FAST SPECTRAL CLUSTERING WITH EFFICIENT LARGE GRAPH CONSTRUCTION

*Wei Zhu, Feiping Nie**

School of Computer Science and Center for OPTIMAL, Northwestern Polytechnical University Xi'an 710072, Shaanxi, P. R. China

Xuelong Li

Center for OPTIMAL, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, P. R. China

ABSTRACT

Spectral clustering has been regarded as a powerful tool for unsupervised tasks despite its excellent performance, the high computational cost has become a bottleneck which limits its application for large scale problems. Recent studies on anchor-based graph can partly alleviate the problem, however, it is still a great challenge to deal with such data with both high performance and high efficiency. In this paper, we propose Fast Spectral Clustering (FSC) to efficiently deal with large scale data. The proposed method first constructs anchor-based similarity graph with Balanced K-means based Hierarchical K-means (BKHK) algorithm, and then performs spectral analysis on the graph. The overall computational complexity is $O(ndm)$, where n is the number of samples, d is the number of features, and m is the number of anchors. Comprehensive experiments on several large scale data sets demonstrate the effectiveness and efficiency of the proposed method.

***Index Terms*—** Spectral clustering, anchor-based graph, balanced k-means based hierarchical k-means

1. INTRODUCTION

Increasing amounts of data bring lots of challenges for conventional spectral clustering [1], which is one of the most popular method used in real life clustering tasks. Specifically, conventional spectral clustering needs two independent steps: first, constructing similarity graph by K-Nearest Neighbors (K-NN); then, performing spectral analysis on the obtained graph. Unfortunately, both the two steps are time consuming for large scale data, and the computational complexity of spectral clustering is $O(n^2d)$ at least.

Recently, anchor-based graph has been widely adopted in spectral based method to speed up the procedure, e.g. scalable semi-supervised learning [2], large scale spectral clustering [3], multiview large scale spectral clustering [4], and large scale spectral based dimensionality reduction [5]. The anchor-based graph is used to replace the K-NN graph to efficiently capture the data structure, and numerical results show that anchor-based graph makes spectral based method achieve pretty high performance as well as relatively low computational cost. The most important step for anchor-based graph is anchor generation, there are two most commonly used strategies to generate anchors, i.e., random selection and k-means generation [3]. Generally speaking, k-means generation strategy achieves good performance by generating representative anchors but has high computational cost [3], random strategy is efficient while the performance cannot be guaranteed, therefore, we reach the horn of a dilemma.

To tackle the problem, we then propose a novel anchor generation method called Balanced K-means based Hierarchical K-means (BKHK) which generates representative anchors with quite low computational cost. Concretely, for a data set $X \in \mathbb{R}^{n\times d}$, BKHK adopts balanced binary tree structure and has a computational complexity of $O(nd\log(m)t)$, where t and m are the number of iterations and anchors, respectively, which is a great reduction compared to k-means with $O(ndmt)$. We then adopt BKHK to perform Fast Spectral Clustering (FSC) with anchor-based graph, the computational complexity of FSC is $O(ndm)$ which makes it suitable for handling large scale data.

Three main contributions of this paper are listed as follows:

1. 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.

2. Based on BKHK, we propose FSC to cluster large scale data. FSC deals with large scale data efficiently with a computational complexity $O(ndm)$, while conventional spectral clustering needs $O(n^2d)$ at least.

3. Comprehensive experiments on several large scale data sets demonstrate the efficiency and the effectiveness of the proposed BKHK and FSC.

*This work is supported by the Fundamental Research Funds for the Central Universities (Grant no. 3102015BJ(II)JJZ01).

2. SPECTRAL CLUSTERING WITH ANCHOR-BASED GRAPH

In this section, we demonstrate the proposed BKHK and FSC. Similar to conventional spectral based method, FSC contains two parts: first, constructing anchor-based similarity graph via BKHK algorithm; then, performing spectral analysis on the graph.

2.1. Anchor-Based Graph Construction with BKHK

Conventional spectral based methods always use K-NN to capture local manifold structure with a high computational complexity. Recent studies adopt anchor-based strategy to construct similarity graph to speed up the algorithm. For detail, they first generate some anchors, and then only calculate the distance between the anchors and original samples. The anchors generation strategy is obviously crucial for the final performance. Random selection strategy is extremely fast, but it cannot guarantee that the selected anchors are good enough to construct the similar graph of the whole data, thus, the performance is usually poor in practice. By contrast, k-means strategy can generate representative anchors, which surely makes it achieve better performance, nevertheless, k-means has high computational complexity, i.e. $O(ndmt)$, which makes traditional k-means strategy almost impossible to apply to large scale data. There are several methods to speed up the k-means procedure, e.g., performing down-sampling on the data or early stopping iterations, however they may also cause performance degradation in many cases.

2.1.1. Balance K-means Based Hierarchical K-means

We propose BKHK to tackle this problem, BKHK can generate representative anchors efficiently especially for large scale data. For detail, BKHK adopts balanced binary tree structure, in other words, BKHK iteratively segment the data into two clusters with same number of samples.

We now show the detail of BKHK algorithm. Let us begin with two class k-means which can be formulated as follows:

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

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:

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

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

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

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

$$\min_{g\in\{0,1\},\mathbf{1}^T g=\kappa} g^T e_1 + (\mathbf{1} - g)^T e_2 \tag{4}$$

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

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

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.

Since the two clusters have same number of samples, we call this algorithm balanced k-means, and we hierarchically perform balanced k-means to perform the BKHK. The computational complexity of BKHK is $O(nd\log(m)t)$, and the algorithm is summarized in Algorithm 1.

Algorithm 1 Balanced K-means Based Hierarchical K-means (BKHK)

Input: Data matrix $X \in \mathbb{R}^{n\times d}$, number of anchors m

while not reach the termination condition **do**
  Initialize the cluster center matrix C.
  while not converge **do**
    Obtain the indicator vector g via solving problem (5).
    Obtain the indicator matrix $G = [g, 1 - g]$.
    Calculate the cluster center for both of the two clusters.
  end while
  Hierarchically perform Algorithm 1 on the obtained two sub-clusters.
end while
Obtain anchor set U by calculating the center of all sub-clusters.

Output: Anchor set U.

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 [6], semi-supervised learning [7, 8], dimensionality reduction [9], RBF networks [10], 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.

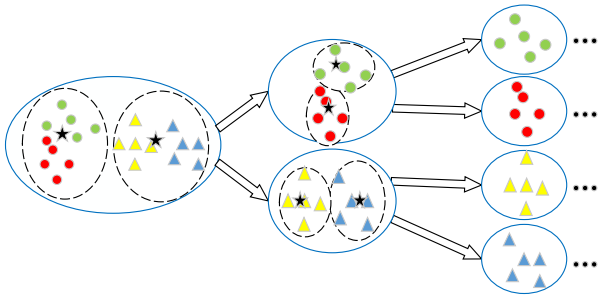


Fig. 1. BKHK adopts binary tree structure, and performs balanced k-means at each node.

2.1.2. Similarity Graph Construction with the Obtained Anchors

The similarity graph construction with the obtained anchors is a well-studied problem [2]. Let U denote the set of the generated anchors, and let $U_{\langle i \rangle}$ denote the set of k-nearest anchors for the i-th sample. Conventional methods usually use kernel based neighbor assignment strategy, e.g. Gaussian kernel $K_t(x_i, u_j) = exp(-\|x_i - u_j\|_2^2/2\sigma^2)$, but kernel based methods always bring extra parameters, e.g. bandwidth σ. Therefore, 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 [11]

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

where $Z \in \mathbb{R}^{n \times m}$ denotes the similarity between the i-th sample and the j-th anchor, $h(x_i, u_j)$ is the distance between the i-th sample and its j-th nearest anchor. To keep it simple, we define $h(x_i, u_j) = \|x_i - u_j\|_2^2$, which is the square of Euclidean distance. Follow [11], 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

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

For detail derivation, see [11].

As we obtain the matrix Z, similarity matrix A then can be obtained by [2]:

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

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

2.2. Spectral Analysis with Anchor-based Graph

It can be verified that the similarity matrix A obtained by solving problem (8) is double stochastic and symmetric, which means that the graph is automatically normalized [3]. Spectral clustering can then be performed by

$$\min_{F^T F = I} Tr(F^T L F) \tag{9}$$

where $F \in \mathbb{R}^{n \times c}$ is indicator matrix, c is the clustering number. Degree matrix $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix with the i-th element is $d_{ii} = \sum_j A_{ij}$, we then get $D = I$. $L \in \mathbb{R}^{n \times n}$ 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.

3. COMPUTATIONAL COMPLEXITY ANALYSIS

Given data $X \in \mathbb{R}^{n \times d}$, the computation complexity of the proposed method can be divided into 3 parts.

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 $O(m^3 + m^2 n)$ 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 $O(ndm)$.

4. EXPERIMENTS

In this section, we experimentally show the effectiveness and efficiency of the proposed method.

Table 1. Data Set Description

Data Set	Samples	Features	Classes
USPS	9298	256	10
Protein	24387	357	3
Connect-4	67557	126	3
MNIST	70000	784	10

4.1. Data Sets

The experiments are conducted on 4 large scale data sets, including handwritten digit (i.e. MNIST [12] and USPS [13]), molecular biology (e.g., Protein [14]), and connect-4 game (e.g., Connect-4 [15]). The detail of all the data sets are summarized in Table 1.

4.2. Comparison Methods

To demonstrate the superiority of the proposed method, we compare it with several state-of-the-art methods stated as follows:

1. **SC**: spectral clustering [16].
2. **LSC-R**: Landmark-based spectral clustering while the landmarks are randomly selected [17].
3. **LSC-K**: Landmark-based spectral clustering while the landmarks are generated by k-means [17].
4. **FSC**: the method proposed in this paper.

We use 5-nearest neighbors to construct graph. For the number of anchors used in LSC-K, LSC-R, and FSC, we set $m = 1024$.

4.3. Evaluation Metric

For all the methods, the clustering results are evaluated by ACCuracy (ACC). We run all methods 10 times, and record the mean results as well as the running time. All the codes in the experiments are implemented in MATLAB R2015b, and run on a Windows 10 machine with 3.20 GHz i5-3470 CPU, 16 GB main memory.

Table 2. Performance (ACC) on 4 data sets (%)

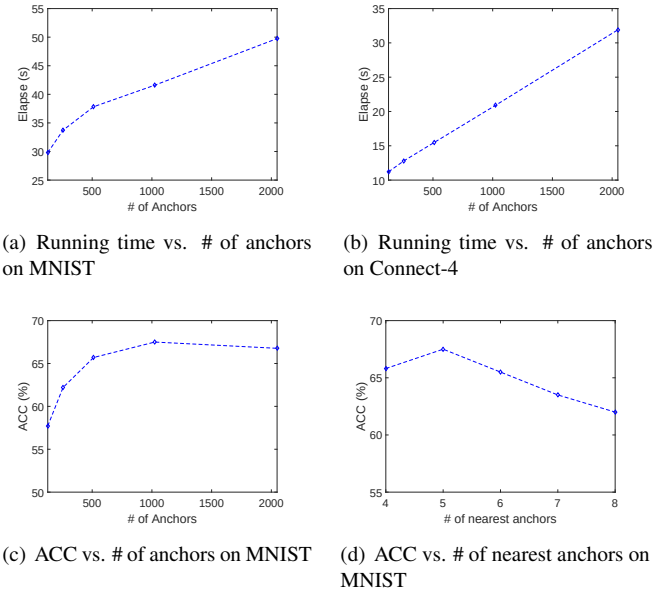
Data Set	SC	LSC-R	LSC-K	FSC
USPS	64.0	57.6	57.8	61.1
Protein	43.9	43.4	43.6	44.2
Connect-4	44.3	38.4	39.2	42.5
MNIST	68.4	63.3	69.5	67.1

Table 3. Running time on 4 data sets, we omit the common final k-means step to provide better comparisons (s)

Data Set	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

4.4. Clustering Results

The performance and running time of all the methods are shown in Table (2) and Table (3), respectively. 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, see Table 2. 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.



(a) Running time vs. # of anchors on MNIST

(b) Running time vs. # of anchors on Connect-4

(c) ACC vs. # of anchors on MNIST

(d) ACC vs. # of nearest anchors on MNIST

Fig. 2. Parameter sensitivity study of FSC.

4.5. Parameter Sensitivity

We now experimentally study the influence of the parameters for running time and performance. Figure 2(a) and Figure 2(b) show that large amounts of anchors does not increase the running time a lot, nonetheless, from Figure 2(c), overly large amounts of anchors are useless for final performance. Figure 2(d) show that FSC is robust to k, i.e. number of nearest anchors, and it is a reasonable choice to set $k = 5$.

5. CONCLUSIONS

In this paper, we first propose a novel anchor generation strategy, called Balanced K-means based Hierarchical K-means (BKHK), BKHK can generate representative anchors with little computational cost compared with k-means strategy. Based on BKHK, we then propose Fast Spectral Clustering (FSC) to deal with large scale data. We then make experiments on several real life large scale data sets to validate the superiority of the proposed BKHK and FSC.

6. REFERENCES

[1] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems 14*, 2001, pp. 849–856.

[2] Wei Liu, Junfeng He, and Shih-Fu Chang, "Large graph construction for scalable semi-supervised learning," in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 679–686.

[3] Deng Cai and Xinlei Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE T. Cybernetics*, vol. 45, no. 8, pp. 1669–1680, 2015.

[4] Yeqing Li, Feiping Nie, Heng Huang, and Junzhou Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 2750–2756.

[5] Deng Cai, "Compressed spectral regression for efficient nonlinear dimensionality reduction," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015, pp. 3359–3365.

[6] Xuelong Li, Di Hu, and Feiping Nie, "Large graph hashing with spectral rotation," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[7] Dengyong Zhou, Jason Weston, Arthur Gretton, Olivier Bousquet, and Bernhard Schölkopf, "Ranking on data manifolds," in *Advances in Neural Information Processing Systems 16*, 2003, pp. 169–176.

[8] Xiaojin Zhu, "Semi-supervised learning literature survey," *Computer Science*, vol. 37, no. 1, pp. 63–77, 2008.

[9] Feiping Nie, Dong Xu, Xuelong Li, and Shiming Xiang, "Semisupervised dimensionality reduction and classification through virtual label regression," *IEEE Trans. Systems, Man, and Cybernetics, Part B*, vol. 41, no. 3, pp. 675–685, 2011.

[10] Friedhelm Schwenker, Hans A. Kestler, and Günther Palm, "Three learning phases for radial-basis-function networks," *Neural Networks*, vol. 14, no. 4-5, pp. 439–458, 2001.

[11] Feiping Nie, Xiaoqian Wang, Michael I. Jordan, and Heng Huang, "The constraint laplacian rank algorithm for graph-based clustering," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[12] Yann Lcun, Leon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[13] Jonathan J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, 1994.

[14] Zheng Rong Yang, "Biological applications of support vector machines," *Briefings in Bioinformatics*, vol. 5, no. 4, pp. 328–338, 2004.

[15] M. Lichman, "UCI machine learning repository," 2013.

[16] Ulrike von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[17] Xinlei Chen and Deng Cai, "Large scale spectral clustering with landmark-based representation," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, 2011.