Machine, Data and Learning (CS7.301)

Spring 2022, IIIT Hyderabad
20 Jan, Thursday (Lecture 6)

Taught by Prof. Praveen Paruchuri

Machine Learning

Avoiding Overfitting

Feature Selection (contd.)

There are three types of methods to carry out feature selection: filter, wrapper and embedded methods.

Filter methods select features on the basis of their scores in various statistical tests for their correlation with the outcome variable. Some such tests are the chi-squared test (for categorical variables) and the Pearson correlation coefficient scores (for continuous variables).

Wrapper methods use a subset of features and train a model with them, and based on the inferences from this, decide to add or remove features. These are computationally expensive. An example of a wrapper method is the recursive feature elimination algorithm.
Recursive feature elimination is a greedy optimisation algorithm, which repeatedly creates models and keeps aside the best or worst performing feature at each iteration. It repeats this process until all the features are exhausted, and then ranks the features.

Embedded methods learn which features best contribute to the accuracy of the model while the model is being created. Regularisation methods are the most common type of embedded feature selection methods.

Dimensionality Reduction

This is a method for transforming high-dimensional data into a lower-dimensional space. Naturally, this leads to a loss of information – the goal is to minimise this loss.

Data Preprocessing

Data mining is the predecessor to data science. A popular methodology in it is CRISP DM, or the cross-industry standard process for data mining. It has the following phases:

- business understanding
- data understanding
- data preparation
- modelling
- evaluation
- deployment

The key to this is high-quality data, which means that cleaning data is very important to the process. It can therefore form a large part of ML.

However, there is no single method for cleaning data, and it frequently involves manual work. The data can have many forms, but usually, as it is the most convenient, it is assumed to have the form of a table of numbers by default.

Feature Engineering

Features can be of many types – binary or categorical, quantitative, or continuous. They can follow many distributions (normal, binomial, poisson, etc.).
Feature engineering is thus the process of creating or improving features. It is based on common sense and domain knowledge.

A part of feature engineering consists of how to deal with missing values. One way is to ignore them; we could also substitute them with fixed values, like the mean or median (this is called imputing).

Another aspect of the process is converting labels to numeric form, which is processable by ML algorithms. This is called label encoding; if the labels are integers, we also refer to it as integer-label encoding. The issue with this is that the model may conclude that there is an ordering.
A common method for integer-label encoding is one-hot encoding or one-of-K encoding, which encodes each data point as a vector with a 1 in one position and 0 in all other positions.

Scaling is another process in feature engineering. When the values of a feature differ by orders of magnitude, it has to be scaled.
One strategy for this is standardisation, wherein we remove the mean and divide by the standard deviation, to get a normal distribution.

Features may sometimes interact with each other. If a, b are two features, a polynomial feature of the form $a^2 + ab + b^2$ can encode their relation, and the ab term is called their *interaction*. We try to avoid higher-order polynomials and the interaction of too many features.

Power transformations are functions to transform numerical features into a more

convenient form (for example, to conform better to a normal distribution). One such function is the Box-Cox transform, which tries to find the best power needed to transform data to a normal distribution.

Binning is another process, wherein feature values are separated into bins. It can lead to a loss of information, but it improves on speed and memory requirements. It can also reduce the chance of overfitting.