Feature extraction and selection

Based on slides Martina Bachlera martin.bachler@igi.tugraz.at, Makoto Miwa
And paper Isabelle Guyon, André Elisseeff: An Introduction to variable and feature selection. JMLR, 3 (2003) 1157-1182
Overview

- Introduction/Motivation
- Basic definitions, Terminology
- Variable Ranking methods
- Feature subset selection

WHY ?
WHAT ?
HOW ?
Problem: **Where to focus attention?**

- A universal problem of intelligent (learning) agents is where to focus their attention.

- What aspects of the problem at hand are important/necessary to solve it?

- Discriminate between the relevant and irrelevant parts of experience.
What is **feature selection**?

- **Feature selection**: Problem of selecting some subset of a learning algorithm’s input variables upon which it should focus attention, while ignoring the rest (DIMENSIONALITY REDUCTION)

- **Humans/animals do that constantly!**
Monkeys performing classification task

Motivational example from Biology

Motivational example from Biology

Monkeys performing classification task

All considered features:
- Eye height
- Eye separation
- Nose length
- Mouth height

How many pairs of features?

Diagnostic features:
- Eye height
- Eye separation

Non-Diagnostic features:
- Nose length
- Mouth height
Motivational example from Biology

Monkeys performing classification task

Results:

- activity of a population of 150 neurons in the anterior inferior temporal cortex was measured

- 44 neurons responded significantly differently to at least one feature

- After Training: 72% (32/44) were selective to one or both of the diagnostic features (and not for the non-diagnostic features)
Why even think about Feature Selection in ML?

- The information about the target class is **inherent in the variables**!

- Naive theoretical view:
  More features
  => More information
  => More discrimination power.

- In practice:
  many reasons why this is not the case!

- Also:
  Optimization is (usually) good, so why not try to optimize the input-coding?
Introduction

- Large and high-dimensional data
  - Web documents, etc...
  - A large amount of resources are needed in
    - Information Retrieval
    - Classification tasks
    - Data Preservation etc...

Dimension Reduction
Dimension Reduction

- preserves information on classification of overweight and underweight as much as possible
- makes classification easier
- reduces data size (2 features → 1 feature)
Dimension Reduction

- Feature Extraction (FE)
  - Generates feature
  - ex.
    - Preserves weight / height

- Feature Selection (FS)
  - Selects feature
  - ex.
    - Preserves weight
Problem Setting

- Each of data $X$ ($n$ samples) is represented by $d$ features
- Data belong to $c$ different classes in supervised learning
- Dimension reduction is to generate or select $p$ features preserving original information as much as possible in some criterion

$$1 < p \approx c \ll d < n$$
Feature Extraction

- Extracts features by projecting data to a lower-dimensional space

  - Unsupervised Method
    - Principal Component Analysis (PCA)
    - Independent Component Analysis (ICA)

  - Supervised Method
    - Linear Discriminant Analysis (LDA)
    - Maximum Margin Criterion (MMC)
    - Orthogonal Centroid algorithm (OC)

- Finds an optimal projection matrix $W$
Principal Component Analysis

- Unsupervised Method
- PCA tries to maximize

\[ J(W) = \text{trace}(W^T C W) \]

- PCA needs Singular Value Decomposition calculation (SVD).
  - time complexity: \( O(n^2 d) \)
  - space complexity: \( O(nd) \)

\( C \): covariance matrix
Linear Discriminant Analysis

Supervised method

Time complexity
$O((n + c)^2d)$

Space complexity
$O(nd)$

$S_b$  Interclass scatter matrix
$S_w$: Intra-class scatter matrix
Feature Selection in ML ? YES!

- Many explored domains have **hundreds** to **tens of thousands** of variables/features with many irrelevant and redundant ones!

- In domains with many features the underlying probability distribution can be very complex and very hard to estimate (e.g. dependencies between variables)!

- Irrelevant and redundant features can „confuse“ learners!

- Limited training data!

- Limited computational resources!

- **Curse of dimensionality**!
Curse of dimensionality

- The required number $m$ of samples (to achieve the same accuracy) grows exponentially with the number of variables! PAC: $m > |\text{Hypothesis_space}|$

- In practice: number of training examples is fixed!

  => the classifier’s performance usually will degrade for a large number of features!

In many cases

  - the information that is lost by discarding variables
  - is made up for by
  - a more accurate mapping/sampling in the lower-dimensional space!
Věta o PAC učení rozhodovacího stromu

Nechť objekty jsou charakterizovány pomocí $n$ binárních atributů a nechť připouštíme jen hypotézy ve tvaru rozhodovacího stromu s maximální délkou větve $k$. Dále nechť $\delta, \varepsilon$ jsou malá pevně zvolená kladná čísla blízká 0. Pokud algoritmus strojového učení vygeneruje hypotézu $\varphi$, která je konzistentní se všemi $m$ příklady trénovací množiny a platí

$$m \geq m_{k-DT}(n) \geq c \left( n^k + \ln \left( \frac{1}{\delta} \right) \right) / \varepsilon$$

pak $\varphi$ je $\varepsilon$-skoro správná hypotéza s pravděpodobností větší než $(1-\delta)$, t.j. chyba hypotézy $\varphi$ na celém definičním oboru konceptu je menší než $\varepsilon$ s pravděpodobností větší než $(1-\delta)$. 
Gene selection from microarray data

- **Variables:**
  - gene expression coefficients corresponding to the amount of mRNA in a patient's sample (e.g. tissue biopsy)

- **Task:** Separate healthy patients from cancer patients

- Usually there are only about 100 examples (patients) available for training and testing (!!!)
- Number of variables in the raw data: 6,000 – 60,000
- Does this work? ([8])

---

Text-Categorization

- Documents are represented by a vector containing word frequency counts (its size ~ number of features is comparable to that of the vocabulary)

- Vocabulary ~ 15,000 words (i.e. each document is represented by a 15,000-dimensional vector)

- Typical tasks:
  - Automatic sorting of documents into web-directories
  - Detection of spam-email
Motivation

- Especially when dealing with a large number of variables there is a need for **dimensionality reduction**!

- Feature Selection can significantly improve a learning algorithm’s performance!
Overview

- Introduction/Motivation
- Basic definitions, Terminology
- Variable Ranking methods
- Feature subset selection
Given a set of features \( F = \{ f_1, \ldots, f_i, \ldots, f_n \} \),

the Feature Selection problem is to find a subset \( F' \subseteq F \) that “maximizes the learners ability to classify patterns”.

\[
\{ f_1, \ldots, f_i, \ldots, f_n \} \xrightarrow{f \text{. selection}} \{ f_{i_1}, \ldots, f_{i_j}, \ldots, f_{i_m} \}
\]

\[
i_j \in \{1, \ldots, n\}; \ j = 1, \ldots, m\]
\[
i_a = i_b \Rightarrow a = b; \ a, b \in \{1, \ldots, m\}\]
Given a set of features \( F = \{ f_1, \ldots, f_i, \ldots, f_n \} \),

the Feature Extraction ("Construction") problem is to map \( F \) to some feature set \( F' \) that maximizes the learner’s ability to classify patterns (design new derived attributes).
Feature Selection – Optimality?

- In theory the goal is to find an optimal feature-subset (one that maximizes the scoring function)

- In real world applications this is usually not possible
  - For most problems it is computationally intractable to search the whole space of possible feature subsets
  - One usually has to settle for approximations of the optimal subset
  - Most of the research in this area is devoted to finding efficient search-heuristics
Optimal feature subset

- Often: Definition of optimal feature subset in terms of classifier’s performance

- The best one can hope for theoretically is the Bayes error rate

- Given a learner $I$ and training data $L$ with features $F = \{f_1, \ldots, f_i, \ldots, f_n\}$ an optimal feature subset $F_{opt}$ is a subset of $F$ such that the accuracy of the learner’s hypothesis $h$ is maximal (i.e. its performance is equal to an optimal Bayes classifier)*.

  - $F_{opt}$ (under this definition) depends on $I$
  - $F_{opt}$ need not be unique
  - Finding $F_{opt}$ is usually computationally intractable

* for this definition a possible scoring function is $1 - true\_error(h)$
Relevance of features

- Relevance of a variable/feature:
  - There are several definitions of relevance in literature:
    - Relevance of 1 variable,
    - Relevance of a variable given other variables,
    - Relevance given a certain learning algorithm,..

- Most definitions are problematic, because there are problems where all features would be declared to be irrelevant.

- The authors of [2] define two degrees of relevance: weak and strong relevance.

- A feature is relevant iff it is weakly or strongly relevant and "irrelevant" (redundant) otherwise.

Relevance of features

- **Strong Relevance** of a variable/feature:
  
  Let $S_i = \{f_1, \ldots, f_{i-1}, f_{i+1}, \ldots, f_n\}$ be the set of all features except $f_i$. Denote by $s_i$ a value-assignment to all features in $S_i$.

  A feature $f_i$ is strongly relevant, iff removal of $f_i$ alone will always result in a performance deterioration of an optimal Bayes classifier.

- **Weak Relevance** of a variable/feature:

  A feature $f_i$ is weakly relevant, iff it is not strongly relevant, and there exists a subset of features $S_i'$ of $S_i$, for which there exists a subset of features $S_i''$, such that the performance of an optimal Bayes classifier on $S_i''$ is worse than on $S_i' \cup \{f_i\}$.
Relevance of features

- Relevance \( \not\implies \) Optimality of Feature-Set
  - Classifiers induced from training data are likely to be suboptimal (no access to the real distribution of the data)
  - Relevance does not imply that the feature is in the optimal feature subset
  - Even “irrelevant” features can improve a classifier’s performance
  - Defining relevance in terms of a given classifier (and therefore a hypothesis space) would be better.
Overview

- Introduction/Motivation
- Basic definitions, Terminology
- Variable Ranking methods
- Feature subset selection
Variable Ranking

Given a set of features $F$

Variable Ranking is the process of ordering the features by the value of some scoring function $S: F \rightarrow \Omega$ (which usually measures feature-relevance)

Resulting set: a permutation of $F$: $F' = \{ f_{i_1}, \ldots, f_{i_j}, \ldots f_{i_n} \}$ with

$$S(f_{i_j}) \geq S(f_{i_{j+1}}); \quad j = 1, \ldots, n-1;$$

The score $S(f_i)$ is computed from the training data, measuring some criteria of feature $f_i$.

By convention a high score is indicative for a valuable (relevant) feature.
A simple method for feature selection using variable ranking is to select the \( k \) highest ranked features according to \( S \).

This is usually not optimal.

but often preferable to other, more complicated methods.

computationally efficient(!): only calculation and sorting of \( n \) scores
Ranking Criteria – Correlation

Correlation Criteria:

- Pearson correlation coefficient

\[ R(f_i, y) = \frac{\text{cov}(f_i, y)}{\sqrt{\text{var}(f_i) \text{ var}(y)}} \]

- Estimate for \( m \) samples:

\[ R(f_i, y) = \frac{\sum_{k=1}^{m} (f_{k,i} - \bar{f_i})(y_k - \bar{y})}{\sqrt{\sum_{k=1}^{m} (f_{k,i} - \bar{f_i})^2 \sum_{k=1}^{m} (y_k - \bar{y})^2}} \]

The higher the correlation between the feature and the target, the higher the score!
Ranking Criteria – Correlation

- $r = 1$: Perfect (linear) correlation
- $r = 0.5$: Intermediate correlation
- $r = 0$: No correlation
- $r = -1$: Perfect (linear) inverse correlation
Correlation Criteria:

- $\rho_{xy} \in [-1,1]$  

- mostly $R(x_i,y)^2$ or $|R(x_i,y)|$ is used 

- measure for the goodness of \textbf{linear} fit of $x_i$ and $y$.  
  (can only detect \textit{linear dependencies} between variable and target.) 

- what if $y = XOR(x_1,x_2)$? 

- often used for microarray data analysis
Ranking Criteria – Correlation

Questions:

- Can variables with **small score** be automatically discarded?

- Can a useless variable (i.e. one with a small score) be useful together with others?

- Can two variables that are useless by themselves can be useful together?)
Ranking Criteria – Correlation

• Can variables with small score be discarded without further consideration? **NO!**

• Even variables with small score can improve class separability!

• Here this depends on the correlation between $x_1$ and $x_2$.

(Here the class conditional distributions have a high covariance in the direction orthogonal to the line between the two class centers)
• Example with high correlation between $x_1$ and $x_2$.

(Here the class conditional distributions have a high covariance in the direction of the two class centers)

• No gain in separation ability by using two variables instead of just one!
Can a useless variable be useful together with others?

YES!
• correlation between variables and target are not enough to assess relevance!

• correlation / covariance between pairs of variables has to be considered too!

(potentially difficult)

• diversity of features
Information Theoretic Criteria

- Most approaches use (empirical estimates of) mutual information between features and the target:

\[
I(x_i, y) = \int \int p(x_i, y) \log \frac{p(x_i, y)}{p(x_i)p(y)} \, dx \, dy
\]

- Case of discrete variables:

\[
I(x_i, y) = \sum_{x_i} \sum_{y} P(X = x_i, Y = y) \log \frac{P(X = x_i, Y = y)}{P(X = x_i)P(Y = y)}
\]

(probabilities are estimated from frequency counts)
Mutual information can also detect non-linear dependencies among variables!

But harder to estimate than correlation!

It is a measure for “how much information (in terms of entropy) two random variables share”
Variable Ranking - SVC

**Single Variable Classifiers**

- Idea: Select variables according to their *individual predictive power*
- criterion: Performance of a classifier built with 1 variable
- e.g. the value of the variable itself
  (set threshold on the value of the variable)
- predictive power is usually measured in terms of error rate (or criteria using fpr, fnr)
- also: combination of SVCs using ensemble methods
  (boosting, ...)


Overview

- Introduction/Motivation
- Basic definitions, Terminology
- Variable Ranking methods
- Feature subset selection
Feature Subset Selection

- Goal:
  - Find the optimal feature subset.
  - (or at least a “good one.”)

- Classification of methods:
  - Filters
  - Wrappers
  - Embedded Methods
Feature Subset Selection

- You need:
  - a measure for assessing the goodness of a feature subset (scoring function)
  - a strategy to search the space of possible feature subsets

- Finding a minimal optimal feature set for an arbitrary target concept is NP-hard

=> Good heuristics are needed!

Feature Subset Selection

- **Filter Methods**
  
  Select subsets of variables as a pre-processing step, independently of the used classifier!!

- Note that Variable Ranking-FS is a filter method
Feature Subset Selection

- Filter Methods
  - usually fast
  - provide generic selection of features, not tuned by given learner (universal)
  - this is also often criticised (feature set not optimized for used classifier)
  - sometimes used as a preprocessing step for other methods
Feature Subset Selection

- **Wrapper Methods**
  - Learner is considered a black-box
  - Interface of the black-box is used to score subsets of variables according to the predictive power of the learner when using the subsets.
  - Results vary for different learners
  - One needs to define:
    - how to search the space of all possible variable subsets?
    - how to assess the prediction performance of a learner?
Feature Subset Selection

Wrapper Methods
Feature Subset Selection

- **Wrapper Methods**
  - The problem of finding the optimal subset is NP-hard!
  - A wide range of heuristic search strategies can be used. Two different classes:
    - **Forward selection**
      (start with empty feature set and add features at each step)
    - **Backward elimination**
      (start with full feature set and discard features at each step)
  - Predictive power is usually measured on a validation set or by cross-validation
  - By using the learner as a black box wrappers are universal and simple!
  - Criticism: a large amount of computation is required.
Feature Subset Selection

- **Embedded Methods**
  - Specific to a given learning machine!
  - Performs variable selection (implicitly) in the process of training
  - E.g. WINNOW-algorithm
  - (linear unit with multiplicative updates)
Important points 1/2

• Feature selection can significantly increase the performance of a learning algorithm (both accuracy and computation time) – but it is not easy!

• One can work on problems with very high-dimensional feature-spaces

• Relevance <-> Optimality

• Correlation and Mutual information between single variables and the target are often used as Ranking-Criteria of variables.
Important points 2/2

- One can not automatically discard variables with small scores – they may still be useful together with other variables.

- Filters – Wrappers - Embedded Methods

- How to search the space of all feature subsets?

- How to assess performance of a learner that uses a particular feature subset?
THANK YOU!
Sources


9. E. Amaldi, V. Kann: The approximability of minimizing nonzero variables and unsatisfied relations in linear systems. (1997)