

CS 670/470 Team Project (Part 1)

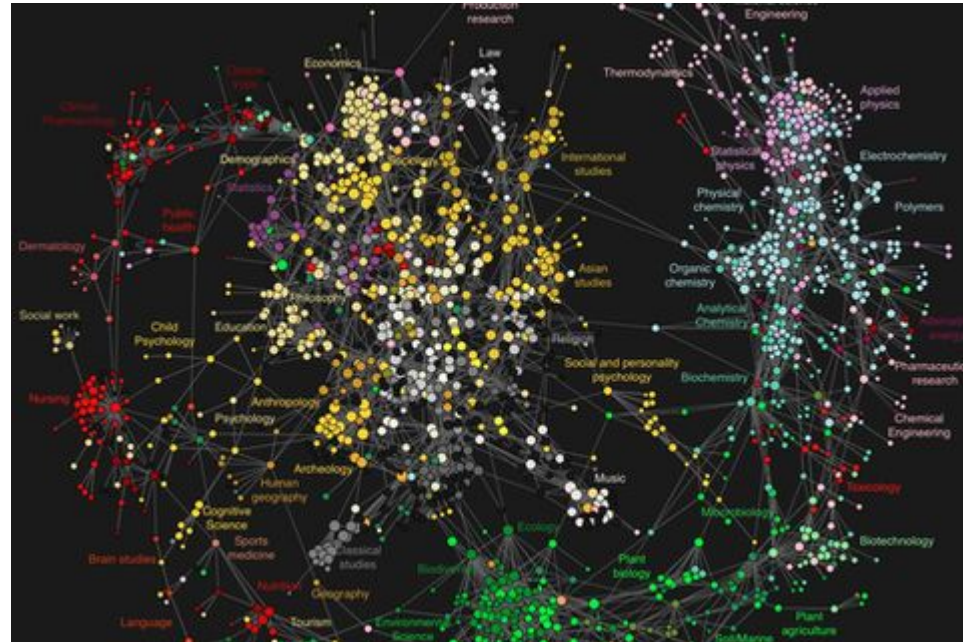
-- Scalable and Accurate OnLine Approach (SAOLA)

Yong Zhuang, Xin Shu, Yu Kui

High - Dimensionality

In machine learning, “dimensionality” simply refers to the number of features (i.e. input variables) in the dataset.

The general question is, when the number of features is very large relative to the number of observations in the dataset, certain algorithms struggle to train effective models.

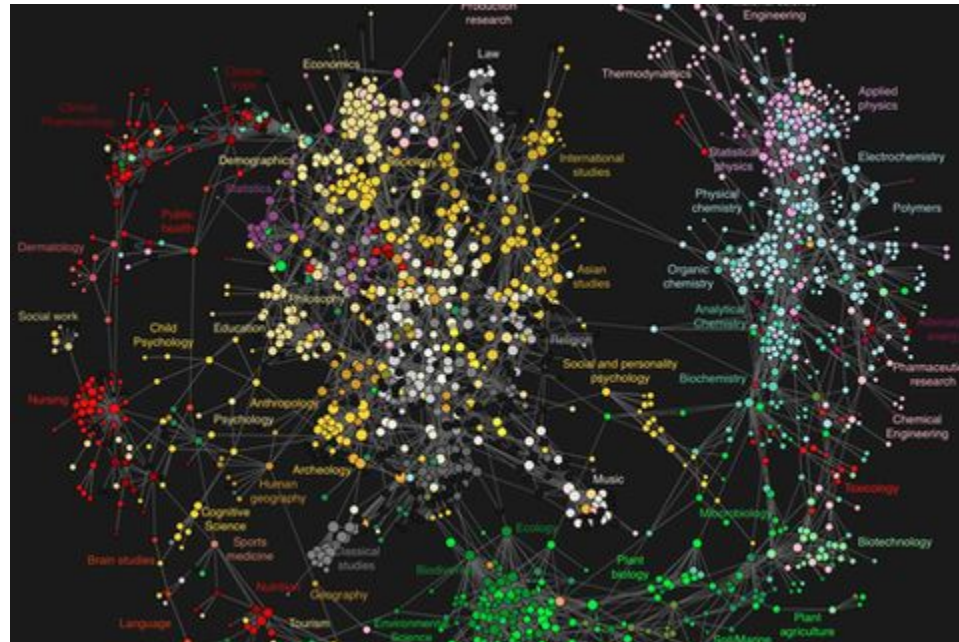
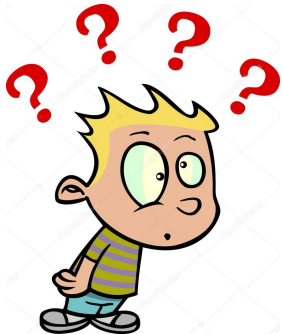


Dimension Reduction

In machine learning and statistics, dimensionality reduction or dimension reduction is the process of reducing the number of random variables under consideration, via obtaining a set of principal variables. It can be divided into feature selection and feature extraction.

--- Wikipedia entry

Which features should you use to create a predictive model?



Feature Selection

Feature selection is also called variable selection or attribute selection.

Feature selection... is the process of selecting a subset of relevant features for use in model construction.
--- Wikipedia entry

Feature selection is itself useful, but it mostly acts as a filter, muting out features that aren't useful in addition to your existing features.

--- Robert Neuhaus in "How valuable do you think feature selection is in machine learning?"

Feature Selection

As many pattern recognition techniques were originally not designed to cope with large amounts of irrelevant features, combining them with feature selection techniques has become a necessity in many applications.

The objective of variable selection is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data.

--- Guyon and Elisseeff in “An Introduction to Variable and Feature Selection”

Feature Selection

The objectives of feature selection are manifold, the most important ones being:

- ❖ To avoid overfitting and improve model performance, i.e. prediction performance in the case of supervised classification and better cluster detection in the case of clustering,
- ❖ To provide faster and more cost-effective models
- ❖ To gain a deeper insight into the underlying processes that generated the data.

The Problem The Feature Selection Solves

- A. Feature selection methods aid you in your mission to create an accurate predictive model. They help you by choosing features that will give you as good or better accuracy whilst requiring less data.
- B. Feature selection methods can be used to identify and remove unneeded, irrelevant and redundant attributes from data that do not contribute to the accuracy of a predictive model or may in fact decrease the accuracy of the model.
- C. Fewer attributes is desirable because it reduces the complexity of the model, and a simpler model is simpler to understand and explain.

Example

Task: predict chances of lung disease

Data: medical history survey

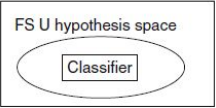
X			
Vegetarian	No	Reduced X	<div>Family history No Smoker Yes</div>
Plays video games	Yes		
Family history	No		
Athletic	No		
Smoker	Yes		
Gender	Male		
Lung capacity	5.8L		
Hair color	Red		
Car	Audi		
...	...		
Weight	185 lbs		

Feature Selection Algorithms

There are three general classes of feature selection algorithms:

1. Embedded methods.
2. Wrapper methods
3. Filter methods

Embedded Methods

Model search	Advantages	Disadvantages	Examples
Embedded 	Interacts with the classifier Better computational complexity than wrapper methods Models feature dependencies	Classifier dependent selection	Decision trees Weighted naive Bayes (Duda <i>et al.</i> , 2001) Feature selection using the weight vector of SVM (Guyon <i>et al.</i> , 2002; Weston <i>et al.</i> , 2003)

Wrapper Methods

Model search

Advantages

Disadvantages

Examples

Wrapper

Deterministic

Simple
Interacts with the classifier
Models feature dependencies
Less computationally
intensive than randomized methods

Risk of over fitting
More prone than randomized
algorithms to getting stuck in a
local optimum (greedy search)
Classifier dependent selection

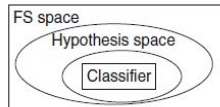
Sequential forward selection
(SFS) (Kittler, 1978)
Sequential backward elimination
(SBE) (Kittler, 1978)
Plus q take-away r
(Ferri *et al.*, 1994)
Beam search (Siedelecky
and Sklansky, 1988)

Randomized

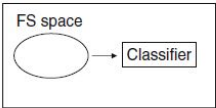
Less prone to local optima
Interacts with the classifier
Models feature dependencies

Computationally intensive
Classifier dependent selection
Higher risk of overfitting
than deterministic algorithms

Simulated annealing
Randomized hill climbing
(Skalak, 1994)
Genetic algorithms
(Holland, 1975)
Estimation of distribution
algorithms (Inza *et al.*, 2000)



Filter Methods

Model search	Advantages	Disadvantages	Examples
Filter 	Univariate		
	Fast Scalable Independent of the classifier	Ignores feature dependencies Ignores interaction with the classifier	χ^2 Euclidean distance <i>i</i> -test Information gain, Gain ratio (Ben-Bassat, 1982)
	Multivariate		
	Models feature dependencies Independent of the classifier Better computational complexity than wrapper methods	Slower than univariate techniques Less scalable than univariate techniques Ignores interaction with the classifier	Correlation-based feature Selection (CFS) (Hall, 1999) Markov blanket filter (MBF) (Koller and Sanami, 1996) Scalable and Accurate On-Line Approach (SAOLA) (Yu, 2014)

Feature Selection

Feature selection... is the process of selecting a subset of relevant features for use in model construction.
--- Wikipedia entry



Feature Selection



A subset of relevant features



Improve

Best Feature Selection



A subset of the most relevant features



How to find the relevant features?

Markov and Markov Property



Born: 14 June 1856 in Ryazan, Russia

Died: 20 July 1922 in Petrograd (now St Petersburg), Russia

Bayes Rule:

$$P(q_t, q_{t-1}, \dots, q_1) = P(q_t \mid q_{t-1}, \dots, q_1) P(q_{t-1}, \dots, q_1)$$

Markov Property:

$$P(q_i \mid q_{i-1}, \dots, q_1) = P(q_i \mid q_{i-1}) \text{ for } i > 1$$

Markov Random Field (MRF)

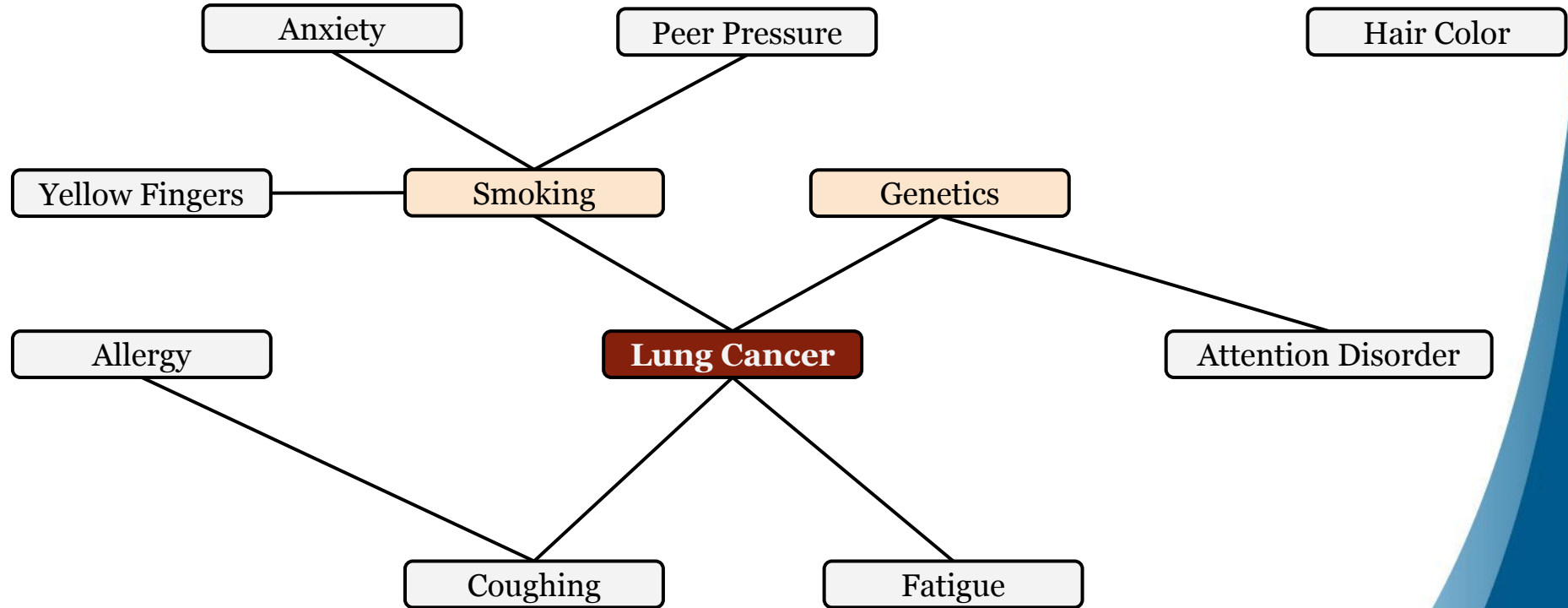
In the domain of physics and probability, a Markov random field (often abbreviated as MRF), Markov network or undirected graphical model is a set of random variables having a Markov property described by an undirected graph.

--- Wikipedia entry

A Markov Random Field (MRF) is a graphical model of a joint probability distribution. It consists of an undirected graph in which the nodes represent random variables. And the edges encode conditional independence relationships.

--- <http://homepages.inf.ed.ac.uk>

MRF Example



Markov Blanket



How to find the most relevant features?

Markov Blanket

In machine learning, ... In a Markov random field, the Markov blanket of a node is its set of neighboring nodes.
--- Wikipedia entry

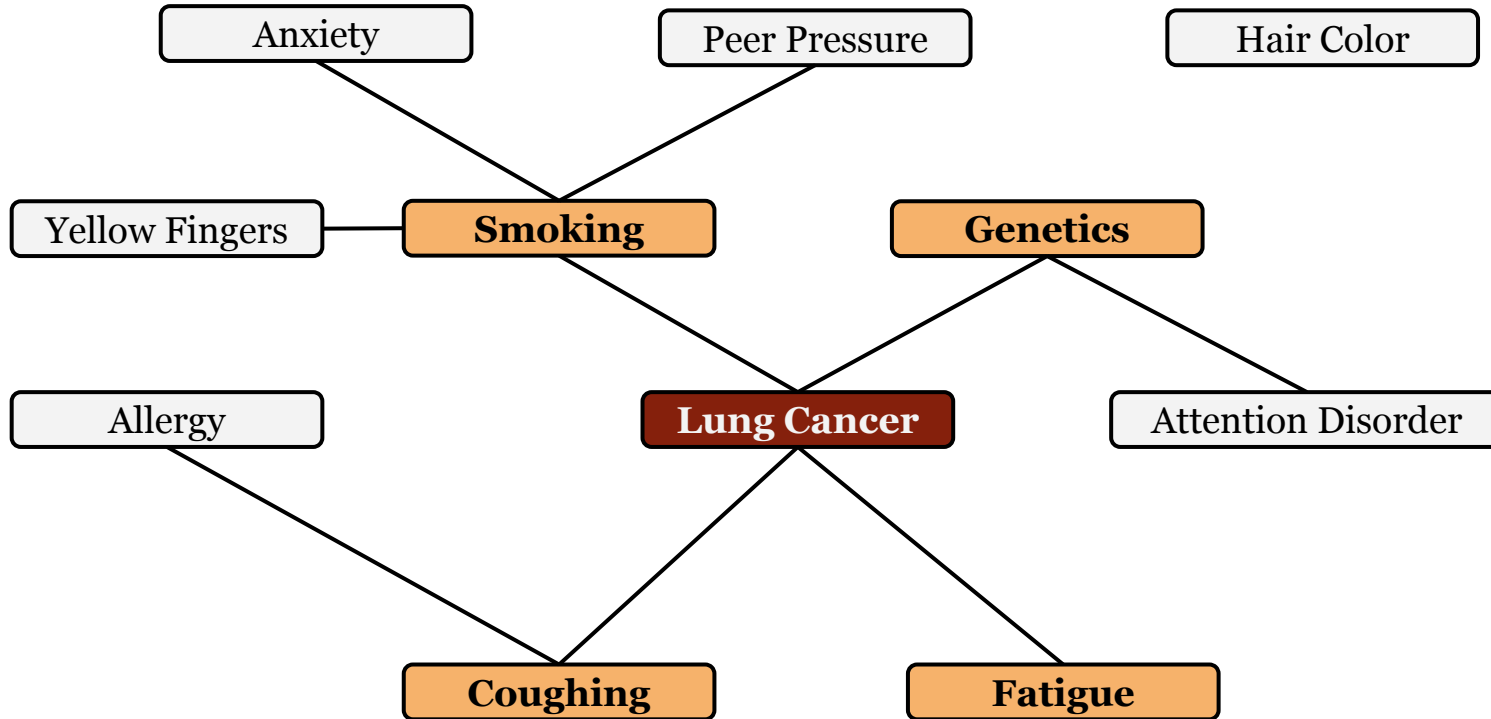
The Markov blanket of a node contains all the variables that shield the node from the rest of the network. This means that the Markov blanket of a node is the only knowledge needed to predict the behavior of that node.

--- Judea Pearl in 1988.



Markov Blanket

Markov Blanket Example



Markov Blanket

Definition 1 (Markov Blankets) A Markov blanket of feature F_i , denoted as $M \subseteq F - \{F_i\}$ makes all other features independent of F_i given M , that is,

$$\forall Y \in F - (M \cup \{F_i\}) \text{ s.t. } P(F_i|M, Y) = P(F_i|M).$$

Markov Blanket In Classification Problem

Problem Definition

Denote D by $D = \{(F_i, C), 1 \leq i \leq P\}$, which is a sequence of features that is presented in a sequential order, where $F_i = \{f_1, f_2, \dots, f_N\}^T$ denotes the i^{th} feature containing N data instances, and C includes N class label instances.

If D can be processed in a sequential scan, that is, one dimension at a time, we can process high dimensional data not only with limited memory, but also without requiring its complete set of features available. The challenge is that, as we process one dimension at a time, at any time t_i , how to online maintain a minimum feature subset $S_{t_i}^*$ of maximizing its predictive performance for classification. Assuming $S \subseteq F$ is the feature set containing all features available till time t_{i-1} and F_i is a new coming feature at time t_i , our problem can be formulated as follows:

$$S_{t_i}^* = \arg \min_{S'} \{ |S'| : S' = \arg \max_{\zeta \subseteq \{S \cup F_i\}} P(C|\zeta) \}. \quad (1)$$

Markov Blanket In Classification Problem

We can further decompose it into the following key steps:

- Determine the relevance of F_i to C . Firstly, we determine whether Eq.(2) holds or not.

$$P(C|F_i) = P(C). \quad (2)$$

If so, F_i is discarded as an irrelevant feature. If not, secondly, we further evaluate whether F_i carries additional predictive information to C given the selected feature set $S_{t_{i-1}}^*$ at t_{i-1} , that is, whether Eq.(3) holds. If Eq.(3) holds, F_i will be discarded.

$$P(C|S_{t_{i-1}}^*, F_i) = P(C|S_{t_{i-1}}^*). \quad (3)$$

- Calculate $S_{t_i}^*$ with F_i 's inclusion. Once F_i is added to $S_{t_{i-1}}^*$, at time t_i , $S_{t_i} = \{S_{t_{i-1}}^*, F_i\}$, we then solve Eq.(4) to prune S_{t_i} to satisfy Eq.(1).

$$S_{t_i}^* = \arg \max_{\zeta \subseteq S_{t_i}} P(C|\zeta). \quad (4)$$

Scalable and Accurate OnLine Approach (SAOLA)

The benefits of SAOLA is when a data set includes extremely high dimensionality in big data analytics, SAOLA can significantly mitigates the expensively computational costs.

Scalable and Accurate On-Line Approach (SAOLA) by employing on-line pairwise comparisons between features in the currently selected feature set once a new coming feature is included

Scalable and Accurate OnLine Approach (SAOLA)

Algorithm 1: The SAOLA Algorithm

Data:

F_i : predictive features; C : the class attribute;

δ_1 : a relevance threshold ($0 \leq \delta_1 \leq 1$);

δ_2 : a correlation bound of $I(F_i; Y)$;

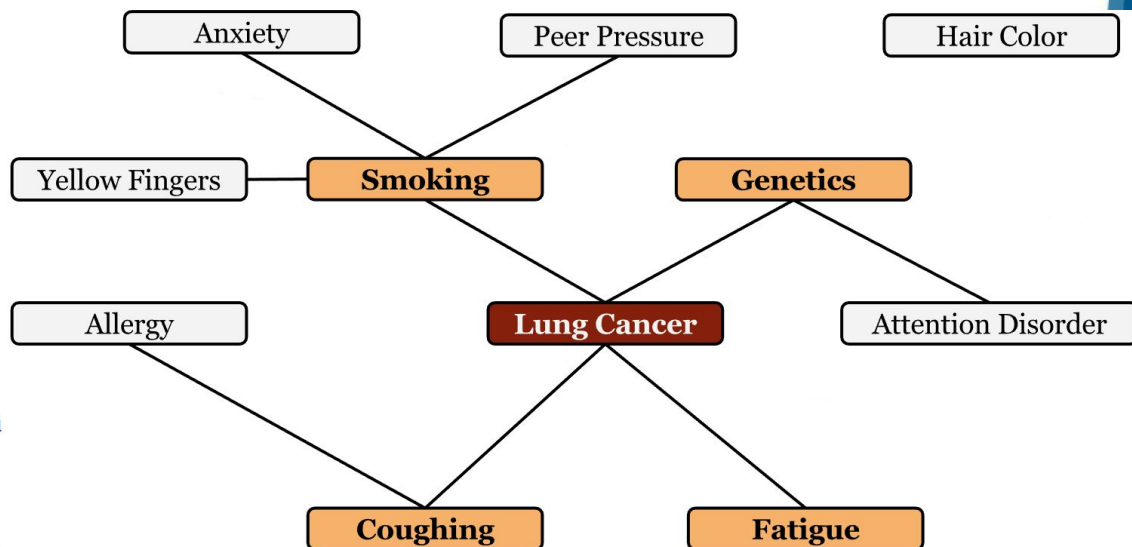
$S_{t_{i-1}}^*$: the selected feature set at time t_{i-1} ;

$S_{t_i}^*$: the selected feature set at time t_i

```

1 repeat
2   Get a new feature  $F_i$  at time  $t_i$ ;
3   /*Solve Eq.(2)*/
4   if  $I(F_i, C) < \delta_1$  then
5     Discard  $F_i$ , and go to Step 18;
6   end
7   for each feature  $Y \in S_{t_{i-1}}^*$  do
8     /*Solve Eq.(3)*/
9     if  $I(Y; C) > I(F_i; C) \ \& \ I(F_i; Y) \geq \delta_2$  then
10      Discard  $F_i$ , go to Step 18;
11    end
12    /*Solve Eq.(4)*/
13    if  $I(F_i; C) > I(Y; C) \ \& \ I(F_i; Y) \geq \delta_2$  then
14       $S_{t_{i-1}}^* = S_{t_{i-1}}^* - Y$ ;
15    end
16  end
17   $S_{t_i}^* = S_{t_{i-1}}^* \cup F_i$ ;
18 until no features are available;
19 Output  $S_{t_i}^*$ ;

```



Scalable and Accurate OnLine Approach (SAOLA)

Algorithm 1: The SAOLA Algorithm

Data:

F_i : predictive features; C : the class attribute;

δ_1 : a relevance threshold ($0 \leq \delta_1 \leq 1$);

δ_2 : a correlation bound of $I(F_i; Y)$;

$S_{t_{i-1}}^*$: the selected feature set at time t_{i-1} ;

$S_{t_i}^*$: the selected feature set at time t_i

1 **repeat**

2 Get a new feature F_i at time t_i ;

3 /*Solve Eq.(2)*/

4 **if** $I(F_i, C) < \delta_1$ **then**

5 | Discard F_i , and go to Step 18;

6 **end**

7 **for each feature** $Y \in S_{t_{i-1}}^*$ **do**

8 /*Solve Eq.(3)*/

9 **if** $I(Y; C) > I(F_i; C) \ \& \ I(F_i; Y) \geq \delta_2$ **then**

10 | Discard F_i , go to Step 18;

11 **end**

12 /*Solve Eq.(4)*/

13 **if** $I(F_i; C) > I(Y; C) \ \& \ I(F_i; Y) \geq \delta_2$ **then**

14 | $S_{t_{i-1}}^* = S_{t_{i-1}}^* - Y$;

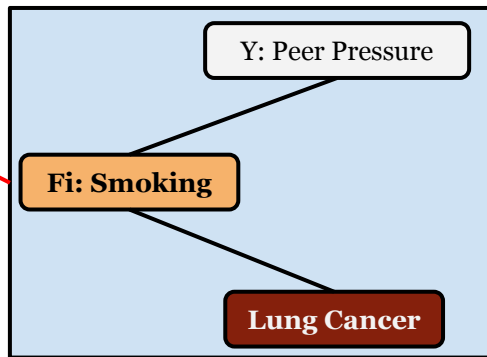
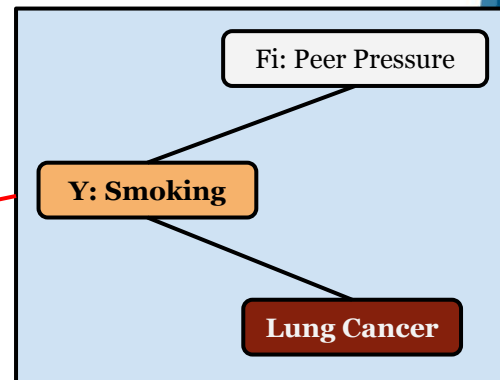
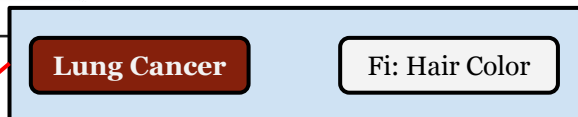
15 **end**

16 **end**

17 $S_{t_i}^* = S_{t_{i-1}}^* \cup F_i$;

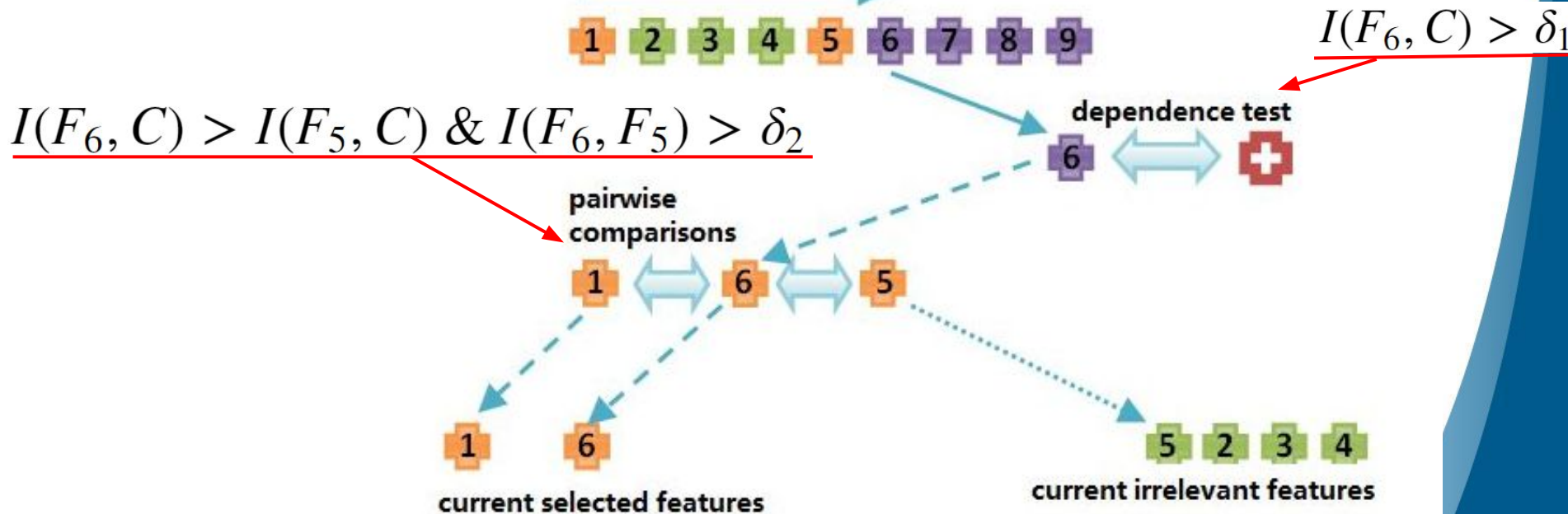
18 **until** no features are available;

19 Output $S_{t_i}^*$;



Scalable and Accurate OnLine Approach (SAOLA)

c: SAOLA



Relevant test:



Selected feature:



Class label:



Accept:



Group:



Irrelevant feature:



Unprocessed feature:



Reject:





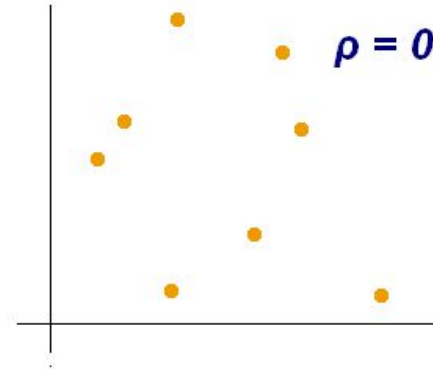
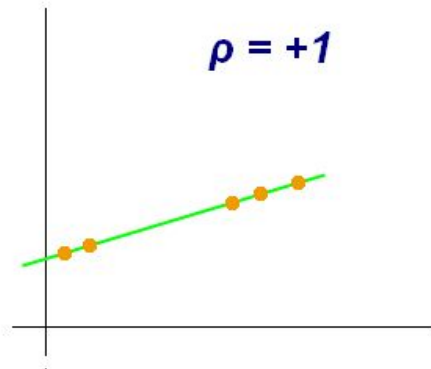
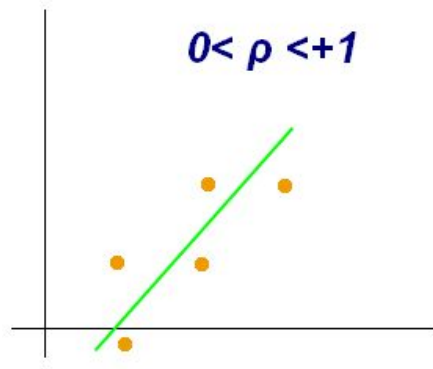
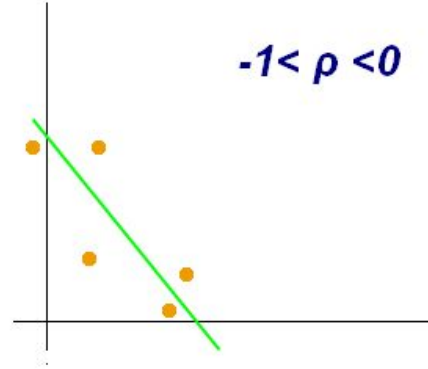
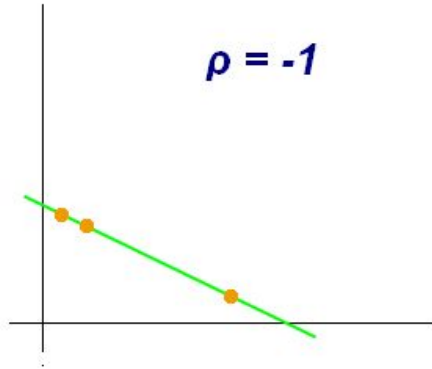
Function I is used to calculate relevance between two features.
How to implement the function I ?

Pearson correlation coefficient

In statistics, the Pearson correlation coefficient (PCC), also referred to as the Pearson's r , Pearson product-moment correlation coefficient (PPMCC) or bivariate correlation, is a measure of the linear correlation between two variables X and Y . It has a value between $+1$ and -1 , where 1 is total positive linear correlation, 0 is no linear correlation, and -1 is total negative linear correlation. It is widely used in the sciences. It was developed by Karl Pearson from a related idea introduced by Francis Galton in the 1880s.

--- Wikipedia entry

Pearson correlation coefficient



SAOLA



How to judge if two features are relevant?

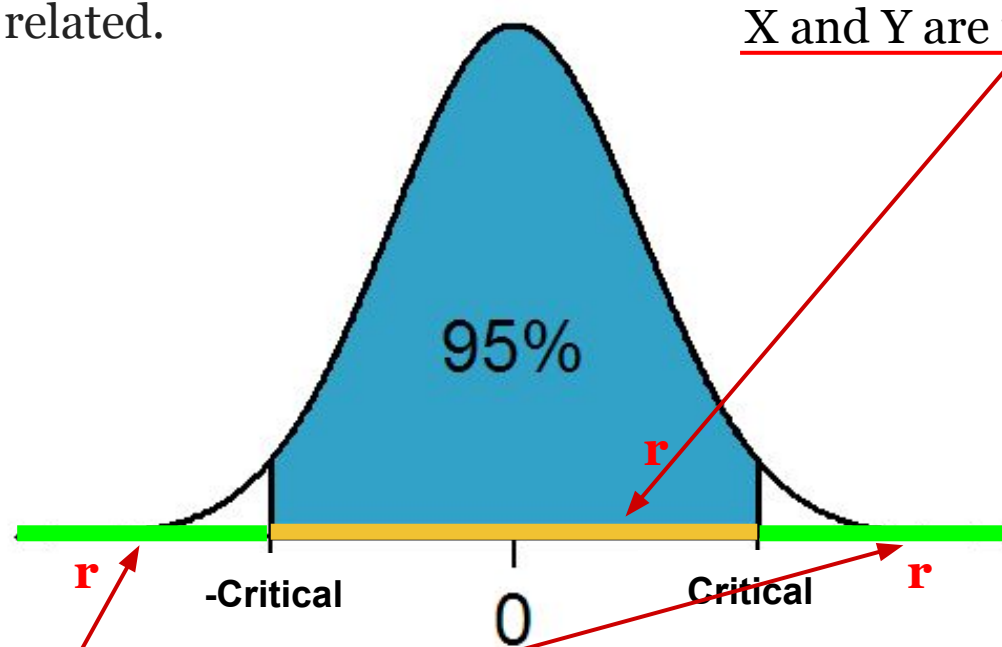
Z-test

A Z-test is any statistical test for which the distribution of the test statistic under the null hypothesis can be approximated by a normal distribution. Because of the central limit theorem, many test statistics are approximately normally distributed for large samples. For each significance level, the Z-test has a single critical value (for example, 1.96 for 5% two tailed). --- Wikipedia entry

Null Hypothesis: We assume that two features X and Y are independent, and their expected Pearson's r is 0.

Confidence Interval

If we can calculate the p value of r , we can judge these two features (X and Y) are irrelevant or related.



X and Y are irrelevant

X and Y are related

Confidence Interval



However, the sampling distribution of Pearson's r may not
always be normally distributed.



We need a transformation for r .

Fisher z transformation

The Fisher z-Transformation is a way to transform the sampling distribution of Pearson's r (i.e. the correlation coefficient) for samples of size N to r' , which has a normal distribution $N(\rho', s_{r'})$,

$$r' = \frac{1}{2} \ln \left(\frac{1+r}{1-r} \right) = \operatorname{arctanh}(r)$$

$$s_{r'} = \frac{1}{\sqrt{N-3}}$$

Fisher z transformation

Pearson correlation coefficient

not normally distributed



Fisher z transformation

normal distribution



...

Standard normal distribution



VS

Standard normal distribution

Confidence Interval of standard normal distribution (1.96)

Fisher z transformation

Pearson correlation coefficient

Not normally distributed



Fisher z transformation

Normal distribution



...

Standard normal distribution

VS

Standard normal distribution

Confidence Interval of standard normal distribution (1.96)

Z Score

The standard score (more commonly referred to as a z-score) is a very useful statistic because it allows us to calculate the probability of a score occurring within our normal distribution and enables us to compare two scores that are from different normal distributions. The standard score does this by converting (in other words, standardizing) scores in a normal distribution to z-scores in what becomes a standard normal distribution.

--- <https://statistics.laerd.com/statistical-guides/standard-score.php>



$$z = \frac{r' - \rho'}{S_{r'}}$$

Z Score

$$z = \frac{r' - \rho'}{S_{r'}}$$



$$r' = \frac{1}{2} \ln \left(\frac{1+r}{1-r} \right) = \text{arctanh}(r)$$

$$S_{r'} = \frac{1}{\sqrt{N-3}}$$

$$\rho' = 0 \quad \text{Based on the Null Hypothesis}$$

Fisher z transformation

Pearson correlation coefficient--- *r is correlation of two features x and y*



Fisher z transformation--- *r' is the Fisher transformation of r*



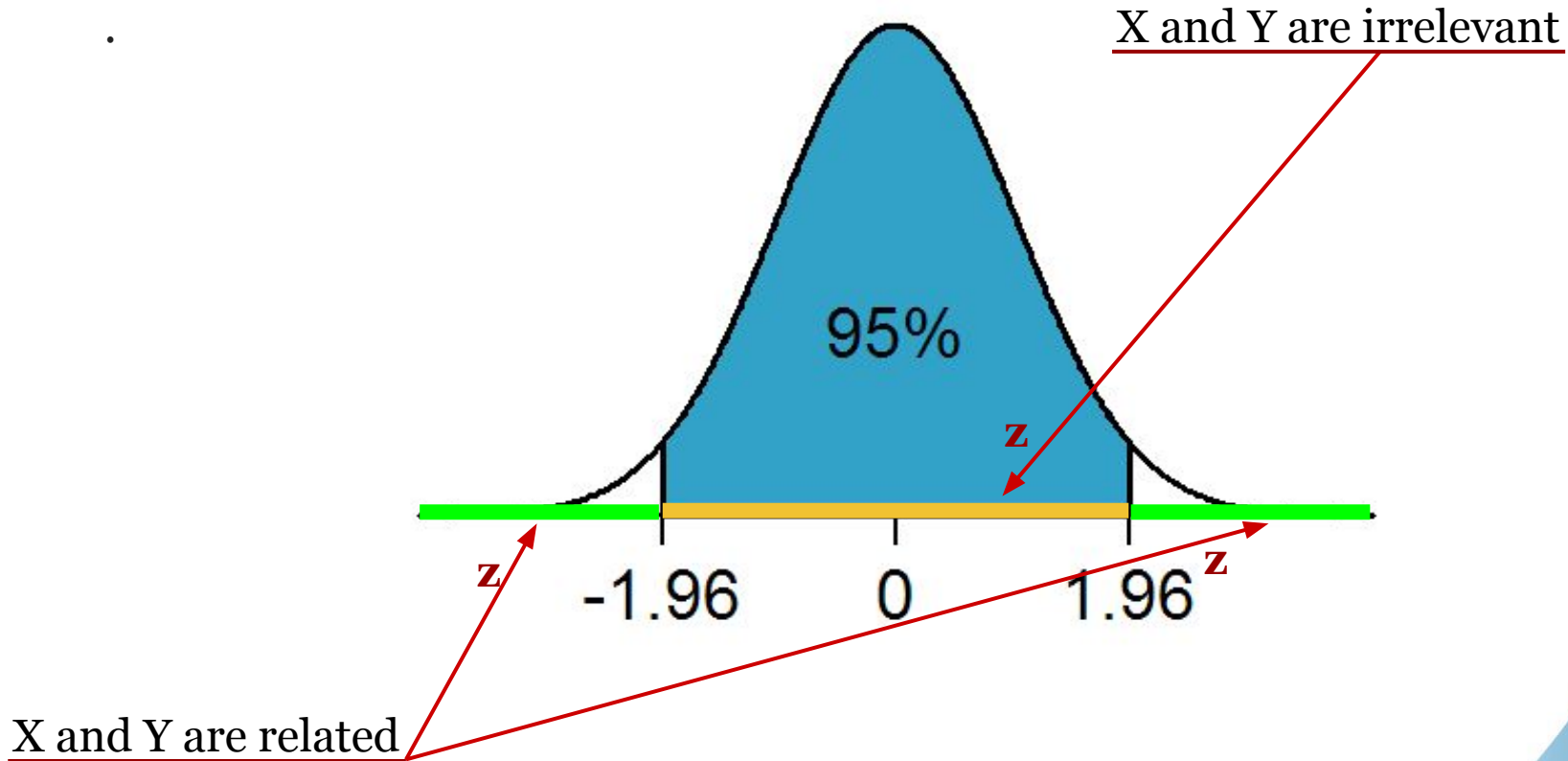
Z Score--- *z is the z-score of r'*

VS

Confidence Interval of standard normal distribution (1.96)

Confidence Interval

If z is in the confidence interval $(-1.96, 1.96)$, that means we 95% believe features x and y are irrelevant, otherwise, they are relevant features.



SAOLA

Algorithm 1: *The SAOLA Algorithm*

Data:

F_i : predictive features; C : the class attribute;

δ_1 : a relevance threshold ($0 \leq \delta_1 \leq 1$);

δ_2 : a correlation bound of $I(F_i; Y)$;

$S_{t_{i-1}}^*$: the selected feature set at time t_{i-1} ;

$S_{t_i}^*$: the selected feature set at time t_i

1 **repeat**

2 Get a new feature F_i at time t_i ;

3 /*Solve Eq.(2)*/

4 **if** $I(F_i, C) < \delta_1$ **then**

5 | Discard F_i , and go to Step 18;

6 **end**

7 **for each** feature $Y \in S_{t_{i-1}}^*$ **do**

8 /*Solve Eq.(3)*/

9 **if** $I(Y; C) > I(F_i; C) \ \& \ I(F_i; Y) \geq \delta_2$ **then**

10 | Discard F_i , go to Step 18;

11 **end**

12 /*Solve Eq.(4)*/

13 **if** $I(F_i; C) > I(Y; C) \ \& \ I(F_i; Y) \geq \delta_2$ **then**

14 | $S_{t_{i-1}}^* = S_{t_{i-1}}^* - Y$;


15 **end**

16 **end**

17 $S_{t_i}^* = S_{t_{i-1}}^* \cup F_i$;

18 **until** no features are available;

19 Output $S_{t_i}^*$;


$$|Z_{f_i_c}| < 1.96$$


$$|Z_{y_c}| > |Z_{f_i_c}| \ \& \ |Z_{f_i_y}| \geq 1.96$$


$$|Z_{f_i_c}| > |Z_{y_c}| \ \& \ |Z_{f_i_y}| \geq 1.96$$

Task

Implement SAOLA in Python.

Function name: Saola (data, label)

Return: M , where M is the markov blanket of the label.

Reference

- ❖ [Towards scalable and accurate online feature selection for big data](https://pdfs.semanticscholar.org/7353/oe3d8f2d3a88b3boe3d7e3c5d992227ec614.pdf)
<https://pdfs.semanticscholar.org/7353/oe3d8f2d3a88b3boe3d7e3c5d992227ec614.pdf>
- ❖ [Pearson correlation coefficient](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.corr.html)
<https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.corr.html>

Thank You!