Feature selection

V. Kitov

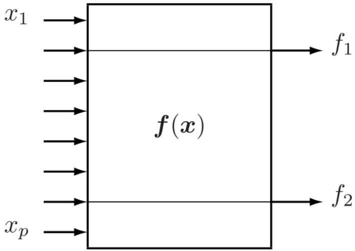
Yandex School of Data Analysis

Imperial College London
Department of Physics

January 2015

Feature selection

Feature selection is a process of selecting a subset of original features with minimum loss of information related to final task (classification, regression, etc.)



(a) feature selector

(b) feature extractor

Applications of feature selection

- increase predictive accuracy of classifier
- increase computational efficiency
- reduce cost of future data collection
- make classifier more interpretable

Types of features

- Let f be the feature, $\chi = \{f_1, f_2, ...f_D\}$ is the full set of features, $S = \chi \backslash \{f\}$
- Strongly relevant feature:
- $$p(y|f, S) \neq p(y|S)$$
- Weakly relevant feature:

$$p(y|f, S) = p(y|S), \text{ but } \exists S' \subset S : \; p(y|f, S') = p(y|S')$$

- Irrelevant feature:

$$\forall S' \subset S : \; p(y|f, S') = p(y|S')$$

Complexity

- We seek optimal subset of m features $\hat{F}_m$
- χ_m is a set of all subsets of features of size m
- It is equal to

$$\hat{F}_m = \arg\max_{F \in \chi_m} J(X)$$

- Requires $\binom{D}{m}$ checks!

Types of feature selection algorithms

- Completeness of search:
  - Optimal
  - Suboptimal
    - deterministic
    - random
- Classifier dependency
  - independent (filter methods)
  - uses classifier output (wrapper methods)
  - is embedded inside classifier (embedded methods)

Properties of each type

- filter methods
  - rely only on measures of dependency between features and output
  - do not take final quality measure into account (like misclassification rate)
  - are computationally efficient
- wrapper methods
  - subsets of variables are evaluated with respect to the quality of final classification
  - give better performance than filter methods
  - more computationally demanding
- embedded methods
  - feature selection is built into the classifier
  - feature selection and model tuning are done jointly
  - example: classification trees, methods with L_1 regularization.

Specification

- Need to specify:
    - quality criteria $J(X)$
    - subset generation method $X_1, X_2, X_3, ...$

Table of Contents

Correlation

- two class:

$$\rho(f, y) = \frac{\sum_i (f_i - \bar{f})(y_i - \bar{y})}{\left[\sum_i (f_i - \bar{f})^2 \sum_i (y_i - \bar{y})^2\right]^{1/2}}$$

- multiclass $\omega_1, \omega_2, ...\omega_C$

$$R^2 = \frac{\sum_{c=1}^{C} \left[\sum_i (f_i - \bar{f})(y_{ic} - \bar{y}_c)\right]^2}{\sum_{c=1}^{C} \sum_i (f_i - \bar{f})^2 \sum_i (y_{ic} - \bar{y}_c)^2}$$

- Properties:
  - simple to compute
  - takes into account only linear relationships

Mutual information

- Entropy of feature X:

$$H(X) = -\sum_x p(x) \ln p(x)$$

- Entropy of X after observing Y:

$$H(X|Y) = -\sum_y p(y) \sum_x p(x|y) \ln p(x|y)$$

- Mutual information - how much Y gives information about X:

$$\begin{aligned} MI(X,Y) &= H(X) - H(X|Y) \\ &= \sum_{x,y} p(x,y) \ln \left[\frac{p(x,y)}{p(x)p(y)}\right] \\ &= H(Y) - H(Y|X) = MI(Y,X) \end{aligned}$$

- MI is Kullback-Leibler divergence, so it is non-negative

Mutual information

- Symmetrical uncertainty:

$$SU(X, Y) = 2\left(\frac{MI(X, Y)}{H(X) + H(Y)}\right)$$

- $SU(X, Y)$ lies between 0 (independence) and 1 (full dependence)
- Properties of MI and SU:
  - identifies arbitrary non-linear dependencies
  - requires calculation of probability distributions
  - continuous variables need to be discretized

Other criteria

- Probabilistic distance: $p(x|\omega_1)$ vs. $p(x|\omega_2)$
- Probabilistic dependence: $p(x|\omega_i)$ vs. $p(x)$
- Metric separability of classes:
  - $S_W = \sum_{c=1}^{C} \frac{N_c}{N} \Sigma_c$, $S_B = \sum_{c=1}^{C} \frac{N_c}{N}(m_j - m)(m_j - m)$
  - metrics: $Tr\{S_W^{-1} S_B\}$, $\frac{Tr\{S_B\}}{Tr\{S_W\}}$

Table of Contents

Complete search with optimal solution

- exhaustive search
- branch and bound method
  - requires monotonicity property:

$$F \subset G : J(F) < J(G)$$

  - example

Incomplete search with suboptimal solution

- Order features with respect to $J(f)$:

$$J(f_1) \geq J(f_2) \geq ... \geq J(f_D)$$

  - select top m

$$\hat{F} = \{f_1, f_2, ... f_m\}$$

  - select best set from nested subsets:
    $S = \{\{f_1\}, \{f_1, f_2\}, ... \{f_1, f_2, ... f_D\}\}$

$$\hat{F} = \arg\max_{F \in S} J(F)$$

Sequential search

- Sequential forward selection algorithm:
  - init: $k = 0$, $F_0 = \emptyset$
  - while k<max_features:
    - $f_{k+1} = \arg\max_{f \in \chi} J(F_k \cup \{f\})$
    - $F_{k+1} = F_k \cup \{f_{k+1}\}$
    - if $J(F_{k+1}) < J(F_{k-1})$: break
  - return F_k
- Variants:
  - sequential backward selection
  - up-k forward search
  - down-p backward search
  - up-k down-p composite search
  - up-k down-(variable step size) composite search

Other

- Random feature sets selection:
  - new feature subsets are generated completely at random
    - does not get stuck in local optimum
    - low probability to locate small optimal feature subset
  - sequential procedure of feature subset creation with inserted randomness
    - may get stuck in local optimum
    - more efficiently locates small optimal feature subsets
- Compositions
- Stability measures:
  - different algorithms
  - different subsamples