Assignment 8

CS-UH-2218: Algorithmic Foundations of Data Science

Assignments are to be submitted in groups of two or three*. Upload the solutions on NYU classes as one PDF file for theoretical assignments and separate source code files for each programming assignment. Submit only one copy per group. Clearly mention the participant names on each file you submit.*

Problem 1 (10 points)**.**
Suppose that we have a dataset in d dimensions and we would like to do k-means clustering for some k that is much smaller than d. Instead of doing k-means on the data set directly, it is much cheaper to first reduce the dimension and then do k-means clustering since most algorithms for k-means clustering (e.g. Lloyd's algorithm) take time proportional to the dimension in every step. However, if we do dimension reducion then we may potentially get a solution whose cost is low for the projected dataset but very large with respect to the original dataset. We will show that this is not the case is we project to the best fit k-dimensional subspace.

We need some (unfortunately, quite a bit of) notation first. Let $\textsc{Opt}$ be the set of optimal k-means centers for the original dataset and let $C_{\textsc{Opt}}$ denote its cost. For any subspace X, we define the following:

- $\textsc{Opt}_X$ denotes the optimal k-centers for the dataset obtained by projecting the original data points orthogonally on X.
- C_X^{orig} denotes the cost of $\textsc{Opt}_X$ with respect to the original dataset.
- C_X^{proj} denotes the cost of $\textsc{Opt}_X$ with respect to the projected dataset.
- E_X denotes the sum of squares of the distances from the (original) data points to X.

Let S be the best fit k-dimensional subspace (obtained via SVD) for the (original) dataset and let T be the subspace through the k centers in $\textsc{Opt}$.

1. Prove that for any subspace X, $C_X^{proj} \leq C_{\textsc{Opt}}$ and $C_X^{orig} = C_X^{proj} + E_X$.
2. Argue that $C_{\textsc{Opt}} = C_T^{orig}$.
3. Show that $E_S \leq E_T$.
4. Conclude that $C_S^{orig} \leq 2 \cdot C_{\textsc{Opt}}$ i.e. $\textsc{Opt}_S$ is a 2-approximation to the optimal solution.

Remark. The proofs above can be done very mechanically but it is important to contemplate about it in order to see through the notation gain insight. Notation is often necessary to make proofs precise and concise. However, it also obfuscates the idea and the reader needs to unpack the idea themselves. Draw some pictures to really understand what is going on and then take a walk and think about it.

Problem 2 (10 points)**.**
In the greedy algorithm we presented in class for computing best fit subspaces, we assumed that the best fit $(k+1)$-dimensional subspace for a data set contains the best fit k-dimensional subspace for the data set. Prove that this assumption is valid.

Hint: Let $\mathbf{v}_1, \cdots, \mathbf{v}_k$ be an orthonormal basis for the best fit k-dimensional subspace V. Let W be the best fit $(k+1)$-dimensional subspace. Show that it is possible to choose an orthonormal basis $\mathbf{w}_1, \cdots, \mathbf{w}_{k+1}$ for W such that $\mathbf{w}_{k+1} \perp \mathbf{v}_1, \cdots, \mathbf{v}_k$. Show that the subspace V' spanned by $\mathbf{v}_1, \cdots, \mathbf{v}_k$ and $\mathbf{w}_{k+1}$ is at least as good as W.

Problem 3 (20 points)**.**
Load the MNIST handwritten digits dataset in sklearn using the following code:

```
from sklearn.datasets import fetch_openml
X, y = fetch_openml('mnist_784', version=1, return_X_y=True)
```

The database consists of 28×28 pixel grayscale images of 70000 handwritten digits. After running the above code, $X[i]$, $i \in \{0, \cdots, 69999\}$ contains the i^{th} image and $y[i] \in \{0, \cdots, 9\}$ contains the digit it represents. $X[i]$ is an array of size $784 = 28 \times 28$ with entries in $\{0, \cdots, 255\}$ (grayscale values). You can print the i^{th} image and its label using the following code:

```
%matplotlib inline
import matplotlib.pyplot as plt
i = 42
plt.imshow(X[i].reshape(28,28))
print(y[i])
```

We can think of this dataset as a set of $n = 70000$ points in the $d = 784$ dimensions.

1. Use the standard scaler (see `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html`) to normalize the data.

2. Do an SVD of the $n \times d$ matrix X using the Randomized SVD algorithm, a fast algorithm suitable for large data (see `sklearn.decomposition.TruncatedSVD`). It it typically used to compute a truncated SVD but we will force it compute almost the full SVD by setting required number of components to $d - 1 = 783$.

3. Plot a graph of the singular values (they are returned in decreasing order).
   What do you see?

4. How many dimensions can we project the dataset to so that explained variance is about 90%? Explain your calculation.

5. Project the dataset to the best fit two dimensional subspace (spanned by the first two right singular vectors). Each data point $X[i] \in \mathbb{R}^{784}$ is mapped to a point $P_i \in \mathbb{R}^2$. To visualize the dataset, plot the points in the plane with colors determined by $y[i]$ (so that we have different colors for distinct digits). This example might help for plotting: `https://matplotlib.org/gallery/shapes_and_collections/scatter.html#sphx-glr-gallery-shapes-and-collections-scatter-py`.

Problem 4 (10 points)**.**
Consider a set of data points in the plane where each point is labelled either 0 or 1. Suppose that the two classes are separable by a circle i.e. there exists a circle so that all points with one of the labels lie inside the circle and all the points with the other label lie outside the circle. Prove that if we map each point (x, y) in the dataset to the point $(x, y, x^2 + y^2)$ in three dimensions then the two classes become linearly separable in three dimensions.

Problem 5 (10 points)**.**
Load the Iris data set (see `http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html`). Use the `train_test_split` function (see `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html`) to split the data set into a training and test sets. Use logistic regression to learn a classifier and report on its performance on the test set.

Problem 6 (10 points)**.**
Let $K(\mathbf{x}, \mathbf{x}')$ be a kernel function defined on pairs of vectors in $\mathbb{R}^d$. Let $\mathbf{x}_1, \cdots, \mathbf{x}_n$ be n vectors in $\mathbb{R}^d$. Consider the $n \times n$ matrix A in which $A_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. Prove that the matrix A is symmetric and positive semidefinite.