Machine Learning

PATRICK HALL

DEPARTMENT OF DECISION SCIENCE

Course Overview

Machine Learning

As a follow up course to Data Mining (DNSC 6279) that will expand on both the theoretical and practical aspects of subjects covered in the pre-requisite course while optionally introducing new materials. Techniques covered may include feature engineering, penalized regression, neural networks and deep learning, ensemble models including stacked generalization and super learner approaches, matrix factorization, model validation, and model interpretation.

Classes will be taught as workshops where groups of students will apply lecture materials to the ongoing Kaggle Advanced Regression and Digit Recognizer contests.

Course Information & Grading Policy

- Course Syllabus
  - https://github.com/jphall663/GWU_data_mining
  - Blackboard
- Recommended Textbook
  - Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani, and Jerome Friedman
  - Pattern Recognition and Machine Learning by Christopher Bishop
  - A Primer on Scientific Programming with Python by Hans Petter Langtangen
- Grading Policy
  - In-Class Participation: 1/3
  - Kaggle Performance: 1/3
  - Public Github Contribution: 1/3

Software

- Anaconda Python
- H2o.ai
- R & R Studio
- SAS 9.4 and Enterprise Miner
- TensorFlow + Keras
- XG Boost

Tentative Course Outline

- Week 1: Linear Models
  - Penalized Linear & Logistic Regression
  - Python and H2o.ai
- Week 2: Neural Networks
  - Multi-Layer Perceptrons and Autoencoders
  - Python and H2o.ai
- Week 3: Deep Learning (Wen Phan, Guest Lecturer)
  - Convolutional Neural Networks
  - TensorFlow, Keras, & H2o.ai Deep Water
- Week 4: Decision Tree Ensembles
  - Bagging (Random Forest), Boosting (GBM), & Stacked Ensembles
  - H2o.ai and XGBoost
- Week 5: Matrix Factorization
  - SVD, PCA, GLRM, & NMF
  - H2o.ai and Python

- Week 6: Embedding Methods
  - Entity Embedding Neural Networks
  - Factorization Machines

OR

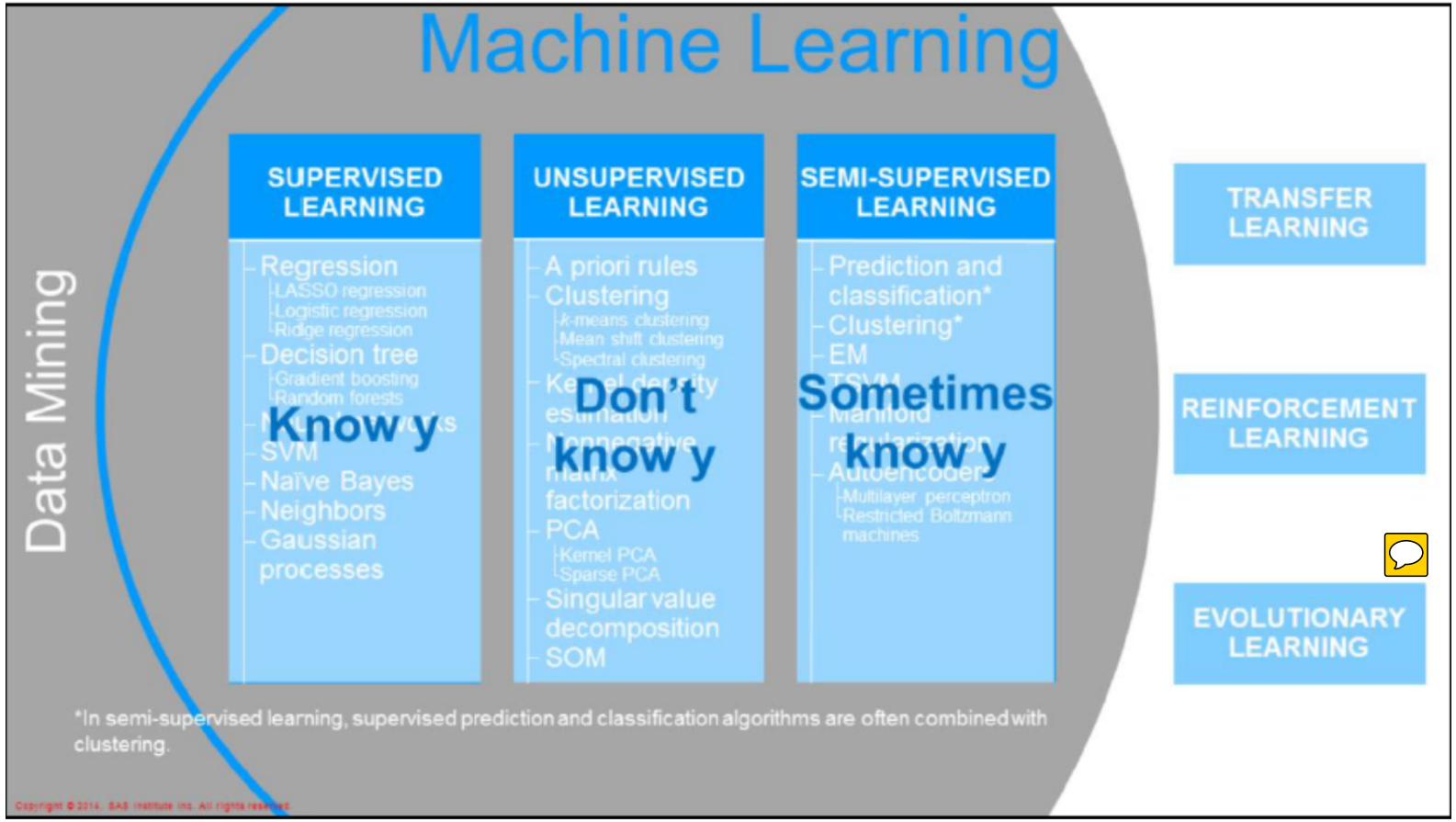
- Week 6: Model Interpretation
  - LIME and LOCO
  - Partial Dependence and ICE

Reminder/Suggestions

- Grading Policy
  - In-Class Participation: 1/3 – possible class sign-in sheet since class-participation is significant part of grade (for the first 2 classes; I am planning attending class sessions for week 3-6; maybe I can take attendance on my end for this?)
  - Kaggle Performance: 1/3 – individual or group for this class? If thinking about group-based project performance then ~3 per group is optimal (based on my group experience) & possible peer-evaluation form at the end of the course?
  - Public Github Contribution: 1/3 – should be more clear on this part

Class Introduction

Machine Learning: Description



Contents

- Background: Linear Model – ESL, CH 3
- Gradient Descent Optimization
- Background: Logistic Model – ESL, CH 4
- Penalized Regression
- Cross-Validation and Parameter Tuning
- Ensemble Models

Linear Regression

Carl Friedrich Gauss
(1777–1855)

Regression: Linear Method

- A traditional regression model:

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j.$$

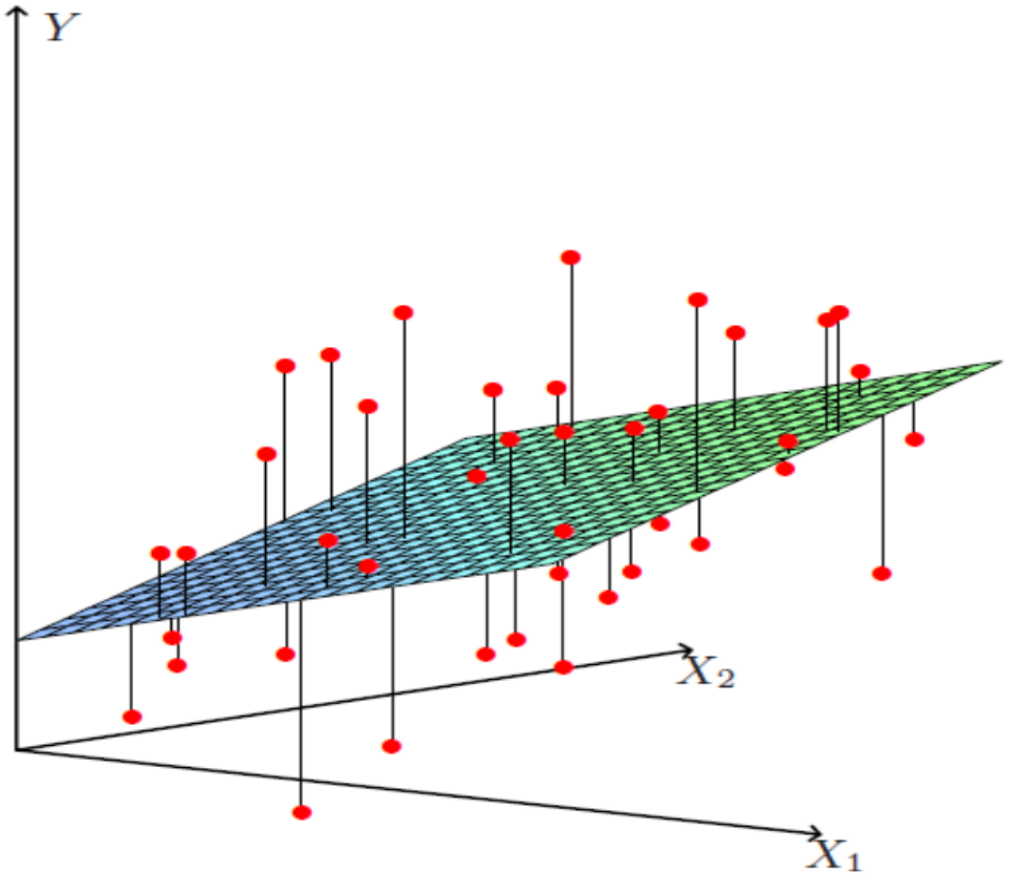
- Basic Model Evaluation: Least-Square Method

$$\begin{aligned} \mathrm{RSS}(\beta) &= \sum_{i=1}^{N} (y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \Big)^2. \end{aligned}$$

- Advantages
  - Simple and ***interpretable*** model description
  - Sometimes outperforms non-linear or more complex models with small number of training cases, low signal-to-nose ratio or sparse data
  - Apply transformation to expand the scope of the model

Regression: Least-Squared Method

Figure 3.1
Visualization of linear least-square fitting that minimizes the sum of squared residuals (SSR) from Y (target)



Gradient Descent Example

Logistic Regression

Regression: Logistic Method

- The logistic regression model arises from the desire to model the posterior probabilities of ***k*** classes via linear function X, while at the same time ensuring that they sum to one and remain in [0,1].
- The logistic model has the form:

$$\log \frac{\Pr(G=1|X=x)}{\Pr(G=K|X=x)} = \beta_{10} + \beta_1^T x$$
$$\log \frac{\Pr(G=2|X=x)}{\Pr(G=K|X=x)} = \beta_{20} + \beta_2^T x$$
$$\vdots$$
$$\log \frac{\Pr(G=K-1|X=x)}{\Pr(G=K|X=x)} = \beta_{(K-1)0} + \beta_{K-1}^T x.$$

Where, the model is specific in terms of K-1 log-odds or logit transformation.

Regression: Logistic Method

- Further calculations show that:

$$\Pr(G = k|X = x) = \frac{\exp(\beta_{k0} + \beta_k^T x)}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \beta_\ell^T x)}, \; k = 1, \ldots, K-1,$$

$$\Pr(G = K|X = x) = \frac{1}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \beta_\ell^T x)}, \qquad \text{(logit function)}$$

- To demonstrate the dependence on the entire parameter set $\theta = \{\beta_{10}, \beta_1^T, \ldots, \beta_{(K-1)0}, \beta_{K-1}^T\}$, we denote the probabilities $\Pr(G = k|X = x) = p_k(x; \theta)$.

- When K=2, above model reduces to a single linear function where it is widely used in situations where binary responses occur frequently, i.e. – patients survive or die or have heart diseases or not.

Fitting Logistic Regression Models

- Let's consider two-class (binary) case:
- Then the two-class g_i via a 0/1 response y_i *where* $y_i = 1$ when $g_i = 1$ *and* $y_i = 0$ when $g_i = 2$.
- Not, let $p_1(x;\theta) = p(x;\theta)$ and $p_2(x;\theta) = 1 - p(x;\theta)$
- The log-likelihood can be expressed as:

$$\begin{aligned}\ell(\beta) &= \sum_{i=1}^{N}\Big\{y_i \log p(x_i;\beta) + (1-y_i)\log(1-p(x_i;\beta))\Big\} \\ &= \sum_{i=1}^{N}\Big\{y_i\beta^T x_i - \log(1+e^{\beta^T x_i})\Big\}.\end{aligned}$$

where, $\beta = \{\beta_{10}, \beta_1\}$. and we assume that the vector of input x_i include the constant term 1 to accommodate the intercept.

Regression: Issues

- Predictive Accuracy - least square estimates often have low bias but large variance
  - Prediction accuracy can sometimes be improved by shrinkage or setting some coefficients to zero – i.e. add bias to reduce the variance to improve the overall prediction accuracy
- Interpretation
  - Often try to determine the smaller subset that exhibit the strongest effect out of a large number of predictors (parameters) – i.e. sacrifice detail to find the general underlying dynamic

Regression: Issues

Requirements	If broken …
Linear relationship between inputs and targets; normal y, normal errors	Inappropriate application/unreliable results; use a machine learning technique; use GLM
N > p	Underspecified/unreliable results; use LASSO or elastic net penalized regression
No strong multicollinearity 	Ill-conditioned/unstable/unreliable results; Use ridge(L2/Tikhonov)/elastic net penalized regression
No influential outliers	Biased predictions, parameters, and statistical tests; use robust methods, i.e. IRLS, Huber loss, investigate/remove outliers
Constant variance/no heteroskedasticity	Lessened predictive accuracy, invalidates statistical tests; use GLM in some cases
Limited correlation between input rows (no autocorrelation)	Invalidates statistical tests; use time-series methods or machine learning technique

Penalized Regression

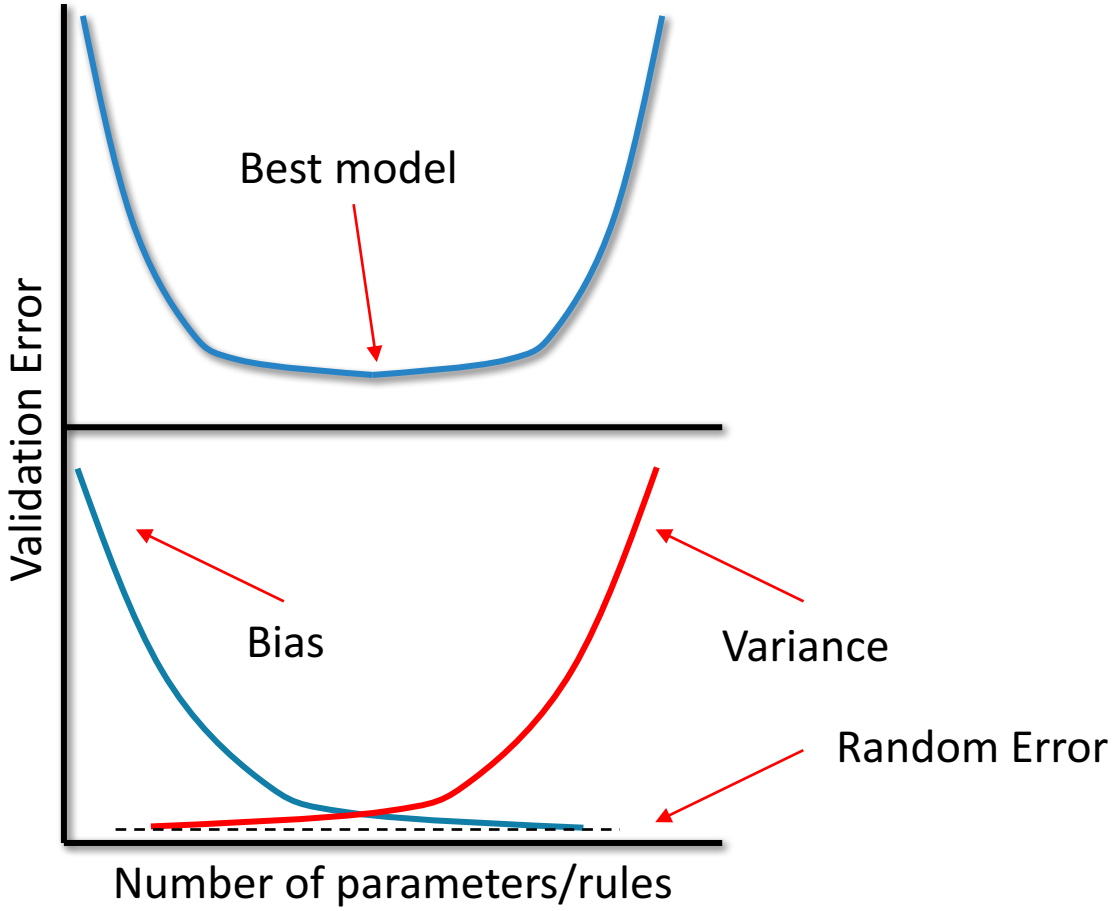
Penalized Linear Regression

- Consider the mean squared error (MSE) of an estimator $\tilde{\theta}$ in estimating θ :

$$\begin{aligned} \mathrm{MSE}(\tilde{\theta}) &= \mathrm{E}(\tilde{\theta} - \theta)^2 \\ &= \mathrm{Var}(\tilde{\theta}) + [\mathrm{E}(\tilde{\theta}) - \theta]^2. \end{aligned}$$

  where the first term is the variance and second term is the squared bias
- The least squares estimator has the smallest MSE of all linear estimators with no bias
- However, there may exists a biased estimator with smaller MSE – little bias for a larger reduction in variance
- Hence any method that shrinks or sets some of the least square coefficients to zero may result in a biased estimate – variable subset selection and ridge selection

The Bias / Variance Trade-off



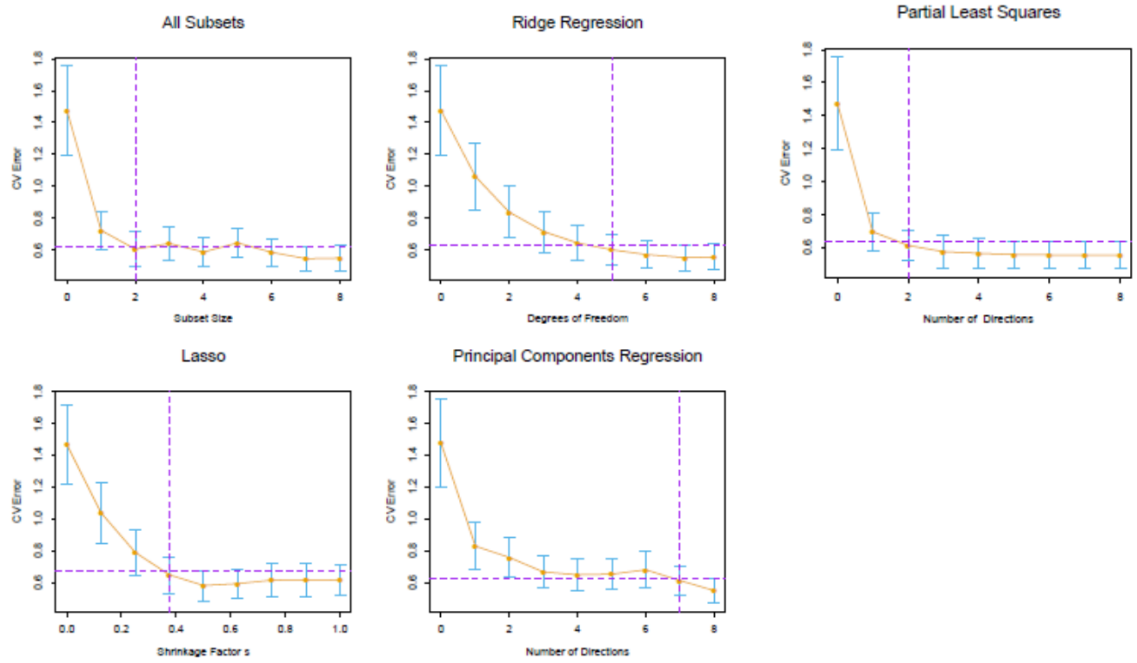
Total Error = Bias + Variance + Random Error = $(\hat{f}(x) - f(x))^2$

Bias = $E[\hat{f}(x)] - f(x)$ or the error that arises from a model's inability to replicate the fundamental phenomena represented by a data set.

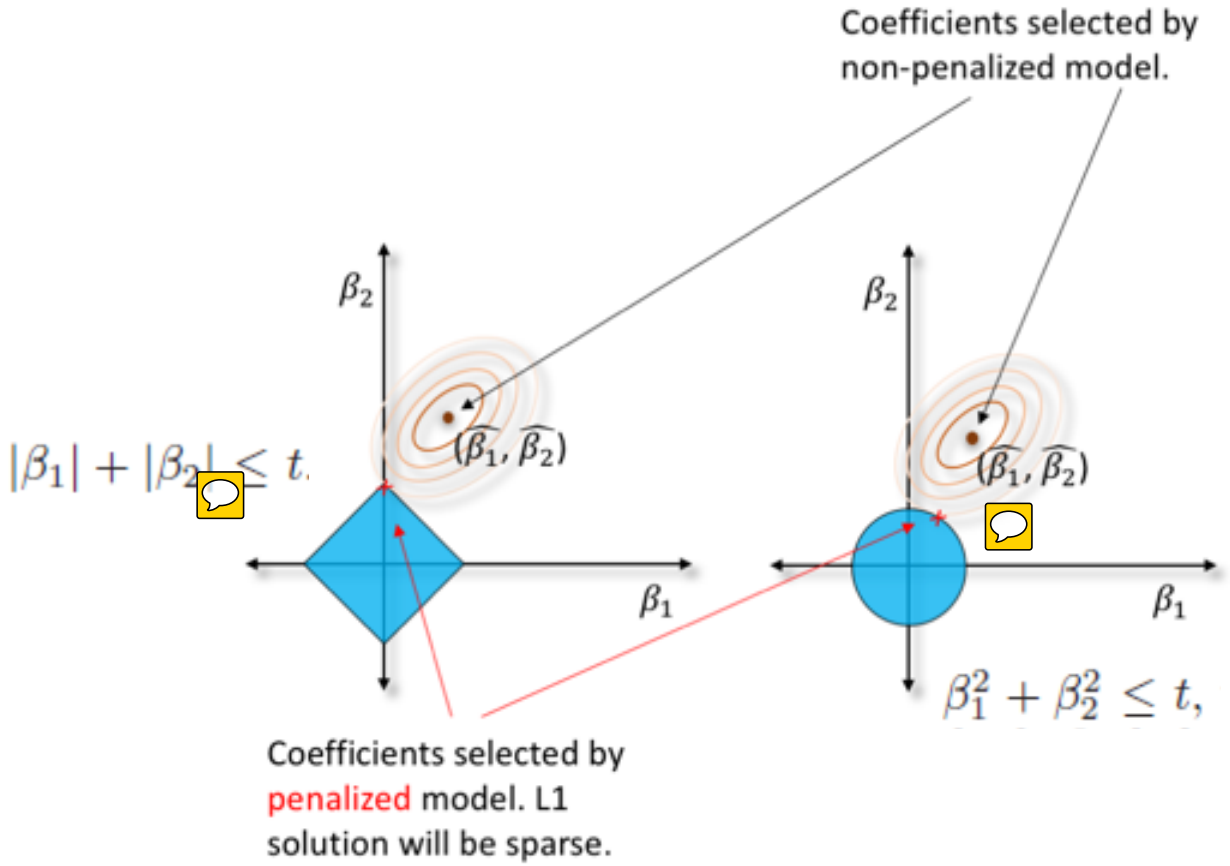
Variance = $(\hat{f}(x) - E[\hat{f}(x)])^2$ or the error that arises from a model's ability to produce differing predictions from the values in a new data set.

Shrinkage Methods

- Subset method is a discrete process, hence exhibits high variance and does not reduce the prediction of error
- Whereas shrinkage methods are more continuous – don't suffer as much high variability
- Types
  - Ridge Regression
  - Lasso Regression
  - Principal Component Regression
  - Partial Least Square Regression



Shrinkage/Regularization Method



Shrinkage Method

- Ridge Regression
  - Shrinks the regression coefficient by imposing a penalty on their size by minimizing a penalized residual sum of squares :

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}.$$

- LASSO – in the equivalent Lagrangian form:

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}.$$

Generalized Linear Model

Generalized Linear Model

- GLM is a flexible generalization of ordinary linear regression that allows for response variables that have error distribution models other than normal distributions

- GLM generalizes linear regression by allowing the linear models to be related to the response variables by a link function and b allowing the magnitude of the variance of each measurement to be a function of its predicted value

- Three components:
  - Probability distribution from the exponential family
  - A linear predictor $\eta = X\beta$
  - A link function E(Y) = $\mu = g^{-1}(\eta)$

Anatomy of a GLM: Link Function

Family/distribution defines mean and variance of **Y**

$$E(\boldsymbol{Y}) = \mu = g^{-1}(\boldsymbol{X}\beta)$$

Linear component

Nonlinear link function between linear component and E(**Y**)

$$Var(\boldsymbol{Y}) = V(\mu) = V(g^{-1}(\boldsymbol{X}\beta))$$

Family/Distribution allows for non-constant variance

Family of Distributions

- **Gaussian** distribution, squared error loss, sensitive to outliers
- **Laplace** distribution, absolute error loss, more robust to outliers
- **Huber** loss, hybrid of squared error & absolute error, robust to outliers
- **Poisson** distribution (e.g., number of claims in a time period)
- **Gamma** distribution (e.g., size of insurance claims)
- **Tweedie** distribution (compound Poisson-Gamma)
- **Binomial** distribution, log-loss for binary classification

Modern Approaches – Elastic Net

Hui Zou and Trevor Hastie
Regularization and variable selection via the elastic net,
Journal of the Royal Statistical Society, 2005

Modern Approaches – Elastic Net

λ - Controls magnitude of penalties. Variable selection conducted by refitting model many times while varying λ. Decreasing λ allows more variables in the model.

L2/Ridge/Tinkhonov Penalty – helps address multicollinearity.

L1/LASSO penalty – for variable selection.

$$\tilde{\beta} = \min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} * \beta_j \right)^2 + \lambda \sum_{j=1}^{p} (\alpha * \beta_j^2 + (1 - \alpha) * |\beta_j|) \right\}$$

Least squares minimization – finds β's for linear relationship.

α - tunes balance between L1 and L2 penalties.

Modern Approaches – Iteratively Reweighted Least Squares

Iteratively Reweighted Least Square complements fitting methods in the presence of the outliers by:

- Initially giving all observations equal weight then…
  - Train the model to estimate the β's and find a linear relationship/equation

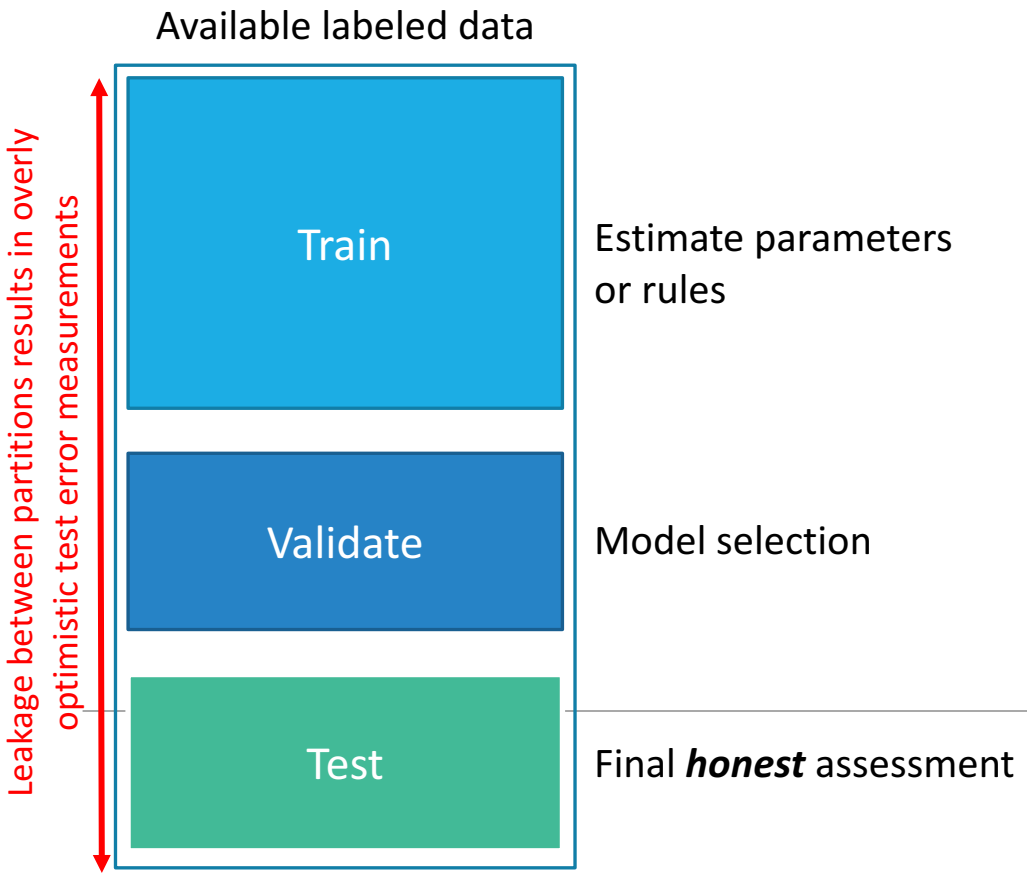
$$\tilde{\beta} = \min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} * \beta_j \right)^2 \right\} \leftarrow \text{"Inner Loop"}$$

  - Calculate the residuals given these β's/ linear equation
  - Re-weight observations that cause high residuals to have a lower impact in the train model
  - Re-train to find new β's/linear equation
  - Continue calculating residuals, re-weighting observations, and re-training until β's become stable and weighted residuals are small…

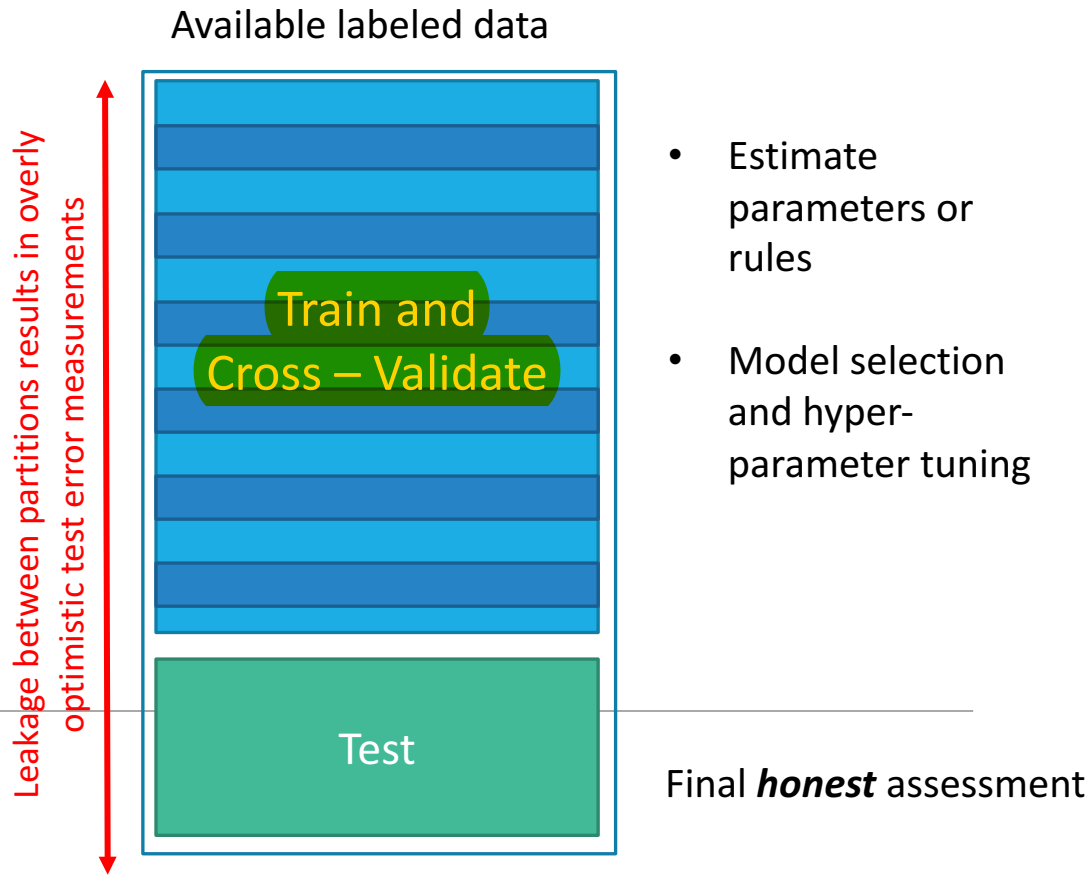
"Outer Loop"

Cross-Validation & Parameter Tuning

Bias/Variance Trade-off in Practice: Honest assessment



Best suited for big data or linear models using traditional forward, backward, or stepwise selection.



- Estimate parameters or rules
- Model selection and hyper-parameter tuning

Nearly always a more generalizable approach, but computationally intensive. Best suited for complex models with many hyper-parameters and small to medium sized data.

Parameter Tuning

Outer most loop(s):

- λ search from λ_{max} (where all coefficients = 0) to $\lambda = 0$
- Grid search on alpha usually not necessary
  - Just try a few: 0, 0.5, 0.95
  - Always keep some L2
  - Set max_predictors, large models take longer
- Models can also be validated:
  - Validation and test partitioning available
  - Cross-validated (k-fold for extra validation, CV predictions available for stacking)

Ensemble Models

Ensemble Models

- Ensemble models – method for combining the posterior probability of two or more predictive models to create a potentially more accurate model
- Ensemble Methods
  - Simple Averaging
  - Top-*t* ensemble selection
  - Hill Climbing ensemble Selection
  - Weighted Averaging
  - Stacking
  - Cluster-based selection

Ensemble Models

- Simple Averaging (simple soft voting) – takes the average of the posterior probability for each response level across the models and then classifies the model based on the level that has the maximum average probability

- Top-*t* Ensemble Selection – take the top *t* models out the M that are generated when the models are ranked by an accuracy measure and uses validation data to determine the best value for *t*
  - Similar to weighted averaging but here, equal weights are assigned to a subset of the available models

- Hill-climbing Ensemble Selection – improvement in the accuracy of adding any given model, i.e. model that most improves the misclassification rate in the validation set. The final ensemble is a selected based on the misclassification rate in the test set
  - Similar to weighted averaging, but here, different weight are assigned to each model depending on how many times a particular model is included in the ensemble

Ensemble Models

- Weighted Averaging – weighted average of the posterior probability for each response level is considered with a model-specific weight applied

- Stacking – uses posterior probability from the various models that are used as inputs and the original response (target) variable that is used as the response
  - Can use linear regression model to generate the weights for a weighted averaging ensemble
  - Can also implement penalized linear regression – LASSO, ridge, and elastic net
  - Other models can also be implemented – decision tree and random forest

- Clustering-based selection – uses a principle-component-based variable clustering algorithm to cluster posterior probabilities that are similar and uses simple averaging to choose the best model from each cluster to combine into an ensemble

Supplemental Materials

Subset Selection: Traditional Methods

- Best-Subset – Subset of size k that gives the smallest residual sum of squares (RSS)
- Forward-stepwise
  - Greedy Algorithm
  - Computational (p>>N) and Statistical (variance-bias trade-off)
- Backward-stepwise
  - N>p
  - Start with full model and sequentially delete parameter(s) that has the least impact on the fit

Subset Selection

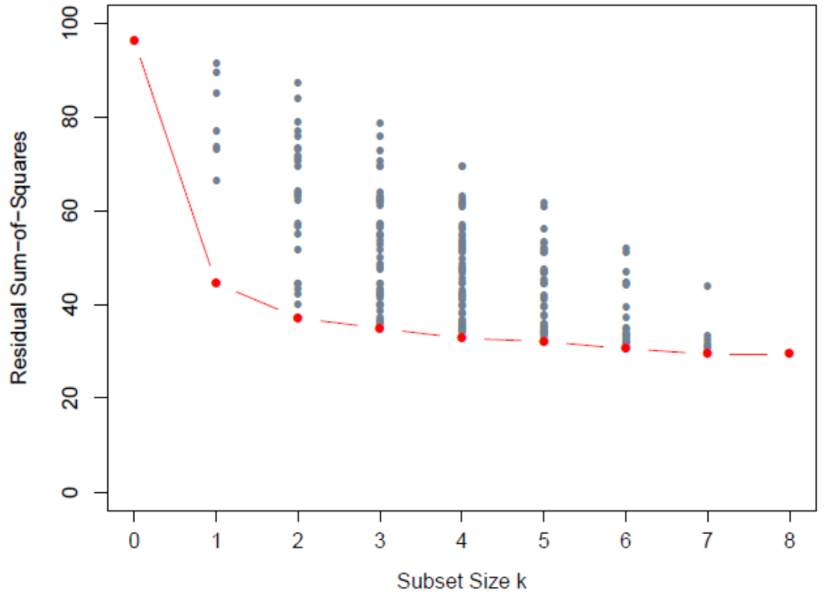


Figure 3.5
All possible subset model for the prostate cancer example

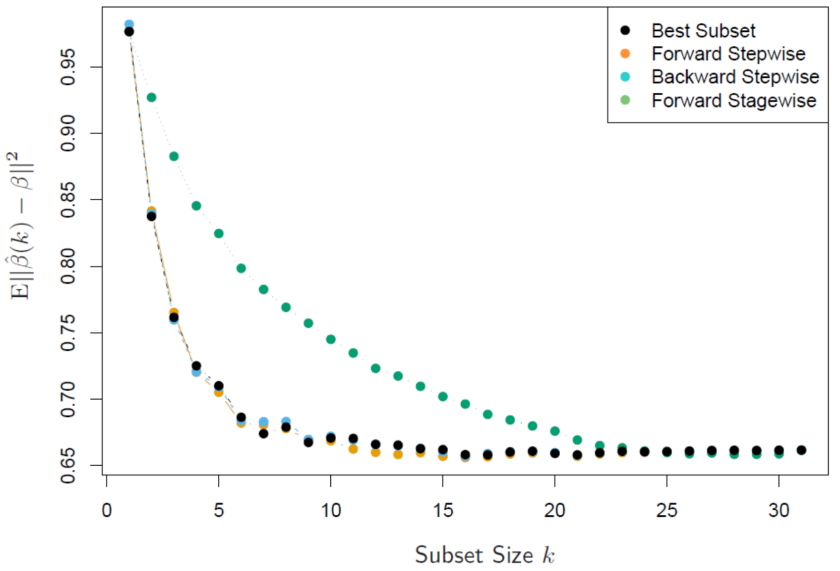


Figure 3.6
N=300 and p=31 (standard real number variables)

Least Angle Regression

Algorithm 3.2 *Least Angle Regression.*

1. Standardize the predictors to have mean zero and unit norm. Start with the residual $\mathbf{r} = \mathbf{y} - \bar{\mathbf{y}}$, $\beta_1, \beta_2, \ldots, \beta_p = 0$.

2. Find the predictor $\mathbf{x}_j$ most correlated with $\mathbf{r}$.

3. Move β_j from 0 towards its least-squares coefficient $\langle \mathbf{x}_j, \mathbf{r} \rangle$, until some other competitor $\mathbf{x}_k$ has as much correlation with the current residual as does $\mathbf{x}_j$.

4. Move β_j and β_k in the direction defined by their joint least squares coefficient of the current residual on $(\mathbf{x}_j, \mathbf{x}_k)$, until some other competitor $\mathbf{x}_l$ has as much correlation with the current residual.

5. Continue in this way until all p predictors have been entered. After $\min(N - 1, p)$ steps, we arrive at the full least-squares solution.

Fitting Logistic Regression Models

- To maximize, set the derivative of the log-likelihood function to zero and solve.
- The ***score equation*** are:

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^{N} x_i (y_i - p(x_i; \beta)) = 0,$$

  Which are p+1 equations nonlinear in β
- To find solution(s) of score equation, use Newton-Raphson algorithm (second derivative):

$$\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = -\sum_{i=1}^{N} x_i {x_i}^T p(x_i; \beta)(1 - p(x_i; \beta)).$$

Fitting Logistic Regression Models

- Logistic equations are typically fit by a maximum likelihood function criterion using conditional likelihood of G given X.
- The conditional likelihood of G given X, Pr(G|X) completely specifies the conditional distribution – the multinomial
- The log-likelihood for N observation is given by:

$$\ell(\theta) = \sum_{i=1}^{N} \log p_{g_i}(x_i; \theta),$$

where $p_k(x_i; \theta) = \Pr(G = k | X = x_i; \theta)$

Fitting Logistic Regression Models

- A single Newton update is $\beta^{\text{new}} = \beta^{\text{old}} - \left(\frac{\partial^2 \ell(\beta)}{\partial\beta\partial\beta^T}\right)^{-1} \frac{\partial \ell(\beta)}{\partial\beta},$

- Where the derivatives are evaluated at β^{old}.
- For convenience, write the score and Hessian (second derivative) in a matrix format

$$\frac{\partial \ell(\beta)}{\partial\beta} = \mathbf{X}^T(\mathbf{y} - \mathbf{p})$$
$$\frac{\partial^2 \ell(\beta)}{\partial\beta\partial\beta^T} = -\mathbf{X}^T\mathbf{W}\mathbf{X}$$

Fitting Logistic Regression Models

- The Newton step is thus

$$\begin{aligned}\beta^{\text{new}} &= \beta^{\text{old}} + (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{y}-\mathbf{p}) \\ &= (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\left(\mathbf{X}\beta^{\text{old}} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})\right) \\ &= (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{z}.\end{aligned}$$

- Where the second and third step is re-expressed as a weighted least square step with the response (adjusted response)

$$\mathbf{z} = \mathbf{X}\beta^{\text{old}} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})$$

- These equations get solved repeatedly (note at each iteration, **p** changes hence **W** and **z**) – ***Iteratively reweighted least squares (IRLS)***

$$\beta^{\text{new}} \leftarrow \arg\min_{\beta}(\mathbf{z}-\mathbf{X}\beta)^T\mathbf{W}(\mathbf{z}-\mathbf{X}\beta).$$

Note, β typically starts at 0 for the iterative procedure, although convergence is not guaranteed. Typically, the algorithm does converge (concavity of the log-likelihood) with possible overshooting.

Model Averaging

- Bayesian Model Averaging
- Let $\mathcal{M}_m,\ m = 1, \ldots, M$ be a set of candidate models for the training set **Z** (these models may have same or different parameter value or different models for the same task).
- Now, suppose ζ is some quantity of interest (prediction *f(x)* at some fixed value of *x*). Then posterior distribution of ζ is given by

$$\Pr(\zeta|\mathbf{Z}) = \sum_{m=1}^{M} \Pr(\zeta|\mathcal{M}_m, \mathbf{Z}) \Pr(\mathcal{M}_m|\mathbf{Z})$$

with the posterior mean

$$\mathrm{E}(\zeta|\mathbf{Z}) = \sum_{m=1}^{M} \mathrm{E}(\zeta|\mathcal{M}_m, \mathbf{Z}) \Pr(\mathcal{M}_m|\mathbf{Z}).$$

- Essentially, Bayesian prediction is a weighted average of the individual predictions with weights proportional to the posterior probability of each model.

Model Averaging

- Committee Method – simple unweighted average of the prediction of each model
  - BIC Criterion – applicable in cases where different models arise from the same parametric models with different parameter values; BIC gives different weight to each model depending on how well it fits and how many parameter it use
  - Frequentist approach – prediction with square-error loss method, seek the weights such that

$$\hat{w} = \underset{w}{\operatorname{argmin}} \, \mathrm{E}_{\mathcal{P}} \left[Y - \sum_{m=1}^{M} w_m \hat{f}_m(x) \right]^2$$

- Population Linear regression of **Y** on $\hat{F}(x)^T \equiv [\hat{f}_1(x), \hat{f}_2(x), \ldots, \hat{f}_M(x)]$:

$$\hat{w} = \mathrm{E}_{\mathcal{P}}[\hat{F}(x)\hat{F}(x)^T]^{-1} \mathrm{E}_{\mathcal{P}}[\hat{F}(x)Y].$$

Model Averaging

- Full regression has smaller error than any single model at the population level

$$\mathrm{E}_{\mathcal{P}}\left[Y - \sum_{m=1}^{M} \hat{w}_m \hat{f}_m(x)\right]^2 \leq \mathrm{E}_{\mathcal{P}}\left[Y - \hat{f}_m(x)\right]^2 \; \forall m$$

Model Averaging

- Stacked Generation (stacking) – stacking estimation of the weights is obtained from the least squares linear regression of y_i on $\hat{f}_m^{-i}(x)$ for m=1,2,…,M.
- Stacking weights are given by

$$\hat{w}^{\mathrm{st}} = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{N} \left[y_i - \sum_{m=1}^{M} w_m \hat{f}_m^{-i}(x_i) \right]^2$$

- Where the final prediction is $\sum_m \hat{w}_m^{\mathrm{st}} \hat{f}_m(x)$.

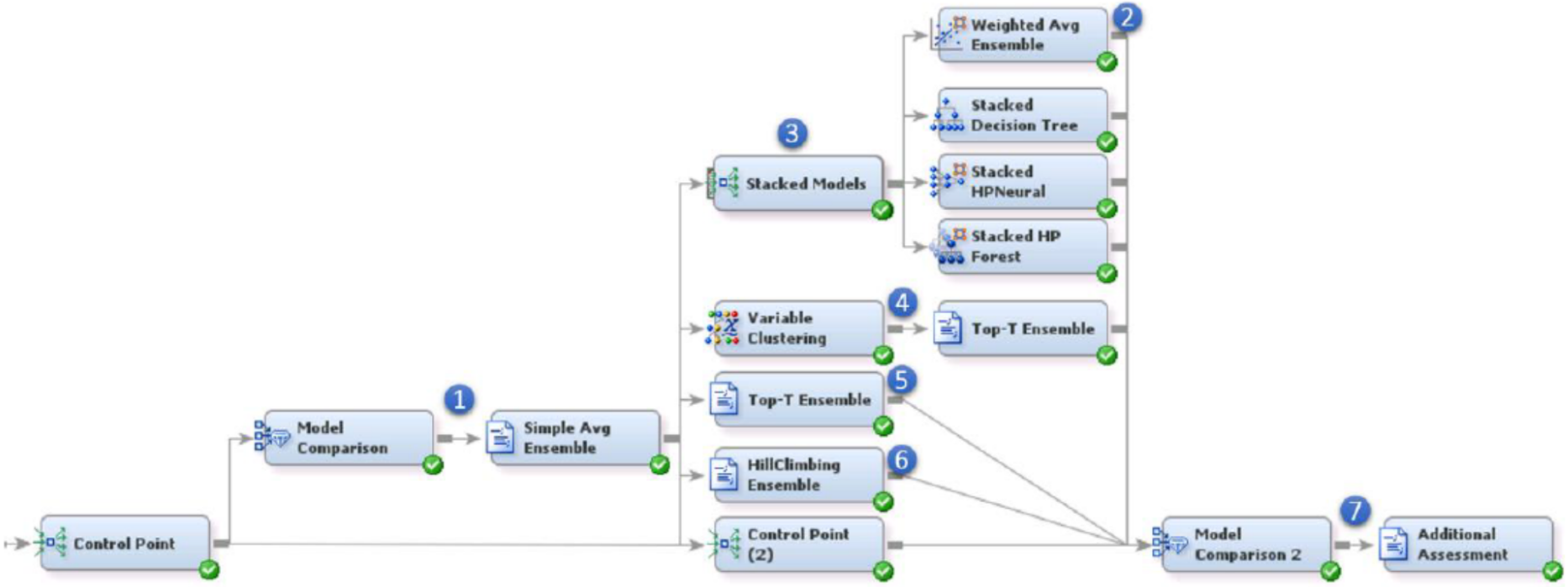
- Note, the cross-validation prediction on $\hat{f}_m^{-i}(x)$ avoids giving unfairly high weight to modes with higher complexity

Ensemble Models

- Ensemble models – method for combining the posterior probability of two or more predictive models to create a potentially more accurate model
- Ensemble Methods
  - Simple Averaging
  - Top-*t* ensemble selection
  - Hill Climbing ensemble Selection
  - Weighted Averaging
  - Stacking
  - Cluster-based selection

Ensemble Models

- Diagram Flow of Ensemble Methods



Ensemble Models

- Simple Averaging (simple soft voting) – takes the average of the posterior probability for each response level across the models and then classifies the model based on the level that has the maximum average probability

- Top-*t* Ensemble Selection – take the top *t* models out the M that are generated when the models are ranked by an accuracy measure and uses validation data to determine the best value for *t*
  - Similar to weighted averaging but here, equal weights are assigned to a subset of the available models

- Hill-climbing Ensemble Selection – improvement in the accuracy of adding any given model, i.e. model that most improves the misclassification rate in the validation set. The final ensemble is a selected based on the misclassification rate in the test set
  - Similar to weighted averaging, but here, different weight are assigned to each model depending on how many times a particular model is included in the ensemble

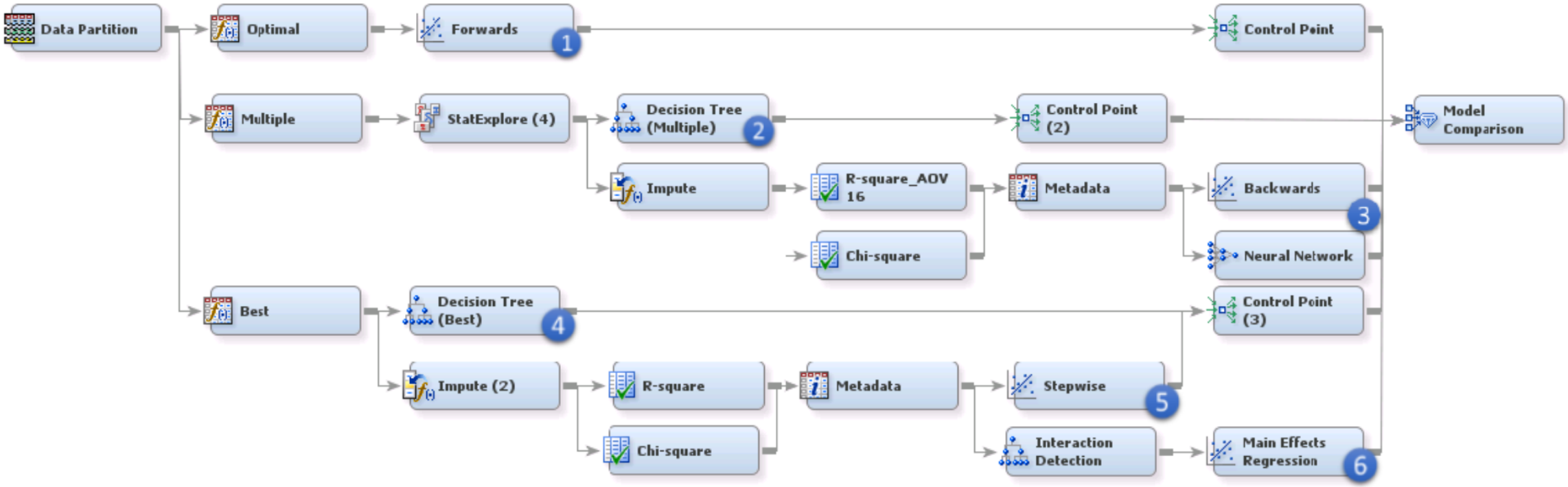
Ensemble Models

- Weighted Averaging – weighted average of the posterior probability for each response level is considered with a model-specific weight applied

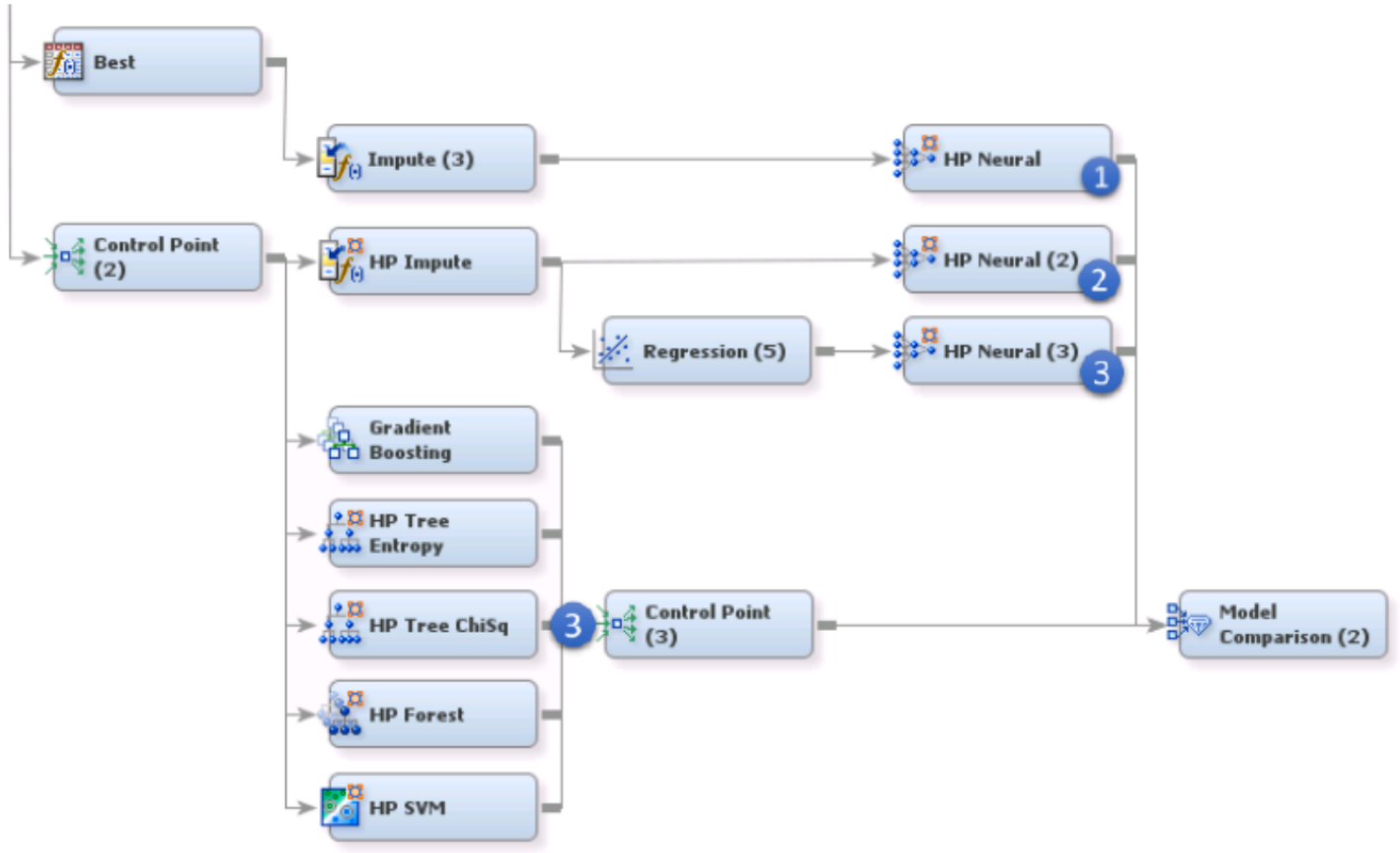
- Stacking – uses posterior probability from the various models that are used as inputs and the original response (target) variable that is used as the response
  - Can use linear regression model to generate the weights for a weighted averaging ensemble
  - Can also implement penalized linear regression – LASSO, ridge, and elastic net
  - Other models can also be implemented – decision tree and random forest

- Clustering-based selection – uses a principle-component-based variable clustering algorithm to cluster posterior probabilities that are similar and uses simple averaging to choose the best model from each cluster to combine into an ensemble

Ensemble Models – Rapid Predictive Modeler



Ensemble Models – Common Practice



Ensemble Model – Comparison

Misclassification Rates for Best-Performing Models and Ensembles

	RPM-Based Model		Common Practice Model		Ensemble Model		
Data Set	Model	Test Misclassification Rate	Model	Test Misclassification Rate	Ensemble	Test Misclassification Rate	Misclassification Improvement (%)
Home Equity	Stepwise Regression	0.1034	Gradient Boosting	0.1006	Stacked Forest	0.0872	13.33%
German Credit	Decision Tree (best transformation)	0.2525	HPSVM	0.2259	Top-T	0.2326	-2.94%
Give Me Credit	Stepwise Regression	0.0640	HPNeural (2-way interactions)	0.0635	Top-T	0.0634	0.17%
PAKDD	Neural Network	0.2606	Several models	0.2609	Top-T	0.2608	-0.08%
Australian	Decision Tree (multiple transformations)	0.1531	HPNeural (best transformation)	0.1435	VarClus Top-T	0.1340	6.67%