Homework 5

201900800302 赵淇涛

1. PCA

Overview

PCA is widely used for dimensionality reduction of data, lossy data compression, and e.t.c..

Specifically, we would find a lower-dimansional projection of data, which minimizes the *reconstruction error* (mean squared error between data points and projections) or alternatively, maximizes the variance of projected data in the *principal subspace*.

Later we shall see the *principal subspace* can be found by calculating the eigenvectors of the data covariance matrix $\mathbf{S}$ and the *reconstruction error* can be given by the eigenvalues corresponding to the eigenvectors of $\mathbf{S}$.

Algorithm in detail

Here the **PCA** algorithm is introduced from a perspective of minimum construction error.

Assume we have data points $\{\mathbf{x}_n\}$ of dimension D, each of which can be represented by a complete orthonormal set of D-dimensional basis vectors $\{\mathbf{u}_i\}$, where $i = 1, \ldots, D$, satisfying $\mathbf{u}_i^T\mathbf{u}_j = \delta_{ij}$,

$$\mathbf{x}_n = \sum_{i=1}^{D} \alpha_{ni}\mathbf{u}_i$$

where α_{ni} are coefficients. Making use of the orthonormality property of basis vectors, the $\mathbf{x}_n$ can be further written as

$$\mathbf{x}_n = \sum_{i=1}^{D} (\mathbf{x}_n^T\mathbf{u}_i)\ \mathbf{u}_i.$$

To reduce the dimensionality of data, we can approximate the data using first M ($M < D$) eigenvectors corresponding to a lower-dimensional subspace and fix the coefficients of other eigenvectors

$$\widetilde{\mathbf{x}}_n = \sum_{i=1}^{M} z_{ni}\mathbf{u}_i + \sum_{i=M+1}^{D} b_i\mathbf{u}_i,$$

where the $\{z_{ni}\}$ depend on the particular data point while the $\{b_i\}$ are constants. To measure the precision of approximation of $\widetilde{\mathbf{x}}_n$ for $\mathbf{x}_n$, an objective function J is defined below.

$$J = \frac{1}{N}\sum_{n=1}^{N} ||\mathbf{x}_n - \widetilde{\mathbf{x}}_n||^2$$

Setting the derivative of J w.r.t. z_{nj} to zero, where $j = 1, \ldots, M$, gives

$$z_{nj} = \mathbf{x}_n^T\mathbf{u}_j.$$

Similarly, setting the derivative of J w.r.t. b_j to zero, where $j = M+1, \ldots, D$, we obtain

$$b_j = \overline{\mathbf{x}}_n^T \mathbf{u}_j.$$

If we substitute for z_{ni} and b_i, and make use of the expansion above, we obtain

$$\mathbf{x}_n - \widetilde{\mathbf{x}}_n = \sum_{i=M+1}^{D} \left\{ (\mathbf{x}_n - \overline{\mathbf{x}})^T \mathbf{u}_i \right\} \mathbf{u}_i.$$

Therefore, the objective function J can be written as

$$J = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=M+1}^{D} (\mathbf{x}_n^T \mathbf{u}_i - \overline{\mathbf{x}}^T \mathbf{u}_i)^2 = \sum_{i=M+1}^{D} \mathbf{u}_i^T \mathbf{S} \mathbf{u}_i,$$

where $\mathbf{S}$ is the covariance matrix of data. Let's consider a simplest case where $D = 2$ and $M = 1$. To minimize the function $J = \mathbf{u}_2^T \mathbf{S} \mathbf{u}_2$, subject to $\mathbf{u}_2^T \mathbf{u}_2 = 1$, we can use Lagrange multiplier method. Hence the objective function to be minimized becomes

$$\widetilde{\mathbf{J}} = \mathbf{u}_2^T \mathbf{S} \mathbf{u}_2 + \lambda_2 (1 - \mathbf{u}_2^T \mathbf{u}_2).$$

The general solution for arbitrary D and M is obtained by choosing the $\{\mathbf{u}_i\}$ to be eigenvectors of the covariance matrix satisfying

$$\mathbf{S}\mathbf{u}_i = \lambda_i \mathbf{u}_i$$

where $i = 1, \ldots, D$.

Finally, the optimal reconstruction error J is given by

$$J = \sum_{i=M+1}^{D} \lambda_i.$$

Hence, basis vectors of the *principal subspace* are first M eigenvectors of the covariance matrix of data corresponding to largest eigenvalues and the minimum distortion measure $\mathbf{J}$ is given by the sum of rest smaller eigenvalues.

Code and process

0. Set up

```
import numpy as np
import matplotlib.pyplot as plt
```

1. Function *generate_data* to produce data points

```
def generate_data(dot_num):
    mean = [2, 1]
    covariance = [[0.5, 0.3], [0.3, 1]]
    return np.random.multivariate_normal(mean, covariance, dot_num)
```

2. Function *PCA* to calculate basis vectors, standardized data and the reconstruction error

```
def PCA(X):
    N, D = X.shape
    temp = np.ones((2, 50))
    mean = np.mean(X, axis=0)
    variance = ((X - mean)**2).sum(axis=0) / N

    # Standarize the data
    X_centered = (X - mean) / variance

    covariance = X.T.dot(X) / N
    w, v = np.linalg.eig(covariance)
    loss = np.min(w)
    return v[:, np.argmax(w)], v[:, np.argmin(w)], X_centered, loss # , loss
```

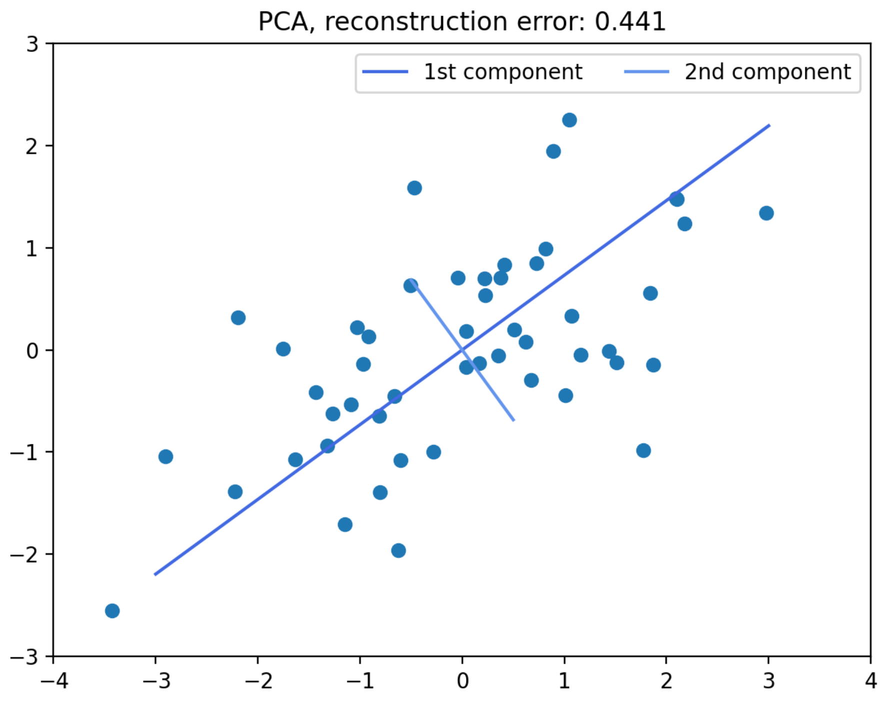
3. Main part

```
np.random.seed()
X = generate_data(50)
u_1, u_2, X_centered, loss = PCA(X)
print("Reconstruction loss", loss)
x_1 = np.arange(-3, 4)
x_2 = np.linspace(-0.75, 0.75)

# Plot the result
plt.figure(figsize=(6, 6))
plt.scatter(X_centered[:, 0], X_centered[:, 1])
plt.plot(x_1, x_1*u_1[1]/u_1[0], c="royalblue", label="1st component")
plt.plot(x_2, x_2*u_2[1]/u_2[0], c="cornflowerblue", label="2nd component")
plt.title("PCA")
plt.legend(loc="best", ncol=4)
plt.xlim(-4, 4)
plt.ylim(-3, 3)
plt.tight_layout()
plt.gca().set_aspect('equal', adjustable='box') # Set x-axis and y-axis to an equal
scale
plt.show()
```

Result and discussion

The result plot shows the standardized data, first principal component as well as the second one. The reconstruction error is also given in the title.

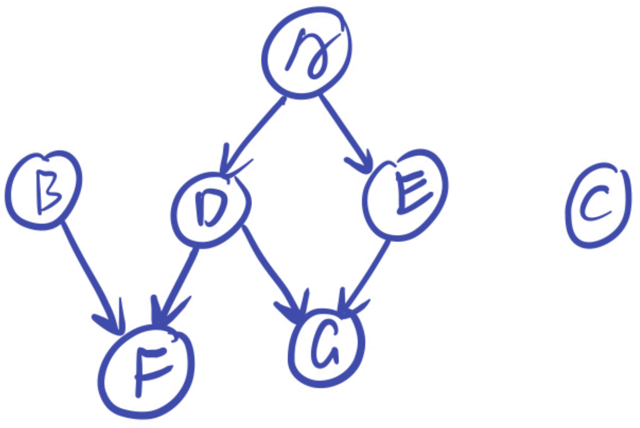


Intuitively, the data is approximately distributed along the vector correspongding to the first principal component with a large variance and therefore the projection of data onto the first principal component is a reasonable approximation.

2. Bayes Nets

(a)

The corresponding directed graph is shown below.



(b)

The factored joint distribution represented by the graph is given by

$$P(A,B,C,D,E,F) = P(A)P(B)P(C|A,B)P(D|B)P(E|C,D)P(F|E)$$

(c)

For the nodes that are not connected to a parent node(s), the needed parameter number is 1. For those connected to one or more parent nodes, the number of parameters is specified by the possible values of parents nodes. That is 2^n (n is the number of parent nodes).

Therefore, provided the formulation above, the total number of required parameters is

$$N = 1 + 1 + 4 + 2 + 4 + 2 = 14.$$

If we make use of the normalization rule of the probability (sum to 1), the number of the parameters can reduce to

~~$N = 14 - 1 = 13.$~~

好像不能用
归一化.

(d)

1. $A \perp\!\!\!\perp B$

   True. There is a v-structure and the child node is not observed.

2. $A \perp\!\!\!\perp B|C$

   False. The child node in the v-structure is observed.

3. $C \perp\!\!\!\perp D$

   False. Node C and D are in a common parent structure and the parent node is not observed.

4. $C \perp\!\!\!\perp D|E$

   False. Node C and D are also in a v-structure, where in this case the child node is observed.

5. $C \perp\!\!\!\perp D|B,F$

   False. Node C and D are in a v-structure, where one of their descendants is observed.

6. $F \perp\!\!\!\perp B$

   False. Node F and B are connected by an unblocked cascading path.

7. $F \perp\!\!\!\perp B|C$

   False. Node F and B are connected by an unblocked cascading path.

8. $F \perp\!\!\!\perp B|C,D$

   True. Both cascading paths between node F and B are blocked.

9. $F \perp\!\!\!\perp B|E$

   True. Both cascading paths between node F and B are blocked by node E.

10. $A \perp\!\!\!\perp F$

    False. Node A and F are connected by an unblocked cascading path.

11. $A \perp\!\!\!\perp F|C$

False. On the one hand, node A and B are not separate given node C in a v-structure. On the other, node B and F are connected by an unblocked cascading path and therefore are not separate. Hence, node A and F are not separate given node C.

12. $A \perp\!\!\!\perp F | D$

    False. Node A and F are connected by an unblocked cascading path.