Notes on the SDP relaxation of k-means

The SDP relaxation of k-means

Iguchi et al.[1] formulated a semidefinite progmming approach to k-means as follows[2]:

$$
\begin{aligned}
\arg\max_{Z} \; & -\mathrm{Tr}(D_2 Z) \\
\text{s.t.} \; & \mathrm{Tr}(Z) = k \\
& Ze = e \\
& Z \geq 0 \text{ element-wise} \\
& Z \text{ is positive semidefinite}
\end{aligned}
$$

Where

- $D_2 = [d_{ij}] = [||x_i - x_j||^2]$
- $x_1, ..., x_n \in \mathbb{R}^q$
- The number of clusters, k, is known
- $e = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^q$

Note that without the SDP relaxation, we have a rigid structure for Z where $z_{ij} = \begin{cases} n_k^{-1} & x_i, x_j \text{ in same cluster } k \\ 0 & \text{else} \end{cases}$

Equating the trace formulation of k-means to kernel k-means

We can see that the data matrix $X = \begin{bmatrix} x_1^\top \\ \vdots \\ x_n^\top \end{bmatrix}$ is not explicitly in the objective, although squared Euclidean distances are. We can rewrite this as a kernel formulation by noting that $D_2 = \kappa(B)$ where B is a kernel matrix:

$$
\begin{aligned}
-\mathrm{Tr}(D_2 Z) &= -\mathrm{Tr}(\kappa(B) Z) \\
&= -\mathrm{Tr}((be^\top - 2B + eb^\top) Z) \\
&= 2\mathrm{Tr}(BZ) - \mathrm{Tr}(be^\top Z) - \mathrm{Tr}(eb^\top Z) \\
&= 2\mathrm{Tr}(BZ) - \mathrm{Tr}(be^\top) - \mathrm{Tr}(Zeb^\top) \\
&= 2\mathrm{Tr}(BZ) - \mathrm{Tr}(be^\top) - \mathrm{Tr}(eb^\top) \\
&= 2\mathrm{Tr}(BZ) - 2\mathrm{Tr}(be^\top) \\
&= 2\mathrm{Tr}(BZ) - 2\mathrm{Tr}(B)
\end{aligned}
$$

[1] https://arxiv.org/abs/1505.04778
[2] the notation is slightly different here

... where $b = \text{diag}(B)$, the vector of diagonal entries of B. Note that if we think of B as a weight matrix for an undirected graph, $\text{Tr}(B) = 0$. Similarly, if we impose that the diagonal entries of B are equal to 1 (e.g., B is a correlation matrix), then $\text{diag}(B) = n$. Either way, $\text{Tr}(B)$ does not depend on Z, so we can ignore it in the objective, and we can see that $\arg\max_Z \text{Tr}(D_2 Z) = \arg\max_Z \text{Tr}(BZ)$, which is just the typical kernel formulation of k-means:

$$\begin{aligned}
&\arg\max_Z \text{Tr}(BZ) \\
&\quad \text{s.t. } \text{Tr}(Z) = k \\
&\qquad\quad z_{ij} = \begin{cases} n_k^{-1} & x_i, x_j \text{ in same cluster } k \\ 0 & \text{else} \end{cases}
\end{aligned}$$

Similarly, we can go from a kernel formulation of k-means to one based on squared Euclidean distances by noting that $D_2 = \tau(B)$. For simplicity of notation, we will rewrite $\arg\max_x f(x) = \arg\max_x 2f(x)$.

$$\begin{aligned}
2\text{Tr}(BZ) &= 2\text{Tr}(\tau(D_2)Z) \\
&= \text{Tr}(-PD_2PZ) \\
&= -\text{Tr}((I - n^{-1}ee^\top)D_2(I - n^{-1}ee^\top)Z) \\
&= -\text{Tr}((D_2 - n^{-1}D_2ee^\top - n^{-1}ee^\top D_2 + n^{-2}ee^\top ee^\top D_2)Z) \\
&= -\text{Tr}(D_2Z) + n^{-1}\text{Tr}(D_2ee^\top Z) + n^{-1}\text{Tr}(ee^\top D_2Z) - n^{-2}\text{Tr}(ee^\top D_2Z) \\
&= -\text{Tr}(D_2Z) + 2n^{-1}\text{Tr}(D_2ee^\top) - n^{-2}\text{Tr}(D_2ee^\top)
\end{aligned}$$

Since the second and third terms do not depend on Z, we can discard them, and we get $\arg\max_Z \text{Tr}(BZ) = \arg\max_Z -\text{Tr}(D_2Z)$.[3]

Equating the SDP relaxation of k-means to the SDP relaxation of ratio cut

The ratio cut objective is:

$$\arg\min_Z \text{Tr}(LZ)$$

where L is the combinatorial graph Laplacian and Z has the same structure as before. If we relax the optimization problem by not enforcing Z to have this structure, we can see that:

$$\arg\min_Z \text{Tr}(LZ) = \arg\max_Z \text{Tr}(L^\dagger Z)$$

where $L^\dagger$ is the generalized inverse of L. Since $L^\dagger$ is positive semidefinite, it can be thought of as a kernel matrix, and we can apply the $\tau(\cdot)$ transformation to it to obtain D_2. In this case, D_2 is the expected commute time of the graph that generated L.

The argmin and argmax equivalence is not true in general if we force Z to have the structure that we want. It also is not true if we apply the SDP constraints (namely $Z \geq 0$ element-wise). One question of interest is under what conditions can we equate the two objectives under the SDP constraints.

[3] We can rewrite $\text{Tr}(D_2ee^\top) = \sum_{i,j} d_{ij}^2 = 2\sum_{i<j} d_{ij}$