Supervised Learning I

Christoph Kern

Mannheim Machine Learning Modules

c.kern@uni-mannheim.de

October 20, 2017

Outline

Introduction

Machine Learning

- Algorithms based on statistical criteria which focus on making predictions based on a data-driven learning process
- "Machine Learning is the field of scientific study that concentrates on induction algorithms and on other algorithms that can be said to "learn"." (Kohavi / Provost 1998)
- Combines Computer Science and Statistics

Statistical Learning

- Machine Learning from a "statistical perspective"

ML Basics

Unsupervised Learning

- Finding patterns in data using a set of input variables X

Supervised Learning

- Predicting an output variable Y based on a set of input variables X
  1. Learn the relationship between input and output using **training data** (with X and Y)
     $$Y = f(X) + \varepsilon$$
  2. Predict the output based on the prediction model (of step 1) for **new test data** ($\sim$only X available)
- continuous Y: regression, categorical Y: classification
- Focus on **prediction** ($\neq$ causation)

Training and test error

Training error

$$\overline{\text{err}} = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{f}(x_i))$$

- Prediction error based on **training data**

Test error

$$\text{Err}_{\mathcal{T}} = \text{E}(L(Y, \hat{f}(X))|\mathcal{T})$$

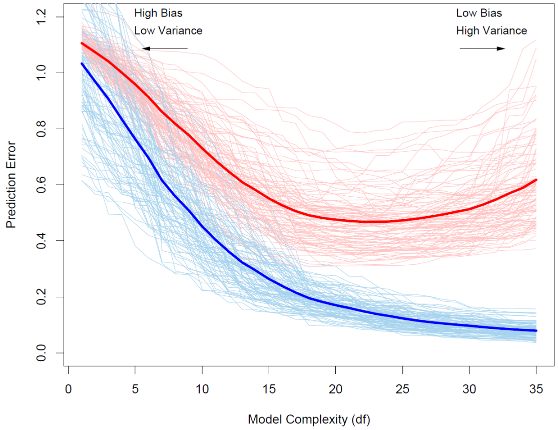
- Prediction error using **test data** (given training data $\mathcal{T}$)

Expected test error decomposition

$$\text{Err}(x_0) = \text{Bias}^2(\hat{f}(x_0)) + \text{Var}(\hat{f}(x_0)) + \text{Var}(\varepsilon)$$

- Minimizing the expected test error
  - Low bias (deviation between $\text{E}(\hat{f}(x_0))$ and $f(x_0)$) **and**
  - Low variance ($\text{Var}(\hat{f}(x_0))$ using different training data)

Figure: Bias-Variance Trade-Off: Training error and test error by model complexity



Hastie et al. 2009

Validation set, test set, CV

Validation set approach

- Training set & validation set
  1. Fit model using one part of training data
  2. Compute test error for the excluded section

$\rightarrow$ Model assessment

- Training set, validation set & test set
  1. Fit models using training part of training data
  2. Choose best model using validation set
  3. Evaluate final model using test set

$\rightarrow$ Model tuning & assessment

Cross-Validation

- LOOCV (Leave-One-Out Cross-Validation)
  1. Fit model on training data while excluding one case
  2. Compute test error for the excluded case
  3. Repeat step 1 & 2 n times

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{f}^{-\kappa(i)}(x_i))$$

- k-Fold Cross-Validation
  1. Fit model on training data while excluding one group
  2. Compute test error for the excluded group
  3. Repeat step 1 & 2 k times (e.g. $k = 5$, $k = 10$)

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^{k} L(y_i, \hat{f}^{-\kappa(i)}(x_i))$$

- Outlook: nested CV, repeated CV, ...

Performance measures

Performance metrics for regression problems

$$r^2 = 1 - RSS/TSS$$

Root mean squared error (RMSE):

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2}$$

Median of squared errors (MEDSE):

$$\text{median}((y_1 - \hat{f}(x_1))^2, \dots, (y_n - \hat{f}(x_n))^2)$$

Mean of absolute errors (MAE):

$$\frac{1}{n}\sum_{i=1}^{n}|(y_i - \hat{f}(x_i))|$$

Performance metrics for classification

Table: Confusion matrix

		Prediction 0	Prediction 1	
Reference	0	True Negatives (TN)	False Positives (FP)	N'
	1	False Negatives (FN)	True Positives (TP)	P'
		N	P	

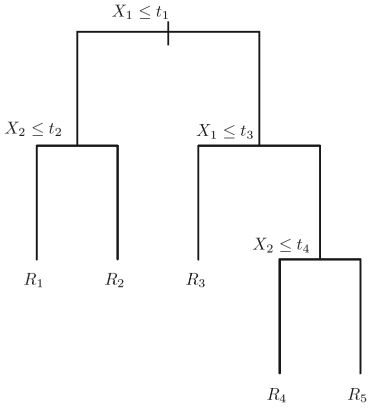
Accuracy: $\frac{TP+TN}{TP+FP+TN+FN}$
Sensitivity / Recall: $\frac{TP}{TP+FN}$
Specificity: $\frac{TN}{TN+FP}$
Positive predictive value / Precision: $\frac{TP}{TP+FP}$
Negative predictive value: $\frac{TN}{TN+FN}$
Outlook: F_1, κ, AUC-ROC, AUC-PR, ...

CART

Classification and Regression Trees (CART)

- Approach for partitioning the predictor space into smaller subregions via "recursive binary splitting"
- Results in a "top-down" tree structure with...
    - Internal nodes within the tree
    - Terminal nodes as endpoints
- Can be applied to regression and classification problems
- Can be used as base learner for ensemble methods

Figure: A small tree



James et al. 2013

Tree growing

Growing a regression tree

Define pairs of regions for all $X_1, X_2, ..., X_p$ predictors and cutpoints c

$$\tau_L(j,c) = \{X|X_j < c\} \text{ and } \tau_R(j,c) = \{X|X_j \geq c\}$$

Find split s which maximizes the reduction in RSS

$$\Delta RSS(s,\tau) = RSS(\tau) - RSS(\tau_L) - RSS(\tau_R)$$

$$RSS(\tau) = \sum_{i \in \tau} (y_i - \hat{y})^2$$

with $\hat{y}$ being the mean of y in node τ_m

Growing a classification tree

Define pairs of regions for all $X_1, X_2, ..., X_p$ predictors and cutpoints c

$$\tau_L(j,c) = \{X | X_j < c\} \text{ and } \tau_R(j,c) = \{X | X_j \geq c\}$$

Find split s which maximizes the reduction in node impurity

$$\Delta I(s,\tau) = I(\tau) - p(\tau_L)I(\tau_L) - p(\tau_R)I(\tau_R)$$

Impurity measures

$$I_{Gini}(\tau) = \sum_{k=1}^{K} \hat{p}_k(1-\hat{p}_k)$$

$$I_{entropy}(\tau) = -\sum_{k=1}^{K} \hat{p}_k \log \hat{p}_k$$

with $\hat{p}_k$ being the proportion of observations from class k in node τ

Algorithm 1: Tree growing process

```
1  Define stopping criteria;
2  Assign training data to root node;
3  if stopping criterion is reached then
4      end splitting;
5  else
6      find the optimal split point;
7      split node into two subnodes at this split point;
8      for each node of the current tree do
9          continue tree growing process;
10     end
11 end
```

A given tree

$$\mathcal{T} = \sum_{m=1}^{M} \gamma_m \cdot 1_{(i \in \tau_m)}$$

consists of a set of $m = 1, 2, \ldots, M$ nodes which can be used for prediction by...

- Regression
  - ...using the mean of y for training observations in τ_m
- Classification
  - ...going with the majority class in τ_m

$\rightarrow$ Prediction surface: Block-wise relationship between features and outcome

Tree pruning

Stopping rules

- Minimum number of cases in terminal nodes
- Decrease in impurity exceeds some threshold

$\rightarrow$ However, worthless splits can be followed by good splits

Cost complexity pruning

$$R_\alpha(\mathcal{T}) = R(\mathcal{T}) + \alpha|\mathcal{T}|$$

- Find the best subtree by balancing tree quality $R(\mathcal{T})$ and complexity $|\mathcal{T}|$
- α controls the penalty on the number of terminal nodes
- α can be chosen through CV

References

James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). *An Introduction to Statistical Learning*. New York, NY: Springer.

Kohavi, R., Provost, F. (1998). Glossary of Terms. *Machine Learning*, 30(2): 271–274.

Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer.