Fast Spectral Clustering with Efficient Large Graph Construction

Wei Zhu, Feiping Nie, and Xuelong Li, *Fellow, IEEE*

zwvews@gmail.com, feipingnie@gmail.com, xuelong_li@ieee.org

Contribution

We propose a novel and efficient anchor generation approach, i.e. BKHK. BKHK has low computational complexity and relatively high performance compared with K-means. It is worthwhile to note that BKHK can be easily portable to other spectral based methods to enhance their ability of dealing with large scale data.

BKHK

Let us begin with two class k-means which can be formulated as follows:

$$\min_{G\in Ind,\mathbf{1}^TG=[\kappa,\iota]} \|X - GC^T\|_F^2 \tag{1}$$

where $C \in \mathbb{R}^{d\times 2}$ is the center of the cluster, $G \in \mathbb{R}^{n\times 2}$ is the index matrix, g_{i1} equals 1 if the i-th sample belongs to the first cluster, or g_{i2} equals 1 otherwise, and $\mathbf{1}$ is the column-vector of all ones. Moreover, κ and ι are the number of samples in these two clusters, and we clearly have $\kappa + \iota = n$. We can simply set $\kappa = \lfloor \frac{n}{2} \rfloor$ to make different clusters have same amount of samples (If n is an odd number, we set $\kappa = \frac{n-1}{2}$). We then rewrite problem (1) as:

$$\min_{G\in Ind,\mathbf{1}^TG=[\kappa,\iota]} \sum_{i=1}^{n}\sum_{k=1}^{2} \|x_i - c_k\|_2^2 g_{ik} \tag{2}$$

where c_k is the k-th column of C. For convenience, we define matrix $E \in \mathbb{R}^{n\times 2}$ and the (i,j)-th entry of E is denoted as $e_{ij} = \|x_i - c_k\|_2^2$, thus we rewrite problem (2) as

$$\min_{G\in Ind,\mathbf{1}^TE=[\kappa,\iota]} Tr(E^TG) \tag{3}$$

Let g denoted the first column of G, since G is index matrix, the second column can be denoted as $(\mathbf{1}-g)$, then problem (3) can be rewritten as

$$\min_{g\in\{0,1\},\mathbf{1}^Tg=\kappa} g^Te_1 + (\mathbf{1}-g)^Te_2 \tag{4}$$

where e_1 and e_2 are the first and second column of E, respectively. Then, we arrive at

$$\min_{g\in\{0,1\},\mathbf{1}^Tg=\kappa} g^T(e_1 - e_2) \tag{5}$$

The solution to problem (5) is intuitively, i.e. we assign $g_i = 1$ when the i-th element of $e_1 - e_2$ is the κ minimum of all its elements.

Graph Construction

We adopt a parameter-free yet effective neighbor assignment method. The neighbor assignment for the i-th sample can be seen as solving following problem

$$\min_{z_i^T\mathbf{1}=1,z_i\geq\mathbf{0}} \sum_{j=1}^{m} h(x_i,u_j)z_{ij} + \gamma\sum_{j=1}^{m} z_{ij}^2, \tag{6}$$

where $Z \in \mathbb{R}^{n\times m}$ denotes the similarity between the i-th sample and the j-th anchor, γ can be set as $\gamma = \frac{k}{2}h(i,k+1) - \frac{1}{2}\sum_{j=1}^{k} h(i,j)$. The solution to problem (6) is

$$z_{ij} = \frac{h(x_i,u_{k+1}) - h(x_i,u_j)}{\sum_{j'=1}^{k}(h(x_i,u_{k+1}) - h(x_i,u_{j'}))}. \tag{7}$$

As we obtain the matrix Z, similarity matrix A then can be obtained by

$$A = Z\Delta^{-1}Z^T, \tag{8}$$

where $\Delta \in \mathbb{R}^{m\times m}$ is a diagonal matrix and the i-th entry is defined as $\sum_{j=1}^{n} z_{ji}$.

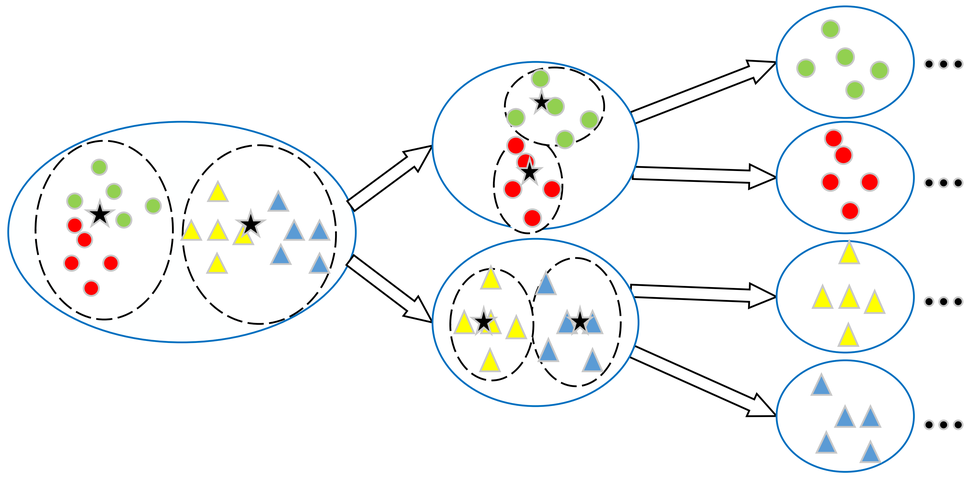
Spectral Analysis

Spectral clustering can then be performed by

$$\min_{F^TF=I} Tr(F^TLF) \tag{9}$$

where $F \in \mathbb{R}^{n\times c}$ is indicator matrix, c is the clustering number. $L \in \mathbb{R}^{n\times n}$ is Laplacian Matrix which is defined by $L = D - A$. Thus, the solution to problem (9) can be obtained by performing eigenvalue decomposition on A. In addition, according to Equation (8), A can be written as $A = BB^T$, where $B = Z\Delta^{-\frac{1}{2}}$, instead of directly performing eigenvalue decomposition on A, we prefer to performing SVD on B to speed up the algorithm.

Illustration of Balanced K-means based Hierarchical K-means



There are two more points we would like to mention, first, BKHK is a pretty efficient method especially for large scale data, and can be easily applied to accelerate other graph based learning methods, e.g. hashing, semi-supervised learning , dimensionality reduction, RBF networks, etc. Second, early stopping and down-sampling can speed up k-means a lot, and can also be adopted by BKHK for extremely large data.

Comparative results on benchmark datasets

ACC (%)	SC	LSC-R	LSC-K	FSC
USPS	64.0	57.6	57.8	61.1
Protein	43.9	43.4	43.6	44.2
Connect-4	44.3	38.4	39.2	42.5
MNIST	68.4	63.3	69.5	67.1

RT (s)	SC	LSC-R	LSC-K	FSC
USPS	5.8	2.1	12.8	3.0
Protein	73.3	16.1	267.4	19.8
Connect-4	398.7	17.7	534.5	20.9
MNIST	242.6	3.4	149.4	41.5

According to the results, we conclude several interesting points. Anchor-based graph can greatly reduce the computational cost (e.g. LSC-R and FSC), nonetheless, inappropriate use even slow down the speed. LSC-K adopts k-means to generate anchors, the high computational complexity of k-means greatly limits the algorithm, particularly, k-means may need lots of iterations to converge in some cases, e.g. we need about 500 seconds to perform k-means on Connect-4. And as mentioned above, LSC-R randomly selects the anchors, which makes it extremely efficient but also with poor performance. By contrast, FSC adopts BKHK to generate anchors, combined with effective non-parameter graph construction method, it achieves pretty high performance with little time cost, and there is no doubt that FSC is the best choice for real life application among all the methods.

Computational Complexity

1. We need $O(nd\log(m)t)$ to obtain m anchors by BKHK algorithms, where t is the iterative number of balanced k-means. 2. We need $O(ndm + nm\log(m))$ to construct graph by anchor-based approach. 3. We need $O(m^3 + m^2n)$ to obtain F by perform SVD on matrix B. 4. We need $O(ndmr)$ to perform k-means for final clustering results, where r is the iterative number.

Considering that $m \ll n$ and t is usually pretty small, the overall computational complexity of FSC is $O(ndm)$.