Foundations of Data Science, Fall 2020

7a. Feature Selection

Prof. Dan Olteanu

DaST Data•(Systems+Theory)

University of Zurich[UZH]

Oct 16, 2020

https://lms.uzh.ch/url/RepositoryEntry/16830890400

https://uzh.zoom.us/j/96690150974?pwd=cnZmMTduWUtCeWoxYW85Z3RMYnpTZz09

Feature Selection: Goal, Premise, and Motivation

Goal: Given a set of possible features, select features relevant to the model to learn

Premise:

- Data contains redundant or irrelevant features

  Removing them does not incur loss of information

- Relevant features may be redundant in the presence of another relevant feature

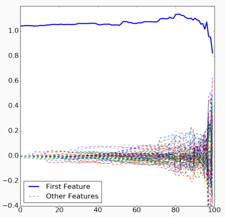
  The two features may be strongly correlated

Why?

- Enhanced generalisation by reducing overfitting
- Avoid the curse of dimensionality
- Shorter training times
- Simplified models that are easier to interpret by users

Feature Selection to Reduce Overfitting

Recall previous example with a small training set and many irrelevant features that is prone to overfitting



- What if we discard irrelevant features and using training set with fewer features?
- Problem with exhaustive search: For n features, there are 2^n subsets of features

---

Feature Selection Methods

Component 1: Search technique for subsets of a given set of features

Component 2: Evaluation measure to score the different feature subsets – methods differ in the measure

1. Wrapper methods: Train a new model for each feature subset
   - Score = count the number of mistakes on hold-out set
   - Expensive, yet usually provides the best performing feature set
   - Example: Forward stepwise selection
2. Filter methods: Use proxy measure as score instead of the actual test error
   - Expose relationships between features
   - Typical scores: mutual information, Pearson correlation coefficient
   - Fast to compute, yet resulting feature set not tuned to a specific type of models
3. Embedded methods: Perform feature selection while constructing the model
   - LASSO: non-zero parameters for corresponding features
   - Elastic net regularisation: combines ℓ_1 and ℓ_2 regularisations

---

Forward Stepwise Selection

For n features, we ask the following sequence of n questions ($1 < i \leq n$):

Q1: What is the best 1-feature model? Let the chosen feature be f_1.

$\vdots$

Q_i: What is the best i-feature model that also has the previously selected features $f_1, \ldots, f_{i-1}$?
Let the newly chosen feature be f_i.

Output: The best seen k-feature model, for any $1 \leq k \leq n$.

Analysis of the algorithm:

- At step i, it trains and tests $n - i + 1$ new models $\Rightarrow O(n^2)$ models to train and test in total
- For linear regression: Same complexity as building one model

Efroymson (1960): Multiple regression analysis.
https://www.github.com/EFavDB/linselect

Feature Selection via Mutual Information

Mutual Information for two random variables X and Y:

$$I(X, Y) = \sum_x \sum_y p(X = x, Y = y) \cdot \log \frac{p(X = x, Y = y)}{p(X = x) \cdot p(Y = y)}$$

Approach:

- Compute mutual information for each feature and the label/target
- Only keep the features that provide information about output
  - Ranking of features instead of finding the best subset
  - Cut-off point using cross-validation

Computational considerations:

- Probabilities $p(X = x)$, $p(Y = y)$ and $p(X = x, Y = y)$ can be empirically obtained from training set
  - $p(X = x)$: fraction of the number of samples with $X = x$ over the number of all samples
  - Variables with continuous domains: first discretise their domains

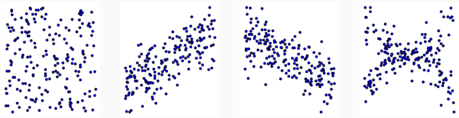
Covariance vs Correlation

Covariance for random variables X and Y measures how the random variables change jointly

$$\mathrm{cov}(X, Y) = \mathbb{E}\left[(X - \mathbb{E}[X]) \cdot (Y - \mathbb{E}[Y])\right]$$

The (Pearson) correlation coefficient normalises the covariance to give a value between -1 and $+1$.

$$\mathrm{corr}(X, Y) = \frac{\mathrm{cov}(X, Y)}{\sqrt{\mathrm{cov}(X, X) \cdot \mathrm{cov}(Y, Y)}}$$



corr = 0.0 corr = 0.7 corr = -0.7 corr = 0.0

Note: Independent variables are uncorrelated, but the converse is not true!