# STAT37710/CMSC35400 Homework 2

University of Chicago, Spring 2018 due Thursday, April 19 at 11:59pm

1. Let $A$ be a symmetric $d \times d$ matrix.
   (a) Show that if $\mathbf{v}$ and $\mathbf{v}'$ are two eigenvectors of $A$ with corresponding eigenvalues $\lambda \neq \lambda'$, then $\mathbf{v}$ is orthogonal to $\mathbf{v}'$.
   (b) Show that if $\mathcal{S}$ is a set of eigenvectors of $A$ with the same eigenvalue $\lambda$ and $\mathcal{S}$ spans a subspace $V$ of $\mathbb{R}^d$ of dimension $k$, then one can find $k$ mutually orthogonal vectors $\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^k$ such that (a) $V = \operatorname{span}\{\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^k\}$, and (b) each $\mathbf{v}^{(i)}$ is an eigenvector of $A$ with eigenvalue $\lambda$.
   (c) Explain why the above two statements imply that $A$ has an eigenvector decomposition of the form

$$A = \sum_{i=1}^{d} \lambda_i \mathbf{v}_i \mathbf{v}_i^\top,$$

   where each $\mathbf{v}_i$ is a unit eigenvector of $A$ and $\lambda_i$ is the corresponding eigenvalue.
   (d) Assume that $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_d$. Show that

$$\underset{\mathbf{w}\in\mathbb{R}^d\setminus\{0\}}{\operatorname{argmin}} \frac{\mathbf{w}^\top A\,\mathbf{w}}{\|\mathbf{w}\|^2} = \mathbf{v}_1 \qquad \text{and} \qquad \underset{\mathbf{w}\in\mathbb{R}^d\setminus\{0\}}{\operatorname{argmax}} \frac{\mathbf{w}^\top A\,\mathbf{w}}{\|\mathbf{w}\|^2} = \mathbf{v}_d.$$

2. Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a dataset of $n$ vectors in $\mathbb{R}^d$ that have already been centered, i.e., $\sum_{i=1}^n \mathbf{x}_i = 0$. Let $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k$ be a set of $k$ mutually orthogonal unit vectors, and $V$ the subspace that they span.
   (a) Given any $\mathbf{x} \in \mathbb{R}^d$, let $\mathbf{x}_V$ be the closest point to $\mathbf{x}$ in $V$, i.e., $\mathbf{x}_V = \operatorname{argmin}_{\mathbf{y}\in V} \|\mathbf{x}-\mathbf{y}\|^2$. Show that $\mathbf{x}_V$ is given by the orthogonal projection of $\mathbf{x}$ to $V$, i.e.,

$$\mathbf{x}_V = \sum_{i=1}^{k} (\mathbf{x} \cdot \boldsymbol{p}_i)\, \boldsymbol{p}_i.$$

   (b) Let $\Phi(V)$ be the mean squared error of projecting $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ to $V$,

$$\Phi(V) = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_i - (\mathbf{x}_i)_V\|^2.$$

   Show that $\Phi(V)$ is minimized by setting $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k$ to be the $k$ leading eigenvectors of the sample covariance matrix $\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top$.

3. (a) Let $K$ be the Gram matrix of $n$ points in $\mathbb{R}^d$ (with $n \geq d$). Show that $\operatorname{rank}(K) \leq d$.
   (b) Let $K \in \mathbb{R}^{n\times n}$ be a (symmetric) positive semi-definite matrix of rank $r$, and let $d \geq r$. Find $n$ points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ such that their Gram matrix is $K$.

4. Define the $n$ dimensional centering matrix $P = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$, where $\mathbf{1}$ is the $n$ dimensional all ones vector $(1, 1, \ldots, 1)^\top$.
   (a) Show that $P$ is a projection operator, i.e. that $P^2 = P$.
   (b) Show that the kernel of $P$ is the line $U = \{\lambda\mathbf{1}\}$, i.e., $P\mathbf{v} = 0$ if and only if $\mathbf{v} \in U$ or $\mathbf{v} = 0$.
   (c) Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a set of $n$ points in $\mathbb{R}^d$, and let $\tilde{G}$ be their centered gram matrix, $\tilde{G}_{i,j} = (\mathbf{x}_i - \boldsymbol{\mu})^\top(\mathbf{x}_j - \boldsymbol{\mu})$, where $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$. Show that

$$\tilde{G} = -\frac{1}{2} P\,D\,P, \tag{1}$$

   where $D_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$.

5. Locally Linear Embedding (LLE) finds an embedding that maps $n$ high dimensional input vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$, to lower dimensional output vectors $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^p$. In the second phase of the algorithm, $(\mathbf{y}_1, \ldots, \mathbf{y}_i)$ are found by minimizing the cost function

$$\Psi(\mathbf{y}_1, \ldots, \mathbf{y}_n) = \sum_{i=1}^{n} \left\| \mathbf{y}_i - \sum_j w_{i,j}\, \mathbf{y}_j \right\|^2$$

given the weights $(w_{i,j})_{i,j}$ found in the first phase. To make the problem well posed, this optimization is performed subject to the constraints

$$\sum_{i=1}^{n} \mathbf{y}_i = 0, \qquad \text{and} \qquad \frac{1}{n} \sum_{i=1}^{n} \mathbf{y}_i \mathbf{y}_i^T = I,$$

where $I$ is the $p \times p$ identity matrix. Derive the eigenvector problem that LLE reduces to.

6. The file `3Ddata.txt` is a dataset of 500 points in $\mathbb{R}^3$ sampled from a manifold with some added noise. The last number in each line is just an index in $\{1, 2, 3, 4\}$ related to the position of the point on the manifold to make the visualization prettier (for example, you can plot those points with index 1 in green, index 2 in yellow, 3 in blue and 4 in red).

   Apply each of the three dimensionality reduction methods PCA, Isomap and LLE to map this data to $\mathbb{R}^2$. To construct the graph (mesh), in each case you can use $k = 10$ nearest neighbors. Plot the results and comment on the differences. For each dimensionality reduction method you need to write your own code (it shouldn't be more than a few lines each) and submit it together with the write-up.

7. The file `train35.digits` contains 2000 images of 3's and 5's from the famous MNIST database of handwritten digits in text format. The size of each image is $28 \times 28$ pixels. Each row of the file is a representation one image, with the $28 \times 28$ pixels flattened into a vector of size 784. A value of 1 for a pixel represents black, and value of 0 represents white. The corresponding row of `train35.labels` is the class label: $+1$ for the digit 3, or $-1$ for the digit 5. The file `test35.digits` contains 200 testing images in the same format as `train35.digits`.

   Implement the perceptron algorithm and use it to label each test image in `test35.digits`. Submit the predicted labels in a file named `test35.predictions`. In the lectures, the perceptron was presented as an online algorithm. To use the perceptron as a batch algorithm, train it by simply feeding it the training set $M$ times. The value of $M$ can be expected to be less than 10, and should be set by cross validation. Naturally, in this context, the "mistakes" made during training are not really errors. Nonetheless, it is intructive to see how the frequency of mistakes decreases as the hypothesis improves. Include in your write-up a plot of the cumulative number of "mistakes" as a function of the number of examples seen.

   Since the data is fairly large, for debugging purposes it might be helpful to run your code on just subsets of the 2000 training test images. Depending on your implementation, each run of each algorithm can take several minutes. It may be helpful to normalize each example to unit norm.