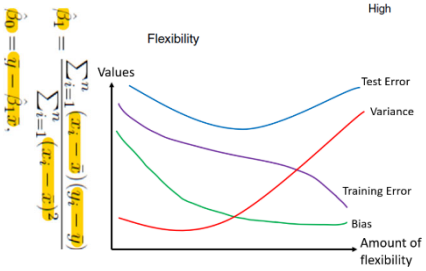
#T0. Review (**Sample space** Ω, the set of all possible outcomes. **Event** E, any subset of the outcomes of Ω, A+A`=Ω, **Union** $A \cup B$, **Intersection** $A \cap B$, **Mutually exclusive** $A \cap B=0$)(**Discrete**, integer coding, **Continuous**, real num coding)(**Discrete prob dist**. = (Discrete uniform, Bernoulli, Binomial, Geometric, Poisson, Negative binomial, Hyper geometric), **Continuous prob dist**. = (Continuous uniform, Exponential, Gamma, Normal))(**Independent RV**, D: p(x,y)=pX(x) · pY(y), F: f(x,y)=fX(x) · fY(y)) (**Type 1** err, reject H0 when H0 is T, **type 2** err, fail to reject h0 when Ha is T, sig lv = P(type 1 err) = α) (**Test**: one sample, z-test, X^2-test; two sample: z-test, t-test, F-test; multivariate: T^2-test)



#T1. Statistical Learning (**Statistics** (prediction in statistical views). Machine learning, Modeling vs. learning from data, Inference vs. prediction, Small vs. big dataset) (Suppose we observe a response Y and p diff predictors X=X1…Xp, we can model y=f(X)+ε, ε=error with mean0)) (**Supervised**, both XY are observed, -> Linear Reg, fir model -> relates the response to the predictors) (**Unsupervised**, only X are observed, lack a response to supervise the analysis) (**Sup**. -> reg and class prob., **reg** -> Y is **continuous/ numerical**, **class** -> Y is **categorical**, select base on response is **quantitative or qualitative** or properly coded) (**Reason** to est. f: **prediction**, **inference**, **pred**: as err mean=0, $\hat{y} = \hat{f}(x)$, f hat = est. of f, y hat = y pred, $\hat{f}$ treat as black box, Predictive modeling. **inference**: relationship %X,Y, e.g. which predictors are associated with response, simple linear relations or complicated, $\hat{f}$ can't treat as black box, Inferential modeling.)(**Est f**, p= #predictors, n= #observations, use train data + statistical method (Parametric Methods and non-) to est. f. **Parametric**: assume shape of f, e.g. linear model, use train data to fix, est. parameters $\beta 0 \ldots \beta p$), **Pros**: simplify est. f, **cons**: model may not match, far from true f -> chose flexible models, but need to est. greater num of parameters. **Non**: don't make explicit assume about the fun form of f, seek est. f that gets as close to the data as possible, **pros**: avoiding assumption, no bias. **Cons**: very large num of observations is req.) (**Restrictive** methods: relatively small range of shape to est. f e.g., linear reg, **Flexible** methods: wider range of possible shape to est. f, e.g., thin0plate splines. **why Restrictive** over flex, 1. Mainly interested in inference, easy to understand the relationship, (better interpretability), 2 obtain more accurate predictions via less flexible, potential overfitting in high flex metho, OF: model follows err/ noise, too closely, will not yield acc. est. of the response on new data) (**Learning process**, **training**, pred on training data. Measure MSE, $MSE = 1/n \sum \left(yi - \hat{f}(xi)\right)^2$, MSE will be small if pred responses are close to true response. **Prediction**, not used in train, measure MSE, small better.) (**Train vs test MSE**, test MSE measures accuracy of the method -> performance measure, as unseen data. Method w/ lowest train MSE =/= have lowest test MSE) (**Test MSE** = bias^2+var+var(ε), bias=error cause by approx. var=uncertainty due to rand of the training data. More flex method=lower bias + higher var, bias-var trade off)

#T2. Linear Regression (**Simple LR**, pred Y by single X, $Y = \beta 0 + \beta 1X + \varepsilon$, 0:intercept, 1:slope, **Est coeff**, goal: fit the data well, minimize $RSS = \sum \left(yi - \left(\hat{\beta}0 + \hat{\beta}1xi\right)\right)^2$, **Assessing Accuracy of coeff est**, population reg line: best linear approx to the T relationship, **Standard err**, avg est coeff diff from T value, **Assessing Model Fitting**, quality of LR fit (in training): $RSE = \sqrt{1/(n-2)\,RSS} = \hat{\sigma}$), est sd of ε, measure lack of fit; $R^2 = 1 - \sum(yi - \hat{y}i)^2 / \sum(yi - \bar{y})^2$, measure the proportion [0,1] of var explained by model, **intervals**, CI for avg response, pred interval of the response, PI is wider than CI.)(**Multiple LR**, $Y = \beta 0 + \beta 1X1 \ldots + \beta jXj + \varepsilon$, 0:intercept, j:slope, **Relationship**, F-test on all coeff, $F = ((TSS - RSS)/p)/(RSS/n - p - 1)$, overall effect of predictors, means at least one is assicuated. **Variable selection**: compare models. Model search, shrinkage. **R^2 problem**, increase when more variables added, even eakly associated. **Adjusted R^2**, $R_a^2 = 1 - (1 - R^2)(N - 1)/(N - p - 1))$ **(Non-measurable predictor**, aka factor, code into multi levels, gender->0(-1),1, e.g. $male = \beta 0, female = \beta 0 + \beta 1$, male as the baseline, n1 as the avg diff %genders)(**Interaction**, … $\beta 3X1X2$, may increase by both, without int. reg line are paralell, with reg line, inparallel, diff in slope and intercept)(**Nonlinear Relationships**, e.g. degre 3=+b1x1^1+b2x1^2+b3x1^3)(**Model Diagnostics**, **Non-linearity**, assume st line, check pattern in residual plots ($ei\ vs\ \hat{y}i$), solution, use non-linear transformations of predictors (x^2 log x sqrt x), **Non-constant Var of Err Terms**, heteroscedasticity, Ir assume err terms has constant var, var(ei)=σ^2), check funnel shape in residual plots ($ei\ vs\ \hat{y}i$), solution, transform respomnse e.g. log Y, sqrt Y, use weighted RSS if info on Var of individual responses is ava. **Outliers**, unusual yi for given xi, check the plot of studentized residuals (abs value>3), solutiuon, err in data collect->correct/remove, real->can't remove. **High Leverage Points**, unusual value of xi, check the plot of leverages (hi>>(p+1)/n), solution, lim the value of x. **Collinearity**, 2+ predictors are closely related to the other, (difficult to determine how each one of the collinear var separately affect the response), incease std err in coeff est., solution, VIF: indicate how serios the colinearity is, 1:no, >5: serious, drop one of the problematic predictoes.)

#T3. Classification (predict qualitative response, classifier, pop class: Log Reg, LDA & QDA, KNN)(**Log Reg**, linear model using binary response, **problem**: interpret -ve value and value >1, **solution**: log fun: $P(Y = 1) = e^{\beta 0 + \beta 1X} / \left(1 + e^{\beta 0 \beta 1X}\right)$,

$$l(\beta_0, \beta_1) = \prod_{i:y_i=1} P(y_i = 1) \prod_{i':y_{i'}=0} 1 - P(y_{i'} = 1)$$

$$P(y_i = 1) = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \qquad P(y_{i'} = 0) = \frac{1}{1 + e^{\beta_0 + \beta_1 x_{i'}}}$$

$$\Pr(Y = k | X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

K: the total number of response classes
π_k: the overall or prior probability of the kth class
$f_k(x)$: the density function of observations from the kth class

$\log\left((Y = 1)/\left(1 - (Y = 1)\right)\right) = \beta 0 + \beta 1X$, linear: b1=avg change in Y for one-unit increase in X, log: simple interpretation does not work, as predictiing the prob P(Y) not Y. **Interpreting b1**, =0, no relation %response-predictor, >0 x gets larger so does the prob of default, <0 x gets larger prob of default smaller, how much larger -> slope. **Est. Coeff**, Maximum likelihood method, est b0 b1 are chosen to max this fun.) (**Multiple Log Reg**, $P(Y = 1) = e^{\beta 0 + \beta 1X1 \ldots + \beta jXj} / \left(1 + e^{\beta 0 + \beta 1X1 \ldots + \beta jXj}\right)$) (**Multinomial Log Reg**, two ways, select a single class as baseline or treat all classes symmetrically.) (**LDA**, assume each predictor var is norm dist, if >1 predictor, all follow a multivariable norm dist. **Why not log reg**, when n is small, the dist of X is approx. normal in each class, LDA is more stable than log reg. LDA is more pop when response has 2+ class, (log reg usually used when only 2 class)(**Bayes Classifier**, cal. P(Y=1|X=x)…P(Y=K|X=x), assign to class w/ largest prob.) (**Gold std of class**, NB class has the best performance in class, IRL, we don't know the dist of predictors in each class, so computing is impossible, unattainable.) (**Idea of LDA**, assume predictor is norm dist, have equal var. Discriminant function, Decision boundary assume pi1=pi2=.5) (**QDA**, LDA assume all class have same var or var-covar mat in case of multiple predictors. LDA perform poorly if assumption is wrong. QDA similar to LDA, except est separate var-covar mat in case of multiple predictors, Discriminant function in quadratic form. **Which better**, QDA allow diff var %class, boundary become quadratic, **QDA** more flex, work best when **var very diff** %class, have enough observations to est the var; **LDA** work best when **var are similar** %class or not enough data to est the var.) (**KNN**, find k in train data that are closest to x0, cal the fraction, assign the observation to class with largest fraction. **Choice of K**, k=1, decision boundary is flex, k grows the bound become less flex, closer to linear, 1/k = lv of flexibility, chose min test err.) (**Log Reg vs LDA**, both linear boundaries, diff: 1.LDA assume norm dist with common var in each class; log reg does not. 2.LDA would do better if norm assumption holds, otherwise log reg better**. KNN vs …**, adv of KNN, KNN better when decision bound is highly nonlinear, Dis-adv, does not tell which prediction are important (no table of coeff). **QDA vs …**, compromise %non-parametric KNN and LDA, Log Reg. **Summary**: linear->LDA/Log Reg, moderately nonlinear->QDA, more complicated->KNN. **Performance**, **T +ve rate** =TP/P, **F +ve rate** (fraction of non-defaulters incorrectly identified as defaulters) =FP/N, **accuracy** (correctly classified) =(TN+TP)/(N+P), **err rate** (incorrectly classified) =(FN+FP)/(N+P) =1- accuracy, **sensitivity** (fraction of defaulters correctly identified as non-defaulters) =TP/P = T +ve rate, **specificity** (fraction of non-defaulters correctly identified) =TN/N =1- F +ve rate, **ROC**, **AUC. Threshold**, usually .5, affect the performance of class.) (**ROC**, ideal=top left corner, overall: AUC, plot by sensitivity/(1- specificity).)

$$P(Y = 1) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}}$$

#T4. Resampling (repeatedly drawing samples from a training set, refitting model on each sample set, refitting model to obtain more info about the fitted model. 1.**Model assessment** (est test err rate of a learning method when no test set) 2. **Model selection** (select model w/ appropriate lev of flex), method: Cross Validating, Bootstrapping.) (**CV**, train vs test err rate, test: avg err when stat learning method is use to pred the response of a new observation) (**Validating Set Approach**, rand split ava data into 2: train vs validation set. Use train set to build model, chose model by lowest test err rate with V set. **Adv**, simple. Easy to implement, **dis-adv**, V MSE can be highly var; only a subset of data is used to fit the model, perform worse when train on fewer data, V set err rate may overestimate the test err rate for the whole set.) (**LOOCV**, split data into train set n-1 and V set 1, repeat n times. **VS Validating Set Approach**, **Adv**, LOOCV less bias, almost entire data set is used, less var MSE, split based on one observation each time. **Dis-adv**, computationally intensive, repeat n times. Short-cut for Least SQ linear/ poly reg only) (**k-fold CV**, k=5/10, fit model w/ k-1 parts, repeat k times, taking out diff part each time, averaging the k diff MSE to get est V err rate. LOOCV is special case of KCV with k=n, both stable, no excessively high bias (V set), no vary high var (LOOCV). **CV usage**, if test data set is not ava, unknown teat err rate, apply k-fold CV, MSE from CV are est of test err rate, select best method by lowest test err (min pt. in est test MSE curve) from methods.) (**Bootstrap**, emulate the process of obtaining new sample sets, est variability of hat a w/o gen additional samples, by obtain distinct data sets by repeatedly sampling observations from the original data set, e.g., 123-> 313, 231, 221…, **Adv**, applied in almost all situations, no complicated maths cal.)

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$$

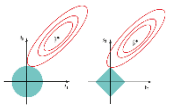
#T5. Linear Model Selection and Regularization (2 reasons for improving the OLS model: Prediction Accuracy, Model Interpretability) (**Prediction Accuracy**, least SQ est has relatively low bias low var when X Y are linear, #observations are way bigger than #predictors (n>>p), when $n \approx p$, OLS fit may have high var -> overfitting and poor est on new obs. When n<p, OLS does not work, the var of these est is inf so this method can't be use. **Model Interpretability**, for large #predictors, many of them have little/no impact on Y, easier to interpret by removing them (set coeff to 0). Solution: **Subset Selection**, subset of p predictors that related to Y then fit the model w/ the subset. Shrinkage, shrinking the est of coeff toward 0, reduce var, some coeff may shrink to exactly 0, preform var section, e.g., Ridge and LASSO. **Dim Reduction**, projecting p predictors onto an M-dim space where M<p and fit the reg model of Y on it, e.g., PCR.) (**Subset Selection**, chose subset with best pred accuracy. **Best Subset Selection**, run reg with each possible combo of predictors pCk for k=1…p, mode l0: model with 0 predicates, model *n*: *best* model with n predicates, then find best model from them. #Of possible model =2^p. **Stepwise Selection**, **Forward**, begin with 0, add 1 until no further improves, **Backward**, begin with all, delete 1, until no further improve. #Of possible model =1+p(p+1)/2, can be use when p is too large. No guarantee to yield the best model. **Backward vs. Forward**, backward requires n>p for full model to fit, forward can be use when n<p, only choice when p is very large.) (**Selecting the Single Best Model**, R^2 bad, p increase, R increase. Indirectly estimate the test err by making an adj to the train err to account for the bias due to overfitting account for the bias. Directly estimate the test err, using a V set approach or CV. **Approach 1**, add penalty to RSS, AIC, BIC, Cp, smaller better, Adjusted R^2, larger better. BIC, heavier penalty with many predictors, mainly for LR. **Approach 2**, V set err or CV err for each model, select one with smallest test err, **adv**, direct estimate of test err, fewer assumptions on the T underlying model.) (**Shrinkage Methods**, contains subset of the predictors through least squares, shrinks the coefficient estimates towards zero. **Ridge**, $\lambda \geq 0$ tuning parameter, add penalty to shrink large βs towards 0, $\lambda = 0$ -> OLS. **Why to 0**, OLS: low bias but high var, when n and p are similar, or n<p, OLS will highly var. Penalty term: increase bias, reduce var. **Adv**, only need to fit 1 model can be use when n<p. **LASSO**, some coefficients end up being set to exactly 0, produce model that high predictive power, simple to interpret. **Ridge vs LASSO**, Ridge when response depends on many predictors, all coeff roughly equal size. LASSO, relatively small #of predictors have substantial coefficients, remaining predictors have very small or 0 coeff. **Tuning Parameter**, by CV.)

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 \leq s \qquad \text{subject to } \sum_{j=1}^{p} |\beta_j| \leq s$$

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$

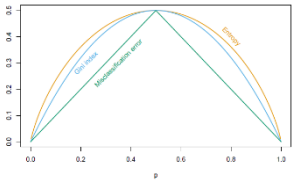
$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| = \text{RSS} + \lambda \sum_{j=1}^{p} |\beta_j|$$



Linear regression $f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$ Regression tree $f(X) = \sum_{m=1}^{M} c_m \cdot 1_{(X \in R_m)}$

#T6. Tree Based Methods (divide the predictor space into distinct regions, for each $\boldsymbol{X}$ that falls in a particular region make prediction using data in that region as training data. **Regression Tree**, split space into 2 regions, Xj>s, Xj<s, for all values of j and s by lowest RSS on the training data. Repeat until too few data to continue, e.g., all region has 5- points. **Values Used for Predictions**, for Rj, best prediction is simply the average of all the responses in the train data that fell in it. **Trees vs. Linear Models**, relationship between the predictors and response is linear, classical linear models e.g., linear regression will outperform regression trees, highly nonlinear and complex, decision trees. **Pros**, easy to explain, more closely mirror human decision-making, can be displayed graphically, easily handle qualitative predictors without the need to create dummy var. **Cons**, don't have the same prediction accuracy as some of the more complicated approaches, very non-robust (small change in the data, large change in the final estimated tree).) (**Classification Tree**, same procedure as in regression tree, use Criteria defined for classification instead of RSS, pmk=proportion of training data in the m th region that are from the k th class. Best prediction is the most commonly occurring class of training observations in that region. **True Pruning**, large tree may tend to overfit, small tree tends to have lower variance and better interpretation. Pruning, cutting off some of the terminal nodes. Grow a very large tree, prune it back, use CV to est test err. Choose subtree with lowest test err. **Improving Trees**, Bagging, Random forests, Boosting. **Bagging**, bootstrap aggregating, bootstrapping: lots of training datasets, averaging: reduces variance. Averaging a set of observations reduces variance. **Regression Trees**: Generate B different bootstrapped training datasets, Construct B regression trees using the training datasets, Average the resulting predictions. Tree not pruned; each individual tree has low bias with high variance. Averaging reduces variance -> lowering both variance and bias. Classification Trees, almost the same, but provide an overall prediction to the most commonly occurring one or average the prob. **Adv**, improves prediction accuracy. **Dis-adv**, hard to interpret, no longer possible represent the resulting statistical learning procedure, no longer clear which variables are most important to the procedure. **Random Forests**, de-correlates the trees. Build a number of decision trees on bootstrapped training sample, For each tree, each time a split in a tree is considered, a random sample of $\boldsymbol{m}$ predictors is chosen as split candidates from among the p predictorsm m=sqrt(p). Bagged trees will look similar, predictions from the bagged trees will be highly correlated. Averaging many highly correlated quantities, more reduction in variance. **Boosting**, grows trees sequentially using info from previously grown trees, does not involve bootstrap. Boosting learns slowly and tends to perform well.)

$$E = 1 - \max_k(\hat{p}_{mk}) \qquad G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \qquad D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$



#T7. Support Vector Machines (class, Hyperplane: 2D, then a line. Feature space, formed by the predictors, p-dim, n-pt. Training data, inputs x, output y, determine classes. Find the hyperplane such that a test point is assigned the correct class. If y*f(x*) is far from 0, test pt lies far from the hyperplane, close to 0, located near HP, less certain which class to assign. **Margin**, perpendicular distance, from every point, smallest of such distances. **Maximal Margin Classifier**, linear boundary, separable, Best separating HP, max min-distance. **Support Vectors**, support vectors, support max margin hyperplane, if they were moved slightly moved slightly then the max margin HP would move as max margin HP would move as well, max margin HP depends directly on the support vectors. **Non-separable Case**, No solution for MMC if the data classes are mixed.

$$\text{class } 1\text{: } f(x) > 0 \qquad \text{class } -1\text{: } f(x) < 0$$

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \; M$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \; \sum_{i=1}^{n} \epsilon_i \leq C,$$

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}$$

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d$$

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$$

Soft margin: almost separate. **Support Vector Classifier**, linear boundary, separable or non non-separable cases, Consider a hyperplane that does not perfectly separate the two does not perfectly separate the two classes, in the interest of Greater robustness and Better classification. Misclassify a few training points for better classifying the remaining points. Soft margin classifier: the margin is soft because it can be violated by some of the training points. $\varepsilon = 0$ margin not violated, correct side, $\varepsilon > 0$, violated, wrong side, $\varepsilon > 1$, boundary crossed. C: tuning parameter, small: narrow margin, raely violated, fit well, low bias, high var, large: wide margin, more violate, fit less hard, low var, high bias. SV: pt. lies on margin or on the wrong side or margin, on the wrong side of the HP, data lie on the correct side will not affect the SVC. **Support Vector Machine**, linear or nonlinear boundary, separable or non non-separable cases, allows us to enlarge the feature space in a way that leads to efficient computations, Decision rule is based on the inner product, K is called the kernel function, Linear kernel, Polynomial kernel, Radial kernel.) **#T8. Principal Components Regression**(**PCA**, unsupervised approach. Dimension reduction, low-dim representation of the data that captures as much of the information as possible; Data visualization, p variables -> uncorrelated principal components PC1 PC2. With p-dimensional feature dimensional feature space not all directions are equally interesting. PCA seek small number of dimensions as interesting measured by variancevarianc as possible. Each PC is a linear combination of the p features. **Properties**, The principal components are uncorrelated, ordered according to the decreasing variance, can be used in further supervised learning. **Usage**, Scaling: scale each variable to have standard deviation 1 before performing PCA. Uniqueness, each PC loading vector is unique (Flipping the sign has no effect). **Proportion of variance explained**, info in a data set lost by projecting the data onto the PCs. Total of PCs=min(n-1,p). PVEs of all PCs sum to 1. **#Of PC**, smallest number of PCs required to explain a sizable amount of the variation in the data. **Principal Components Regression**, Use the selected PCs as the predictors in a linear regression selected PCs as the predictors in a linear regression model fit using least squares. PCR is not a feature selection method, selecting PCs by cross validation by CV, it works well when the first few PCs are sufficient to capture most of the variation in the predictors as well as the relationship with the response.