Homework 2

CS420-Machine learning, Shikui Tu, Spring 2018

* Name:Zhiwen Qiang Student ID:515030910367 Email: qlightman@163.com

1 PCA algorithm

1.1 Algorithm 1

1.1.1 Pseudo code

```
Algorithm 1: Algorithm 1
input  : Data set $D = \{x1, \ldots, x_N\}, x_t \in R^{n*1}, \forall t$ as input
output: the first principal component $w$
Centralize the sample: $x_i \leftarrow x_i - \frac{1}{m}\sum_{i=1}^{m} x_i$;
Calculate the covariance matrix of the sample: $XX^T$;
Eigendecomposition of the covariance matrix: $XX^T$;
Obtain the corresponding eigenvector of the maximum eigenvalue.
```

1.1.2 Computational Details

$$J(w) = \frac{1}{N}\sum_{t=1}^{N} ||x_t - (x_t^T w)w||^2 = x_t^T x_t - w^T(x_t x_t^T)w \tag{1.1}$$

Where $J(w)$ is the Mean Square Error (MSE).

Using Lagrange multiplier λ, we have

$$L(x_t, w) = J(x_t, w) - \lambda(w^T w - 1) \tag{1.2}$$

$$\frac{\partial J(w)}{\partial w} - \lambda\frac{\partial(w^T w - 1)}{\partial w} = -2(\sum_x w) - \lambda * 2w = 0 \tag{1.3}$$

That is:

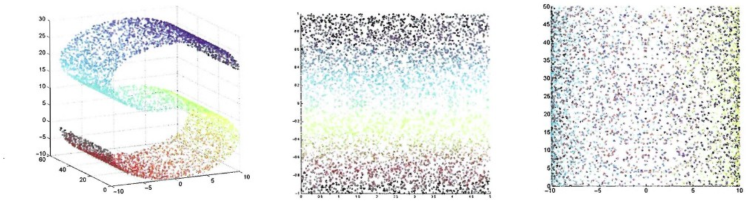
$$\sum_x w = (-\lambda)w \tag{1.4}$$

1.1.3 Advantages

- It is an unsupervised method, which means that no information about groups is used in the dimension reduction.
- It is the simplest of the true eigenvector-based multivariate analyses.

1.1.4 Limitations

- The results depend on the scaling of the variables.
- It assumes that the mapping function form high dimension to low dimension is linear, so it cannot reveal the intrinsic low dimension structure when dealing with non-linear mapping problem. As is shown in figure 1.1, the result obtianed from PCA lost the original structure.



(a) 三维空间中的观察　(b) 本真二维结构　(c) PCA 降维结果

Figure 1.1: PCA results lack the original structrue.

1.2 Algorithm 2

1.2.1 Pseudo code

Algorithm 2: Algorithm 2

```
input  : Data set $X = \{x1, \ldots, x_N\}, x_t \in R^{n*1}, \forall t$ as input
output: the first principal component $w$
Calculate the distance matrix $D \in R^{m*m}$, where $dist_{ij}$ is the distance between sample $x_i$ to $x_j$;
Calculate $dist^2_{i\cdot}$, $dist^2_{\cdot j}$, $dist^2_{\cdot\cdot}$ according to equation 1.9 to 1.11.
Calculate matrix $B$ according to equation 1.12
Eigendecomposition of matrix $B$;
$\hat{\Lambda}$ is the diagonal matrix composed by the maximum eigenvalue $d$, $\hat{V}$ is the corresponding eigenvector matrix.
The first principal component $w = \hat{V}\hat{\Lambda}^{1/2} \in R^{m*1}$
```

1.2.2 Computational Details

Let $B = Z^T Z \in R^{m*m}$, where B is the matrix being dimensionality reduced, $b_{ij} = z_i^T z_j$, we have:

$$dist^2_{ij} = ||z_i||^2 + ||z_j||^2 - 2z_i^T z_j = b_{ii} + b_{jj} - 2b_{ij} \tag{1.5}$$

After being centralize, it is easy to know:

$$\sum_{i=1}^{m} dist^2_{ij} = tr(B) + mb_{jj} \tag{1.6}$$

$$\sum_{j=1}^{m} dist_{ij}^2 = tr(B) + mb_{ii} \tag{1.7}$$

$$\sum_{i=1}^{m} \sum^{m} j = 1 dist_{ij}^2 = 2mtr(B) \tag{1.8}$$

Let:

$$dist_{i.}^2 = \frac{1}{m} \sum_{j=1}^{m} dist_{ij}^2 \tag{1.9}$$

$$dist_{.j}^2 = \frac{1}{m} \sum_{i=1}^{m} dist_{ij}^2 \tag{1.10}$$

$$dist_{..}^2 = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} dist_{ij}^2 \tag{1.11}$$

From equation 1.5 to 1.11, we have:

$$b_{ij} = -\frac{1}{2}(dist_{ij}^2 - dist_{i.}^2 - dist_{.j}^2 + dist_{..}^2) \tag{1.12}$$

1.2.3 Advantages

- It is a form of non-linear dimensionality reduction, which means it can reveal the intrinsic structure.
- It is also an unsupervised method, which means that no information about groups is used in the dimension reduction.

1.2.4 Limitations

It require more training resources than Algorithm 1.

2 Factor Analysis (FA)

2.1 E-step:

$$\begin{gathered} p^{old}(y|x) = \frac{G(y|0, I)G(x|Ay + \mu, \sigma^2 I)}{G(x|\mu, AA^T + \sigma^2 I)} \\ E[y|x] = Wx \qquad W = A^T(AA^T + \sigma^2 I)^{-1} \\ E[yy^T|x] = I - WA + Wxx^T W^T \end{gathered}$$

2.2 M-step:

$$Q = \int p^{old}(y|x) \cdot \ln[G(y|0, I)G(x|Ay + \mu, \sigma^2 I)]dy$$

In order to $max(Q(p^{old}(y|x), \Theta))$, we have:

$$A^{new} = (\sum_{t=1}^{N} x_t(E[y|x_t])^T)(\sum_{t=1}^{N} E[yy^T|x_t])^{-1}$$

$$\sigma^{2^{new}} = \frac{1}{Nd} Tr\Bigg\{ \sum_{t=1}^{N} \{x_t x_t^T - A^{new} E[y|x_t] x_t^T\} \Bigg\}$$

3 Independent Component Analysis (ICA)

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.

$$y = w^T x = w^T As = (w^T A)s = z^T s \tag{3.1}$$

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

4 Causal discovery algorithms

4.1 Dataset description

The data I use is a database with so far 108 two-variable cause-effect pairs created at Max-Planck-Institute for Biological Cybernetics in Tuebingen, Germany. It can be accessed via http://webdav.tuebingen.mpg.de/cause-effect/ . The datafiles are .txt-files and contain two variables, one is the cause and the other the effect. For every example there exists a description file where you can find the ground truth and how the data was derived. For example, in the pair0001.txt file, the two variables are altitude, temperature, the ground truth is that altitude is the cause, temperature is effect.

4.2 Problem details

The problem here is to find which is the cause and which the effect of the 108 two-variable cause-effect pairs.

4.3 Algorithm details

Here I use the algorithm described in Information-geometric causal inference, Daniusis et al. (2010); Janzing et al. (2012). The author of the paper first proposed two Postulate:

1. If $X \to Y$, the distribution of X and the function f mapping X to Y are independent since they correspond to independent mechanisms of nature.
2. Let ε_X and ε_Y define exponential families of smooth reference distributions for X and Y , respectively. Let u denote the projection of p_X onto ε_X and u_f its image under f. If $X \to Y$, then

$$D(p_Y||\varepsilon_Y) = D(p_X||\varepsilon_X) + D(u_f||\varepsilon_Y). \tag{4.1}$$

The author then conclude that the consequence of Postulate 2 and the positivity of relative entropy is that if $X \to Y$,

$$C_{X\to Y} := D(p_X||\varepsilon_X) - D(p_Y||\varepsilon_Y) < 0. \tag{4.2}$$

On the other hand, if $Y \to X$,We have

$$C_{Y\to X} := D(p_Y||\varepsilon_Y) - D(p_X||\varepsilon_X) < 0. \tag{4.3}$$

Therefore, we have the defination of the Information Geometric Causal Inference algorithm:

Causal Inference method (IGCI): Given $C_{X\to Y}$, infer that X causes Y if $C_{X\to Y} < 0$, or that Y causes X if $C_{X\to Y} > 0$.

4.4 Results

4.4.1 Results presented in the paper

The author in the paper provide comparative results of IGCI method (using two different reference measures) and two other causal inference methods that are suit- able for inferring the causal direction between pairs of variables: the Additive Noise (AN) model and an implementation of the Post-NonLinear (PNL) model. The results in shown in table 1.

Table 1: Results for CauseEffectPairs data set (51 pairs)

Method	Decisions(%)	Accuracy(%)
IGCI (uniform)	100	78
IGCI (Gaussian)	100	76
AN (Gaussian)	20	100
PNL	82	95

4.4.2 Results produced myself

It is worthy noting that the CauseEffectPairs data set is growing and I implemented the algorithm in a 108 two-variable cause-effect pairs. The matlab code along with the dataset is in the 515030910367_qiangzhiwen_hw2 zip file. The environment I use is matlab 2015b, just execute the *main.m* file and you can see the results.

Table 2: Results for CauseEffectPairs data set (108 pairs)

Method	Decisions(%)	Accuracy(%)
IGCI (uniform,entropy)	100	65
IGCI (uniform,integral)	100	65
IGCI (Gaussian,entropy)	100	61
IGCI (Gaussian,integral)	100	59

The results is shown in table 2. I use two different reference measures and two different estimators of IGCI method to evaulate the results. We can see the accuracy is lower than the result in the paper. So I run the code on the inital dataset(51 pairs), the result is simailar compared to the paper. So I think the reason is that the additional pairs (57 pairs) have high noise level than the initial 51 pairs.

5 Causal tree reconstruction

First we give a algorithm to reconstruct rooted trees. The detail in shown in algorithm 3.

Algorithm 3: Rooted trees reconstruction algorithm

```
   input  : Data set $X = \{x1, \ldots, x_N\}, x_t \in R^{n*1}, \forall t$ as input
   output: the first principal component $w$
1  $T_c = T_i$($T_c$ is a subtree of $T_i$ to which $x_{i+1}$ 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 $s = 2$ then
4      let $v$ be the root of $T_c$ and $x_j$,$x_k$ its two leaves.
5  if $s > 2$ then
6      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 $v$
7  Ask for the leader of the triple $(x_{i+1}, x_j, x_k)$. if $s > 2$ then
8      return to 1
9  Define a partition of $T_c$ into two subtrees: $T_{c1}$ rooted at $v$ with all the
   descendants of $v$ and $T_{c2} = T_c - T_{c1}$ in which $v$ is considered a leaf. if
   $x_{i+1}$ is the leader of $(x_{i+1}, x_j, x_k)$ then
10     set $T_c = T_{c2}$
11 if there is no leader then
12     $T_c = T_{c1}$ from which the two sons of $v$ whose descendants are $x_j$ and
       $x_k$ are removed with all their descendants.
13 if $x_j$ or $x_k$ is the leader then
14     set $T_c =$ the subtree of $T_{c1}$ rooted at that son of $v$ which is the
       ancestor of $x_k$ or $x_j$.
15 if $s = 2$ then
16     return to 1
17 if $x_j$ or $x_k$ is the leader then
18     add a new node on the edge of $x_k$ or $x_j$, and make it the father of
       $x_{i+1}$.
19 if $x_{i+1}$ is the leader then
20     add a new root and make $x_{i+1}$ and the old root $v$ its sons.
21 if there is no leader then
22     make $x_{i+1}$ a son of $v$.
23 return to 1
```

Then we prove that the rooted-tree proceduce can be adapted to unrooted trees as well.

Let T be an unrooted tree in which the degree of every node is at least three, and let u, v, w, x be any quadruple of leaves. We say that x pairs with u relative to (v, w) if the path from x to u is edge-disjoint relative to the path form v to w.

Remove a leaf x of T and examine the remaining tree T_1 as rooted at the node x_1 with which x is adjacent in T. Then we can use algorithm 3 to reconstruct T_1 rooted at x_1 and finally add x as a son of x_1 to obtian the unrooted tree T.