Dimensionality reduction
Feature selection

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Dimensionality reduction. Motivation.

- ML methods are sensitive to the dimensionality $d$ of data
- **Question:** Is there a lower dimensional representation of the data that captures well its characteristics?
- **Objective of dimensionality reduction:**
  - Find a lower dimensional representation of data
- **Two learning problems:**
  - Supervised
    - $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$
    - $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$
  - Unsupervised
    - $D = \{x_1, x_2, \ldots, x_n\}$
    - $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$
- **Goal:** replace $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$
  with $x_i'$ of dimensionality $d' < d$
Dimensionality reduction for classification

• Classification problem example:
  \[ D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \]
  \[ x_i = (x_i^1, x_i^2, \ldots, x_i^d) \]
  \[ f: x \rightarrow y \]
  – Assume the dimension \( d \) of the data point \( x \) is very large

• Problems with high dimensional input vectors
  – A large number of parameters to learn, if a dataset is small this can result in:
    • A large variance of estimates and overfit
  – it becomes hard to explain what features are important in the model (too many choices, some can be substitutable)

---

Dimensionality reduction

• Solutions:
  – Selection of a smaller subset of inputs (features) from a large set of inputs; train classifier on the reduced input set
  – Combination of high dimensional inputs to a smaller set of features \( \phi_h(x) \); train classifier on new features
Feature selection

How to find a good subset of inputs/features?

• **We need:**
  – A criterion for ranking good inputs/features
  – Search procedure for finding a good set of features

• **Feature selection process can be:**
  – **Dependent on the learning task**
    • e.g. classification
    • Selection of features affected by what we want to predict
  – **Independent of the learning task**
    • Unsupervised methods
    • may lack the accuracy for classification/regression tasks

Task-dependent feature selection

**Assume:** **Classification problem:**
  – x – input vector, y - output

**Objective:** Find a subset of inputs/features that gives/preserves most of the output prediction capabilities

**Selection approaches:**

• **Filtering approaches**
  – Filter out features with small predictive potential
  – Done before classification; typically uses univariate analysis

• **Wrapper approaches**
  – Select features that directly optimize the accuracy of the multivariate classifier

• **Embedded methods**
  – Feature selection and learning closely tied in the method
  – Regularization methods, decision tree methods
Feature selection through filtering

Assume:

Classification problem:
- $x$ – input vector, $y$ - output

- How to select the features/inputs?
  Univariate analysis
  - Pretend that only one input $x_k$, exists
  - Calculate a score reflecting how well $x_k$ predicts the output $y$ alone
  - Repeat the above analysis and scores for all inputs
  - Pick the inputs best scores
    (or eliminate/filter the inputs with the worst scores)

Feature scoring for classification

- Scores for measuring the differential expression
  - T-Test score (Baldi & Long)
    - Based on the test that two groups come from the same population
    - Null hypothesis: is mean of class 0 = mean of class 1
Feature scoring for classification

Scores for measuring the differential expression

• Fisher Score

\[ Fisher(i) = \frac{(\mu^{(+)} - \mu^{(-)})^2}{\sigma^{(+)} + \sigma^{(-)}} \]

- **AUROC score**: Area under Receiver Operating Characteristic curve

Feature scoring

• Correlation coefficients
  - Measures linear dependences

\[ \rho(x_k, y) = \frac{Cov(x_k, y)}{\sqrt{Var(x_k)Var(y)}} \]

• Mutual information
  - Measures dependences
  - Needs discretized input values

\[ I(x_k, y) = \sum_i \sum_j \tilde{P}(x_k = j, y = i) \log_2 \frac{\tilde{P}(x_k = j, y = i)}{\tilde{P}(x_k = j)\tilde{P}(y = i)} \]
Feature set scoring

Problems:
• Univariate score assumptions:
  – Only one input and its effect on \( y \) is incorporated in the score
  – Effects of two features on \( y \) are considered to be independent

Partial solution:
• Correlation based feature selection
  • Idea: good feature subsets contain features that are highly correlated with the class but independent of each other
  • Assume a set of features \( S \). Then
    \[
    M(S) = \frac{k \bar{r}_{yx}}{\sqrt{k + k(k + 1) \bar{r}_{xx}}}
    \]
  • Average correlation between \( x \) and class \( y \) \( \bar{r}_{yx} \)
  • Average correlation between pairs of \( x \) \( \bar{r}_{xx} \)

Feature selection

Problems:
• Many inputs and low sample size
  – if many random features, and not many instances we can learn from, the features with a good differentially expressed score may arise simply by chance
  – The probability of this happening can be quite large
• Techniques to address the problem:
  – reduce FDR (False discovery rate) and
  – FWER (Family wise error).