Lecture 12: Course overview

Alan Lee

Department of Statistics
STATS 784 Lecture 12

September 1, 2017

Outline

Today's agenda

In this lecture we present an overview of the material we have covered in the last 6 weeks.

The data mining process

- Definitions: size, information discovery, actionable insights
- Bias still can be a problem
- CRISP-CM, SEMMA
- Multidisciplinary skills
- Statistical learning (prediction, classification, unsupervised learning)

Prediction: general principles

- Data follows model $y = f(x) +$ error relating target y to features x
- Choose predictor $\hat{f}$ from some class of functions (e.g. linear)
- Minimize criterion $\frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}(x_i))$ for some loss function L (e.g. least squares)

Prediction Error

- Conditional: $E_{X,Y}[L(Y, \hat{f}_Z(X))]$
- Unconditional: $E_Z[E_{X,Y}[L(Y, \hat{f}_Z(X))]]$
- Fixed x: $E_Z[E_Y[L(Y, \hat{f}_Z(x))]]$
- Last is equal to $\sigma^2 + \text{BIAS}^2 + \text{VARIANCE}$
- Bias/variance tradeoff

Estimating PE

- Training (apparent) error underestimates
- Test set estimate (requires test set, approaches conditional PE as test set increases in size)
- Cross-validation - can be biased upward
- Bootstrap

Cross-validation

- Divide data set into k "folds" of similar size (random splits)
- For each fold, use the fold as a test set and the rest as a training set
- Fit model and calculate the test set error
- Repeat for each fold in turn, average
- Repeat for different random splits, average.
- Large k, less biased, more variable

Bootstrap

- err+opt:
  $\overline{err} + \overline{PE(boot, train) - PE(boot, boot)}$.
- 0.632 estimate: $(1 - 0.632)\overline{\text{err}} + 0.632\epsilon^{(0)}$
- $\epsilon^{(0)} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|C_i|}\sum_{b\in C_i}^{n}(y_i - f_b(x_i))^2$
- Last one based on out-of-bag samples

Complexity

- Model too complex: Apparent error small, test error large
- Predictor has small bias but big variance
- Need to tune models to avoid under/overfitting

Classes of functions

- Linear: simple, not very flexible, (tune by variable selection)
- Gams: more flexible, tune by variable selection
- PPR: more flexible, not very interpretable, tune by number of ridge functions
- MARS: more flexible, not very interpretable, tune by number of basis functions
- NN: more flexible, not very interpretable, tune by number of hidden layer units
- Tree: more flexible, interpretable, tune by number of terminal nodes (cp)

Trees

- Recursive partitioning
- Choosing splits
- Defining the function
- Role of cp

Boosting and bagging

- Boosting with squared error loss, fit model to residuals, add a small proportion of predictor to previous
- Boosting with “classification” loss, fit model to gradient of loss function
- Boosting trees: lots of small trees
- Bagging: fit model to different bootstrap samples, average or “majority vote”
- Random forests: Bagging with trees, (mtry, depth of trees)
- Use big trees to reduce bias, reduce variance with mtry

Classification

- Bayes classifier: assign to class for which $P(C_j|x)$ is a maximum
- Model $P(C_j|x)$ with logistic, tree, NN, RF etc
- Can use Bayes' Theorem: choose class with maximum $P(x|C_j)\pi_j$
- QDA and LDA
- Trees: choose splits to get biggest decrease in node impurity (Gini Index)
- Support vector machines

SVM's

- Dividing feature space with linear boundary
- Criterion for choosing the boundary
- Primal and dual formulations
- Enlarging feature space
- Kernels

Regularization

- Ridge: minimize $||y - Xb||^2$ subject to $\sum_{j=1}^{p} b_j^2 \leq s$
- Lasso: minimize $||y - Xb||^2$ subject to $\sum_{j=1}^{p} |b_j| \leq s$
- Shrinks coefs (Lasso can zero)
- Can improve PE
- Handles case $p > n$

Preprocessing

- Centering/scaling
- Symmetrization with Box-Cox transform
- Feature engineering with PCA
- Variable screening with correlations
- Zero-variance predictors

Imputation

- Simple methods
- Multiple imputation: reflects uncertainty in imputation process
- Requires modeling conditional distributions
- `mi` and `mice`
- Imputation using random forests (`missForest`).