21.9 Binary Heap:
Is a complete-binary tree[def. 21.6] where every parent is smaller/larger (min-heap/max-heap) than its children.

Tries/Prefix Trees

Definition 21.10 Prefix Tree:
Is a tree special kind of tree where each node can have multiple children. It is usually used for prefix lookup of words, where words with the same prefix share the same nodes. It can reduce lookup time from $\mathcal{O}(M \log N)$ for a word of size M with N total words to $\mathcal{O}(M)$. Special terminating nodes are used to indicate if a prefix is an actual word.

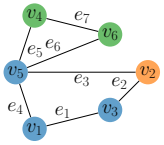
1. Graph Layering

Definition 21.11 Graph Layering:

Given a graph $\mathcal{G}$ a layering of the graph is a partition of its node set $\mathcal{V}$[def. 18.4] into subsets

$$\{\mathcal{V}_1, \dots, \mathcal{V}_L\} \subseteq \mathcal{V}$$

$$\text{s.t.} \quad \mathcal{V} = \mathcal{V}_1 \cup \dots \cup \mathcal{V}_L \quad (21.2)$$



2. Bisection Algorithms

2.1. Local Approaches

2.2. Global Approaches

2.2.1. Spectral Decomposition

Definition 21.12 Graph Laplacian (Matrix) $\mathbf{L}(\mathcal{G})$:
Given a graph with n vertices and m edges has a graph laplacian matrix defined as:

$$\mathbf{L} = \mathbf{A} - \mathbf{D} \qquad l_{ij} := \begin{cases} -1 & \text{if } i \neq j \text{ and } e_{ij} \in \mathcal{E} \\ 0 & \text{if } i \neq j \text{ and } e_{ij} \notin \mathcal{E} \\ \deg(v_i) & \text{if } i = j \end{cases} \quad (21.3)$$

Corollary 21.2 title:

2.2.2. Inertial Bisection