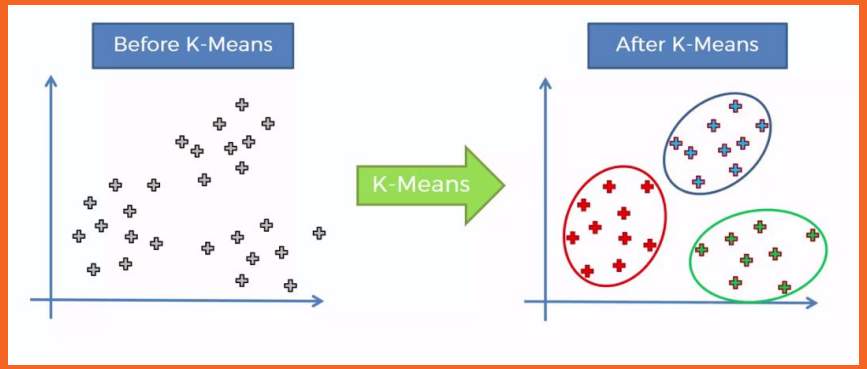
Spectral Relaxation for K-means Clustering

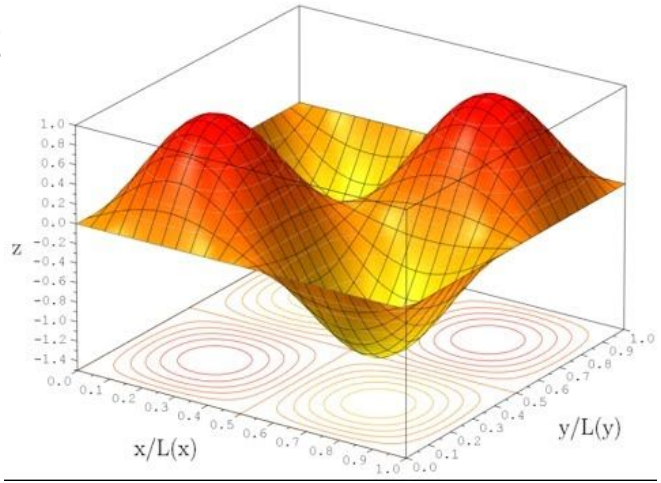


-Presentation by Sakshi Jain (B20ME065)

What is Spectral ?

Spectral methods are a class of techniques used in applied mathematics and scientific computing to numerically solve certain differential equations, potentially involving the use of the fast fourier transform.

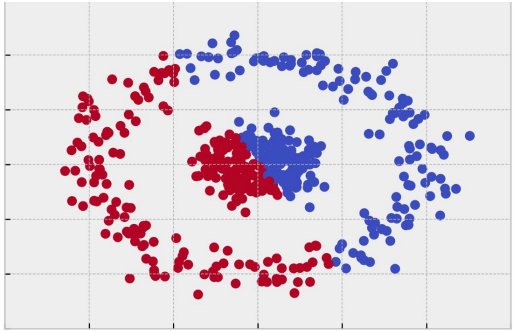
Spectral methods are computationally less expensive than finite element methods, but become less accurate for problems with complex geometries and discontinuous coefficients.



What is Spectral Clustering?

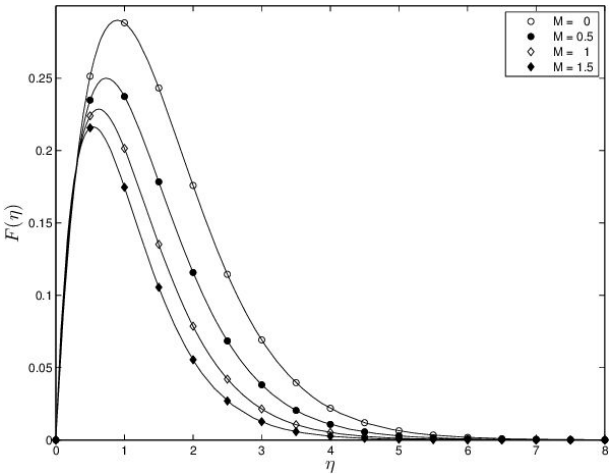
Spectral clustering techniques make use of the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions.

The weighted kernel *k*-means problem can be reformulated as a spectral clustering (graph partitioning) problem and vice versa.



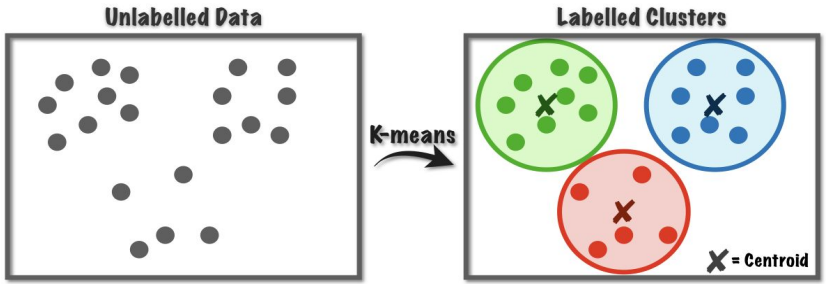
What is Spectral Relaxation?

The spectral relaxation method (SRM) is based on simple iteration schemes formed by reduction of the order of the momentum equation followed by a rearrangement of the resulting governing nonlinear equation systems.The SRM does not require any evaluation of derivatives, perturbation, and linearization.The method is accurate, effective, and easy and can be used for solving a wide variety of highly nonlinear systems of similarity boundary layer equations



What is Kmeans ?

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms.
In Kmeans, clusters are represented by the centre of mass of their members. It uses an algorithm in which each data point is assigned to the nearest cluster centre and then computing the centroid of its member data vectors and assigning that point as the cluster centre.



Kmeans Algorithm Sample

```
def Kmeans(X,x,y):

  cluster_1 = []
  cluster_2 = []
  cluster_3 = []
  cluster_4 = []

  for i in range(X.shape[0]):
    a = [X.iloc[i,1],X.iloc[i,2]]
    d = []
    for j in range(K):
      b = [x[j],y[j]]
      d.append(Euclidean_distance(a,b,2))
    e = min(d)
    f = d.index(e)

    if f==0:
      cluster_1.append(i)
    elif f==1:
      cluster_2.append(i)
    elif f==2:
      cluster_3.append(i)
    elif f==3:
      cluster_4.append(i)

clustering = [cluster_1,cluster_2,cluster_3,cluster_4]

centroid_xx,centroid_yy = [0]*K, [0]*K
for i in range(K):
  p = 0
  q = 0
  for j in clustering[i]:
    p += X.iloc[j,1]
    q += X.iloc[j,2]
  if len(clustering[i]) != 0:
    centroid_xx[i] = round(p/len(clustering[i]),4)
    centroid_yy[i] = round(q/len(clustering[i]),4)
  else:
    centroid_xx[i] = x[i]
    centroid_yy[i] = y[i]

return [centroid_xx,centroid_yy,clustering]
```

What's new?

How Kmeans is related to Spectral relaxation?

Another approach is to minimize the sum-of-squares cost function using the coordinate descent method

Spectral Relaxation Approach

The data-vectors are m dimentional and there are n such data-vectors. The m-by-n data vector matrix

$$A = [a_1, a_2, ..., a_n]$$

A_i is m-by-s_i matrix,i.e., the ith cluster contains the data vectors in A_i,

$$A_i = [a_1^{(i)}, a_2^{(i)}, ..., a_{s_i}^{(i)}]$$

The sum-of-squares cost function is defined as:

$$ss(\Pi) = \sum_{i=1}^{k} \sum_{s=1}^{s_i} (||a_s^{(i)} - m_i||)^2, \qquad m_i = \sum_{s=1}^{s_i} \frac{a_s^{(i)}}{s_i}$$

$$ss(\Pi) = \sum_{i=1}^{k} \left(trace(A_i^T A) - \left(\frac{e^t}{\sqrt{s_i}} \right) A_i^T A \left(\frac{e}{\sqrt{s_i}} \right) \right)$$

$$ss(\Pi) = trace(A^T A) - trace(X^T A^T A X)$$

where X is n-by-k orthonormal matrix.

$$X = \begin{pmatrix} s_1 \\ s_2 \\ . \\ s_k \end{pmatrix} \begin{pmatrix} \frac{e}{\sqrt{s_1}} & & \\ & \frac{e}{\sqrt{s_2}} & \\ . & . & . \\ & & \frac{e}{\sqrt{s_k}} \end{pmatrix} \qquad (1)$$

We need to minimize the cost function which is equivalent to

$$max\{trace(X^T A^T A X)\}$$

And it is easy to see that

$$trace(X^T A^T A X) = \sum_{i=1}^{k} \frac{x_i^T A^T A x_i}{x_i^T x_i} = \sum_{i=1}^{k} \frac{(||Ax_i||)^2}{(||x_i||)^2}$$

Left-hand-side can be easily written as the weighted sum of the squared Euclidean norm of the mean vector of each cluster.

$$min\{ss(\Pi)\} \geq trace(A^T A) - max\{trace(X^T A^T A X)\} = \sum_{i=k+1}^{min\{m,n\}} \sigma_i^2(A)$$

where $\sigma_i(A)$ is the i largest singular value of A.

Let H be a symmetric matrix with eigenvalues as:

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$$

and the corresponding eigenvectors $U = [u_1, u_2, ..., u_n]$. Then

$$\lambda_1 + \lambda_2 + ... + \lambda_k = max_{X^T X = I_k} trace(X^T H X)$$

Spectral Relaxation

Finally,

$$min_{\Pi} ss(\Pi) \geq max_a \sum_{i=k+1}^{min\{m,n\}} \sigma_i^2(A - ae^T)$$

$$ss(\Pi) = \frac{1}{2} \sum_{i=n-k+1}^{n} \lambda_i(W)$$

where $W = (||a_i - a_j||)_{i,j=1}^{n}$ and λ_i is the eigenvalue.

Cluster Assignment Using Pivoted QR Decomposition

The gram matrix of A is:

$$A^TA = \begin{pmatrix} A_1^TA_1 & 0 & ... & 0 \\ 0 & A_2^TA_2 & ... & 0 \\ . & . & . & . \\ 0 & & 0 & ...A_k^TA_k \end{pmatrix} + E = B + E$$

If the overlaps among the clusters represented by the submatrices A_i are small, then the norm of E will be small as compare with the block diagonal matrix B in the above equation. Let the largest eigenvector of $A_i^TA_i$ be y_i , and $A_i^TA_iy_i = \mu_i y_i, \quad ||y_i|| = 1, \quad i = 1,2,...,k.$

Let the eigenvalues and eigenvectors of AT A be:

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n, \quad A^TAx_i = \lambda_i x_i$$

Assume that there is a gap between the two eigenvalue sets $\{\mu_1, \mu_2, ..., \mu_k\}$ and $\{\lambda_{k+1}, ..., \lambda_n\}$, i.e.,

$$0 < \delta = min\{|\mu_i - \lambda_i|, |i = 1, ..., k, j = k+1, ..., n\}$$

By Davis-Kahan $\sin(\theta)$ theorem,

$$X_k^T = [x_1, x_2, ..., x_k] = Y_kV + O(||E||)$$

where V is an k-by-k orthogonal matrix. Ignoring the $O(||E||)$ term, we get that,

$$X_k^T = [y_{11}v_1, ..., y_{1s_1}v_1, ..., y_{k1}v_k, ..., y_{ks_k}v_k]$$

where $y_i^T = [y_{i1}, ..., y_{is_i}]$ and $V^T = [v_1, ..., v_k]$

When QR decomposition is applied with column pivoting to X_k^T, with a permutation matrix P, such that

$$X_k^TP = QR = Q[R_{11}, R_{12}],$$

where Q is a k-by-k orthogonal matrix, and R_{11} is a k-by-k upper triangular matrix.We then compute the matrix

$$\hat{R} = R_{11}^{-1}[R_{11}, R_{12}]P^T = [I_k, R_{11}^{-1}R_{12}]P^T$$

Then the cluster membership of each data vector is determined by the row index of the largest element in absolute value of the corresponding column of $\hat{R}$

Experimental Results

From experiments, the following conclusions are made:

- ➔ **The two clustering algorithms p-QR and p-Kmeans are comparable to each other, and both are better and sometimes substantially better than K-means in case of binary clustering.**
- ➔ **Both p-QR and p-Kmeans perform better than Kmeans**
- ➔ **For data sets with small overlaps, p-QR performs better than p-Kmean**

THANK YOU