Machine Learning
Lecture 3

Lecture by Dr. Shai Fine
Typeset by Steven Karas

2017-11-05
Last edited 21:01:02 2017-11-05

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.

HW2 HW2 is due a month from today. Pairs are strongly recommended. Triplets are allowed, but have no bonus points and must complete all bonus assignments.

It is strongly recommended to be very conservative when selecting features.

We can choose the exact amount of records for the test and cross validation data sets, but they should fall within the guidelines.

Agenda

- Data cleansing

1 Data Cleansing

Noise can find its way into our data, and we need tools to help remove or deal with it. Generally speaking, we only try to treat white noise (uniformly distributed spectral density).

1.1 Outlier Detection

Outliers are observations that deviate strongly from other observations in the sample. We can classify outliers using statistical methods, or with a simple distance metric.

1.1.1 Metric based approaches

Common metrics are normalized distance, etc.

- Density - observations with fewer than t neighbors within a distance d
- Distance - the top n observations whose distance to the kth nearest neighbor is greatest
- Local Outlier Factor - define local density as a function of the distance of the nearest n neighbors, and compare the local density to those neighbors
- Class Outlier Factor - as above, but restricted to classes

1.2 Attribute/Feature Construction

- Grouping categories - Age groups, moving to less granular groups
- Data representation/distribution - Log scaling, polynomial scaling
- Meta-features - Image contours, phoneme in speech signal
- Domain knowledge - semantic knowledge

1.3 Normalization

1.3.1 Min-Max

Good for transforming low kurtosis (tends to uniform) data to fall within a specific range.

$$v' = \frac{v - \min v}{\max v - \min v}(\max_{\text{new}} - \min_{\text{new}}) + \min_{\text{new}}$$

1.3.2 Z-score

Good for symmetric data with long tails.

$$v' = \frac{v - \mu_v}{\sigma_v}$$

1.3.3 Decimal Scaling

Good for unknown distributions.

$$v' = \frac{v}{10^k}$$

Where k is the smallest integer such that $\max |v'| < 1$.

1.4 Distribution Transformation

Algorithms work best against normally distributed data.

For example, convert lognormal distributions to normal with $v' = \exp(v - \min v)$, or a geometric distribution to normal with $v' = \log(v - \min v + 1)$

1.5 Discretization

Some algorithms work better against categorical data, or at least quicker.

Binning, with or without class information, preprocessed or dynamic, global or local. There are many ways to do this.

1.6 Nominal to Numeric Conversion

Some models can only take nominal values, others can only take numeric values.

Binary If there are only two nominal values, it can be useful to use $\{0, 1\}$ or $\{-1, 1\}$.

Multi valued unordered ID-like variables can usually be thrown out. It can be useful to map down into a smaller set of nominal values.

Sparse representation - create a presence feature per nominal value (0,1 usually represents a don't care value, whereas -1,1 is typically used for a exists/does not exist semantic)

For n nominal values, we only need $\log n$ binary features to represent the values, but may be susceptible to model noise. However, this can be limiting on our modeling architecture (e.g. a CNN cannot take a union of multiple nominal values).

Ordered Convert values to numbers and preserve natural order.

1.7 Imbalanced class sizes

Class frequency can be extremely unbalanced: customer churn, medical diagnoses, fraud.

Most algorithms can handle up to an order of magnitude or two difference in class frequency.

Generally speaking, training against an artificially balanced training set works well.

Downsampling Sample each class independently.

Bootstrap Sampling with replacement based on class frequency.

Weighted training Weighting error by the frequency of the classes when training.

2 Feature Selection

Filter methods ranks features or feature subsets, and uses the highest ranked features.

Wrapper methods use a classifier to assess features or subsets.

Embedded methods are specific to a learning model. Performs feature selection implicitly in the course of training (e.g. decision trees, winnow).

2.1 Filter Methods

We need to define a ranking function for each feature subset. Pearson or mutual information is common.

Pearson correlation Useful for linear relationships.

Mutual Information

$$I(f_k, y) = \int_{f_k} \int_y \Pr(f_k, y) \log \frac{\Pr(f_k, y)}{\Pr(f_k) \Pr(y)} df_k dy$$

This measures the shared information, and can detect any form of statistical dependency.

2.1.1 Subset Selection

Finding a minimally small feature set is NP-Hard.

Sequential Forward Selection These algorithms add or remove features, but have a tendency to locally converge.

Start with no features, and transition to the state that improves the rank the most by adding a feature.

Termination condition is not well defined.

Sequential Backwards Selection As SFS, but start from all features, and remove them.

3 Next week

There will be a special lecture before the regular lecture next week that will cover Python.