Data Analysis 3

Dimension Reduction, Classfication and Feature Selection

Jan Platoš

October 28, 2018

Department of Computer Science
Faculty of Electrical Engineering and Computer Science
VŠB - Technical University of Ostrava

Table of Contents

Dimension Reduction

Classification

Feature Selection

Decision trees

Rule-based classification

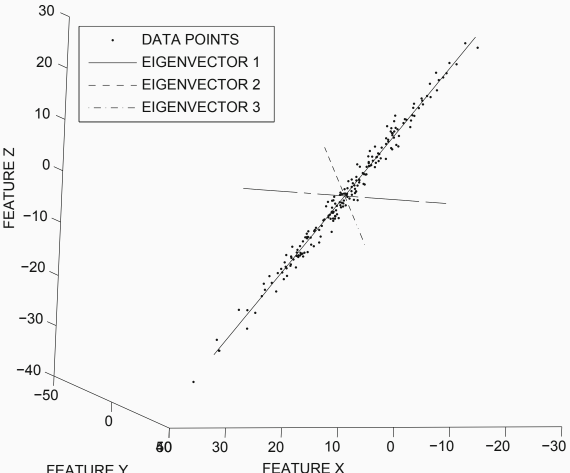
Dimension Reduction

Dimension Reduction

- Highly dimension data brings problems with clustering/classification.
- Many features are noisy or noise itself.
- Many features correlates with another features.
- Feature selection:
    - Select features according a measure and removes is from the dataset.
    - Measure is based on a mathematical principle (Variance, Entropy, etc.)
- Dimension Reduction:
    - Search for optimal mapping between original dimension into defined amount of dimensions.
    - Each new dimension is a linear/non-linear combination of original features.

Principal Component Analysis (PCA)

- The goal of PCA is to rotate the data into an axis-system where the greatest amount of variance is captured in a small number of dimensions.



Principal Component Analysis (PCA)

- The PCA for the input matrix D is computed as:

$$C = \frac{D^T D}{n} - \overline{\mu}^T \overline{\mu}$$

- C is a covariance matrix of D, n is the number of points of the D, μ is the mean vector.

$$C = P \Delta P^T$$

- P contains orthonormal eigenvectors and Δ contain eigenvalues.

$$D' = DP$$

- D' is transformed matrix in the terms of new axis P.

Singular Value Decomposition (SVD)

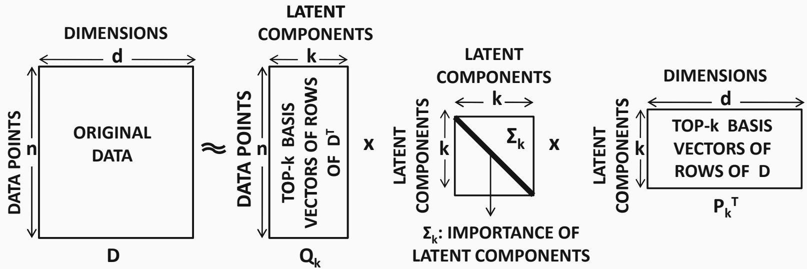
- Generalization of the PCA.

$$D = U\Sigma V^T$$

- where U contains left singular vectors, Σ contains singular values and V^T contains right singular vectors.
- The presented decomposition is proven to be optimal.
- Reducing the Σ to k coefficients leads to best approximation of the matrix D

$$D \approx U_k\Sigma_k V_k^T$$

Singular Value Decomposition



DIMENSIONS

d

DATA POINTS

n

ORIGINAL DATA

D

≈

LATENT COMPONENTS

k

DATA POINTS

n

TOP-k BASIS VECTORS OF ROWS OF D^T

Q_k

x

LATENT COMPONENTS

k

LATENT COMPONENTS

k

Σ_k

Σ_k: IMPORTANCE OF LATENT COMPONENTS

x

DIMENSIONS

d

LATENT COMPONENTS

k

TOP-k BASIS VECTORS OF ROWS OF D

P_k^T

Non-negative Matrix Factorization (NMF or NNMF)

- A factorization methods which works and produces only non-negative elements.

$$D = WH$$

- W contains weights and H contains basis vectors.
- Due to non-negativity the basis vectors as well as weights may be easily interpreted.
- The NMF inherits clustering property, where close vectors are clustered together.
- The cost function is defined usually as a Frobenius norm:

$$E = \|D - WH\|_F$$

$$\|A\|_F = \sqrt{\sum_i \sum_j |a_{ij}|^2}$$

Classification

Classification

Basic questions:

- What it is?
- What it needs?
- What it produces?

Classification

Basic questions:

- What it is?
- What it needs?
- What it produces?

Definition

Given a set of training data points, each of which is associated with a class label, determine the class label of one or more previously unseen test instances.

Classification

Phases of classification:

- Training phase - construction of models from the training instances.
- Testing phase - determining class labels of one or more training instances.

Output of classification:

- Label prediction - one fixed label is predicted.
- Numerical score - numerical evaluation of each label assignment to the instance.

Feature Selection

Feature Selection

- Selection of the attributes subset for classification.
- Three types of models:
  1. Filter models – crisp mathematical criterion is used to evaluate each subset of attributes.
  2. Wrapper models – the model is run on each candidate subset to evaluate its efficiency.
  3. Embedded models – The model information is used to prune irrelevant attributes.

Feature Selection - Filter models

Gini index:

- Measures the discriminative power of a particular attributes subset.
- Usually used to categorical data/discretized numerical data.

Feature value index:

$$G(v_i) = 1 - \sum_{j=1}^{k} p_j^2$$

- $v_1, v_2, \ldots, v_r$ are r values of a particular attribute.
- p_j is the fraction of points that contains attribute v_i that belong to the class j for k possible classes.

Feature index:

$$G = \frac{1}{n} \sum_{i=1}^{r} n_i G(v_i)$$

- n is the number of input points and n_i is the number of point with the value v_i.

Feature Selection - Filter models

Entropy:

- Measures the information gain from fixing a specific attribute value.

Feature value entropy:

$$E(v_i) = -\sum_{j=1}^{k} p_j \log(p_j)$$

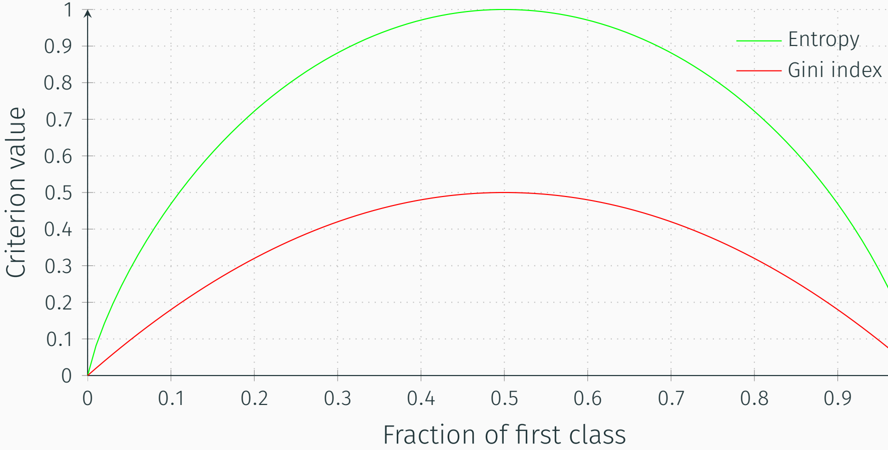
- $v_1, v_2, \ldots, v_r$ are r values of a particular attribute.
- p_j is the fraction of points that contains attribute v_i that belong to the class j for k possible classes.

Feature entropy:

$$E = \frac{1}{n} \sum_{i=1}^{r} n_i E(v_i)$$

- n is the number of input points and n_i is the number of point with the value v_i.

Feature Selection - Filter models



Feature Selection - Filter models

Fisher Score:

- Naturally designed for numeric attributes.
- Measures the the ratio of the average interclass separation to the average intraclass separation.

$$F = \frac{\sum_{j=1}^{k} p_j(\mu_j - \mu)^2}{\sum_{j=1}^{k} p_j \sigma_j^2}$$

- p_j is the fraction of data points belonging to class j.
- μ_i, σ_j is the mean and standard deviation of data points belonging to class j for a particular feature.
- μ is the global mean of the data points on the feature being evaluated.

Feature Selection - Wrapper models

- Different classification models are more accurate with *different* sets of features.
- Filter models are agnostic to the particular classification algorithm being used.
- The characteristics of the specific classification algorithm is used to select features.
    - Linear classifier work more effectively with a set of features where the classes are best modeled with linear separators.
    - Distance based classifier works well with features in which distances reflect class distributions.
- A specific classification algorithm is used as an input to the feature selection.
- Wrapper models then optimize the feature selection process to the classification algorithm.
- The basic strategy in wrapper models is to iteratively refine a current set of features F by successively adding features to it.

Feature Selection - Wrapper models

- The algorithm starts with empty feature set $F = \emptyset$.
- The strategy may be summarized as follows:
    - Create an augmented set of features F by adding one or more features to the current feature set.
    - Use a classification algorithm A to evaluate the accuracy of the current set of features F.
    - Use the accuracy to either accept or reject the augmentation of F.
- The augmentation of F can be performed in many different ways.
    - Greedy strategy - the set of features in the previous iteration is augmented with an additional feature with the greatest discriminative power with respect to a filter criterion).
    - Random sampling - features may be selected for addition via random sampling.

Feature Selection - Wrapper models

- The accuracy of the classification algorithm A is used to determine the acceptance/rejection of the features.
- The rejected features are removed from the set and another augmentation is tested.
- This approach is continued until there is no improvement in the current feature set for a defined minimum number of iterations.
- The final set of featured is sensitive to the choice of the algorithm A.

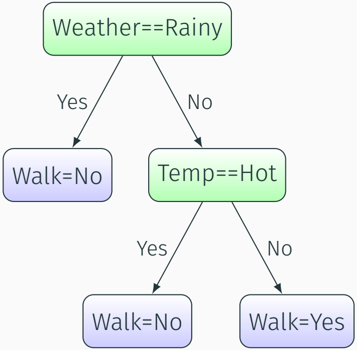
Decision trees

Decision Trees

- Classification is modeled using hierarchical decisions on the features that are arranged in tree-like structure.
- The decision at a particular node, called split criterion, is a relational condition on one or more features and their values.
- The goal is to identify a split criterion that minimizes the mixing of classes in each branch.
- Works on binary, numeric and categorical attributes.
- Each sub-space (region) is recursively split until terminal conditions are reached.
- Univariate or Multivariate split is possible.

Decision Trees

Weather	Temp	Walk?
Sunny	Cold	Yes
Sunny	Warm	Yes
Sunny	Hot	No
Cloudy	Cold	Yes
Cloudy	Warm	Yes
Cloudy	Hot	No
Rainy	Cold	No
Rainy	Warm	No
Rainy	Hot	No



Decision Trees

Split Criteria:

- The goal is to maximize separation of the different classes among the children nodes.
- Binary attribute – only one type of split is possible.
- Categorical attribute with r values
    - r-way split,
    - binary split on 21 possibilities,
    - binary split on r possibilities (one-to-rest strategy).
- Numeric attribute
    - A split is made between two values with < or <= relation.
    - All values or selected values only may be tested.

Decision Trees - Split Criteria

Definitions:

- S is a set of points in a branch of a tree.
- $|S|$ is size of the set (number of points in a set).
- r-way split has r subsets $S_1, \ldots, S_r$ of set S.
- k is the number of classes.

Decision Trees - Split Criteria

Error rate:

- On a set:

$$Err(S) = 1 - p$$

  - where the p is a fraction of points that belongs to the dominant class from S.

- On r-way split:

$$Err(S \Rightarrow S_1, \ldots, S_r) = \sum_{i=1}^{r} \frac{|S_i|}{|S|} (1 - p)$$

Decision Trees - Split Criteria

Gini index:

- On a set:

$$G(S) = 1 - \sum_{j=1}^{k} p_j^2$$

  - where the p_j is a fraction of points that belongs to the class j from S.

- On r-way split:

$$G(S \Rightarrow S_1, \ldots, S_r) = \sum_{i=1}^{r} \frac{|S_i|}{|S|} G(S_i)$$

Decision Trees - Split Criteria

Entropy:

- On a set:

$$E(S) = -\sum_{j=1}^{k} p_j \log_2 (p_j)$$

  - where the p_j is a fraction of points that belongs to the class j from S.

- On r-way split:

$$E(S \Rightarrow S_1, \ldots, S_r) = \sum_{i=1}^{r} \frac{|S_i|}{|S|} E(S_i)$$

Decision Trees

Stopping criterion:

- Very difficult to stop during the tree growth.
- Single class in a leaf node is the final condition.
- Such tree has 100% precision on Training data.
- But, such tree is over-fitted (unable to generalize to unseen data).
- Over-fitting is done by lower nodes with less number of points.

Decision Trees

Pruning:

- Shallow trees are more preferable is they produces the same error on training data.
- Nodes/Trees are evaluated using a criterion that penalizes the more complex tress without satisfactory improvement in precision.
- Usually a holdout set (e.g. 20% of training set) is used for pruning.
- A node is prunes is its removing improves the precision on the holdout.
- A leaf node are pruned iteratively until no node should be removed.

Rule-based classification

Rule-based classification

- A generalization of the Decision Trees.
- A set of rules in a form:

IF Condition THEN Conclusion

- *Condition* or *Antecedent* is a combination of relational, set and logical operators over features.
- *Conclusion* or *Consequent* is a class label.
- A rule cover the training instance is the condition match the instance.

Rule-based classification

Rule types:

- Mutually exclusive rules
  - Each rule covers disjoin set of instances.
  - Each instance trigger at most one rule.
- Exhaustive rules
  - The entire data space is covered by at least one rule.
  - Simple exhaustive rule assign dominant class do anything (catch-all).
- Non mutually exclusive rules brings problems with rule evaluation.

Rule-based classification

Rule ordering:

- Ordered rules
  - Rules are ordered by priority, such as quality measure.
  - Rules may be ordered by class-based principle.
  - Only the first triggered rule vote, its consequent is the result.
  - The rare classes are usually ordered first.
- Unordered rules
  - There is no priority on rules.
  - The dominant class of the all triggered rules is selected.
  - Simplifies the learning phase.

Rule-based classification

Rule generation:

- The goal is to generate rules that covers the instances from the training data.
- Two major algorithm exists:
  - Generation using Decision Trees.
  - Sequential Covering Algorithm.

Rule-based classification - Rule generation

Rule generation using Decision Trees:

- Trees are used for generation of the rules.
- Each leaf node represent one rule with its sequence of splits that lead to this leaf from root.
- The pruning is not made on tree, but on rules.
- Each rule is processed separately and pruned to get the most precise rule on the holdout set.
- The pruning process is more flexible because any part of the antecedent may be pruned.
- Duplicate rules are removed.
- The rules after pruning are not mutually exclusive.
- The ordering of the rules is necessary.
- Rare classes and less complex rules or rules with less false positives are prioritized.

Rule-based classification - Rule generation

Sequential Covering Algorithm:

- An algorithm for creation of ordered set of rules.
- An 2-step iterative algorithm:
  - **Learn-one-rule** – select particular class and determine the "best" rule from the current training instances S with this class as a consequent. Add this rule to the bottom of the ordered rule list.
  - **Prune training data** – Remove training instances in S that are covered by the rule generated in previous step. The detection is based on the antecedent only, that consequent of the instances is ignored.

Rule-based classification - Rule generation - Sequential Covering Algorithm

The ordering of the generated rules:

- Class-based ordering
  - All rules for particular class are put together.
  - Rare classes may be prioritizes.
  - All rules for this particular class are generated continuously, until a termination criterion is met.
  - For k-class problem, $k-1$ rule sets is generated and the final catch-all rule covers the last class.
- Quality-based ordering
  - The rule are selected according a measure, such as confidence or support.
  - The catch-all rule corresponds to the dominant class among remaining instances.
  - The quality of very difficult to measure.

Rule-based classification - Rule generation - Sequential Covering Algorithm

Learn-one-rule step:

- Iterative algorithm that grows a rule with best conjunct according the quality measure.
- The simplest quality is the precision/accuracy.
- Each split choice (conjunct) is evaluated the same was as it is in trees.
- Several best options may be maintained to reduce the possibility of the mistakes and suboptimal rules.
- The ideal quality measure must combine accuracy and coverage, e.g. Laplace smoothing, like-hood ratio statistics, FOIL information gain.

Rule-based classification - Rule generation

Rule pruning:

- An Minimum description length (MDL) principle is one option.
- A penalty based on MDL may be used in rule-growth phase.
- An holdout set is another good principle.
- A greedy algorithm may be used for conjunct evaluation.

Questions?