Recitation 11

Lecturer: Regev Schweiger *Scribe: Regev Schweiger*

11.1 Probabilistic View of PCA

Consider the following model. Let $\mathbf{z}_1, \ldots, \mathbf{z}_n$ be points in $\mathbb{R}^d$, and assume W is a full rank matrix over $\mathbb{R}^{m \times d}$. Assume that we do not observe the points $\mathbf{z}_1, \ldots, \mathbf{z}_n$, and we do not know what is W (but we know d). Instead, assume we observe the points $\mathbf{x}_i = W\mathbf{z}_i + \epsilon_i$, where $\epsilon_i \sim N(0, \sigma^2 I_m)$, i.e., it is a multivariate normal noise. Thus, we obtain points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in m dimensions, however these points are truly points in d dimensions with additional normal noise. We would like to treat the problem as a likelihood inference problem. Let us write down the log likelihood of the above formulation:

$$\ell(Z, W; X) = -mn \log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} \|\mathbf{x}_i - W\mathbf{z}_i\|^2.$$

Thus, maximizing the log likelihood is equivalent to solving the following:

$$(\hat{Z}, \hat{W}) = \arg\min_{Z,W} \sum_{i=1}^{n} \|\mathbf{x}_i - W\mathbf{z}_i\|^2.$$

For any fixed W we get a simple linear regression problem, since $\mathbf{x}_i$ is fixed. The solution to the linear regression is given by the Normal equations:

$$\hat{\mathbf{z}_i} = (W^t W)^{-1} W^t \mathbf{x}_i.$$

Note that since W is of full rank, we know that $W^t W$ is nonsingular. Note that this assumption is not limiting since if W is not full rank we can change the problem by assuming that the points $\mathbf{z}_i$ arise from a smaller dimension. Plugging the regression solution into the likelihood we get that we need to minimize the following:

$$\hat{W} = \arg\min_{W} \sum_{i=1}^{n} \|\mathbf{x}_i - W(W^t W)^{-1} W^t \mathbf{x}_i\|^2.$$

Let $W = USV^t$ be the singular value decomposition of W. It is easy to see that $W(W^t W)^{-1} W^t = US(S^t S)^{-1} S^t U^t = U_d U_d^t$, where U_d is the $m \times d$ matrix consisting of the first d columns of U. Thus, we get that maximizing the likelihood is equivalent to maximizing the following:

$$\hat{U_d} = \arg\min_{U_d} \sum_{i=1}^{n} \|x_i - U_d U_d^t x_i\|^2.$$

Therefore, we obtain the exact formulation of PCA. The probabilistic formulation allows for a few natural extensions to PCA. First, we can deal with PCA with missing data using the Expectation Maximization algorithm. Second, we can now easily model a mixture of PCAs, where the assumption is that every point was sampled from a mixture of lower dimension hyperplanes.

It is worth noting that we $\mathbf{z}_1, \ldots, \mathbf{z}_n$ as parameters, i.e., the assumption is that these are fixed points. It is possible to add the assumption about the distribution of these points as well. Specifically, the case in which $\mathbf{z}_i \sim N(0, \tau^2)$ has been studied in the literature. This variation of PCA is referred to as probabilistic PCA. The optimization is similar to the original PCA in that it is sufficient to find the eigenvalues and eigenvectors of XX^t in order to compute the maximum likelihood estimate for W, however the maximum likelihood estimate of W is not the PCA solution.

It is interesting to note that the probabilistic interpretation of PCA is analogous to the probabilistic interpretation of regression. Consider the case of linear regression of one variable, i.e., under the model $x_2 = ax_1 + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$. Instead of using linear regression, we can use PCA by considering a different model, i.e., $\binom{x_1}{x_2} = \binom{a_1}{a_2} z + \delta$, where a_1, a_2, z are unobserved parameters, and $\delta \sim N(0, \tau^2 I_2)$. As we showed above, the solution to this problem is the PCA, reducing the two dimensions (x_1, x_2) into one dimension. The difference between the two resulting optimizations is shown in Figures 11.1 and 11.2.

11.2 Spectral Clustering

Spectral clustering is a method for clustering data based on a similarity measure between every two samples. Spectral clustering works for general similarity measures. Given a set of data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and some notion of similarity $s_{i,j} \geq 0$ between all pairs of data points $\mathbf{x}_i$ and $\mathbf{x}_j$, the intuitive goal of clustering is to divide the data points into several groups such that points in the same group are similar and points in different groups are dissimilar to each other. If we do not have more information than similarities between data points, a nice way of representing the data is in form of the similarity graph $G = (V, E)$. Formally, we are given a graph with non-negative edge weights defined by a symmetric matrix $W \in R_+^{n \times n}$. The weights correspond to the similarity between the different vertices. We define the weighted degree of a vertex i as $d_i = \sum_{j=1}^{n} w_{ij}$. We also define D as the diagonal matrix consisting of the weighted vertex degrees in the diagonal:

$$D = \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix}$$

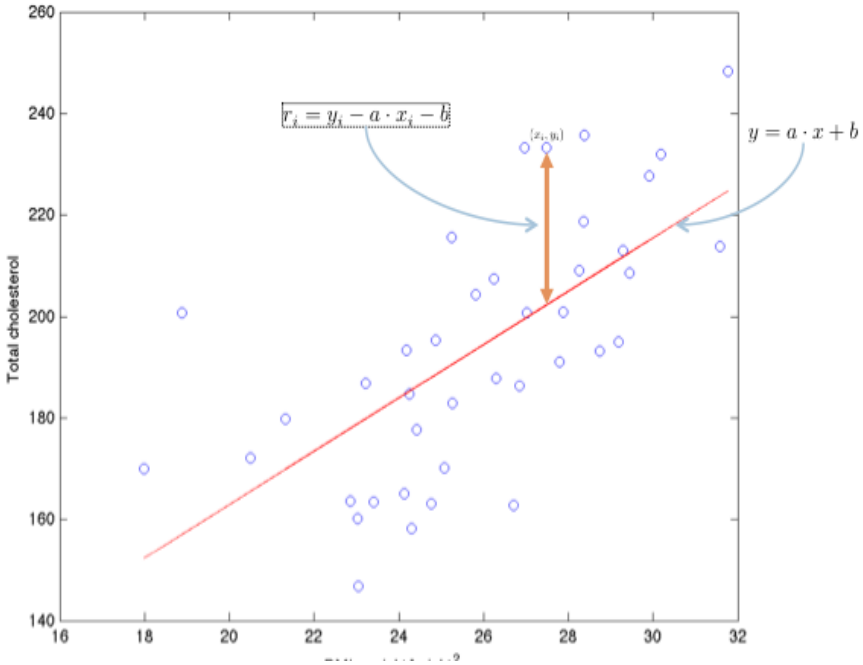

Figure 11.1: Linear regression: minimize the square of the residuals

We now define the Laplacian of the graph G as $L = D - W$. Consider the quadratic form of the Laplacian:

$$v^t(D - W)v = \sum_{i=1}^{n}\sum_{j=1}^{n} v_i v_j (D_{ij} - w_{ij}) = \sum_{i<j}(v_i - v_j)^2 w_{ij}$$

Minimizing the quadratic form results in the intuition of having v_i close to v_j when w_{ij} is large. Thus, we are interested in finding the smallest eigenvalue. The smallest eigenvalue, however, is not interesting, since the corresponding eigenvector is simply $(1, 1, \ldots, 1)$. Thus, instead, we are searching for the second smallest eigenvalue. Put differently, we are searching for a unit vector v, such that $\sum_{i=1}^{n} v_i = 0$, which minimizes $v^t L v$ (since the eigenvector with the second smallest eigenvalue is the vector with the smallest eigenvalue from the set of vectors orthogonal to the eigenvector with the smallest eigenvalue).

The spectral clustering algorithm is presented in Algorithm 1.

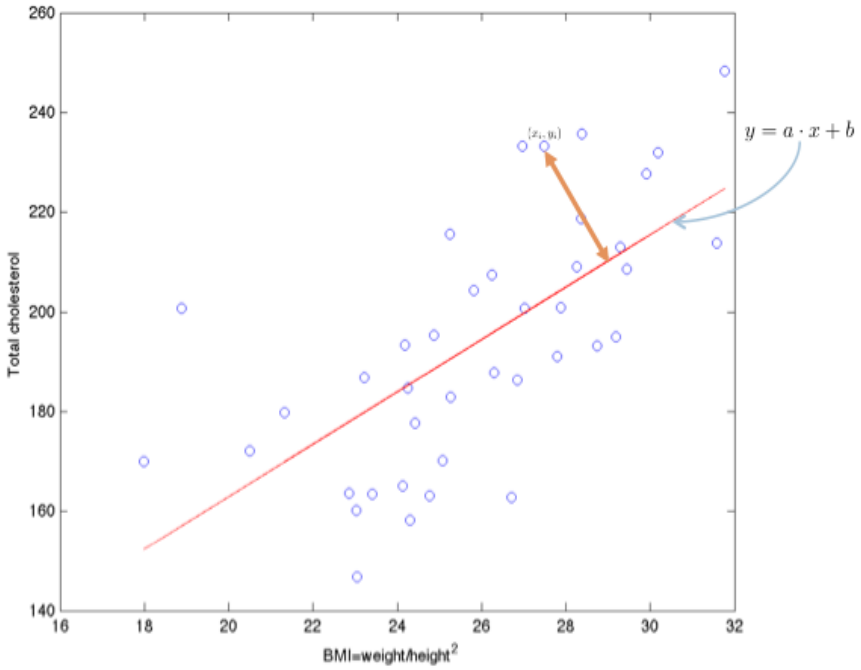

Figure 11.2: PCA: minimize the square of the distances from the line

Algorithm 1 Spectral clustering

Input: Similarity matrix S, of size $n \times n$, number k of clusters to construct.
Construct a similarity graph Let W be its weighted adjacency matrix.
Compute the Laplacian L.
Compute the first $k+1$ eigenvectors $u_1, \ldots, u_{k+1}$ of L, corresponding to the $k+1$ smallest eigenvalues.
Let U be the $n \times k$ matrix containing the vectors $u_2, \ldots, u_k$ as columns.
Cluster the rows of U for k-means into clusters.

As opposed to PCA, spectral clustering works for any similarity measure, as long as the graph weights are positive. Moreover, spectral clustering has the following natural interpretation. Consider the following score for a cut in the graph:

$$score(A, B) = \frac{\sum_{i \in A, j \in B} w_{ij}}{\sum_{i \in A, j \in V} w_{ij}} + \frac{\sum_{i \in A, j \in B} w_{ij}}{\sum_{i \in B, j \in V} w_{ij}}.$$

The minimum normalized cut problem searches for a partition of the graph into two clusters A, B, so that $score(A, B)$ is minimized. This criterion is reasonable since we are often interested in a balanced minimum cut, i.e., a small cut that has many vertices in both A

and B. The problem is NP-hard, and under the assumption that the graph is regular (D is the identity matrix times a constant), it is possible to show that spectral clustering is a relaxation of the normalized cut problem, that is, if we limit the values of v_i to take only two possible values we will be able to solve the normalized cut problem exactly (the proof is not shown, but you can see Shi and Malik, "Normalized Cuts and Image Segmentation", 2000, for more information).

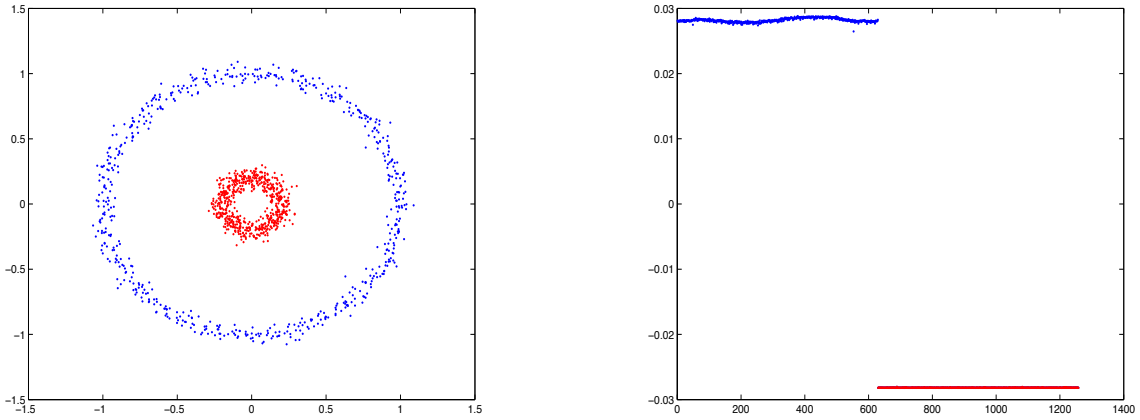

Figure 11.3: Spectral clustering: on the left are the points in the original space, and on the right are the points in the space defined by the second smallest eigenvector of the Laplacian (the y-axis is the eigenvector value of the point, and the x-axis is simply the point ordinal number in the original set).

Here is an illustration of the power of spectral clustering. Consider the set of points given in the left subfigure of Figure 11.3. We can define the similarity between every two points as $w_{ij} = e^{-10\|x_i - \mathbf{x}_j\|}$; note that we have large values for w_{ij} when x_i is close to x_j. The right subfigure of Figure 11.3 shows the values of the second smallest eigenvector of the Laplacian. Clearly, it'd be easy to cluster the blue and red points using this eigenvector.