Stat260/CS294: Spectral Graph Methods **Lecture 23 - 04/16/2015**

Lecture: Some Statistical Inference Issues (2 of 3)

Lecturer: Michael Mahoney *Scribe: Michael Mahoney*

Warning: these notes are still very rough. They provide more details on what we discussed in class, but there may still be some errors, incomplete/imprecise statements, etc. in them.

23 Convergence and consistency questions

Last time, we talked about whether the Laplacian constructed from point clouds converged to the Laplace-Beltrami operator on the manifold from which the data were drawn, under the assumption that the unseen hypothesized data points are drawn from a probability distribution that is supported on a low-dimensional Riemannian manifold. While potentially interesting, that result is a little unsatisfactory for a number of reasons, basically since one typically does not test the hypothesis that the underlying manifold even exists, and since the result doesn't imply anything statistical about cluster quality or prediction quality or some other inferential goal. For example, if one is going to use the Laplacian for spectral clustering, then probably a more interesting question is to ask whether the actual clusters that are identified make any sense, e.g., do they converge, are they consistent, etc. So, let's consider these questions. Today and next time, we will do this in two different ways.

- Today, we will address the question of the consistency of spectral clustering when there are data points drawn from some space $\mathcal{X}$ and we have similarity/dissimilarity information about the points. We will follow the paper "Consistency of spectral clustering," by von Luxburg, Belkin, and Bousquet.
- Next time, we will ask similar questions but for a slightly different data model, i.e., when the data are from very simple random graph models. As we will see, some of the issues will be similar to what we discuss today, but some of the issues will be different.

I'll start today with some general discussion on: algorithmic versus statistical approaches; similarity and dissimilarity functions; and embedding data in Hilbert versus Banach spaces. Although I covered this in class briefly, for completeness I'll go into more detail here.

23.1 Some general discussion on algorithmic versus statistical approaches

When discussing statistical issues, we need to say something about our model of the data generation mechanism, and we will discuss one such model here. This is quite different than the algorithmic perspective, and there are a few points that would be helpful to clarify.

To do so, let's take a step back and ask: how are the data or training points generated? Here are two possible answers.

- **Deterministic setting.** Here, someone just provides us with a fixed set of objects (consisting, e.g, of a set of vectors or a single graph) and we have to work with this particular set of data. This setting is more like the algorithmic approach we have been adopting when we prove worst-case bounds.
- **Probabilistic setting.** Here, we can consider the objects as a random sample generated from some unknown probability distribution P. For example, this P could be on (Euclidean or Hilbert or Banach or some other) space $\mathcal{X}$. Alternatively, this P could be over random graphs or stochastic blockmodels.

There are many differences between these two approaches. One is the question of what counts as "full knowledge." A related question has to do with the objective that is of interest.

- In the deterministic setting, the data at hand count as full knowledge, since they are all there is. Thus, when one runs computations, one wants to make statements about the data at hand, e.g., how close in quality is the output of an approximation algorithm to the output of a more expensive exact computation.
- In the probabilistic setting, complete or full knowledge is to know P exactly, and the finite sample contains only noisy information about P. Thus, when we run computations, we are only secondarily interested in the data at hand, since we are more interested in P, or relatedly in what we can say if we draw another noisy sample from P tomorrow.

Sometimes, people think of the deterministic setting as the probabilistic setting, in which the data space equals the sample space and when one has sampled all the data. Sometimes this perspective is useful, and sometimes it is not.

In either setting, one simple problem of potential interest (that we have been discussing) is clustering: given a training data $(x_i)_{i=1,\dots,n}$, where x_i correspond to some features/patterns but for which there are no labels available, the goal is to find some sort of meaningful clusters. Another problem of potential interest is classification: given training points $(x_i, y_i)_{i=1,\dots,n}$, where x_i correspond to some features/patterns and y_i correspond to labels, the goal is to infer a rule to assign a correct y to a new x. It is often said that, in some sense, in the supervised case, *what* we want to achieve is well-understood, and we just need to specify *how* to achieve it; while in the latter case both *what* we want to achieve as well as *how* we want to achieve it is not well-specified. This is a popular view from statistics and ML; and, while it has some truth to it, it hides several things.

- In both cases, one specifies—implicitly or explicitly—an objective and tries to optimize it. In particular, while the vague idea that we want to predict labels is reasonable, one obtains very different objectives, and thus very different algorithmic and statistical properties, depending on how sensitive one is to, e.g., false positives versus false negatives. Deciding on the precise form of this can be as much of an art as deciding on an unsupervised clustering objective.
- The objective to be optimized could depend on just the data at hand, or it could depend on some unseen hypothesized data (i.e., drawn from P). In the supervised case, that might be obvious; but even in the unsupervised case, one typically is not interested in the output per se, but instead in using it for some downstream task (that is often not specified).

All that being said, it is clearly easier to validate the supervised case. But we have also seen that the computations in the supervised case often boil down to computations that are identical to computations that arise in the unsupervised case. For example, in both cases locally-biased spectral ranking methods arise, but they arise for somewhat different reasons, and thus they are used in somewhat different ways.

From the probabilistic perspective, due to randomness in the generation of the training set, it is common to study ML algorithms from this statistical or probabilistic point of view and to model the data as coming from a probability space. For example, in the supervised case, the unseen data are often modeled by a probability space of the form

$$((\mathcal{X} \times \mathcal{Y}), \sigma(\mathcal{B}_{\mathcal{X}} \times \mathcal{B}_{\mathcal{Y}}), P)$$

where $\mathcal{X}$ is the feature/pattern space and $\mathcal{Y}$ is the label space, $\mathcal{B}_{\mathcal{X}}$ and $\mathcal{B}_{\mathcal{Y}}$ are σ-algebras on $\mathcal{X}$ and $\mathcal{Y}$, and P is a joint probability distribution on patterns and labels. (Don't worry about the σ-algebra and measure theoretic issues if you aren't familiar with them, but note that P is the main object of interest, and this is what we were talking about last time with labeled versus unlabeled data.) The typical assumption in this case is that P is unknown, but that one can sample $\mathcal{X} \times \mathcal{Y}$ from P. On the other hand, in the unsupervised case, there is no $\mathcal{Y}$, and so in that case the unseen data are more often modeled by a probability space of the form

$$(\mathcal{X}, \mathcal{B}_{\mathcal{X}}, P),$$

in which case the data training points $(x_i)_{i=1,\ldots,n}$ are drawn from P.

From the probabilistic perspective, one is less interested in the objective function quality on the data at hand, and instead one is often interested in finite-sample performance issues and/or asymptotic convergence issues. For example, here are some questions of interest.

- Does the classification constructed by a given algorithm on a finite sample converge to a limit classifier at $n \to \infty$?
- If it converges, is the limit classifier the best possible; and if not, how suboptimal is it?
- How fast does convergence take place, as a function of increasing n?
- Can we estimate the difference between finite sample classifier and the optimal classifier, given only the sample?

Today, we will look at the convergence of spectral clustering from this probabilistic perspective. But first, let's go into a little more detail about similarities and dissimilarities.

23.2 Some general discussion on similarities and dissimilarities

When applying all sorts of algorithms, and spectral algorithms in particular, MLers work with some notion either of similarity or dissimilarity. For example, spectral clustering uses an adjacency matrix, which is a sort of similarity function. Informally, a dissimilarity function is a notion that is somewhat like a distance measure; and a similarity/affinity function measures similarities and is sometimes thought about as a kernel matrix. Some of those intuitions map to what we have been discussing, e.g., metrics and metric spaces, but in some cases there are differences.

Let's start first with dissimilarity/distance functions. In ML, people are often a little less precise than say in TCS; and—as used in ML—dissimilarity functions satisfy some or most or all of the following, but typically at least the first two.

- (D1) $d(x,x) = 0$
- (D2) $d(x,y) \geq 0$
- (D3) $d(x,y) = d(y,x)$
- (D4) $d(x,y) = 0 \Rightarrow x = y$
- (D5) $d(x,y) + d(y,z) \geq d(x,z)$

Here are some things to note about dissimilarity and metric functions.

- Being more precise, a *metric* satisfies all of these conditions; and a *semi-metric* satisfies all of these except for (D4).
- MLers are often interested in dissimilarity measures that do *not* satisfy (D3), e.g., the Kullback-Leibler "distance."
- There is also interest in cases where (D4) is not satisfied. In particular, the so-called cut metric—which we used for flow-based graph partitioning—was a semi-metric.
- Condition (D4) says that if different points have distance equal to zero, then this implies that they are really the same point. Clearly, if this is not satisfied, then one should expect an algorithm should have difficulty discriminating points (in clustering, classification, etc. problems) which have distance zero.

Here are some commonly used methods to transform non-metric dissimilarity functions into proper metric functions.

- If d is a distance function and $x_0 \in \mathcal{X}$ is arbitrary, then $\tilde{d}(x,y) = |d(x,x_0) - d(y,x_0)|$ is a semi-metric on $\mathcal{X}$.
- If $(\mathcal{X}, d)$ is a finite dissimilarity space with d symmetric and definite, then
$$\tilde{d} = \begin{cases} d(x,y) + c & \text{if } x \neq y \\ 0 & \text{if } x = y \end{cases},$$
with $c \geq \max_{p,q,r \in \mathcal{X}} |d(p,q) + d(p,r) + d(r,q)|$, is a metric.
- If D is a dissimilarity matrix, then there exists constants h and k such that the matrix with elements $\tilde{d}_{ij} = \left(d_{ij}^2 + h\right)^{1/2}$, for $i \neq j$, and also $\bar{d}_{ij} = d_{ij} + k$, for $i \neq j$, are Euclidean.
- If d is a metric, so are $d + c$, $d^{1/r}$, $\frac{d}{d+c}$, for $c \geq 0$ and $r \geq 1$. If $w : \mathbb{R} \rightarrow \mathbb{R}$ is monotonically increasing function s.t. $w(x) = 0 \iff x = 0$ and $w(x+y) \leq w(x) + w(y)$; then if $d(\cdot,\cdot)$ is a metric, then $w(d(\cdot,\cdot))$ is a metric.

Next, let's go to similarity functions. As used in ML, similarity functions satisfy some subset of the following.

- (S1) $s(x,x) > 0$

- (S2) $s(x,y) = s(y,x)$
- (S3) $s(x,y) \geq 0$
- (S4) $\sum_{ij=1}^{n} c_i c_j s(x_i, x_j) \geq 0$, for all $n \in \mathbb{N}, c_i \in \mathbb{R}, x_i \in \mathcal{X}$ PSD.

Here are things to note about these similarity functions.

- The non-negativity is actually *not* satisfied by two examples of similarity functions that are commonly used: correlation coefficients and scalar products
- One can transform a bounded similarity function to a nonnegative similarity function by adding an offset: $s(x,y) = s(x,y) + c$ for come c.
- If S is PSD, then it is a kernel. This is a rather strong requirement that is mainly satisfied by scalar products in Hilbert spaces.

It is common to transform *similarities to dissimilarities*. Here are two ways to do that.

- If the similarity is a scalar product in a Euclidean space (i.e., PD), then one can compute the metric
$$d(x,y)^2 = \langle x-y, x-y \rangle = \langle x,x \rangle - 2 \left(x,y \right\rangle + \langle y,y \rangle .$$
- If the similarity function is normalized, i.e., $0 \leq s(x,y) \leq 1$, and $s(x,x) = 1$, for all x, y, then $d = 1 - s$ is a distance.

It is also common to transform *dissimilarities to similarities*. Here are two ways to do that.

- If the distance is Euclidean, then one can compute a PD similarity
$$s(x,y) = \frac{1}{2} \left(d(x,0)^2 + d(y,0)^2 - d(x,y)^2 \right),$$
where $0 \in \mathcal{X}$ is an arbitrary origin.
- If d is a dissimilarity, then a nonnegative decreasing function of d is a similarity, e.g., $s(x,y) = \exp\left(-d(x,y)^2/t\right)$, for $t \in \mathbb{R}$, and also $s(x,y) = \frac{1}{1-d(x,y)}$.

These and related transformations are often used at the data preprocessing step, often in a somewhat ad hoc manner. Note, though, that the use of any one of them implies something about what one thinks the data "looks like" as well as about how algorithms will perform on the data.

23.3 Some general discussion on embedding data in Hilbert and Banach spaces

Here, we discuss embedding data (in the form of similarity or dissimilarity functions) into Hilbert and Banach spaces. To do so, we start with an informal definition (informal since the precise notion of dissimilarity is a little vague, as discussed above).

Definition 1. *A space $(\mathcal{X}, d)$ is a dissimilarity space or a metric space, depending on whether d is a dissimilarity function or a metric function.*

An important question for distance/metric functions, i.e., real metrics that satisfy the above conditions, is the following: when can a given metric space $(\mathcal{X}, d)$ be embedded *isometrically* in Euclidean space $\mathcal{H}$ (or, slightly more generally, Hilbert space $\mathcal{H}$). That is, the goal is to find a mapping $\phi : \mathcal{X} \rightarrow \mathcal{H}$ such that $d(x, y) = \|\phi(x) - \phi(y)\|$, for all $x, y \in \mathcal{X}$. (While this was something we relaxed before, e.g., when we looked at flow-based algorithms and looked at relaxations where there were distortions but they were not too too large, e.g., $O(\log n)$, asking for isometric embeddings is more common in functional analysis.) To answer this question, note that distance in Euclidean vector space satisfies (D1)–(D5), and so a necessary condition for the above is the (D1)–(D5) be satisfied. The well-known Schoenberg theorem characterizes which metric spaces can be isometrically embedded in Hilbert space.

Theorem 1. *A metric space $(\mathcal{X}, d)$ can be embedded isometrically into Hilbert space iff $-d^2$ is conditionally positive definite, i.e., iff*

$$-\sum_{ij=1}^{\ell} c_i c_j d^2(x_i, x_j) \geq 0$$

for all $\ell \in \mathbb{N}, x_i, x_j \in \mathcal{X}, c_i, c_j \in \mathbb{R}$, with $\sum_i c_i = 0$.

Informally, this says that Euclidean spaces and Hilbert spaces are not "big enough" for arbitrary metric spaces. (We saw this before when we showed that constant degree expanders do not embed well in Euclidean spaces.) More generally, though, isometric embeddings into certain Banach spaces can be achieved for arbitrary metric spaces. (More on this later.) For completeness, we have the following definition.

Definition 2. *Let X be a vector space over $\mathcal{C}$. Then X is a* normed linear space *if for all $f \in X$, there exists a number, $\|f\| \in \mathbb{R}$, called the norm of f s.t.: (1) $\|f\| \geq 0$; (2) $\|f\| = 0$ iff $f = 0$; (3) $\|cf\| = |c|\|f\|$, for all scalar c; (4) $\|f + g\| \leq \|f\| + \|g\|$. A* Banach space *is a complete normed linear space. A* Hilbert space *is a Banach space, whose norm is determined by an inner product.*

This is a large area, most of which is off topic for us. If you are not familiar with it, just note that RKHSs are particularly nice Hilbert spaces that are sufficiently heavily regularized that the nice properties of $\mathbb{R}^n$, for $n < \infty$, still hold; general infinite-dimensional Hilbert spaces are more general and less well-behaved; and general Banach spaces are even more general and less well-behaved. Since it is determined by an inner product, the norm for a Hilbert space is essentially an ℓ_2 norm; and so, if you are familiar with the ℓ_1 or ℓ_∞ norms and how they differ from the ℓ_2 norm, then that might help provide very rough intuition on how Banach spaces can be more general than Hilbert spaces.

23.4 Overview of consistency of normalized and unnormalized Laplacian spectral methods

Today, we will look at the convergence of spectral clustering from this probabilistic perspective. Following the von Luxburg, Belkin, and Bousquet paper, we will address the following two questions.

- Q1: Does spectral clustering converge to some limit clustering if more and more data points are sampled and as $n \rightarrow \infty$?

- Q2: If it does converge, then is the limit clustering a useful partition of the input space from which the data are drawn?

One reason for focusing on these questions is that it can be quite difficult to determine what is a cluster and what is a good cluster, and so as a more modest goal one can ask for "consistency," i.e., that the clustering constructed on a finite sample drawn from some distribution converges to a fixed limit clustering of the whole data space when $n \to \infty$. Clearly, this notion is particularly relevant in the probabilistic setting, since then we obtain a partitioning of the underlying space $\mathcal{X}$ from which the data are drawn.

Informally, this will provide an "explanation" for why spectral clustering works. Importantly, though, this consistency "explanation" will be very different than the "explanations" that have been offered in the deterministic or algorithmic setting, where the data at hand represent full knowledge. In particular, when just viewing the data at hand, we have provided the following informal explanation of why spectral clustering works.

- Spectral clustering works since it wants to find clusters s.t. the probability of random walks staying within a cluster is higher and the probability of going to the complement is smaller.
- Spectral clustering works since it approximates via Cheeger's Inequality the intractable expansion/conductance objective.

In both of those cases, we are providing an explanation in terms of the data at hand; i.e., while we might have an underlying space $\mathcal{X}$ in the back of our mind, they are statements about the data at hand, or actually the graph constructed from the data at hand.

The answer to the above two questions (Q1 and Q2) will be basically the following.

- Spectral clustering with the normalized Laplacian is consistent under very general conditions. For the normalized Laplacian, when it can be applied, then the corresponding clustering does converge to a limit.
- Spectral clustering with the non-normalized Laplacian is not consistent, except under very specific conditions. These conditions have to do with, e.g., variability in the degree distribution, and these conditions often do *not* hold in practice.
- In either case, if the method converges, then the limit does have intuitively appealing properties and splits the space $\mathcal{X}$ up into two pieces that are reasonable; but for the non-normalized Laplacian one will obtain a trivial limit if the strong conditions are not satisfied.

As with last class, we won't go through all the details, and instead the goal will be to show some of the issues that arise and tools that are used if one wants to establish statistical results in this area; and also to show you how things can "break down" in non-ideal situations.

To talk about convergence/consistency of spectral clustering, we need to make statements about eigenvectors, and for this we need to use the spectral theory of bounded linear operators, i.e., methods from functional analysis. In particular, the information we will need will be somewhat different than what we needed in the last class when we talked about the convergence of the Laplacian to the hypothesized Laplace-Beltrami operator, but there will be some similarities. Today, we are going to view the data points as coming from some Hilbert or Banach space, call in $\mathcal{X}$, and from these data points we will construct an empirical Laplacian. (Next time, we will consider graphs that are directly constructed via random graph processes and stochastic block models.) The main step today will be to establish the convergence of the eigenvalues and eigenvectors of random graph Laplacian matrices for growing sample sizes. This boils down to questions of convergence of random Laplacian matrices constructed from sample point sets.

(Note that although there has been a lot of work in random matrix theory on the convergence of random matrices with i.i.d. entries or random matrices with fixed sample size, e.g., covariance matrices, this work isn't directly relevant here, basically since the random Laplacian matrix grows with the sample size n and since the entries of the random Laplacian matrix are not independent. Thus, more direct proof methods need to be used here.)

Assume we have a data space $\mathcal{X} = \{x_1 \ldots, x_n\}$ and a pairwise similarity $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, which is usually symmetric and nonnegative. For any fixed data set of n points, define the following:

- the Laplacian $L_n = D_n - K_n$,
- the normalized Laplacian $L'_n = D_n^{-1/2} L_n D_n^{-1/2}$, and
- the random walk Laplacian $L''_n = D_n^{-1} L_n$.

(Although it is different than what we used before, the notation of the von Luxburg, Belkin, and Bousquet paper is what we will use here.) Note that here we assume that $d_i > 0$, for all i. We are interested in computing the leading eigenvector or several of the leading eigenvectors of one of these matrices and then clustering with them.

To see the kind of convergence result one could hope for, consider the second eigenvector $(v_1, \ldots, v_n)^T$ of L_n, and let's interpret is as a function f_n on the discrete space $\mathcal{X}_n = \{X_1, \ldots, X_n\}$ by defining the function $f_n(X_i) = v_i$. (This is the view we have been adopting all along.) Then, we can perform clustering by performing a sweep cut, or we can cluster based on whether the value of f_n is above or below a certain threshold. Then, in the limit $n \rightarrow \infty$, we would like $f_n \rightarrow f$, where f is a function on the entire space $\mathcal{X}$, such that we can threshold f to partition $\mathcal{X}$.

To do this, we can do the following.

1. Choose this space to be $C(\mathcal{X})$, the space of continuous functions of $\mathcal{X}$.
2. Construct a function $d \in C(\mathcal{X})$, a degree function, that is the "limit" as $n \rightarrow \infty$ of the discrete degree vector $(d_1, \ldots, d_n)$.
3. Construct linear operators U, U', and U'' on $C(\mathcal{X})$ that are the limits of the discrete operators L_n, L'_n, and L''_n.
4. Prove that certain eigenfunctions of the discrete operates "converge" to the eigenfunctions of the limit operators.
5. Use the eigenfunctions of the limit operator to construct a partition for the entire space $\mathcal{X}$.

We won't get into details about the convergence properties here, but below we will highlight a few interesting aspects of the limiting process. The main result they show is that in the case of normalized spectral clustering, the limit behaves well, and things converge to a sensible partition of the entire space; while in the case of unnormalized spectral clustering, the convergence properties are much worse (for reasons that are interesting that we will describe).

23.5 Details of consistency of normalized and unnormalized Laplacian spectral methods

Here is an overview of the two main results in more detail.

Result 1. (Convergence of normalized spectral clustering.) Under mild assumptions, *if the first r eigenvalues of the limit operator U' satisfy $\lambda_i \neq 1$ and have multiplicity one*, then

- the same hold for the first r eigenvalues of L'_n, as $n \to \infty$;
- the first r eigenvalues of L'_n converge to the first r eigenvalues of U';
- the corresponding eigenvectors converge; and
- the clusters found from the first r eigenvectors on finite samples converge to a limit clustering of the entire data space.

Result 2. (Convergence of unnormalized spectral clustering.) Under mild assumptions, *if the first r eigenvalues of the limit operator U do not lie in the range of the degree function d and have multiplicity one*, then

- the same hold for the first r eigenvalues of L_n, as $n \to \infty$;
- the first r eigenvalues of L_n converge to the first r eigenvalues of U;
- the corresponding eigenvectors converge; and
- the clusters found from the first r eigenvectors on finite samples converge to a limit clustering of the entire data space.

Although both of these results have a similar structure ("if the inputs are nice, then one obtains good clusters"), the "niceness" assumptions are very different: for normalized spectral clustering, it is the rather innocuous assumption that $\lambda_i \neq 1$, while for unnormalized spectral clustering it is the much stronger assumption that $\lambda_i \in \text{range}(d)$. This assumption is necessary, as it is needed to ensure that the eigenvalue λ_i is isolated in the spectrum of the limit operator. This is a requirement to be able to apply perturbation theory to the convergence of eigenvectors. In particular, here is another result.

Result 3. (The condition $\lambda \notin \text{range}(d)$ is necessary.)

- There exist similarity functions such that there exist no nonzero eigenvectors outside of range(d).
- In this case, the sequence of second eigenvalues of $\frac{1}{n}L_n$ converge to $\min d(x)$, and the corresponding eigenvectors do *not* yield a sensible clustering of the entire data space.
- For a wide class of similarity functions, there exist only finitely many eigenvalues r_0 outside of range(d), and the same problems arise if one clusters with $r > r_0$ eigenfunctions.
- The condition $\lambda \notin \text{range}(d)$ refers to the limit and cannot be verified on a finite sample.

That is, unnormalized spectral clustering can fail completely, and one cannot detect it with a finite sample.

The reason for the difference between the first results is the following.

- In the case of normalized spectral clustering, the limit operator U' has the form $U' = I - T$, where T is a compact linear operator. Thus, the spectrum of U' is well-behaved, and all the eigenvalues $\lambda \neq 1$ are isolated and have finite multiplicity.

- In the case of unnormalized spectral clustering, the limit operator U has the form $U = M - S$, where M is a multiplication operator, and S is a compact integral operator. Thus, the spectrum of U is not as nice as that of U', since it contains the interval range(d), and the eigenvalues will be isolated only if $\lambda_i \neq$ range(d).

Let's get into more detail about how these differences arise. To do so, let's make the following assumptions about the data.

- The data space $\mathcal{X}$ is a compact metric space, $\mathcal{B}$ is the Borel σ-algebra on $\mathcal{X}$, and P is a probability measure on $(\mathcal{X}, \mathcal{B})$. We draw a sample of points $(X_i)_{i \in \mathbb{N}}$ i.i.d. from P. The similarity function $k : X \times X \to \mathcal{R}$ is symmetric, continuous, and there exists an $\ell > 0$ such that $k(x, y) > \ell$, for all $x, y \in \mathcal{X}$. (The assumption that f is bounded away from 0 is needed due to the division in the normalized Laplacian.)

For $f : \mathcal{X} \to \mathbb{R}$, we can denote the range of f by range(f). Then, if $\mathcal{X}$ is connected and f is continuous then range$(f) = [\inf_x f(x), \sup_x f(x)]$. Then we can define the following.

Definition 3. *The* restriction operator $\rho_n : C(\mathcal{X}) \to \mathbb{R}^n$ *denotes the random operator which maps a function to its values on the first n data points, i.e.,*

$$\rho_n(f) = (f(X_1), \ldots, f(X_n))^T .$$

Here are some facts from spectral and perturbation theory of linear operators that are needed.

Let E be a real-valued Banach space, and let $T : E \to E$ be a bounded linear operator. Then, an *eigenvalue* of T is defined to be a real or complex number λ such that

$$Tf = \lambda f, \text{ for some } f \in E.$$

Note that λ is an eigenvalue of T iff the operator $T - \lambda$ has a nontrivial kernel (recall that if $L : V \to W$ then $\ker(L) = \{v \in V : L(v) = 0\}$) or equivalently if $T - \lambda$ is *not* injective (recall that $f : A \to B$ is injective iff $\forall a, b \in A$ we have that $f(a) = f(b) \Rightarrow a = b$, i.e., different elements of the domain do not get mapped to the same element). Then, the *resolvent* of T is defined to be

$$\rho(T) = \{\lambda \in \mathbb{R} : (\lambda - T)^{-1} \text{ exists and is bounded}\},$$

and the *spectrum* of T id defined to be

$$\sigma(T) = \mathbb{R} \setminus \rho(T).$$

This holds very generally, and it is the way the spectrum is generalized in functional analysis.

(Note that if E is finite dimensional, then every non-invertible operator is not injective; and so $\lambda \in \sigma(T) \Rightarrow \lambda$ is an eigenvalue of T. If E is infinite dimensional, this can fail; basically, one can have operators that are injective but that have no bounded inverse, in which case the spectrum can contain more than just eigenvalues.)

We can say that a point $\sigma_{iso} \subset \sigma(T)$ is *isolated* if there exists an open neighborhood $\xi \subset \mathbb{C}$ of σ_{iso} such that $\sigma(T) \cap \xi = (\sigma_{iso})$. If the spectrum $\sigma(T)$ of a bounded operator T in a Banach space E consists of isolated parts, then for each isolated part of the spectrum, a *spectral projection P_{iso}* can be *defined* operationally as a path integral over the complex plane of a path Γ that encloses

σ_{iso} and that separates it from the rest of $\sigma(T)$, i.e., for $\sigma_{iso} \in \sigma(T)$, the corresponding spectral projection is

$$P_{iso} = \frac{1}{2\pi i} \int_{\Gamma} (T - \lambda I)^{-1} \, d\lambda,$$

where Γ is a closed Jordan curve in the complex plane separating σ_{iso} from the rest of the spectrum. If λ is an isolated eigenvalue of $\sigma(T)$, then the dimension of the range of the spectral projection P_λ is defined to be the *algebraic multiplicity* of λ, (for a finite dimensional Banach space, this is the multiplicity of the root λ of the characteristic polynomial, as we saw before), and the *geometric multiplicity* is the dimension of the eigenspace of λ.

One can split up the spectrum into two parts: the *discrete spectrum* $\sigma_d(\mathrm{T})$ is the part of $\sigma(T)$ that consists of isolated eigenvalues of T with finite algebraic multiplicity; and the *essential spectrum* is $\sigma_{ess}(T) = \sigma(T) \setminus \sigma_d(T)$. It is a fact that the essential spectrum cannot be changed by a finite-dimensional or compact perturbation of an operator, i.e., for a bounded operator T and a compact operator V, it holds that $\sigma_{ess}(T+V) = \sigma_{ess}(T)$. The important point here is that one can define spectral projections only for isolated parts of the spectrum of an operator and that these isolated parts of the spectrum are the only parts to which perturbation theory can be applied.

Given this, one has perturbation results for compact operators. We aren't going to state these precisely, but the following is an informal statement.

- Let $(E, \|\cdot\|_E)$ be a Banach space, and $(T_n)_n$ and T bounded linear operators on E with $T_n \to T$. Let $\lambda \in \sigma(T)$ be an isolated eigenvalue with finite multiplicity m, and let $\xi \subset \mathbb{C}$ be an open neighborhood of λ such that $\sigma(T) \cap \xi = \{\lambda\}$. Then,
  - eigenvalues converge,
  - spectral projections converge, and
  - if λ is a simple eigenvalue, then the corresponding eigenvector converges.

We aren't going to go through the details of their convergence argument, but we will discuss the following issues.

The technical difficulty with proving convergence of normalized/unnormalized spectral clustering, e.g., the convergence of $(v_n)_{n\in\mathbb{N}}$ or of $(L'_n)_{n\in\mathbb{N}}$, is that for different sample sized n, the vectors v_n have different lengths and the matrices L'_n have different dimensions, and so they "live" in different spaces for different values of n. For this reason, one can't apply the usual notions of convergence. Instead, one must show that there exists functions $f \in C(\mathcal{X})$ such that $\|v_n - \rho_n f\| \to 0$, i.e., such that the eigenvector v_n and the restriction of f to the sample converge. Relatedly, one relates the Laplacians to some other operator such that they are all defined on the same space. In particular, one can define a sequence (U_n) of operators that are related to the matrices (L_n); but each operator (U_n) is defined on the space $C(\mathcal{X})$ of continuous functions on $\mathcal{X}$, independent of n.

All this involves constructing various functions and operators on $C(\mathcal{X})$. There are basically two *types* of operators, integral operators and multiplication operators, and they will enter in somewhat different ways (that will be responsible for the difference in the convergence properties between normalized and unnormalized spectral clustering). So, here are some basic facts about integral operators and multiplication operators.

Definition 4. *Let $(\mathcal{X}, \mathcal{B}, \mu)$ be a probability space, and let $k \in L_2(\mathcal{X} \times \mathcal{X}, \mathcal{B} \times \mathcal{B}, \mu \times \mu)$. Then,* *the function* $S : L_2(\mathcal{X}, \mathcal{B}, \mu) \to L_2(\mathcal{X}, \mathcal{B}, \mu)$ *defined as*

$$Sf(x) : \int_{\mathcal{X}} k(x,y) f(y) d\mu(y)$$

is an integral operator *with kernel k.*

If $\mathcal{X}$ is compact and k is continuous, then (among other things) the integral operator S is compact.

Definition 5. *Let* $(\mathcal{X}, \mathcal{B}, \mu)$ *be a probability space, and let* $d \in L_\infty(\mathcal{X}, \mathcal{B}, \mu)$*. Then a* multiplication operator $M_d : L_2(\mathcal{X}, \mathcal{B}, \mu) \to L_2(\mathcal{X}, \mathcal{B}, \mu)$ *is*

$$M_d f = fd.$$

This is a bounded linear operator; but if d is non-constant, then the operator M_d is *not* compact.

Given the above two different types of operators, let's introduce specific operators on $C(\mathcal{X})$ corresponding to matrices we are interested in. (In general, we will proceed by identifying vectors $(v_1, \ldots, v_n)^T \in \mathbb{R}^n$ with functions $f \in C(\mathcal{X})$ such that $f(v_i) = v_i$ and extending linear operators on $\mathbb{R}^n$ to deal with such functions rather than vectors.) Start with the unnormalized Laplacian: $L_n = D_n - K_n$, where $D = \mathrm{diag}(d_i)$, where $d_i = \sum_{ij} K(x_i, x_j)$.

We want to relate the degree vector $(d_1, \ldots, d_n)^T$ to a function on $C(\mathcal{X})$. To do so, define the true and empirical degree functions:

$$\begin{aligned} d(x) &= \int k(x,y) dP(y) \in C(\mathcal{X}) \\ d_n(x) &= \int k(x,y) dP_n(y) \in C(\mathcal{X}) \end{aligned}$$

(Note that $d_n \to d$ as $n \to \infty$ by a LLN.) By definition, $d_n(x_i) = \frac{1}{n} d_i$, and so the empirical degree function agrees with the degrees of the points X_i, up to the scaling $\frac{1}{n}$.

Next, we want to find an operator acting on $C(\mathcal{X})$ that behaves similarly to the matrix D_n on $\mathbb{R}^n$. Applying D_n to a vector $f = (f_1, \ldots, f_n)^T \in \mathbb{R}^n$ gives $(D_n f)_i = d_i f_i$, i.e., each element is multiplied by d_i. So, in particular, we can interpret $\frac{1}{n} D_n$ as a multiplication operator. Thus, we can define the true and empirical multiplication operators:

$$\begin{aligned} M_d &: C(\mathcal{X}) \to C(\mathcal{X}) \qquad & M_d f(x) = d(x) f(x) \\ M_{d_n} &: C(\mathcal{X}) \to C(\mathcal{X}) \qquad & M_{d_n} f(x) = d_n(x) f(x) \end{aligned}$$

Next, we will look at the matrix K_n. Applying it to a vector $f \in \mathbb{R}^n$ gives $(K_n f)_i = \sum_j K(x_i, x_j) f_j$. Thus, we can define the empirical and true integral operator:

$$\begin{aligned} S_n &: C(\mathcal{X}) \to C(\mathcal{X}) \qquad & S_n f(x) = \int k(x,y) f(y) dP_n(y) \\ S &: C(\mathcal{X}) \to C(\mathcal{X}) \qquad & S_n f(x) = \int k(x,y) f(y) dP(y) \end{aligned}$$

With these definitions, we can define the *empirical unnormalized graph Laplacian*, $U_n : C(\mathcal{X}) \to C(\mathcal{X})$, and the *true unnormalized graph Laplacian*, $U : C(\mathcal{X}) \to C(\mathcal{X})$ as

$$\begin{aligned} U_n f(x) &= M_{d_n} f(x) - S_n f(x) = \int k(x,y) \left(f(x) - f(y) \right) dP_n(y) \\ U f(x) &= M_d f(x) - S f(x) = \int k(x,y) \left(f(x) - f(y) \right) dP(y) \end{aligned}$$

For the normalized Laplacian, we can proceed as follows. Recall that v is an eigenvector of L'_n with eigenvalue v iff v is an eigenvector of $H'_n = D^{-1/2}K_nD^{-1/2}$ with eigenvalue $1-\lambda$. So, consider H'_n, defined as follows. The matrix H'_n operates on a vector $f = (f_1, \ldots, f_n)^T$ as $(H'_nf)_i = \sum_j \frac{K(x_i,x_j)}{\sqrt{d_id_j}}$. Thus, we can define the normalized empirical and true similarity functions

$$\begin{aligned} h_n(x,y) &= k(x,y)/\sqrt{d_n(x)d_n(y)} \\ h(x,y) &= k(x,y)/\sqrt{d(x)d(y)} \end{aligned}$$

and introduce two integral operators

$$\begin{aligned} T_n : C(\mathcal{X}) \to C(\mathcal{X}) \qquad & T_nf(x) = \int h_n(x,y)f(y)dP_n(y) \\ T : C(\mathcal{X}) \to C(\mathcal{X}) \qquad & Tf(x) = \int h(x,y)f(y)dP(y) \end{aligned}$$

Note that for these operators the scaling factors $\frac{1}{n}$ which are hidden in P_n and d_n cancel each other. Said another way, the matrix H'_n already has $\frac{1}{n}$ scaling factor—as opposed to the matrix K_n in the unnormalized case. So, contrary to the unnormalized case, we do not have to scale matrices H'_n and H_n with the $\frac{1}{n}$ factor.

All of the above is machinery that enables us to transfer the problem of convergence of Laplacian matrices to problems of convergence of sequences of operators on $C(\mathcal{X})$.

Given the above, they establish a lemma which, informally, says that under the general assumptions:

- the functions d_n and d are continuous, bounded from below by $\ell > 0$, and bounded from above by $\|k\|_\infty$,
- all the operators are bounded,
- all the integral operators are compact,
- all the operator norms can be controlled.

The hard work is to show that the empirical quantities converge to the true quantities; this is done with the perturbation result above (where, recall, the perturbation theory can be applied only to isolated parts of the spectrum). In particular:

- In the normalized case, this is true if $\lambda \neq 1$ is an eigenvalue of U' that is of interest. The reason is that $U' = I - T'$ is a compact operator.
- In the unnormalized case, this is true if $\lambda \notin \mathrm{range}(d)$ is an eigenvalue of U that is of interest. The reason is that $U = M_d - S$ is *not* a compact operator, unless M_d is a multiple of the identity.

So, the key difference is the condition under which eigenvalues of the limit operator are isolated in the spectrum: for the normalized case, this is true if $\lambda \neq 1$, while for the non normalized case, this is true if $\lambda \notin \mathrm{range}(d)$.

In addition to the "positive" results above, a "negative" result of the form given in the following lemma can be established.

Lemma 1 (Clustering fails if $\lambda \notin \mathrm{range}(d)$ is violated.)**.** *Assume that $\sigma(U) - \{0\} \cup range(d)$ with eigenvalue 0 having multiplicity 1, and that the probability distribution P on $\mathcal{X}$ has no point masses.*

Then the sequence of second eigenvectors of $\frac{1}{n}L_n$ *converges to* $\min_{x\in\mathcal{X}} d(x)$*. The corresponding eigenfunction will approximate the characteristic function of some* $x \in \mathcal{X}$*, with* $d(x) = \min_{x\in\mathcal{X}} d(x)$ *or a linear combination of such functions.*

That is, in this case, the corresponding eigenfunction does *not* contain any useful information for clustering (and one can't even check if $\lambda \in \mathrm{range}(d)$ with a finite sample of data points).

While the analysis here has been somewhat abstract, the important point here is that this is *not* a pathological situation: a very simple example of this failure is given in the paper; and this phenomenon will arise whenever there is substantial degree heterogeneity, which is very common in practice.