Introduction to Machine Learning, Spring 2023
Homework 4

(Due Friday, May. 5 at 11:59pm (CST))

April 22, 2023

1. [20 points] Consider a dataset of n observations $\mathbf{X} \in \mathbb{R}^{n\times d}$, and our goal is to project the data onto a subspace having dimensionality p, $p < d$. Prove that PCA based on projected variance maximization is equivalent to PCA based on projected error (Euclidean error) minimization.

   Solution:

   Suppose $\mathbf{X}$ has been centralized. Let $\mathbb{V} \in \mathbb{R}^{d\times p}$ represent the projected matrix and $\mathbb{V}^T\mathbb{V} = \mathbb{I}$. Then we have two different optimization goals. The one based on projected variance maximization is

$$\textit{maximize} \qquad \|\mathbb{X}\mathbb{V}\mathbb{V}^T\|_F^2 = Tr(\mathbb{V}\mathbb{V}^T\mathbb{X}^T\mathbb{X}\mathbb{V}\mathbb{V}^T) = Tr(\mathbb{X}^T\mathbb{X}\mathbb{V}\mathbb{V}^T)$$

   , the other one based on projected error minimization is

$$\textit{minimize} \qquad \|\mathbb{X} - \mathbb{X}\mathbb{V}\mathbb{V}^T\|_F^2 = Tr((\mathbb{X} - \mathbb{X}\mathbb{V}\mathbb{V}^T)^T(\mathbb{X} - \mathbb{X}\mathbb{V}\mathbb{V}^T))$$

   And we have

$$\begin{aligned} Tr((\mathbb{X} - \mathbb{X}\mathbb{V}\mathbb{V}^T)^T(\mathbb{X} - \mathbb{X}\mathbb{V}\mathbb{V}^T)) &= Tr((\mathbb{I} - \mathbb{V}\mathbb{V}^T)^T\mathbb{X}^T\mathbb{X}(\mathbb{I} - \mathbb{V}\mathbb{V}^T)) \\ &= Tr(\mathbb{X}^T\mathbb{X}(\mathbb{I} - \mathbb{V}\mathbb{V}^T)) \\ &= Tr(\mathbb{X}^T\mathbb{X}) - Tr(\mathbb{X}^T\mathbb{X}\mathbb{V}\mathbb{V}^T) \end{aligned}$$

   Since $Tr(\mathbb{X}^T\mathbb{X})$ is a constant value, so projected variance maximization is equivalent to projected error minimization.

2. [30 points] Let's see how well you remember k-means clustering, also known as Lloyd's Algorithm. As usual, the input is a set of n sample points, $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_n \in \mathbb{R}^d$ and an integer k. There are no input labels; we want to assign each sample point $\mathbf{X}_j$ a label $y_j \in \{1, 2, \cdots, k\}$, which can be interpreted as "$\mathbf{X}_j$ is assigned to cluster y_j." The means of the clusters are written $\mu_1, \mu_2, \cdots, \mu_k$.

(a) What is the cost function that k-means clustering tries to minimize? Write it in terms of the $\mathbf{X}_j$'s, the y_j's, and the μ_j's. (Not the formula for the within-cluster variation, please; a formula that uses the means. We are flexible about what notation you use to sum the terms in the cost function; please explain it if it's not obvious.) [8 points]

Solution: One way to write it is

$$\sum_{i=1}^{k} \sum_{y_j=i} \|X_j - \mu_i\|^2$$

Another fine way is

$$\sum_{j=1}^{n} \|X_j - \mu_{y_j}\|^2$$

(b) Consider the step of the algorithm where the labels y_j are held fixed while the cluster means μ_j are updated. I asserted in lecture that it is easy to show with calculus that if we want to minimize the cost function, we should choose each μ_j to be the mean (centroid) of the sample points assigned to cluster i. Please do that calculus and show that this claim is correct. (Make sure you explain your notation for counting the points in a cluster.) Show your work and don't skip any steps of the derivation. [10 points]

Solution: Let n_j be the number of sample points assigned to cluster j. If we use the first version of the cost function, we have

$$\frac{\partial J}{\partial \mu_i} = \sum_{y_j=i} (\mu_i - X_j)$$

Setting that derivative to zero gives us

$$\begin{aligned}
\sum_{y_j=i} \mu_i &= \sum_{y_j=i} X_j \\
n_j \mu_i &= \sum_{y_j=i} X_j \\
\mu_i &= \frac{1}{n_j} \sum_{y_j=i} X_j
\end{aligned}$$

which indeed is the mean of the sample points assigned to cluster i.

If we use the second version of the cost function, we have

$$\frac{\partial J}{\partial \mu_i} = \frac{\partial}{\partial \mu_i} \sum_{j=1}^{n} \|X_j - \mu_{y_j}\|^2 = \sum_{y_j=i} (\mu_i - X_j)$$

and the rest of the derivation proceeds in the same way.

(c) Consider the step of the algorithm where the cluster means μ_j are held fixed while the labels y_j are updated. Sometimes a sample point $\mathbf{X}_j$ has several cluster means that are equally close. In class I said, "If there's a tie, and one of the choices is for $\mathbf{X}_j$ to stay in the same cluster as the previous iteration, always take that choice." What could conceivably go wrong if you don't follow that advice? [6 points]

Solution: If a sample point keeps switching back and forth between two (or more) cluster means that are equally close, the k-means algorithm might never terminate even though it has can make no further progress in reducing the cost function.

(d) In which of the following cases should you prefer k-nearest neighbors over k-means clustering? For all the four options, you have access to images $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_n \in \mathbb{R}^d$ [6 points]

A: You do not have access to labels. You want to find out if any of the images are very different from the rest, i.e., are outliers.

B: You have access to labels $y_1, y_2, \cdots, y_n$ telling us whether image i is a cat or a dog. You want to find out whether the distribution of cats is unimodal or bimodal. You already know that the distribution of cats either has either one or two modes, but that's all you know about the distribution.

C: You have access to labels $y_1, y_2, \cdots, y_n$ telling us whether image i is a cat or a dog. You want to find out whether a new image z is a cat or a dog.

D: You have access to labels $y_1, y_2, \cdots, y_n$ telling us whether image i is a cat or a dog. Given a new image z, you want to approximate the posterior probability of z being a cat and the posterior probability of z being a dog.

Solution: A: You don't have access to labels, so you cannot use k-nearest neighbors.

B: In order to do this, you would first filter out all the dogs, so you're left with cat images only. Then you perform k-means clustering twice, once with $k = 1$ and again with $k = 2$. If the total distance between the data points and the cluster centers are markedly lower in the second case than the first, that would suggest that the data is bimodal.

C: You want to perform classification, so you must use k-NN. k-means is a clustering algorithm and not a classification algorithm.

D: In this case, you must use k-NN. Instead of finding the plurality class in the k points closest to z, you find the class histogram of those points.