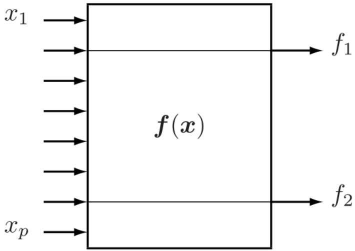
Feature selection

Victor Kitov

v.v.kitov@yandex.ru

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:

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)
    - methods with L1 regularization

Types of features[1]

Define f - the feature, $F=\{f_1,f_2,...f_D\}$ - full set of features, $\tilde{F}=F\backslash\{f\}$.

- **Strongly relevant feature:**

$$p(y|f,\tilde{F})\neq p(y|\tilde{F})$$

- **Weakly relevant feature:**

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

- **Irrelevant feature:**

$$\forall S\subset\tilde{F}:\ p(y|f,S)=p(y|S)$$

[1] Propose an example for each feature type.

Types of features[1]

Define f - the feature, $F=\{f_1, f_2, ...f_D\}$ - full set of features, $\tilde{F}=F\backslash\{f\}$.

- **Strongly relevant feature:**

$$p(y|f,\tilde{F})\neq p(y|\tilde{F})$$

- **Weakly relevant feature:**

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

- **Irrelevant feature:**

$$\forall S\subset\tilde{F}:\ p(y|f,S)=p(y|S)$$

Aim of feature selection

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

[1] Propose an example for each feature type.

Categorization of feature selection algorithms

- Completeness of search:
  - Complete
    - exhaustive search complexity is 2^D.
    - may be not exhaustive under certain conditions on $J(S)$[2]
  - Suboptimal
    - deterministic
    - random (deterministic with randomness / completely random)
- Integration with final predictor
  - independent (filter methods)
  - uses predictor quality (wrapper methods)
  - is embedded inside predictor (embedded methods)

[2] $J(S)$ is a score of feature subset S.

Table of Contents

Individual feature importances approach

- Estimate importances for individual features $I(f_1), I(f_2), ...I(f_D)$.
- Generate feature subset based on importances.

1. Individual feature importances approach
    - Feature subset generation
    - Feature importance estimation

Incomplete search with suboptimal solution

- Order features with respect to feature importances $I(f)$:

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

option 1: select top m

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

option 2: 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
  - when features are correlated, it will take many redundant features

1. Individual feature importances approach
    - Feature subset generation
    - Feature importance estimation

Application of feature importances

- Feature importances can be used:
    - for feature selection
    - for rescaling features for adapting their impact on the model:
        - e.g.: in K-NN, in linear methods with regularization
    - for adapting feature sampling probability in random forest, extra random trees.

Correlation

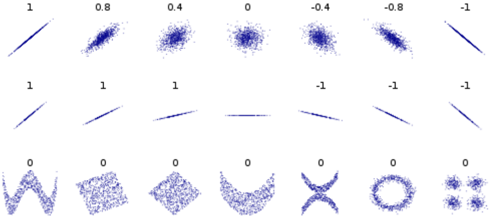
- two class:

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

- multiclass - average correlations with individual classes.
- Benefits:
  - simple to compute
  - applicable both to continuous and discrete features/output.
  - does not require calculation of probability density function.

Correlation for non-linear relationship

- **Correlation captures only linear relationship.**
- *Example: consider X-random variable, with* $\mathbb{E}X = 0$, $\mathbb{E}X^3 = 0$ *and random variable* $Z = X^2$. *Then* X, Z *are uncorrelated but dependent.*
- Other examples of data and its correlation:



- Correlation between ranks.

Defintitions

- Entropy[3] of random variable Y:

$$H(Y) := -\sum_{y} p(y)\ln p(y)$$

- Conditional 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 $Q(x)$:
$KL(P||Q) := \sum_x P(x)\ln\frac{P(x)}{Q(x)}$

[3] measures level of uncertainty of r.v. Y

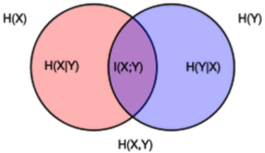
Mutual information

Mutual information measures how much r.v. X and Y share information between each other:

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

Properties:

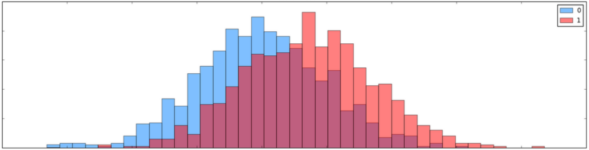
- $MI(X,Y) = MI(Y,X)$
- $MI(X,Y) = KL(p(x,y)||p(x)p(y)) \geq 0$
- X, Y- independent $<=>$ $MI(X,Y) = 0$ (for discrete r.v.)
- $MI(X,Y) = H(Y) - H(Y|X)$
- $MI(X,Y) \leq \min\{H(X), H(Y)\}$
- X completely identifies Y, then $MI(X,Y) = H(Y) \leq H(X)$



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

importance based on probabilistic distance



Measure of feature f importance - distance between $p(f|y=0)$ and $p(f|y=1)$, e.g. total variation:

$$\int |p(x|y=1) - p(x|y=0)|\, dx$$

Relief criterion: 1-NN

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

for each object x_n, y_n:
    calculate nearest neighbour of the same class $x_{s(n)}$

    calculate K nearest neighbour of class $x_{d(n)}$

for each feature f_i in $f_1, f_2, ...f_D$:
    calculate importance $R(f_i) = \frac{1}{N} \sum_{n=1}^{N} \frac{|x_n^i - x_{d(n)}^i|}{|x_n^i - x_{s(n)}^i|}$

OUTPUT:
    feature importances R

Relief criterion: K-NN

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

for each object x_n, y_n:
calculate K nearest neighbours of the same class y_n:
$x_{s(n,1)}, x_{s(n,2)}, ...x_{s(n,K)}$
calculate K nearest neighbours of class other than y_n:
$x_{d(n,1)}, x_{d(n,2)}, ...x_{d(n,K)}$

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

OUTPUT:
feature importances R

Tree feature importances

- Tree feature importances (*clf.feature_importances_* in sklearn).
    - Consider feature f
    - Let $T(f)$ be the set of all nodes, relying on feature f when making split.
    - efficiency of split at node t: $\Delta I(t) = I(t) - \sum_{c \in childen(t)} \frac{n_c}{n_t} I(c)$
    - feature importance of f: $\sum_{t \in T(f)} n_t \Delta I(t)$

- Alternative: difference in decision tree prediction quality for
    1. original validation set
    2. validation set with j-th feature randomly shuffled

Feature importances from linear model

- Feature importances from linear classification:
  1. fit linear classifier with regularization to data
     - features should be normalized
  2. retrieve w (*clf.coef_* in scikit-learn)
  3. importance of feature f_i is equal to $|w_i|$.

Table of Contents

Simultaneous feature selection specification

- Need to specify:
    - quality criteria $J(S)$ for any feature subset S
        - typically: quality of model with these features (wrapper approach)
    - feature subset generation method $S_1, S_2, S_3, ...$

2. Simultaneous feature selection specification
   - Sequential search subset generation
   - Genetic search subset generation

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

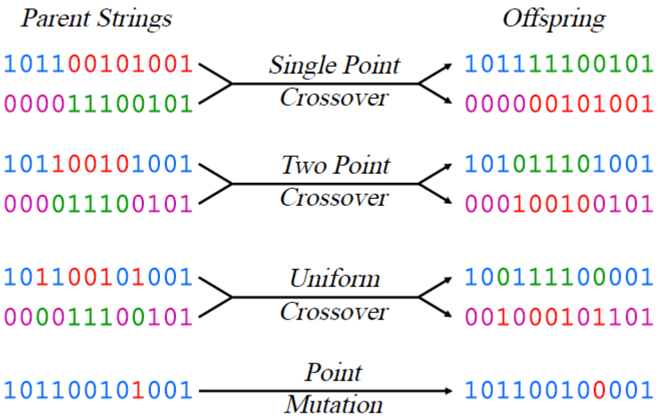
Genetic[4] algorithms

- 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 $b_i=\begin{cases}b_i^1 & \text{with probability } 1-\alpha\\ \neg b_i^1 & \text{with probability } \alpha\end{cases}$

  for some small α.

[4] Name inspired by genetic inheritance in biology.

Genetic operations: demo



Genetic algorithms

INPUT:
population size B and expanded population size B'
parameters of mutation and crossover
maximum number of iterations T, minimum quality change ΔJ

ALGORITHM:
generate B feature sets $S_1, S_2, ...S_B$ randomly.
set $t = 1$, $P^0 = \{S_1, S_2, ...S_B\}$, $J^0 = J(P^0)$

while $t <= T$ and $|J^t - J^{t-1}| > \Delta J$:
modify P^{t-1} using crossover and mutation:
$S_1', S_2', ...S_{B'}' = \text{modify}(P^{t-1}|\theta)$
order transformed sets by decreasing quality:
$J(S'^t_{i(1)}) \geq J(S'^t_{i(2)}) \geq ...J(S'^t_{i(B')})$
set next population to consist of best representatives:
$P^t = \{S'_{i(1)}, S'_{i(2)}, ...S'_{i(B)}\}$
set $J^t = \max_{S \in P^t} J(S)$
$t = t + 1$

OUTPUT: suboptimal set of feature sets P^t

Modifications of genetic algorithm

- Preserve best features and best feature subsets:
  - Augment P'^{t} with K best representatives from P^{t-1}.
  - Make mutation probability lower for good features (that frequently appear in inside representatives).
- Increase breadth of search:
  - Crossover between more than two parents
- To prevent convergence to local optimum:
  - simultaneously modify several populations and allow rare random transitions between them.

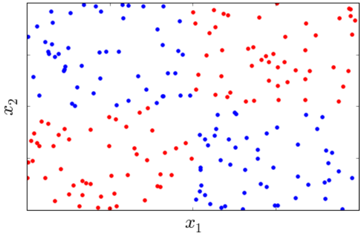
Importance in context

Individually features do not affect y:

$$p(y|x^2) = p(y), \quad p(y|x^2) = p(y)$$

but may be relevant together:

$$p(y|x^1, x^2) \neq p(y)$$



Which methods will extract features relevant in context but itrrelevant individually?