July 11, 2014

- Less ambitious and choose Q_1^T that leaves 1st row unchanged.
- When Q_1 is multiplied on the right, it will leave the 1st col unchanged.

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \xrightarrow{Q_1^T \times} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} \xrightarrow{\times Q_1} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}$$

$A \qquad Q_1^T A \qquad Q_1^T A Q_1$

- Apply the same idea to other cols:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix} \text{upper Hessenberg}$$

$Q_1^TAQ_1 \qquad Q_2^TQ_1^TAQ_1Q_2 \qquad Q_3^TQ_2^TQ_1^TAQ_1Q_2Q_3$

- $Q = Q_1\, Q_2 \ldots Q_{n-2}$ and $Q^T A Q$ = upper Hessenberg
- Complexity:

  flops(Reduction to Hess) ~ $10/3\ n^3$

  flops(Reduction to tridiag) ~ $4/3\ n^3$

Tridiagonalization Algorithm

```
for  k = 1, 2, . . . , n-2
   x  =  A(k+1:n, k)
   v_k  =  sign(x_1)||x||e_1 + x
   v_k  =  v_k / ||v_k||
   for  j = k, k+1, . . . , n
      A(k+1:n, j)  =  A(k+1:n, j) - 2 v_k (v_k^T A(k+1:n, j))        } Q_k^T ×
   end
   for  i = 1, 2, . . . , n
      A(i, k+1:n)  =  A(i, k+1:n) - 2 (A(i, k+1:n) v_k) v_k^T        } × Q_k
   end
end
```

Symmetric case

- If $A = A^T$, then

  $$(Q^T A Q)^T = Q^T A Q \qquad \text{is also symmetric}$$

- A symmetric Hessenberg matrix → tridiagonal matrix.

Two-phase process (sym. case)

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\text{Phase 1}} \underset{T = Q^T A Q}{\begin{bmatrix} \times & \times & & \\ \times & \times & \times & \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix}} \xrightarrow[\text{QR iteration}]{\text{Phase 2}} \underset{D}{\begin{bmatrix} \times & & & \\ & \times & & \\ & & \times & \\ & & & \times \end{bmatrix}}$$

Application: Image Segmentation

Spectral Clustering

- Let $G = (V, E)$ be an undirected graph where $V = \{v_1, \ldots, v_n\}$ set of vertices and $E = \{e_{ij}\}$ set of edges with e_{ij} = edge between v_i and v_j.

- G is weighted if edge e_{ij} has a weight $w_{ij} \geq 0$.

- $W = (w_{ij})$ = weighted adjacency matrix of the graph.

- The degree of a vertex v_i:

$$d_i = \sum_{j=1}^{n} w_{ij}$$

- $D = \mathrm{diag}(d_i)$ = degree matrix.

- Given $A \subset V$, indicator vector $\mathbf{1}_A = (x_1, \ldots, x_n)$ is defined such that $x_i = 1$ if $v_i \in A$ and $x_i = 0$ otherwise.

- Given two subsets A, B, define

$$W(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

- Size of a subset $A \subset V$:

$$|A| = \text{number of vertices in A}$$

$$vol(A) = \sum_{i \in A} d_i$$

Graph Laplacians

- Unnormalized graph Laplacian matrix:

$$L = D - W$$

- Theorem: L satisfies the following properties:

  1. For any vector x,

  $$x^T L x = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (x_i - x_j)^2$$

  2. L is symmetric and positive semi-definite.

  3. The smallest eigenvalue of L is 0 and the corresponding eigenvector is the constant one vector **1**.

  4. L has n non-negative eigenvalues $0 = \lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$.

- Theorem: The multiplicity k of the eigenvalue 0 of L equals the number of connected components $A_1, \ldots, A_k$ in the graph.

- Note: Suppose the graph G is a 2D mesh and $w_{ij} = 1$. Then L becomes the usual 2D Laplacian matrix.

- Normalized graph Laplacian matrix:

$$\hat{L} = I - D^{-1/2} W D^{-1/2}$$

- Theorem: L^ satisfies the following properties:

1. For any vector x,

$$x^T \hat{L} x = \tfrac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}} \right)^2$$

2. The smallest eigenvalue of L^ is 0 and the corresponding eigenvector is $D^{1/2}$ **1**.

3. L^ is positive semi-definite and has n non-negative eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.

- Theorem: The multiplicity k of the eigenvalue 0 of L^ equals the number of connected components $A_1, \ldots, A_k$ in the graph.

Graph cut

- Given a graph G with adjacency matrix W, find a partition of G such that the edges between the partitions have a very low weight.

- Given a partition $A_1, \ldots, A_k$, define

$$cut(A_1, \ldots A_k) = \tfrac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A}_i) \qquad \overline{A}_i = \text{complement of A}$$

- Not good; solution often = separate one vertex from the other.

- Require $A_1, \ldots, A_k$ to be reasonably large. Define

$$RatioCut(A_1, \ldots A_k) = \tfrac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{|A_i|} = \sum_{i=1}^{k} \frac{cut(A_i, \overline{A}_i)}{|A_i|}$$

$$Ncut(A_1, \ldots A_k) = \tfrac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{vol(A_i)} = \sum_{i=1}^{k} \frac{cut(A_i, \overline{A}_i)}{vol(A_i)}$$

- NP hard problem

- Approximate RatioCut for k = 2:

$$\min_{A} RatioCut(A, \overline{A})$$

- Given a subset $A \subset V$, define $x = (x_1, \ldots, x_n)$

$$x_i = \begin{cases} \sqrt{|\overline{A}| / |A|} & \text{if } v_i \in A \\ -\sqrt{|A| / |\overline{A}|} & \text{if } v_i \in \overline{A} \end{cases}$$

- Then one can prove that

$$x^T L x = |V| \cdot RatioCut(A, \overline{A})$$

$$\sum_{i=1}^{n} x_i = 0 \qquad i.e. \qquad x^T 1 = 0$$

$$\|x\|^2 = n$$

- The min problem becomes:

$$\min_{A} x^T L x$$

subject to x_i as defined above, $x \perp 1$, and $||x|| = \text{sqrt}(n)$.

- Relaxed min problem:

$$\min_{x \in R^n} x^T L x$$

subject to $x \perp 1$, and $||x|| = \text{sqrt}(n)$.

- Solution of the relaxed min problem: the eigenvector corresponding to the second smallest eigenvalue of L.

- Approximate Ncut for k = 2:

$$\min_{A} Ncut(A, \overline{A})$$

- Given a subset $A \subset V$, define $x = (x_1, \ldots, x_n)$

$$x_i = \begin{cases} \sqrt{vol(\overline{A}) / vol(A)} & if \;\; v_i \in A \\ -\sqrt{vol(A) / vol(\overline{A})} & if \;\; v_i \in \overline{A} \end{cases}$$

- Then one can prove that

$$x^T L x = vol(V) \cdot Ncut(A, \overline{A})$$

$$\sum_{i=1}^{n} d_i x_i = 0 \qquad i.e. \qquad (Dx)^T 1 = 0$$

$$x^T D x = vol(V)$$

- The min problem becomes:

$$\min_{A} x^T L x$$

subject to x_i as defined above, $Dx \perp \mathbf{1}$, and $x^T Dx = vol(V)$.

- Relaxed min problem:

$$\min_{x \in R^n} x^T L x$$

subject to $Dx \perp \mathbf{1}$, and $x^T Dx = vol(V)$.

- Define y = $D^{1/2}$ x. The relaxed problem becomes:

$$\min_{x \in R^n} y^T D^{1/2} L D^{1/2} y \equiv \min_{x \in R^n} y^T \hat{L} y$$

- subject to y $\perp$ $D^{1/2}$ **1**, and $||y||^2$ = vol(V).

- Solution of the relaxed min problem: the eigenvector corresponding to the second smallest eigenvalue of L^.

K-means clustering

- Clustering for k = 2. From the solution vector x (or y), we need to find a partition. For example,

$$\begin{cases} v_i \in A & \text{if } x_i \geq 0 \ (\text{or } y_i \geq 0) \\ v_i \in \bar{A} & \text{if } x_i < 0 \ (\text{or } y_i < 0) \end{cases}$$

- It does not work for k > 2.

- K-mean clustering: Given a set of n data points $\{p_j\}$, find partitions $A_1, A_2, \ldots, A_k$ which solve the min problem

$$\min_{\{A_i\}} \sum_{i=1}^{k} \sum_{p \in A_i} \|p - \mu_i\|_2^2$$

1. Start with an initial guess for the k means $\{\mu_i\}$.
2. Assign p to A_i if p is closest to μ_i.
3. Update $\{\mu_i\}$ using the new partitions $\{A_i\}$.
4. Repeat (1)-(3).

- For the case k = 2. Consider $\{x_i\}$ as n points in R. Apply the k-means algorithm to cluster the points into 2 groups.