ACM/CMS 107 Linear Analysis & Applications **Fall 2016**

Assignment 1: Data Analysis **Due:** 13th October 2016

For this assignment the following MATLAB code will be required:

http://www.mdunlop.org/cms107/assignment1.zip

An Introduction to Clustering

A common problem in unsupervised learning is that of clustering: grouping a collection of data or objects into clusters, such that objects within a cluster are more similar to one another than they are to objects outside of the cluster. Typical applications include image segmentation, anomaly detection and social network analysis. Figure 1 illustrates the idea of clustering in the case of planar data: 800 points are given distributed in the unit square, and a clustering algorithm attempts to classify these points into four clusters based on the distance between them. This is a simple example for illustration – in general the data may be very high dimensional, and there may be nonlinear structures underlying the data that need to inferred.

One approach to clustering is to find a partition of the data into k clusters such that the within-cluster variance is minimal, i.e. to find a minimizer of the functional

$$I(S_1, \ldots, S_k) = \sum_{j=1}^{k} \sum_{x \in S_j} \|x - \mu_j\|_2^2$$

which is defined on partitions of the data. Here μ_j is the mean of the points in S_j. This method is known as *k-means clustering*, and whilst computationally expensive to solve exactly, there are a number of heuristic algorithms that make the computation feasible [1].

We focus on *spectral clustering*, which involves looking at the eigenvectors and eigenvalues of a graph Laplacian associated with the data. It is outlined in the next section, and described in more detail in the tutorial [2]. In this assignment we first consider clustering for synthetic data, so that there are 'true' clusters that are to be determined, generated from a known statistical model. We then look at an example with real data: classifying political party affiliation from voting records.

Spectral Clustering

We are given data $x_1, \ldots, x_n \in \mathbb{R}^d$. We can think of these data points as the vertices of an undirected graph, which we will call the *similarity graph* and denote[1] G. Assume that the edge between vertices x_i and x_j has weight $w_{ij} \geq 0$, where w_{ij} represents the similarity of the points x_i and x_j. A typical way to define these weights is by

$$w_{ij} = \exp\left(-\frac{d(x_i, x_j)}{2\ell^2}\right) \tag{1}$$

[1] We will denote by G both the entire graph and just its vertices, as this should not lead to any ambiguity in the contexts considered here.

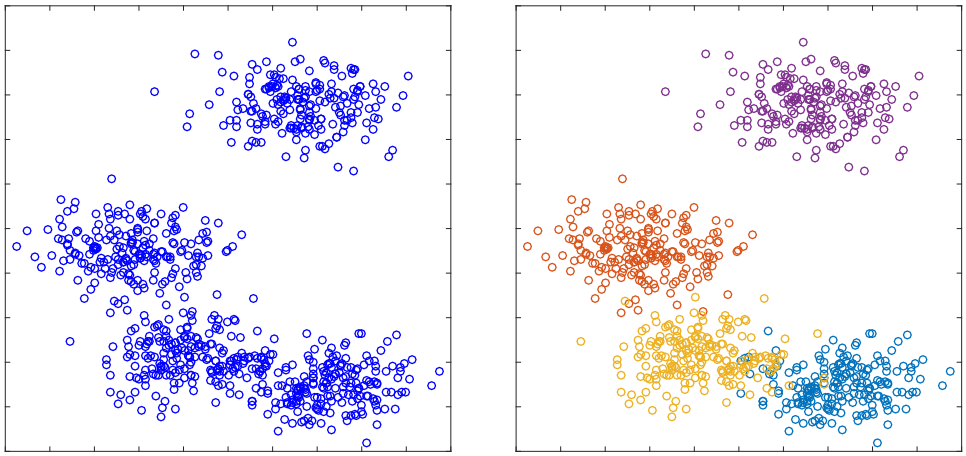

Figure 1: An example of clustered data in the plane (left) and an attempt to partition these data into four clusters (right).

for some metric d on $\mathbb{R}^d$ and scalar $\ell > 0$. The parameter ℓ is a length-scale parameter related to the size and separation distance between clusters; note that the smaller the value of ℓ, the greater the sensitivity of w_{ij} to the distance between the points x_i and x_j. In particular, note that in the limit $\ell \to 0$ the weights become one or zero depending on whether the two points being considered are coincident or not.

Define also the degree δ_i of the vertex x_i by

$$\delta_i = \sum_{k=1}^{n} w_{ij}.$$

We form two matrices $W, D \in \mathbb{R}^{n\times n}$ using the weights and the degrees:

$$W_{ij} = w_{ij}, \quad D_{ij} = \begin{cases} \delta_i & i = j \\ 0 & i \neq j \end{cases}.$$

Using these matrices we can form three further matrices, called graph Laplacians:

Definition 1 (Graph Laplacians)**.** *Let $D, W \in \mathbb{R}^{n\times n}$ be the degree and weight matrices above. Define the following three graph Laplacian matrices:*

- $L := D - W$
- $L_{\text{sym}} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$
- $L_{\text{rw}} := D^{-1} L = I - D^{-1} W$

L is called the unnormalized graph Laplacian*, L_{sym} is called the* symmetric normalized graph Laplacian *and L_{rw} is called the* random walk normalized graph Laplacian*.*

The eigenvalues and eigenvectors of these Laplacians turn out to be a very useful tool for the identification of clusters, as will be seen in the problems that follow.

Problem 1. Theory (25 points)

We focus here on the unnormalized Laplacian L; similar results hold for L_{sym} and L_{rw}. In what follows we assume nothing about the form of the weights w_{ij} except that they are non-negative. Additionally, given a subset of vertices $A \subseteq G$ we define $\mathbb{1}_A \in \mathbb{R}^n$, the indicator vector of A, by

$$(\mathbb{1}_A)_j = \begin{cases} 1 & x_j \in A \\ 0 & x_j \notin A \end{cases}.$$

(a) Show that

(i) for any $f \in \mathbb{R}^n$,

$$\langle f, Lf \rangle = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2;$$

(ii) L is symmetric and positive semi-definite; and

(iii) the smallest eigenvalue of L is 0, with corresponding constant eigenvector $\mathbb{1}_G$.

Deduce that L has n non-negative real eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.

(b) Given two vertices $x_i, x_j \in G$, we will say that there exists a path between x_i and x_j if there is a set of vertices $x_{m_1}, \ldots, x_{m_p} \in G$ such that the weights

$$w_{im_1}, w_{m_1m_2}, w_{m_2m_3}, \ldots, w_{m_{p-1}m_p}, w_{m_pj}$$

are all strictly positive. We will say that $A \subseteq G$ is a connected component of G if both $x_i, x_j \in A$ implies that there exists a path between x_i and x_j, and if only one of $x_i, x_j \in A$ implies that there does not exist a path between x_i and x_j. We will say that G has k connected components if there is a disjoint collection of connected components $A_1, \ldots, A_k \subseteq G$ whose union is G.

(i) Consider Figure 2, representing a graph with 5 vertices. Which of the following are connected components?

- $\{x_1, x_4, x_5\}$
- $\{x_1, x_2\}$
- $\{x_1, x_2, x_3\}$

How many connected components does the graph have?

(ii) Let $A \in \mathbb{C}^{(n_1+n_2)\times(n_1+n_2)}$ have block diagonal form

$$A = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$$

for $A_1 \in \mathbb{C}^{n_1 \times n_1}$ and $A_2 \in \mathbb{C}^{n_2 \times n_2}$. Assume that A_1 has eigenvalues $\lambda_1^1, \ldots \lambda_{n_1}^1$ with corresponding eigenvectors $v_1^1, \ldots, v_{n_1}^1 \in \mathbb{C}^{n_1}$, and A_2 has eigenvalues $\lambda_1^2, \ldots \lambda_{n_2}^2$ with corresponding eigenvectors $v_1^2, \ldots, v_{n_2}^2 \in \mathbb{C}^{n_2}$. What are the eigenvalues and eigenvectors of A? How does this generalize to a larger number of blocks?

(iii) Show that the 0 eigenvalue of L has geometric multiplicity 1 if and only if the graph G has one connected component.

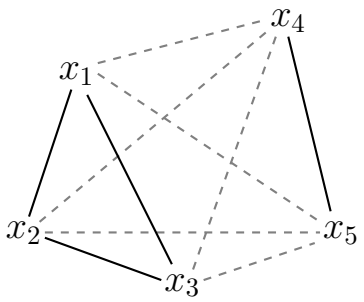

Figure 2: An example of a graph with weighted edges. A dashed line between x_i and x_j corresponds to $w_{ij} = 0$; a solid line corresponds to $w_{ij} > 0$.

(iv) Show that the 0 eigenvalue of L has geometric multiplicity k if and only if the graph G has k connected components $\{A_m\}_{m=1}^k$. Show that in this case the eigenspace of the 0 eigenvalue is spanned by the indicator vectors $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$.

Part (b) of the exercise above tells us that when the similarity graph G has k connected components, these components can be perfectly identified given knowledge of the first k eigenvectors of the graph Laplacian L. With a good choice of weights w_{ij}, these connected components in G should correspond to the clusters in $\mathbb{R}^d$. Choice of such weights is not trivial however, as will be seen in the following problems. Nonetheless, even with imperfect weights this provides useful heuristics for determining the clusters.

Problem 2. Synthetic Data I (35 points)

(a) Implement the three graph Laplacians L, L_{sym} and L_{rw} in MATLAB. Specifically, given an array of data $X \in \mathbb{R}^{n\times d}$, with each row of X representing a data point $x_i \in \mathbb{R}^d$, define a weight matrix $W = \{w_{ij}\}_{i,j=1}^n$ using the form (1) for the weights. Allow for the choices of metric

$$d_{p,q}(x, y) = \|x - y\|_p^q, \quad p \in \{1, 2, \infty\},\ q \in \{1, 2\} \tag{2}$$

and any length-scale parameter $\ell > 0$. Then define the corresponding degree matrix D, and return the three graph Laplacians as given in Definition 1.

(b) Load `data2.mat`. The data sets `X1`, `X2`, `X3` arise from a number of points distributed into a number of clusters in the plane, embedded in $\mathbb{R}^{100}$ and then perturbed by noise in all directions.

(i) Using the `eig` function, compute the eigenvectors of the three Laplacians for a variety of choices of p, q and $\ell > 0$, using the data `X1`. Problem 1(b) told us that information about clusters should lie in the first few eigenvectors – by looking at these, attempt to infer the number of clusters and their elements. How sensitive is the clustering to the choice of these parameters, and the choice of Laplacian?

(ii) Repeat the above using `svd` instead of `eig`. Do you notice any difference in the results? What could be an explanation for this?

(iii) The three sets of data `X1`, `X2`, `X3` arise from the same clusters, perturbed by increasing levels of noise. How does the noise level affect the ability to determine the clusters?

(iv) Illustrate the output of some of the above clustering via a projection of the clustered data onto its first two components.

Problem 3. Real Data (35 points)

(a) Load `data3.mat`. The matrix `X` contains 435 rows, each corresponding to the voting record of an individual representative. Entries of ± 1 correspond to voting for or against a bill, and entries of 0 correspond to abstention. The aim of this problem is to separate the voters into two clusters, representing their political party affiliation.

In what follows we assume that the weights take the form (1) with choice of metric given by (2).

(i) Experiment with classification of the voters into two clusters, representing their political party affiliation. In particular, compare the effect of using the distance $d_{2,q}$ versus the distance $d_{\infty,q}$, for $q \in \{1, 2\}$.

(ii) Is the clustering more sensitive with respect to the choice of length scale ℓ or with respect to choices of p, q in $d_{p,q}$? How would you choose appropriate parameters?

(iii) As in 2(b), compare the effect of using `svd` versus `eig`.

Problem 4. Synthetic Data II (5 points)

(a) Load `data4.mat`. This is a toy example to illustrate the effect of choice of metric when the data are related via a nonlinear geometric structure. All data `X` lie on the curve $\mathcal{C} = \{r(t)\}_{t\in[0,10\pi]}$, where

$$r(t) = (\rho(t)\cos(t), \rho(t)\sin(t))$$

and $\rho : \mathbb{R} \to \mathbb{R}$ is given by

$$\rho(t) = 2 + \sin(t\sqrt{2}).$$

The natural distance between points is then the arc-length along the curve:

$$d_{\mathcal{C}}(x, y) = \left| \int_{r^{-1}(x)}^{r^{-1}(y)} \|r'(t)\|_2 \, \mathrm{d}t \right|.$$

(i) Verify that $d_{\mathcal{C}}$ defines a metric on the set $\mathcal{C}^*$ of points of $\mathcal{C}$ where $\mathcal{C}$ does not self-intersect:

$$\mathcal{C}^* = \{x \in \mathcal{C} \,|\, \#r^{-1}(\{x\}) = 1\}.$$

(ii) Implement the metric $d_{\mathcal{C}}$ in MATLAB. Verify that your implementation gives

$$\ell(\mathcal{C}) = d_{\mathcal{C}}\big((2,0), (2+\sin(10\pi\sqrt{2}), 0)\big) = \int_0^{10\pi} \|r'(t)\|_2 \, \mathrm{d}t \approx 70.6.$$

Hint: for the implementation of the preimage the `min` function may be of use.

(iii) Experiment with classifying the clusters using both $d_{\mathcal{C}}$ and the Euclidean distance $d_{2,1}$. Which is more effective? Does any choice of metric $d_{p,q}$ appear to be more effective than $d_{\mathcal{C}}$?

(iv) What would be an appropriate metric to use if the data points only lie close to the curve $\mathcal{C}$, and not necessarily on it?

Remark: In practice the set $\mathcal{C}$ will not be known, and so determination of an appropriate metric is a large part of the problem. In general the data may be concentrated around an unknown submanifold of a much higher dimensional space, but be corrupted by noise in all dimensions of the containing space.

References

[1] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[2] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.