Machine Learning
Lecture 6

Lecture by Dr. Shai Fine
Typeset by Steven Karas

2017-11-26
Last edited 21:05:10 2017-11-26

Disclaimer These lecture notes are based on the lecture for the course Machine Learning, taught by Dr. Shai Fine at IDC Herzliyah in the fall semester of 2017/2018. Sections may be based on the lecture slides written by Dr. Shai Fine.

Homework 2 It is strongly suggested to submit hw2 on time, because it will delay publishing hw3, as it contains the "correct" list of features to select.

Agenda

- Binary to Multiclass Classification
- Neural networks

1 Short review

1.1 Naive Bayesian Classification

Assume feature independence, and calculate:

$$\Pr(x \mid C_k) \approx \prod_{i=1}^{d} \Pr(x_i \mid C_k)$$

This is a bad approximation, but works well enough.

Note that this creates linear decision boundaries in the probability domain, which are nonlinear in the feature space.

1.2 Decision Trees

Tree where each internal node represents a decision. Decisions may be univariate, or multivariate. Each decision partitions the feature space into separate branches of the tree. Leaves represent classification labels, or regressions either as an average or a local fit.

Note that the decision boundaries are orthogonal to the axes of the feature space.

2 Binary to Multiclass

Recall that KNN and naive bayes are capable of solving multiclass problems natively.

Transform the multiclass problem into a set of binary problems. Train a classifier for each problem separately, then combine the prediction results.

Note that it can be extremely helpful to use classifiers that output a confidence level, which can be used to combine their results. However, this is out of the scope of this course.

2.1 One vs Rest (One-Hot)

Binary classifiers per class. Simplest to implement, but sensitive to binary classification error.

2.2 One vs One (All-pairs)

$\frac{n(n-1)}{2}$ classifiers, and choose the class that wins the most pairwise matches.

2.3 Error Correct Output Code (ECOC)

Presented by Dietterich and Bakiri in 1995

Construct each class as a codeword that maximizes class separation. Some codes maximize column separation, which decorrelates binary classification error regions.

Train classifiers for each of the n codeword elements independently.

Uses Hamming distance as a distance metric on the classifier results, and decides on the class with the smallest Hamming distance.

3 Evaluation and Model Selection

We want to estimate the error of a given model, or compare the expected errors of two models:

3.1 Empirical methods

Compare models based on metrics defined ahead of time, and select a model type and specifically trained model according to these metrics.

3.1.1 Validation set method

Set aside validation[1] data (typically 20-30% of the original data set). Train the model on the training set, and once trained, validate the performance of the model on the validation set.

This is easy to implement, but doesn't train your model on the full data.

3.1.2 k-Fold Cross Validation

This method is for evaluating the model type (e.g. DT vs SVM vs CNN), which should be retrained on the full training set for prediction purposes.

Randomly partition data X into k disjoint equally sized subsets $X_1, \ldots, X_k$.

For each set X_i, train on $X \setminus X_i$, and calculate the validation error $\text{Err}_i = \sum_{x \in X_i} I[h_i(x) \neq c_t(x)]$.

The overall error is the empirical average: $\text{Err} = \frac{1}{k} \sum_i \text{Err}_i$.

There are some methods for parallelizing this process.

Choosing k If k is too small, we have low confidence in the error estimation. If k is too large, the test set is larger, and we train the model more times. There is a proof that $k = |X|$, called Leave-One-Out cross validation, gives us the best estimation of error, but takes a very long time to run.

3.1.3 Receiver Operating Characteristic (ROC)

Historically came from signal detection, then used in biomedical fields. Used heavily in machine learning for assessing binary classifiers.

Specificity $\text{TN}/(\text{TN} + \text{FP})$

Sensitivity $\text{TP}/(\text{TP} + \text{FN})$. Also called the recall, or the true positive rate.

Precision $\text{TP}/(\text{TP} + \text{FP})$

False Positive Rate $\text{FP}/(\text{TN} + \text{FP}) = 1 - \text{Specificity}$

Two dimensional graph - True positive rate (y-axis) is plotted against the False positive rate (x-axis). Sometimes other metrics are graphed, but TPR vs FPR is the most common. A graphical example can be found on slide 45.

Can be used to choose a confidence threshold for a binary classifier.

[1]Older literature refers to this set as the test set, and the test set as the validation set

Comparing models By using area under the ROC curve (AUC) as a quality metric, we can compare models. Note that the AUC must be between 0.5 and 1.

3.2 Theoretical methods

Sample complexity, computational complexity, generalization bounds, and information density.

3.3 Model Selection

When selecting a model, we prefer the simpler

Occam's Razor

> Pluralitas non est ponenda sine necessitas. Plurality should not be posited without necessity. – William of Occam 14th century

Simplicity in our context is when we minimize both the training and the validation error together.

4 Next week

???