Multivariate Analysis and Statistical Learning PC Algorithm's implementation

Authors: Alex Foglia, Tommaso Puccetti

Università degli Studi di Firenze

21/12/2018

Theoretical references (1)

- Bayesian Networks can be rappresented as a **directed acyclic graph (DAG)**
- ”acyclic” means that there are no paths starting from a node v that ends with v itself, $\forall v \in G$

Theoretical references (2)

Let $G = (V, E)$ be a DAG relative to a finite set $X = \{X_v \forall v \in V\}$ of casual variables, then:

$$\forall u, v \in V \; non \; adjacent \; | v \in nd(u) \Rightarrow u \perp\!\!\!\perp v | nd(u) - v$$

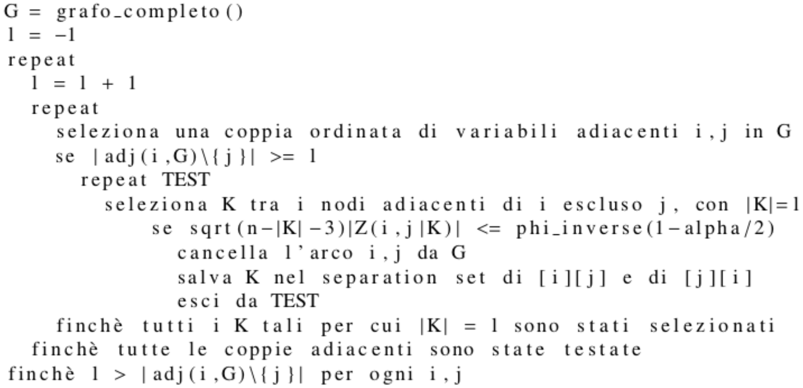
Where $nd(u)$ is the set of **non-descendant** n of n, that are all those nodes u' for which there is no path from u to u'.

PC-Algorithm

Given a set of variables with a joint Gaussian probability distribution, it is possible to learn the DAG closer to the sample through the use of **PC-Algorithm**.
It is composed of two sub-functions that solve two different problems:

1. The construction of the skeleton (or **Moral Graph**)
2. The construction of the DAG from a given skeleton

Pseudocode: skeleton generation



```
G = grafo_completo()
l = -1
repeat
  l = l + 1
  repeat
    seleziona una coppia ordinata di variabili adiacenti i,j in G
    se |adj(i,G)\{j}| >= l
      repeat TEST
        seleziona K tra i nodi adiacenti di i escluso j, con |K|=l
            se sqrt(n-|K|-3)|Z(i,j|K)| <= phi_inverse(1-alpha/2)
              cancella l'arco i,j da G
              salva K nel separation set di [i][j] e di [j][i]
              esci da TEST
    finchè tutti i K tali per cui |K| = l sono stati selezionati
  finchè tutte le coppie adiacenti sono state testate
finchè l > |adj(i,G)\{j}| per ogni i,j
```

PC-Algorithm for the skeleton: implementation (1)

- If the corr_matrix parameter is setted to ”None” then call the ***tocor()*** function that returns the *correlation* matrix from sigma matrix.
- Initialize the *adjacency matrix* G of the *complete graph* with all the elments equals to 1, and set the diagonal elements to 0.
- Instantiate the function ***adj()*** that takes a node x and return the neighbour of x.

INSERIRE CODICE CORR

Skeleton generation: implementation (2)

1. Set counter *l* to zero.
2. Iterating all the elements of the matrix, append in the act_ind list all of the couples x,y of nodes that has an edge in common.
3. Taken a couple x,y in act_ind, create the set of neighbor relative to x using the ***adj()*** function.
4. Calculate the set of all possible combinations of nodes from the neighbor's set, with lenght equal to l.
5. Do the **independence test** for x,y data K, for all K in neighbor set:
    - if $pvalue < alpha$ then delete the edge x,y and y,x from the adjacency matrix.
6. Repeat for every elements of the adjacency matrix.

Skeleton generation: implementation (3)