# Kernel K-means and Spectral Clustering

**Max Welling**
Department of Computer Science
University of Toronto
10 King's College Road
Toronto, M5S 3G5 Canada
*welling@cs.toronto.edu*

## Abstract

This is a note to explain kernel K-means and its relation to spectral clustering.

## 1 Kernel K-means

The objective in K-means can be written as follows:

$$C(z,\mu) = \sum_i ||x_i - \mu_{z_i}||^2 \tag{1}$$

where we wish to minimize over the assignment variables $z_i$ (which can take values $z_i = 1,..,K$, for all data-cases $i$, and over the cluster means $\mu_k,\ k = 1..K$. It is not hard to show that the following iterations achieve that,

$$z_i = \arg\max_k ||x_i - \mu_k||^2 \tag{2}$$

$$\mu_k = \frac{1}{N_k} \sum_{i \in C_k} x_i \tag{3}$$

where $C_k$ is the set of data-cases assigned to cluster k.

Now, let's assume we have defined many features, $\phi(x_i)$ and wish to do clustering in feature space. The objective is similar to before,

$$C(z,\mu) = \sum_i ||\phi(x_i) - \mu_{z_i}||^2 \tag{4}$$

We will now introduce a $N \times K$ assignment matrix, $Z_{nk}$, each column of which represents a data-case and contains exactly one 1 at row $k$ if it is assigned to cluster $k$. As a result we have $\sum_k Z_{nk} = 1$ and $N_k = \sum_n Z_{nk}$. Also define $L = \text{diag}[1/\sum_n Z_{nk}] = \text{diag}[1/N_k]$. Finally define $\Phi_{in} = \phi_i(x_n)$. With these definitions you can now check that the matrix $M$ defined as,

$$M = \Phi Z L Z^T \tag{5}$$

consists of $N$ columns, one for each data-case, where each column contains a copy of the cluster mean $\mu_k$ to which that data-case is assigned.

Using this we can write out the K-means cost as,

$$C = \mathbf{tr}[(\Phi - M)(\Phi - M)^T] \tag{6}$$

Next we can show that $Z^T Z = L^{-1}$ (check this), and thus that $(ZLZ^T)^2 = ZLZ^T$. In other words, it is a projection. Similarly, $I - ZLZ^T$ is a projection on the complement space. Using this we simplify eqn.6 as,

$$\begin{aligned} C &= \mathbf{tr}[\Phi(I - ZLZ^T)^2 \Phi^T] & (7)\\ &= \mathbf{tr}[\Phi(I - ZLZ^T)\Phi^T] & (8)\\ &= \mathbf{tr}[\Phi\Phi^T] - \mathbf{tr}[\Phi ZLZ^T \Phi^T] & (9)\\ &= \mathbf{tr}[K] - \mathbf{tr}[L^{\frac{1}{2}} Z^T K Z L^{\frac{1}{2}}] & (10)\end{aligned}$$

where we used that $\mathbf{tr}[AB] = \mathbf{tr}[BA]$ and $L^{\frac{1}{2}}$ is defined as taking the square root of the diagonal elements.

Note that only the second term depends on the clustering matrix Z, so we can we can now formulate the following equivalent kernel clustering problem,

$$\max_Z \mathbf{tr}[L^{\frac{1}{2}} Z^T K Z L^{\frac{1}{2}}] \tag{11}$$

$$\text{such that: } Z \text{ is a binary clustering matrix.} \tag{12}$$

This objective is entirely specified in terms of kernels and so we have once again managed to move to the ”dual” representation. Note also that this problem is very difficult to solve due to the constraints which forces us to search of binary matrices.

Our next step will be to approximate this problem through a relaxation on this constraint. First we recall that $Z^T Z = L^{-1} \Rightarrow L^{\frac{1}{2}} Z^T Z L^{\frac{1}{2}} = I$. Renaming $H = ZL^{\frac{1}{2}}$, with $H$ an $N \times K$ dimensional matrix, we can formulate the following relaxation of the problem,

$$\max_H \mathbf{tr}[H^T K H] \tag{13}$$

$$\text{subject to } H^T H = I \tag{14}$$

Note that we did not require $H$ to be binary any longer. The hope is that the solution is close to some clustering solution that we can then extract a posteriori.

The above problem should look familiar. Interpret the columns of $H$ as a collection of $K$ mutually orthonormal basis vectors. The objective can then be written as,

$$\sum_{k=1}^{K} \mathbf{h}_k^T K \mathbf{h}_k \tag{15}$$

By choosing $\mathbf{h}_k$ proportional to the $K$ largest eigenvectors of $K$ we will maximize the objective, i.e. we have

$$K = U \Lambda U^T, \quad \Rightarrow \quad H = U_{[1:K]} R \tag{16}$$

where $R$ is a rotation inside the eigenvalue space, $RR^T = R^T R = I$. Using this you can now easily verify that $\mathbf{tr}[H^T K H] = \sum_{k=1}^{K} \lambda_k$ where $\{\lambda_k\},\ k = 1..K$ are the largest K eigenvalues.

What is perhaps surprising is that the solution to this relaxed kernel-clustering problem is given by kernel-PCA! Recall that for kernel PCA we also solved for the eigenvalues of $K$. How then do we extract a clustering solution from kernel-PCA?

Recall that the columns of $H$ (the eigenvectors of $K$) should approximate the binary matrix $Z$ which had a single $1$ per row indicating to which cluster data-case $n$ is assigned. We could try to simply threshold the entries of $H$ so that the largest value is set to $1$ and the remaining ones to $0$. However, it often works better to first normalize $H$,

$$\hat{H}_{nk} = \frac{H_{nk}}{\sqrt{\sum_k H_{nk}^2}} \tag{17}$$

All rows of $\hat{H}$ are located on the unit sphere. We can now run a simple clustering algorithm such as K-means on the data matrix $\hat{H}$ to extract K clusters. The above procedure is sometimes referred to as "spectral clustering".

Conclusion: Kernel-PCA can be viewed as a nonlinear feature extraction technique. Input is a matrix of similarities (the kernel matrix or Gram matrix) which should be positive semi-definite and symmetric. If you extract two or three features (dimensions) you can use it as a non-linear dimensionality reduction method (for purposes of visualization). If you use the result as input to a simple clustering method (such as K-means) it becomes a nonlinear clustering method.