Homework Assignment 6

Kexin Huang

April 29, 2018

1. (a) Why is it more efficient to process data points if they are lower-dimensional vectors? State one reason.

One reason is that when the data is a large high-dimensional and sometimes sparse matrix, it asks for a larger weight vector so the gradient descent is on larger weight so it kind of slow. However, lower-dimensional vectors give a principal representation so it contains crucial information and it is also smaller which leads to a smaller weight matrix, therefore, it is more efficient. Also, in one special case, when the input is a sparse large matrix, there are huge number of 0s that don't contain much information but still need to be processed so it is quite inefficient.

(b) What is a potential trouble of reducing the dimensionality of input vectors before training a classifier? State one reason.

Although dimensionality reduction preserves the principal information of the input, there is still reconstruction error. So it might lose crucial information containing in the original matrix and create bias that lead the classifier to have bias.

2. (a) Given a training set $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, show that the reconstruction error of principal component analysis (PCA) could be written down as

$$\frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}_n - \hat{\mathbf{x}_n}\|_2^2 = \sum_{j=q+1}^{d} \mathbf{w}_i^\top \mathbf{C}\mathbf{w}_i,$$

where $\mathbf{w}_i$ is the i-th principal component or the eigenvector of the input covariance matrix $\mathbf{C}$.

Ans: Suppose we reduce from dimension d to q. Therefore, $\mathbf{x}_n \in R^d$ and $\mathbf{x} \in R^{dxn}$ and $\hat{\mathbf{x}_n} \in R^q$ and $\hat{\mathbf{x}} \in R^{qxn}$. And we know $\mathbf{x} = WZ$ where $W \in R^{dxd}$ and $Z \in R^{dxn}$. We also know $\hat{\mathbf{x}} = W'Z'$ where W' is the top q columns of W, so $W' \in R^{dxq}$ and similarly, $Z' \in R^{qxn}$ and Z' is the top q rows from Z. From there, we can see that $\mathbf{x}_n = WZ_n$ where W is the whole weight vector and Z_n is the nth column of matrix Z, i.e. nth code vector. And we also see that $\hat{\mathbf{x}_n} = W'Z'_n$ where W' is the weight vector for reduced dimensions and Z'_n is nth column of Z'. Let's denote $\mathbf{x}_n^i$as the ith value of $\mathbf{x}_n$. We can derive $\mathbf{x}_n^i = \sum_{j=1}^{d} W_i^j * Z_n^i$ and similarly, $\hat{\mathbf{x}_n^i} = \sum_{j=1}^{q} W_i^j * (Z')_n^i$ where W_i^j means jth element in ith row of W, similar for Z.

Therefore, we have,
$\frac{1}{N}\sum_{n=1}^{N}\|\mathbf{x}_n-\hat{\mathbf{x}_n}\|_2^2$
$=\frac{1}{N}\sum_{n=1}^{N}\|\mathbf{x}_n-\hat{\mathbf{x}}_n\|_2^2$
$=\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{d}\sum_{j=1}^{d}[W_i^j*Z_n^j]^2-\sum_{j=1}^{q}[W_i^j*(Z')_n^j]^2$
$=\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{d}\sum_{j=q+1}^{d}[W_i^j*Z_n^j]^2+\sum_{j=1}^{q}[W_i^j*Z_n^j]^2-\sum_{j=1}^{q}[W_i^j*(Z')_n^j]^2$
$=\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{d}\sum_{j=q+1}^{d}[W_i^jZ_n^j]^2$
$=\frac{1}{N}\sum_{n=1}^{N}\sum_{j=q+1}^{d}[W^jZ^j]^2$ where W^j is the jth column of W.
$=\frac{1}{N}\sum_{n=1}^{N}\sum_{j=q+1}^{d}[Z^j]^2$because $[W^j]^2=1$ And we know $Z^j=(W^j)^TX$
$=\frac{1}{N}*N*\sum_{j=q+1}^{d}[(W^j)^TX]^2$
$=\sum_{j=q+1}^{d}[(W^j)^TXX^T(W^j)]$because it is an element product
$=\sum_{j=q+1}^{d}[(W^j)^TXX^T(W^j)]$
$=\sum_{j=q+1}^{d}[(W^j)^TC(W^j)]$

(b) Show that

$$\Sigma=\mathbf{W}^\top\mathbf{C}\mathbf{W}$$
$$\iff\sigma_j^2=\mathbf{w}_j^\top\mathbf{C}\mathbf{w}_j,\ \text{for all}\ j=1,\ldots,d,$$

where $\mathbf{W}$ is the weight matrix of PCA, $\mathbf{C}$ is the input covariance matrix, and

$$\Sigma=\operatorname{diag}(\sigma_1^2,\ldots,\sigma_q^2)=\begin{bmatrix}\sigma_1^2 & 0 & \cdots & 0\\ 0 & \sigma 2^2 & \cdots & 0\\ \vdots & 0 & \cdots & \vdots\\ \vdots & \vdots & \cdots & \vdots\\ 0 & 0 & \cdots & \sigma_q^2\end{bmatrix}$$

is the covariance matrix of the code vectors.

Ans: forward way: $\mathbf{W}^\top\mathbf{C}\mathbf{W}=\mathbf{W}^\top\mathbf{x}\mathbf{x}^\top\mathbf{W}=(\mathbf{W}^\top\mathbf{x})(\mathbf{x}^\top\mathbf{W})=ZZ^\top$ And denote Z^i as the ith row of this q x N matrix.

$$ZZ^\top=\begin{bmatrix}Z^1(Z^1)^\top & Z^1(Z^2)^\top & \cdots & Z^1(Z^q)^\top\\ Z^2(Z^1)^\top & Z^2(Z^2)^\top & \cdots & Z^2(Z^q)^\top\\ \vdots & Z^3(Z^2)^\top & \cdots & \vdots\\ \vdots & \vdots & \cdots & \vdots\\ Z^q(Z^1)^\top & \cdots & \cdots & Z^q(Z^q)^\top\end{bmatrix}$$

Therefore, we have $Z^i(Z^j)^\top=\sigma_i^2$ when i=j and $Z^i(Z^j)^\top=0$ when i !=j. Therefore, $\sigma_i^2=Z^i(Z^i)^\top=\mathbf{W}_i^\top X*X^\top\mathbf{W}_i$ where $\mathbf{W}_i$ is the i th column of W in $R\in dXq$.

Similarly, in the other way, we have $\sigma_j^2 = \mathbf{w}_j^\top \mathbf{C}\mathbf{w}_j,$ for all $j = 1, \ldots, d$ and $\mathbf{w}_j^\top \mathbf{C}\mathbf{w}_j = Z^j(Z^j)^\top$. Therefore, we have $\sigma_j^2 = Z^j(Z^j)^\top$. Also as Z is the code vector for PCA, we know $Cov(Z_i, Z_j) = 0$ when i !=j. Therefore, we have $\mathbf{W}^\top \mathbf{x}\mathbf{x}^\top \mathbf{W} = \Sigma'$

$$\Sigma' = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma 2^2 & \cdots & 0 \\ \vdots & 0 & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \sigma_q^2 \end{bmatrix}$$

Therefore, $\Sigma' = \Sigma$, we have $\Sigma = \mathbf{W}^\top \mathbf{C}\mathbf{W}$

3. (Programming Assignment) Complete the implementation of PCA and NMF using Python and scikit-learn. The completed notebooks must be submitted together with the answers to the questions above.

When submitting Jupyter notebooks, make sure to save printed outputs as well.

PCA `https://github.com/nyu-dl/Intro_to_ML_Lecture_Note/blob/master/homeworks/hw6_pca.ipynb`

NMF `https://github.com/nyu-dl/Intro_to_ML_Lecture_Note/blob/master/homeworks/hw6_nmf.ipynb`