1 Spectral Theorem

Goal: Prove that symmetric matrices are unitarily(orthogonally?) diagonalizable with real spectrum: eigenvalues are real and eigenvectors can be chosen to be orthonormal.

Proof. Let's begin by supposing A is orthongonally diagonalizable. This implies that $A = PDP^T$, where D is a diagonal matrix and P is a unitary matrix ($P^TP = I = PP^T$). We can also write $D = P^TAP$ Then we get

$$PDP^T = PP^TAPP^T = IAI = A$$

and

$$A^T = (PDP^T)^T = (P^T)^TD^TP^T = PDP^T = A$$

Thus, $A^T = A$ and we have A is symmetric when A is orthongonally diagonalizable.

Now, let's examine what we can say about A when A is symmetric. Let's suppose that A has an eigenvalue λ with corresponding eigenvector $\vec{v}$. So, $\lambda\vec{v} = A\vec{v}$. Taking the complex conjugates and noting that A has real entries,

$$A\overline{\vec{v}} = \overline{A\vec{v}} = \overline{A\vec{v}} = \overline{\lambda\vec{v}} = \overline{\lambda\vec{v}}$$

Because A is symmetric,

$$\overline{\vec{v}^T}A = (\overline{\vec{v}}A)^T = (\overline{\lambda\vec{v}})^T = \overline{\lambda\vec{v}}^T$$

So, we have $A\vec{v} = \lambda\vec{v}$ and $\overline{\vec{v}^T}A = \overline{\lambda}\overline{\vec{v}^T}$, which gives us

$$\overline{\vec{v}^T}\lambda\vec{v} = \overline{\vec{v}^T}A\vec{v} = \overline{\lambda}\overline{\vec{v}^T}\vec{v} = \overline{\vec{v}^T}\overline{\lambda}\vec{v}$$

So, $\lambda\overline{\vec{v}^T}\vec{v} = \overline{\lambda}\overline{\vec{v}^T}\vec{v}$. We can rewrite this as $(\lambda - \overline{\lambda})(\overline{\vec{v}^T}\vec{v}) = 0$. But, since $\vec{v}$ is an eigenvector, we know that $\vec{v}$ and $\overline{\vec{v}^T}$ cannot be $\vec{0}$. So, we must have $\lambda - \overline{\lambda} = 0$ and $\lambda = \overline{\lambda}$. Thus, λ is real.

We can also show that any two eigenvectors corresponding to distinct eigenvalues of A are orthogonal when A is symmetric. Let's let $\lambda\vec{v} = A\vec{v}$ and $\mu\vec{u} = A\vec{u}$. Then, since $A = A^T$ and $\vec{x}\cdot\vec{y} = \vec{x}^T\vec{y}$, we have

$$\lambda\vec{v}\cdot\vec{u} = A\vec{v}\cdot\vec{u} = \vec{v}^TA\vec{u} = \mu\vec{v}^T\vec{u} = \mu\vec{v}\cdot\vec{u}$$

This gives us $(\lambda - \mu)(\vec{v}\cdot\vec{u}) = 0$, and since λ and μ are distinct, we have $\vec{v}\cdot\vec{u} = 0$ and we know that $\vec{v}$ and $\vec{u}$ must be orthogonal.

Now, to complete the proof, we want to show that if A is symmetric, then A is orthogonally diagonalizable. Much to my chagrin, we will do so using a proof by induction. Let's start off with our base case, a 1×1 matrix. I think it's pretty clear that this is orthogonally diagonalizable, because it's already diagonal. So now, let's suppose that $(n-1) \times (n-1)$

matrices are diagonalizable and attempt to show that $n \times n$ matrices are as well. Let λ_1 be an eigenvalue of the $n \times n$ matrix A and let $\vec{v}_1$ be it's corresponding eigenvector. Assume WLOG that $\vec{v}_1$ is a unit vector. Then, we can use $\vec{v}_1$ to construct an orthonormal basis for A consisting of vectors $\vec{v}_1, ..., \vec{v}_n$, where n is the dimension of A. Using these vectors, we can form the orthogonal matrix $P = [\vec{v}_1, ..., \vec{v}_n]$. We can write

$$B = P^T AP = P^T[A\vec{v_1} \; ... \; A\vec{v_n}] = P^T[\lambda_1\vec{v}_1 \; A\vec{v}_2 \; ... \; A\vec{v}_n]$$

Because $\vec{v_1}$ is a unit vector, we'll have $\vec{v_1}^T\lambda_1\vec{v}_1 = \lambda_1$ in B_{11}. Additionally, since all v_i's are orthogonal, the rest of the v_{1i} column will be zeros. The B matrix will look something like this:

λ_1	some numbers
0	A_1

where A_1 is an $(n-1) \times (n-1)$ matrix. Now let's consider B^T.

$$B^T = (P^T AP)^T = P^T A^T (P^T)^T = P^T AP = B$$

Since A is symmetric. This means that B must also be symmetric, which can only happen if B is of the form

λ_1	0
0	A_1

and A_1 is symmetric. Huzzah! We can now invoke our induction hypothesis on A_1! So, A must be diagonalizable (definitely skipping some algebra here, but oh whale).

■

2 A Tutorial on Spectral Clustering

2.1 The Basics

Because my graph theory class was a weakly connected graph theory class.

- **Similarity matrix:** Given a set of data points x_i, $i = 1, ..., n$, we can construct a similarity graph $G = (V, E)$ where each x_i corresponds to a vertex v_i. Two vertices v_i and v_j are connected by a edge with weight s_{ij} if s_{ij} is greater than a certain threshold. Here, s_{ij} is the similarity between x_i and x_j.

- **Weighted adjacency matrix:** If each edge between vertices v_i and v_j has $w_{ij} \geq 0$, the weighted adjacency matrix of our graph is $W = (w_{ij})_{i,j=1,...,n}$. If $w_{ij} = 0$, v_i and v_j are not connected.
- **Degree of a vertex:** The degree of a vertex $v_i \in V$ is $d_i = \sum_{j=1}^{n} w_{ij}$.
- **Degree matrix:** The diagonal matrix D where the degrees $d_1, ..., d_n$ are the diagonal entries.
- **Indicator vector:** Given $A \subset V$, the indicator $\mathbb{1}_A = (f_1, ..., f_n)'$, where $f_i = 1$ if $v_i \in A$ and 0 otherwise.
- **Weight Matrix:** For two sets $A, B \subset V$, we define the weight matrix $W(A, B)$ as $\sum_{i \in A, j \in B} w_{ij}$.
- **Size of subset $A \subset V$:**
  - $|A|$:= the number of vertices in A
  - $\text{vol}(A) := \sum_{i \in A} d_i$.
- **Different Similarity Graphs:**
  - **ε-neighborhood graph:** Connect all pts whose pairwise distances are smaller than ε.
  - **k-nearest neighbor graph:** connect vertices with their k-nearest neighbors. Two types:
    - *k-nearest neighbor*
    - *mutual k-nearest neighbor*
  - **Fully connected:** Connect all points with positive similarity, weight the edges by s_{ij}

2.2 Graph Laplacians

Important items to note:

- Assume G is an undirected, weighted graph weight matrix W and $w_{ij} = w_{ji} \geq 0$
- Eigenvectors of a matrix will not necessarily be normalized, but $\mathbb{1}$ and $3\mathbb{1}$ will be considered the same eigenvector.
- "First k eigenvectors" refers to the eigenvectors corresponding to the k-smallest eigenvalues in ascending order.

Types of Laplacians:

- **Unnormalized:** $L = D - W$
  - $\forall \vec{f} \in \mathbb{R}^n$, $\vec{f}' L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{i,j}(f_i - f_j)^2$
  - L symmetric, positive semi-definite
  - Smallest e-val is 0 corresponding to e-vec $\mathbb{1}$
  - has n non-negative, real-values eigenvalues (not necessarily distinct)
  - Multiplicity k of e-val 0 equals the number of connected components $A_1, ..., A_k$ in G, and the eigenspace of 0 is spanned by $\mathbb{1}_{A_1}, ..., \mathbb{1}_{Ak}$.
- **Normalized:** Share the listed properties unless specified
  - **Symmetric Laplacian:** $L_{sym} := D^{-1/2} L D^{-1/2}$
  - **Random Walk Laplacian:** $L_{rw} := D^{-1} L$
  - $\forall \vec{f} \in \mathbb{R}^n$, $\vec{f}' L_{sym} f = \frac{1}{2} \sum_{i,j=1}^{n} w_{i,j} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$
  - λ is an e-val of L_{rw} with e-vec $\vec{u}$ iff λ is an eigenvalue of Ł$_{sym}$ with e-vec $w = D^{1/2}\vec{u}$
  - λ is an e-val of L_{rw} with e-vec $\vec{u}$ iff λ and $\vec{u}$ solve the generalized eigenvector problem $L\vec{u} = \lambda D\vec{u}$
  - 0 is an e-val of L_{rw} with corresponding e-vec $\mathbb{1}$. 0 is an e-val of L_{sym} with corresponding e-vec $D^{1/2}\mathbb{1}$.
  - both are positive semi-definite with n non-negative real-valued e-vals.
  - Multiplicity k of e-val 0 equals the number of connected components $A_1, ..., A_k$ in G for both. For L_{rw}, the eigenspace of 0 is spanned by the vectors $\mathbb{1}_{A_i}$, and for L_{sym} the eigenspace of 0 is spanned by the vectors $D^{1/2}\mathbb{1}_{A_i}$.

2.3 Spectral Clustering Algorithms

Input similarity matrix $S \in \mathbb{R}^{n \times n}$ and number of clusters k

- **Unnormalized spectral clustering:**
  - Construct similarity graph, with W being the weighted adjacency matrix
  - Compute L and the first k e-vecs of L
  - Set $U \in \mathbb{R}^{n \times k}$ as the matrix with the u_i's as its columns
- **Normalized spectral clustering (Shi et al):**
  - Construct similarity graph, with W being the weighted adjacency matrix
  - Compute L

  - Compute the first k generalized e-vecs of $L\vec{u} = \lambda D\vec{u}$
  - Set $U \in \mathbb{R}^{n\times k}$ as the matrix with the u_i's as its columns

- **Normalized spectral clustering (Ng et al):**
  - Construct similarity graph, with W being the weighted adjacency matrix
  - Compute L_{sym}, first k e-vecs
  - Set $U \in \mathbb{R}^{n\times k}$ as the matrix with the u_i's as its columns and form T by normalizing rows of U

- For $i = 1, ..., n$, $\vec{y_i} \in \mathbb{R}^k$ is the i^{th} row of U (or T)
- cluster y_i's with k-means for each of these

2.4 Graph Cut POV

Goal of clustering is to group similar points (i.e. points with high weights between one another). Essentially, we want to find optimal ways to partition the graph, so we can view clustering as an approximation of graph partitioning problems.

- **Mincut problem:** want to choose a parition $A_1, ..., A_k$ that minimizes $\text{cut}(A_1, ..., A_k := \frac{1}{2}\sum_{i=1}^{k} W(A_i, \overline{A}_i)$
  - This ends up being kinda useless a lot of the time because it will often just isolate one vertex from the rest

To overcome the problems with mincut, we require our A_i's to be reasonably large using either RatioCut or Ncut:

- **RatioCut:** $\text{RatioCut}(A_1, ..., A_k) := \frac{1}{2}\sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A}_i)}{|A_i|}$
- **Ncut:** $\text{Ncut}(A_1, ..., A_k) := \frac{1}{2}\sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A}_i)}{\text{vol}A_i}$

These produce more balanced clusters, but they also make the mincut problem NP hard ☹, which is where spectral clustering can come in to help relax them. Relaxing RatioCut leads to unnormalized spectral clustering and relaxing Ncut leads to normalized spectral clustering.

- **Minimizing RatioCut:** Let H be the matrix of k indicator vectors, where the i^{th} entry of the vector is $1/\sqrt{|A_j|}$ if $v_i \in A_j$ and 0 otherwise. By way of algebra, we find that the problem of minimizing the ratio cut for an arbitrary k turns into $\min_{A_1,...,A_n} \text{Tr}(H'LH)$ subject to $H'H = I$ and H being defined with the indicator vectors as above. We relax by allowing for arbitrary values in entries of H.
  - this becomes a version of the Rayleigh-Ritz Theorem

  - H becomes U from unnormalized spec clustering, and to convert the relaxed version of the problem back to a discrete problem, we use k-means. This is just the unnormalized spectral clustering alg!

- **Minimizing Ncut:** Let H be the matrix of k indicator vectors, where the i^{th} entry of the vector is $1/\sqrt{|A_j|}$ if $v_i \in A_j$ and 0 otherwise. By way of algebra, we find that the problem of minimizing the Ncut for an arbitrary k turns into $\min_{A_1,\ldots,A_n} \text{Tr}(H'LH)$ subject to $H'DH = I$ and H being defined with the indicator vectors as above. To get the relaxed problem, we relax the discreteness condition and substitute $T = D^{1/2}H$, where $T'T = I$.

  - T contains the first k e-vectors of L_{sym}
  - The soln H consists of the first k e-vectors of L_{rw} (or $L\vec{u} = \lambda D\vec{u}$)
  - Shi's normalized clustering!

Unfortunately, we can't really ensure that these relaxed approaches provide a quality solution. But apparently, that isn't why we use spectral relaxation. Rather, we like the simple linear algebra problems that come from it.

2.5 Random Walks POV

We can explain spectral clustering by thinking of random walks on the similarity graph. We want to find a partition of the graph so that if we go on a long random walk, we mostly stay in that partition. We can define the transition matrix P as follows;

$$P = D^{-1}W$$

We also have $L_{rw} = I - P$, so λ is an e-val of L_{rw} iff $1 - \lambda$ is an e-val of P. So, the largest e-vals of P and smallest of L_{rw} can be used for clustering!

- **Ncut with transition probabilities:** Let G be connected and non-bipartite. We can express Ncut as follows:

  - $\text{Ncut}(A, \overline{A}) = P(\overline{A}|A) + P(A|\overline{A})$

I was going to put something in here about commute distance, but then Max sent us a paper where the author of this tutorial trashed it, so I'm going to skip over it for now...

2.6 Perturbation Theory POV

The goal of this POV (though I don't pretend to understand it at a very high level) seems to be to allow us some flexibility in how well-conditioned our eigenvalue problem is.

2.7 Future directions

I'm definitely still trying to get a good grasp of the basics, but a simple? thing I'm interested in is the impact of our choice of metric. I'm also be interested in a deeper investigation into the random walks/probabilistic POV.