Machine Learning Assignment 2

Shenggan Cheng
Shanghai Jiao Tong University
515030910094
chengshenggan@sjtu.edu.cn

Abstract

This report is for the assignment 2 in SJTU Machine Learning Course. And it has five parts for five problems in this assignment.

1. PCA algorithm

1.1. Problem Discription

Give at least two algorithms that could take data set $X = \{x_1, ..., x_N\}, x_t \in R^{n\times 1}, \forall t$ as input, and output the first principal component w. Specify the computational details of the algorithms, and discuss the advantages or limitations of the algorithms.

1.2. Algorithm 1

Algorithm 1 is to minimize the error decribed in figure 1. Then using lagrangian multiplier to find the w that minimize the error $\sum e_t$, and this w is the first principal component. The detail of the algorithm is decribed below.

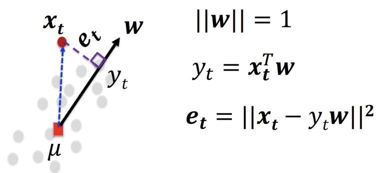


Figure 1. The define of error in Algorithm 1

Algorithm 1 Lagrangian Multiplier

```
1: $J(w) = \frac{1}{N}\sum_{t=1}^{N} ||x_t - (x_t^T w)w||$
2: $L(w, \lambda) = J(w) - \lambda(w^T w - 1)$
3: $\frac{\partial L}{\partial w} = \frac{\partial J}{\partial w} - \lambda\frac{\partial(w^T w-1)}{\partial w} = 0$
4: $\sum_x w = -\lambda w$
5: return the $w_i$ which has the largest $\lambda_i$
```

1.3. Algorithm 2

From matrix factorization perspectives, we can use SVD to get the PCA results.

Algorithm 2 Lagrangian Multiplier

```
1: $\sum_x = \frac{1}{N}\sum^N XX^T = U\Lambda U^T$
2: $X = UDV^T, XX^T = UD^2U^T$
3: $U = [\mu_1, \mu_2, ..., \mu_k]$
4: return the consponse $v_k$ with the largest $\mu_k$
```

1.4. Algorithm 3

The computation procedure of the network can be fomulated as:

$$X' = W^T W X$$

And we will minimize the difference of X' and X, so:

$$min(||X' - X||_2 = ||(W^T W - I)X||_2)$$

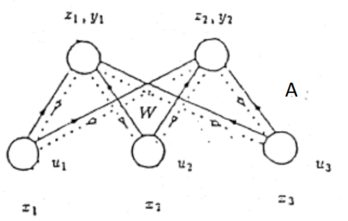


Figure 2. Least Mean Square Error Reconstruction (LMSER)

1.5. Comparison

Algorithm 1 has low computation complexity but can not compute all components. While Algorithm 2 can compute all components but has high computation complexity. Algorithm 3 is simple to control the number of principal components to output but can be cost much computation.

2. Factor Analysis

2.1. Problem Discription

Calculate the Bayesian posterior $p(y|x)$ of the Factor Analysis model $x = Ay + \mu + e$, with $p(x|y) = G(x|Ay+\mu, \Sigma_e), p(y) = G(y|0, \Sigma_y)$, where $G(z|\mu, \Sigma)$ denotes Gaussian distribution density with mean μ and covariance matrix Σ.

2.2. Solution

According the Bayesian formula, bayesian posterior $p(y|x)$ can be computed as:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

$p(x|y)$ and $p(y)$ are given as $G(x|Ay + \mu, \Sigma_e)$ and $G(y|0, \Sigma_y)$. Than we will calculate the $E(x)$ and $Cov(x)$ to get $p(x)$.

$$E(x) = E(Ay + \mu + e) = E(y) + \mu + E(e) = \mu$$

$$\begin{aligned} Cov(x) &= Cov(Ay + \mu + e) \\ &= E((Ay + \mu + e - E(x))(Ay + \mu + e - E(x))^T) \\ &= \Sigma_y A^T A + \Sigma_e \end{aligned}$$

Then we get the Bayesian posterior $p(y|x)$:

$$p(y|x) = \frac{G(x|Ay + \mu, \Sigma_e)G(y|0, \Sigma_y)}{G(x|\mu, \Sigma_y A^T A + \Sigma_e)}$$

3. Independent Component Analysis

Explain why maximizing non-Gaussianity could be used as a principle for ICA estimation.

3.1. Solution

In signal processing, independent component analysis (ICA) is a computational method for separating a multivariate signal into additive subcomponents.

The Central Limit Theorem, a classical result in probability theory, tells that the distribution of a sum of independent random variables tends toward a Gaussian distribution, under certain conditions.

Thus, a sum of two independent random variables usually has a distribution that is closer to Gaussian than any of the two original random variables.

$$y = W^T x = W^T As = z^T s$$

$z^T s$ is more Gaussian than any of the s_i. Therefore, we could take w that maximizes the non-Gaussianity.

From the aspect of information theory, we can give another explaination, gaussianity mens largest entropy among all random variables of equal variance. Below is the proof of this statement.

We use Lagrangian Multiplier and obtain the objective function:

$$\begin{aligned} J(p) = &- \int_{\infty}^{\infty} p(x) lnp(x) dx \\ &+\lambda_0(\int_{\infty}^{\infty} p(x)dx - 1) \\ &+\lambda_1(\int_{\infty}^{\infty} p(x)dx - \mu) \\ &+\lambda_2(\int_{\infty}^{\infty} x^2 p(x)dx - \sigma^2) \end{aligned}$$

and partial derivatives

$$\frac{\partial}{\partial p(x)dx} J(p) = -lnp(x) - 1 + \mu_0 + \mu_1 x$$
$$\frac{\partial^2}{\partial p(x)^2 dx} J(p) = -\frac{1}{p(x)}$$

leading to

$$p(x) = e^{(\mu_0 - 1) + \mu_1 x + \mu_2 x^2}$$

So, we proof that gaussianity mens largest entropy among all random variables of equal variance.

4. Causal Discovery Algorithms

Apply one causal discovery algorithm on a real world problem. You need to specify the details of the problem, collect the data by yourself or from a public website, briefly summarize what algorithm you use, and explain the results.

4.1. Solution

As for applying one causal discovery algorithm on a real world problem, I chooese Titanic Disaster Dataset from Kaggle. The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.

We choose some of the featrue in this causal discovery algorithm experiment. The data dictionaty can describe in the below table.

Variable	Definition
survival	Survival
Pclass	Ticket class
Sex	Sex
Age	Age in years
SibSp	siblings/spouses aboard the Titanic
Parch	parents/children aboard the Titanic
Fare	Passenger fare
Embarked	Port of Embarkation

And I choose a software named Tetrad to complete the causal discovery experiment. And I choose PC ALL, FCI and LiNGAM as the causal discovery algorithm. And the results of this three algorithms can see in figures below.

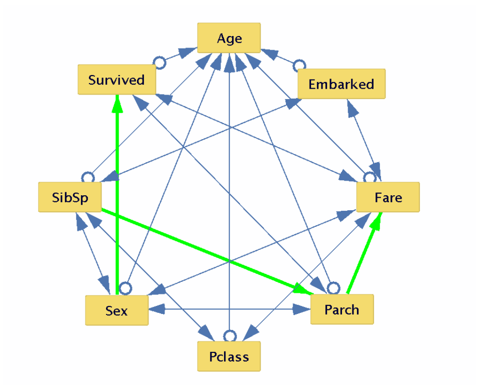


Figure 3. Results of PC ALL

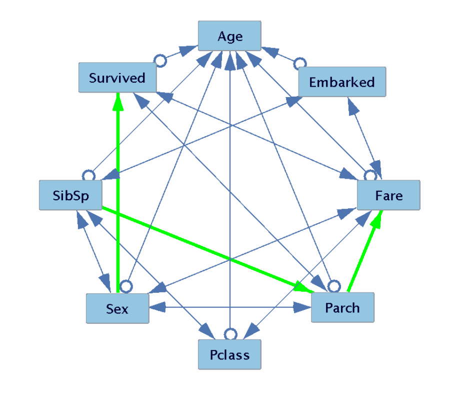


Figure 4. Results of FCI

As we can see, there three type of edges in figures.

$A - -- > B$ means that A is a cause of B. It may be a direct or indirect cause that may include other measured variables. Also, there may be an unmeasured confounder of A and B.

$A < -- > B$ means that there is an unmeasured confounder (call it L) of A and B. There may be measured variables along the causal pathway from L to A or from L to B.

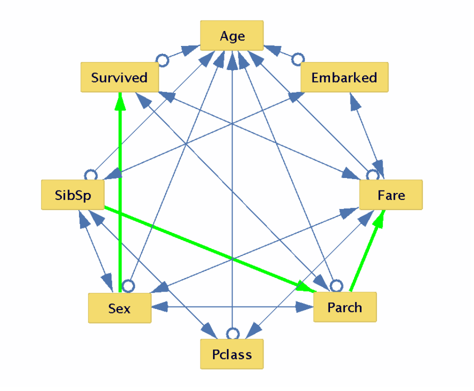


Figure 5. Results of LiNGAM

$A \circ -- > B$ means that either A is a cause of B (i.e, A –¿ B) or there is an unmeasured confounder of A and B (i.e, A ¡-¿ B) or both.

So, we can find that three algorithms gives the barely same results. For example, Sex is a cause of Survival, while Survival and Fare share an unmeasured confounder. And there many interesting points which cen be found in the figures.

5. Causal Tree Reconstruct

Give an algorithm to reconstruct a causal tree based on star-decomposable necessary and sufficient conditions.

5.1. Solution

Let T be a rooted tree with n leaves $x_1, x_2, ..., x_n$. A leaf x_i is said to the leader of the triple(x_i, x_j, x_k) if the path from the root to x_i does not contain the deepest common ancestor of x_j and x_k. If a triple does not have a leader then the deepest common ancestor of all three leaves is also a common ancestor of any two of them.

Then we present an algorithm to reconstruct a tree where the available information is the leader of every triple of leaves. And try to minimize the number of triples.

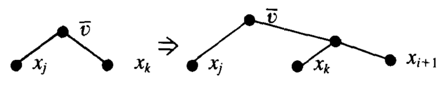
Algorithm add(integer i) begin

1. $T_c = T_i(T_c$ is a subtree of T_i to whichx_{i+1} is to be added. It becomes progressively smaller by eliminating those sections of T_i known not to contain x_{i+1})

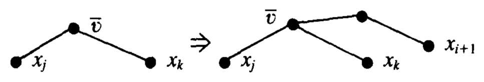
2. $s :=$ the number of leaves in T_c

3. If $s = 2$ let v be the root of T_c and x_j, x_k its teo leaves.

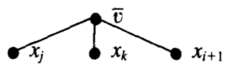
```
4.  If $s > 2$ select as $v$ any node of $T_c$ for which $\frac{s}{k+1} < des(v) <= \frac{sk}{k+1}$ and let $x_j, x_k$ be two leaves whose common ancestor is $v$.
5.  Ask for the leader of the triple $(x_{i+1}, x_j, x_k)$
6.  If $x > 2$ then begin
7.  Define a partition of $T_c$ into teo subtrees:$T_{c_1}$ rooted at $v$ with all the descendants of $v$ and $T_{c_2} = T_c - T_{c_1}$ in which $v$ is considered a leaf
8.  If $x_{i+1}$ is the leader of $(x_{i+1}, x_j, x_k)$ then set $T_c = T_{c_2}$
9.  If there is no leader, set $T_c = T_{c_1}$ from which the two sons of $v$ whose descendants are $x_j$ and $x_k$ are removed with all their descendants.
10. If $x_j$(or $x_k$) is the leader, set $T_c =$ the subtree of $T_{c_1}$ rooted at that son of $v$ which is the ancestor of $x_k$(or $x_j$)
11. Go to 2 END
12. If $s = 2$ then begin
13. If $x_j$(or $x_k$) is the leader add a new node on the edge of $x_k$ or $x_j$, respectively, and make it the father $x_{i+1}$.
```



```
14. If $x_{i+1}$ is the leader add a new root and make $x_{i+1}$ and the old root $v$ its sons.
```



```
15. If there is no leader, make $x_{i+1}$ a son of $v$.
```



```
16. END END add
```