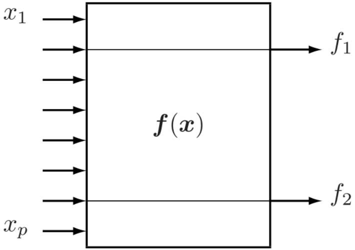
Feature selection

Victor Kitov

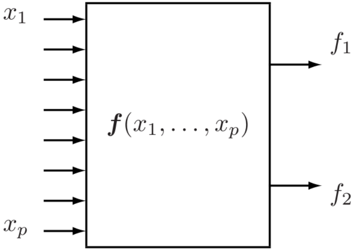
Yandex School of Data Analysis

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, $F=\{f_1,f_2,...f_D\}$ - full set of features, $S=F\backslash\{f\}$.

- **Strongly relevant feature:**

$$p(y|f,S)\neq p(y|S)$$

- **Weakly relevant feature:**

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

- **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.

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 predictor (embedded methods)

Predictor 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.

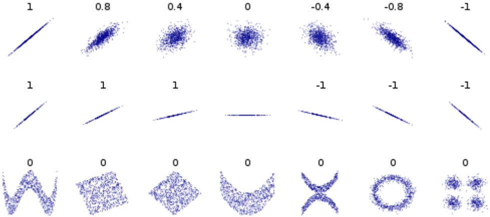
Table of Contents

1. Filter methods
   - Probability measures
   - Context relevant measures
2. Feature subsets generation

Correlation

- two class:

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



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:

$$H(Y|X) = -\sum_{x} p(x) \sum_{y} p(y|x) \ln p(y|x)$$

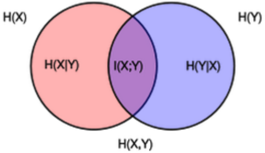
Mutual information

Mutual information measures how much X gives information about Y:

$$MI(X,Y) = \sum_{x,y} p(x,y) \ln \left[\frac{p(x,y)}{p(x)p(y)} \right]$$

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 $MI(X,Y) = H(Y) \leq H(X)$

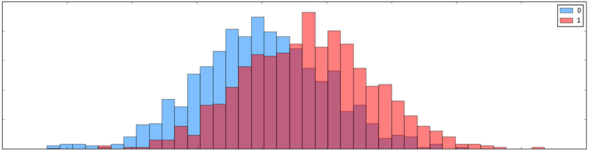


Mutual information for feature selection

- 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

1. Filter methods
   - Probability measures
   - Context relevant measures

Relevance based on probabilistic distance



Measure of feature f relevance - distance between $p(f|\omega_1)$ and $p(f|\omega_2)$

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: $\frac{1}{2}\left(\int(f(x)-g(x))^2dx\right)^{1/2}$

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

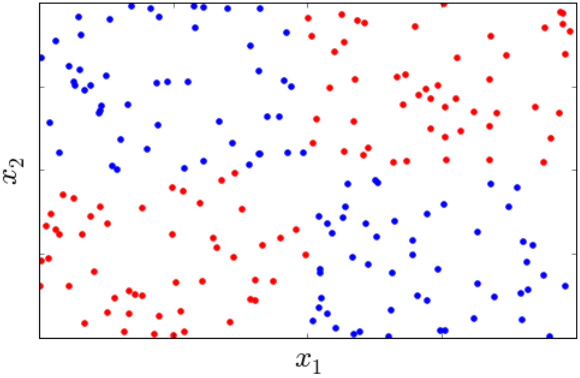
- Kolmogorov: $\sup_x|F(x)-G(x)|$
- Kantorovich: $\int|F(x)-G(x)|dx$

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 $(x_1, y_1), (x_2, y_2), ...(x_N, y_N)$
    Number of neighbours $K$
    Distance metric $d(x, x')$ # usually Euclidean

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

for each feature $f_i$ in $f_1, f_2, ...f_D$:
    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 $R$
```

Table of Contents

1 Filter methods

2 Feature subsets generation
- Deterministic feature selection
- Randomised feature selection

2 Feature subsets generation

- Deterministic feature selection
- Randomised feature selection

Incomplete search with suboptimal solution

- Consider not all but only the most promising feature subsets.
- 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: $S = \{\{f_1\}, \{f_1, f_2\}, ...\{f_1, f_2, ...f_D\}\}$

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

- 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 $k < max\_features$:
        - $f_{k+1} = \arg\max_{f \in F} J(F_k \cup \{f\})$
        - $F_{k+1} = F_k \cup \{f_{k+1}\}$
        - 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

- Deterministic feature selection
- Randomised feature selection

Genetic algorithms

- Each feature set $F = \{f_{i(1)}, f_{i(2)}, ...f_{i(K)}\}$ is represented using binary vector $[b_1, b_2, ...b_D]$ where $b_i = \mathbb{I}[f_i \in F]$
- Genetic operations:
  - $crossover(b^1, b^2) = b$, where $b_i = \begin{cases} b_i^1 & \text{with probability } \frac{1}{2} \\ b_i^2 & \text{otherwise} \end{cases}$
  - $mutation(b^1) = b$, where $b_i = \begin{cases} b_i^1 & \text{with probability } 1 - \alpha \\ \neg b_i^1 & \text{with probability } \alpha \end{cases}$

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 $\Delta J$

ALGORITHM:
generate $B$ feature sets randomly: $P^0 = \{S_1^0, S_2^0, ...S_B^0\}$, set $t = 1$
while $t <= T$ and $|J^t - J^{t-1}| > \Delta J$:
    modify $P^{t-1}$ using crossover and mutation:
        $P'^t = S'^t_1, S'^t_2, ...S'^t_{B'} = \text{modify}(P^{t-1}|\theta)$
    order transformed sets by decreasing quality:
        $J(S'^t_{i(1)}) \geq J(S'^t_{i(1)}) \geq ...J(S'^t_{i(B')})$
    get $B$ best representatives:
        $S_1^t, S_2^t, ...S_B^t = \text{best\_representatives}(P'^t, B)$
    set next population to consist of best representatives:
        $P^t = \{S^t_{i(1)}, S^t_{i(2)}, ...S^t_{i(B)}\}$
    $J^t = J^t(S^t_{i(1)})$
    $t = t + 1$
```

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.