Feature selection

© Victor Kitov
v.v.kitov@yandex.ru

Summer school on Machine Learning in High Energy Physics
in partnership with

Yandex

August 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

- Why feature selection?
  - increase predictive accuracy of classifier
  - improve optimization stability by removing multicollinearity
  - increase computational efficiency
  - reduce cost of future data collection
  - make classifier more interpretable

- Not always necessary step:
  - some methods have implicit feature selection
    - decision trees and tree-based (RF, ERT, boosting)
  - regularization
Types of features

Define \( f \) - the feature, \( F = \{ f_1, f_2, \ldots, f_D \} \) - full set of features, \( S = F \setminus \{ f \} \).

- **Strongly relevant feature:**
  \[ p(y|f, S) \neq p(y|S) \]

- **Weakly relevant feature:**
  \[ p(y|f, S) = p(y|S), \text{ but } \exists S' \subset S : p(y|f, S') \neq p(y|S') \]

- **Irrelevant feature:**
  \[ \forall S' \subset S : p(y|f, S') = p(y|S') \]

**Aim of feature selection**

Find minimal subset \( S \subset F \) such that \( P(y|S) \approx P(y|F) \), i.e. leave only relevant and non-redundant features.
Refining redundant features

Consider a set of interrelated random variables \(Z = \{z_1, z_2, ...z_D\}\)

**Definition 1**

Subset \(S\) of \(Z\) is called a *Markov blanket of \(z_i\)* if

\[P(z_i|S, Z) = P(z_i|S)\]

- For Markov network Markov blanket consists of all nodes connected to \(Y\).
- For Bayesian network Markov blanket consists of: parents, children and children other parents.

Only features from Markov blanket of \(y\) inside set \(\{f_1, f_2, ...f_D, y\}\) are needed.

Markov blanket may be found by special algorithms such as IAMB.
Specification

Need to specify:

- quality criteria $J(X)$
- subset generation method $S_1, S_2, S_3, ...$
Types of feature selection algorithms

- Completeness of search:
  - Complete
    - exhaustive search complexity is $C_D^d$ for $|F| = D$ and $|S| = d$.
  - Suboptimal
    - deterministic
    - random (deterministic with randomness / completely random)

- Integration with predictor
  - independent (filter methods)
  - uses predictor quality (wrapper methods)
  - is embedded inside classifier (embedded methods)
Classifier dependency types

- **filter methods**
  - rely only on general measures of dependency between features and output
  - more universal
  - 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.
Table of Contents

1 Filter methods
   - Probability measures
   - Context relevant measures

2 Feature subsets generation
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 \(\omega_1, \omega_2, ... \omega_C\) (micro averaged \(\rho(f, y_c)\) \(c = 1, 2, ... C\).)

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

- Benefits:
  - simple to compute
  - applicable both to continuous and discrete features/output.
  - does not require calculation of p.d.f.
Correlation for non-linear relationship

- Correlation captures only linear relationship
- Example: $X \sim \text{Uniform}[-1, 1], \ Y = X^2$:

$$\mathbb{E} \{(X - \mathbb{E}X)(Y - \mathbb{E}Y)\} = \mathbb{E} \{X(X^2 - \mathbb{E}X^2)\} = \mathbb{E}X^3 - \mathbb{E}X \mathbb{E}X^2 = 0$$

- Other examples of data and its correlation:
Entropy

- Entropy of random variable $Y$:

$$H(Y) = - \sum_y p(y) \ln p(y)$$

- level of uncertainty of $Y$
- proportional to the average number of bits needed to code the outcome of $Y$ using optimal coding scheme ($- \ln p(y)$ for outcome $y$).

- Entropy of $Y$ after observing $X$:

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

For two p.d.f. $P(x)$ and $Q(x)$ Kullback-Leibler divergence $KL(P||Q)$ equals $\sum_x P(x) \ln \frac{P(x)}{Q(x)}$

- Properties:
  - defined only for $P(x)$ and $Q(x)$ such that $Q(x) = 0 \Rightarrow P(x) = 0$
  - $KL(P||Q) \geq 0$
  - $P(x) = Q(x) \forall x$ if and only if $KL(P||Q) = 0$ (for discrete r.v.)
  - $KL(P||Q) \neq KL(Q||P)$
Kullback-Leibler divergence

- Symmetrical distance: \( KL_{sym}(P||Q) = KL(P||Q) + KL(Q||P) \)
- Information theoretic meaning:
  - true data distribution \( P(x) \)
  - estimated data distribution \( Q(x) \)

\[
KL(P||Q) = - \sum_x P(x) \ln Q(x) + \sum_x P(x) \ln P(x)
\]

- \( KL(P||Q) \) shows how much longer will be the average length of the code word.
Mutual information

Mutual information measures how much $X$ gives information about $Y$:

$$MI(X, Y) = H(Y) - H(Y|X)$$

$$= \sum_{x,y} p(x, y) \ln \left( \frac{p(x, y)}{p(x)p(y)} \right)$$

Properties:

- $MI(X, Y) = MI(Y, X)$
- $MI(X, Y) = KL(p(x, y), p(x)p(y)) \geq 0$
- $MI(X < Y) \leq \min \{H(X), H(Y)\}$
- $X, Y$- independent, then $MI(X, Y) = 0$
- $X$ completely identifies $Y$, then $MI(X, Y) = H(Y) \leq H(X)$
Mutual information for feature selection

- Normalized variant: \( NMI(X, Y) = \frac{MI(X, Y)}{H(Y)} \) equals
  - zero, when \( P(Y|X) = P(Y) \)
  - one, when \( X \) completely identifies \( Y \).

- Properties of \( MI \) and \( NMI \):
  - identifies arbitrary non-linear dependencies
  - requires calculation of probability distributions
  - continuous variables need to be discretized
Filter methods

- Probability measures
- Context relevant measures
Probabilistic distance

Measure of relevance: $p(x|\omega_1)$ vs. $p(x|\omega_2)$
Examples of distances

Distances between probability density functions $f(x)$ and $g(x)$:

- **Total variation**: $\frac{1}{2} \int |f(x) - g(x)| \, dx$,
- **Euclidean**: $\frac{1}{2} \left( \int (f(x) - g(x))^2 \, dx \right)^{1/2}$
- **Hellinger**: $\left( \frac{1}{2} \int \left( \sqrt{f(x)} - \sqrt{g(x)} \right)^2 \, dx \right)^{1/2}$
- **Symmetrical KL**: $\int (f(x) - g(x)) \ln \frac{f(x)}{g(x)} \, dx$

Distances between cumulative probability functions: $F(x)$ and $G(x)$:

- **Kolmogorov**: $\sup_x |F(x) - G(x)|$
- **Kantorovich**: $\int |F(x) - G(x)| \, dx$
- **$L_p$**: $\left( \int |F(x) - G(x)|^p \, dx \right)^{1/p}$
Other

Multiclass extensions:
Suppose, we have a distance score $J(\omega_i, \omega_j)$. We can extend it to multiclass case using:

$$J = \max_{\omega_i, \omega_j} J(\omega_i, \omega_j)$$

$$J = \sum_{i < j} p(\omega_i)p(\omega_j)J(\omega_i, \omega_j)$$

Comparison with general p.d.f:
We can also compare $p(x|\omega_i)$ vs. $p(x)$ using $J = \sum_{i=1}^{C} p(\omega_i)D(p(x|\omega_i), p(x))$:

- Chernoff: $J = \sum_{i=1}^{C} p(\omega_i) \left\{ -\log \int p^s(x|\omega_i)p^{1-s}(x) \right\} dx$
- Bhattacharyya: $J = \sum_{i=1}^{C} p(\omega_i) \left\{ -\log \int (p(x|\omega_i)p(x))^{1/2} \right\} dx$
- Patrick-Fisher: $J = \sum_{i=1}^{C} p(\omega_i) \left\{ \int [p(x|\omega_i) - p(x)]^2 dx \right\}^{1/2}$
1 Filter methods

- Probability measures
- Context relevant measures
Relevance in context

Individually features may not predict the class, but may be relevant together:

\[ p(y|x_1) = p(y), \; p(y|x_2) = p(y), \; \text{but} \; p(y|x_1, x_2) \neq p(y) \]
Relief criterion

**INPUT:**
Training set \((x_1, y_1), (x_2, y_2), \ldots (x_N, y_N)\)
Number of neighbours \(K\)
Distance metric \(d(x, x')\) # usually Euclidean

for each pattern \(x_n\) in \(x_1, x_2, \ldots x_N\):
    calculate \(K\) nearest neighbours of the same class \(y_i\):
    \(x_{s(n,1)}, x_{s(n,2)}, \ldots x_{s(n,K)}\)
    calculate \(K\) nearest neighbours of class different from \(y_i\):
    \(x_{d(n,1)}, x_{d(n,2)}, \ldots x_{d(n,K)}\)

for each feature \(f_i\) in \(f_1, f_2, \ldots f_D\):
    calculate relevance \(R(f_i) = \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{|x_{n}^{i} - x_{d(n,k)}^{i}|}{|x_{n}^{i} - x_{s(n,k)}^{i}|}\)

**OUTPUT:**
feature relevances \(R\)
Cluster measures

General idea of cluster measures

Feature subset is good if observations belonging to different classes group into different clusters.
Cluster measures

Define:

- $z_{ic} = \mathbb{I}[y_i = \omega_c]$, $N$-number of samples, $N_i$-number of samples belonging to class $\omega_i$.
- $m = \frac{1}{N} \sum_i x_i$, $m_c = \frac{1}{N_c} \sum_i z_{ic} x_i$, $j = 1, 2, \ldots, C$.
- Global covariance: $\Sigma = \frac{1}{N} \sum_i (x - m)(x - m)^T$.
- Intraclass covariances: $\Sigma_c = \frac{1}{N_c} \sum_i z_{ic} (x_i - m_c)(x_i - m_c)^T$.
- Within class covariance: $S_W = \sum_{c=1}^C \frac{N_c}{N} \Sigma_c$.
- Between class covariance: $S_B = \sum_{c=1}^C \frac{N_c}{N} (m_j - m)(m_j - m)$.

Interpretation

Within class covariance shows how samples are scattered within classes.
Between class covariance shows how classes are scattered between each other.
Theorem 1

Every real symmetric matrix $A \in \mathbb{R}^{n \times n}$ can be factorized as

$$A = U\Sigma U^T$$

where $\Sigma$ is diagonal and $U$ is orthogonal. $\Sigma = \text{diag}\{\lambda_1, \lambda_2, ... \lambda_n\}$ and $U = [u_1, u_2, ... u_n]$ where $\lambda_i, i = 1, 2, ... n$ are eigenvalues and $u_i \in \mathbb{R}^{n \times 1}$ are corresponding eigenvectors.

- $U^T$ is basis transform corresponding to rotation, so only $\Sigma$ reflects scatter.
- Aggregate measures of scatter $\text{tr } \Sigma = \sum_i \lambda_i$ and $\det \Sigma = \prod_i \lambda_i$
- Since $\text{tr } [P^{-1}BP] = \text{tr } B$ and $\det [P^{-1}BP] = \det B$, we can estimate scatter with $\text{tr } A = \text{tr } \Sigma$ and $\det A = \det \Sigma$
Clusterization quality

- Good clustering: $S_W$ is small and $S_B, \Sigma$ are big.
- Cluster discriminability metrics:

$$\text{Tr}\{S_W^{-1} S_B\}, \quad \frac{\text{Tr}\{S_B\}}{\text{Tr}\{S_W\}}, \quad \frac{\text{det} \Sigma}{\text{det} S_W}$$

Resume

- Pairwise feature measures
  - fail to estimate relevance in context of other features
  - are robust to curse of dimensionality

- Context aware measures:
  - estimate relevance in context of other features
  - prone to curse of dimensionality
Table of Contents

1 Filter methods

2 Feature subsets generation
   • Randomised feature selection
Complete search with optimal solution

- exhaustive search
- branch and bound method
  - requires monotonicity property:
    \[ F \subset G : J(F) < J(G) \]
  - when property does not hold, becomes suboptimal
Example
Incomplete search with suboptimal solution

- Order features with respect to $J(f)$:
  
  $$J(f_1) \geq J(f_2) \geq \ldots \geq J(f_D)$$

- select top $m$
  
  $$\hat{F} = \{f_1, f_2, \ldots, f_m\}$$

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

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

- Comments:
  
  - simple to implement
  - if $J(f)$ is context unaware, so will be the features
  - example: when features are correlated, it will take many redundant features
Sequential search

- **Sequential forward selection algorithm:**
  - init: \( k = 0, \ F_0 = \emptyset \)
  - while \( k < \text{max\_features} \):
    - \( f_{k+1} = \arg \max_{f \in F} J(F_k \cup \{f\}) \)
    - \( F_{k+1} = F_k \cup \{f_{k+1}\} \)
    - if \( J(F_{k+1}) < J(F_k) \): break
    - \( k = k + 1 \)
  - 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
2 Feature subsets generation
   - Randomised feature selection
Randomization

- 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
    - more prone to getting stuck in local optimum (though less than deterministic)
    - more efficiently locates small optimal feature subsets
Randomization

- Means of randomization:
  - initialize an iterative algorithm with random initial features
  - apply algorithm to sample subset
  - at each iteration of sequential search look through random subset of features
  - genetic algorithms
Genetic algorithms

- Each feature set $F = \{f_{i(1)}, f_{i(2)}, \ldots f_{i(K)}\}$ is represented using binary vector $[b_1, b_2, \ldots b_D]$ where $b_i = \mathbb{I}[f_i \in F]$

- Genetic operations:
  - $crossover(b^1, b^2) = b$, where $b_i = \begin{cases} b^1_i & \text{with probability } \frac{1}{2} \\ b^2_i & \text{otherwise} \end{cases}$
  - $mutation(b^1) = b$, where $b_i = \begin{cases} b^1_i & \text{with probability } 1 - \alpha \\ -b^1_i & \text{with probability } \alpha \end{cases}$
Genetic algorithms

**INPUT:**
- size of population $B$
- size of expanded population $B'$
- parameters of crossover and mutation $\theta$
- maximum number of iterations $T$, minimum quality change $\Delta Q$

**ALGORITHM:**
generate $B$ feature sets randomly: $P^0 = \{S^0_1, S^0_2, \ldots S^0_B\}$, set $t = 1$

while $t <= T$ and $|Q^t - Q^{t-1}| > \Delta Q$:
- modify $P^{t-1}$ using crossover and mutation:
  $P^t = S'_1, S'_2, \ldots S'_{B'} = \text{modify}(P^{t-1}|\theta)$
- order transformed sets by decreasing quality:
  $Q(S'_{i(1)}) \geq Q(S'_{i(2)}) \geq \ldots Q(S'_{i(B')})$
- get $B$ best representatives:
  $S'_1, S'_2, \ldots S'_{B} = \text{best\_representatives}(P^t, B)$
- set next population to consist of best representatives:
  $P^t = \{S^t_{i(1)}, S^t_{i(2)}, \ldots S^t_{i(B)}\}$
  $Q^t = Q^t(S^t_{i(1)})$
- $t = t + 1$
Modifications of genetic algorithm

- Augment $P^t$ with $K$ best representatives from $P^{t-1}$ to preserve attained quality
- Allow crossover only between best representatives
- Make mutation probability higher for good features (that frequently appear in best representatives)
- Crossover between more than two parents
- Simultaneously modify several populations and allow rare random transitions between them.
Feature selection using:
- $L_1$ regularization
- feature importances of trees

Compositions
- different algorithms
- different subsamples

Stability measures
- Jaccard distance: $D(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$
- for $K$ outputs: $\frac{2}{K(K-2)} \sum_{i<j} D(S_i, S_j)$

Feature selection compositions yield more stable selections.