On-line Clustering: K-means and Spectral Clustering

December 19, 2017

1 Introduction

Clustering is one of the most common techniques employed in data analysis, with many applications such as image processing and social science or psychology. In virtually every scientific field dealing with empirical data, people attempt to get a first impression on their data by trying to identify groups of similarity in their data [1]. The problem can become very complex when it comes to modern high dimensional data such as images, documents, or multimedia. The data may also arrive as an infinite stream. However different clustering method often yield different results, yet there is actually no single correct answer to the various ways one can cluster a set of data. While some clustering results are reasonable, some obviously does not make sense.

In this project we investigate two popular clustering algorithms: K-means and spectral clustering. K-means is very effective in distinguishing different groups of datasets that is obviously apart. On the other hand Spectral clustering can identify more sophisticated data linkage. Our final objective is to come up with our own clustering method. We try to group the data into the most seemingly reasonable clusters.

1.1 K-means Clustering

The K-means clustering is a basic problem in unsupervised machine learning field. In this setting we are given n vectors in Euclidian space. We want to partition all the vectors into k clusters $S_1, ..., S_k$ to minimize

$$\sum_{i=1}^{k} \sum_{v \in S_i} \|v - c_i\|_2^2. \tag{1}$$

where $c_i = (1/|S_i|) \sum_{v \in S_i} v$ are the centers of the clusters S_i. This problem was originally proposed for the off-line setting. So all the vectors are known before we start to find the suitable clustering. Even in the off-line setting it is still difficult to find a good solution in a limited amount of time. Lloyd's algorithm [2] is a popular solution for k-means clustering. The k-means++ [3] provides an expected $O(log(k))$ approximation. Several improvements of k-means clustering algorithm was made later on [4–6].

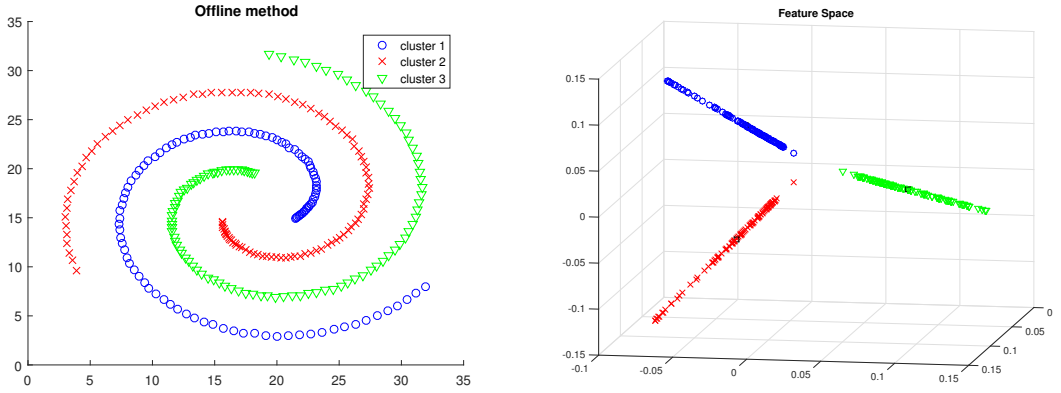
1.2 Spectral Clustering

K-means clustering cannot handle the case that input data is not linearly separated. In order to solve this problem, spectral clustering provides us a method to cluster non-linearly separable data by calculating eigenvectors of the similarity (or affinity) matrix. Today, spectral clustering is the most popular clustering method. Compared to traditional approach, the results generated by spectral clustering is better accepted. When performing spectral clustering on a data set we follow the operations listed below [7]: ① Eigenvectors corresponding to the smallest several eigenvalues of the symmetric normalized Laplacian matrix L_{sym}, defined by (2), are calculated.

$$L_{sym} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{2}$$

where A is the affinity matrix of input data, and D is the corresponding degree matrix with $D_{ii} = \sum_j A_{ij}$. ② K-means algorithm is applied on the eigenvectors generated from previous step. Calculation of eigenvectors

could be regarded as a process of mapping original data into another feature space, like kernel K-means, in which each cluster of data is linearly separated from others. For example, Fig. 1 presents the idea of spectral clustering, in which the data points are mapped to the feature space. Notably, k-means algorithm can easily solve the clustering problem in this feature space.



(a) result of spectral clustering (b) feature space representation

Figure 1: Spectral clustering of Spiral dataset

1.3 On-line Algorithms

Even though k-means and spectral clustering algorithms perform perfectly while operating unsupervised feature learning, the fact that value of k is required to be known advanced motivate the development of on-line algorithm. Contrasted to traditional off-line k-means algorithm, on-line k-means operates on a sequence of data with arbitrary order one by one, making it more appropriate to practical cases. Here is the review of the on-line k-means algorithm [8], where V is a stream of data. k is the initialization for the number of clusters. When a new data arrives, on-line k-means will either add it into an existing cluster or create a new cluster.

Algorithm 1 On-line K-means algorithm

input: V, k.
$C \leftarrow$ first $k+1$ distinct vectors in V; $n = k+1$
$w^* \leftarrow \min_{v,v' \in C} \frac{\|v - v'\|^2}{2}$; $r \leftarrow 1$; $q_1 \leftarrow 0$; $f_1 = \frac{w^*}{k'}$.
for $y \in$ the remainder of $\mathcal{Y}$ **do**
    $D(v, C) = \min_{c \in C} \|v - c\|$.
    with probability $p = \min\left(\frac{D^2(v,C)}{f_r}, 1\right)$
        $C \leftarrow C \cup \{v\}$; $q_r \leftarrow q_r + 1$.
    if $q_r \geq k'$ **then**
        $r \leftarrow r + 1$; $q_r \leftarrow 0$; $f_r \leftarrow 2 \cdot f_{r-1}$.
    end if
    output: class $c_v = \arg\min_{c \in C} \|v - c\|$
end for

2 Proposed Algorithms

2.1 Relationship between K-means and Spectral Clustering

As stated in [9], the problem of kernel k-means clustering is equivalent to spectral clustering. In particular, we note that the kernel mapping ϕ is actually determined by the data distribution itself. More specifically, ϕ is established by the affinity matrix, the degree matrix, and the Laplacian matrix. All the above matrices can be computed given enough data.

From experiments, it is observed that, given sufficiently large data set S, the output of the kernel mapping of a sample, i.e., the feature vector $v_i = \phi(s_i)$, $s_i \in S$, is continuous and nearly deterministic (with little noise). Thus, we propose that the mapping ϕ is also deterministic for each data set. Accordingly, a neural network can be employed to approximate ϕ. Once we have the approximation $\widetilde{\phi} \approx \phi$, the on-line k-means algorithm [8] can be modified directly into an on-line kernel k-means algorithm.

Therefore, our proposed on-line spectral clustering has two steps. Different from traditional unsupervised learning algorithms, first we will have a data set S for training the neural network $\widetilde{\phi}$. Second, each time we see a data point y from stream $\mathcal{Y}$, we process it by the kernel mapping ϕ, i.e., let $z = \widetilde{\phi}(y)$. The rest follows the similar procedure as in the on-line k-means algorithm.

2.2 Pseudo-code

Given a train set of points $S = \{s_1, \cdots, s_n\}$ in $\mathbb{R}^l$, we first build a neural network to approximate the kernel mapping $\phi(s_i)$: $\mathbb{R}^l \to \mathbb{R}^k$, where k is the number of clusters that we are looking for.

Algorithm 2 Labeling the data and train the neural network

1: Construct the affinity matrix $A \in \mathbb{R}^{n \times n}$, where $A_{ij} = \exp\left(-\frac{\|s_i - s_j\|^2}{2\sigma^2}\right)$ and $A_{ii} = 0$.
2: Build the diagonal degree matrix $D \in \mathbb{R}^{n \times n}$, where D_{ii} is the sum of A's i-th row.
3: Form the Laplacian matrix $L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$.
4: Compute the k largest eigenvectors of L as $x_1, x_2, \cdots, x_k$.
5: Create the matrix $X = [x_1, x_2, \cdots, x_k] \in \mathbb{R}^{n \times k}$. Each row of X is the label of the original data point, i.e., we can rewrite $X = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$, where $v_i = \phi(s_i)$.
6: Using the inputs s_i and the labels v_i, we can train a neural network $\widetilde{\phi}$ that approximates the mapping ϕ.

The reason why we replace the spectral clustering procedure with a neural network is that the calculation of eigenvectors of the Laplacian matrix is extremely time consuming on large dataset. To make the algorithm perform fast enough as an on-line algorithm, a neural network is trained to map data into a new feature space. Then, combined this neural network with the following modified on-line k-means, our new on-line spectral clustering algorithm is illustrated as Algorithms 3. $\mathcal{Y}$ is a stream of data, and $\mathcal{Y} \supset S$. k' is the initialization for the number of clusters, and $k' \leq k$. γ is a parameter inversely proportional to the probability that new clusters are created.

Algorithm 3 On-line spectral clustering algorithm

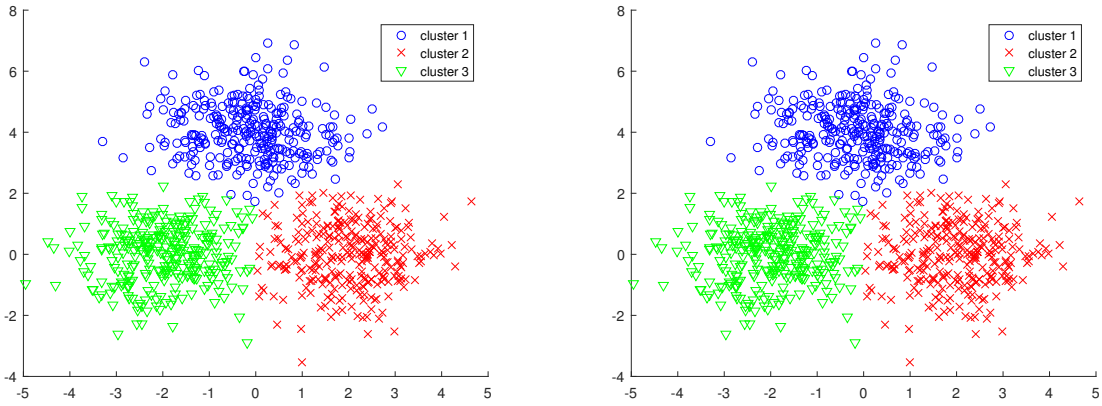
input: $\mathcal{Y}$, k', γ.
Choose first k' distinct vectors, $y_1, \cdots, y_{k'}$ in $\mathcal{Y}$, and use the learned kernel function $z_i = \widetilde{\phi}(y_i)$ for $i = 1, \cdots, k'$. Define the initial cluster centers $C \leftarrow z_i$ for $i = 1, \cdots, k'$.
$w^* \leftarrow \min_{z,z' \in C} \frac{\|z - z'\|^2}{2}$; $r \leftarrow 1$; $q_1 \leftarrow 0$; $f_1 = \frac{w^*}{k'}$.
for $y \in$ the remainder of $\mathcal{Y}$ **do**
    $z \leftarrow \widetilde{\phi}(y)$; $D(z, C) = \min_{c \in C} \|z - c\|$.
    with probability $p = \min\left(\frac{D^2(z,C)}{f_r}, 1\right)$
        $C \leftarrow C \cup \{z\}$; $q_r \leftarrow q_r + 1$.
    if $q_r \geq k'$ **then**
        $r \leftarrow r + 1$; $q_r \leftarrow 0$; $f_r \leftarrow \gamma \cdot f_{r-1}$.
    end if
    output: class $c_y = \arg\min_{c \in C} \|z - c\|$
end for

3 Empirical Evaluation

Since we are evaluating an on-line algorithm with the off-line versions, the elapse time and accuracy are presented. Due to the existence of randomness in the on-line algorithm, the performance will be influenced by the sequence of data. The accuracy listed below is the best we can achieve.

3.1 K-means Clustering: toy example

In this toy example, there are 900 data points needed to be clustered. We run both on-line and off-line k-means clustering algorithm. The elapse time for on-line and off-line k-means clustering algorithm are: 0.017891 seconds and 0.643246 seconds.



(a) result of on-line k-means clustering

(b) result of off-line k-means clustering

Figure 2: Clustering result of toy examples

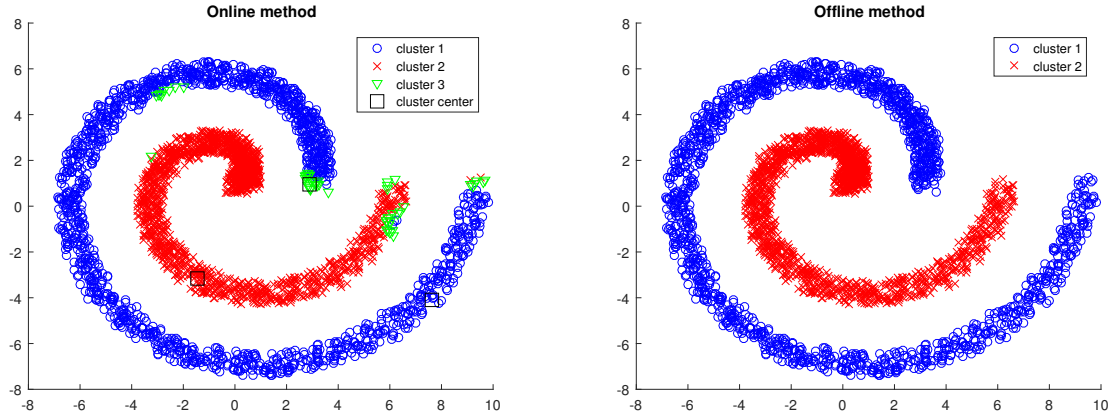
3.2 K-means Clustering: MNIST

We tested off-line and on-line K-means clustering on 18623 images extracted from MNIST dataset. The images are hand-written digits of 0,1 and 2. The elapse time for on-line and off-line k-means clustering algorithm are 0.765315 seconds and 1.324828 seconds. Clustering accuracies are 74% and 92%.

Considering the great performance of auto-encoders on dimensionality reduction. We also proposed to use an auto-encoder to pre-process the images and run the off-line K-means clustering afterwards. The auto-encoder has hidden layers 1000-500-250-50. To save running time it is only trained on 1000 images. According to our experiment this method can achieve more than 95 % accuracy. The great performance of our algorithm shows that using neural networks to pre-process high dimensional datasets could not only provide dimensionality reduction but also improve clustering performance.

3.3 Spectral Clustering: toy example

In this toy example, there are 3000 data points needed to be clustered. We run both on-line and off-line spectral clustering algorithm. The elapse time for on-line and off-line spectral clustering algorithm are: 0.723042 seconds and 26.801105 seconds.



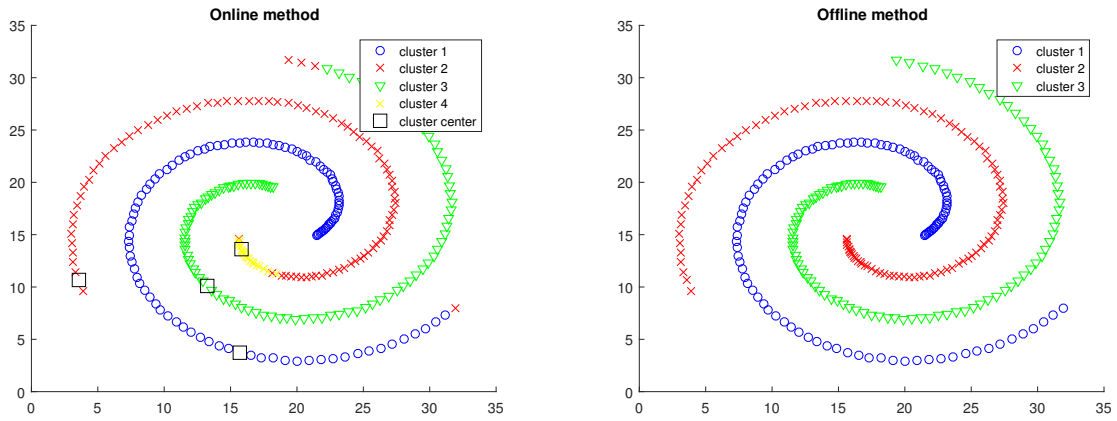
(a) result of on-line spectral clustering

(b) result of off-line spectral clustering

Figure 3: Clustering result of toy examples

3.4 Spectral Clustering: Spiral Dataset [10]

In the spiral dataset, there are 300 data points needed to be clustered. We run both on-line and off-line spectral clustering algorithm. The elapse time for on-line and off-line spectral clustering algorithm are: 0.193304 seconds and 0.222835 seconds.



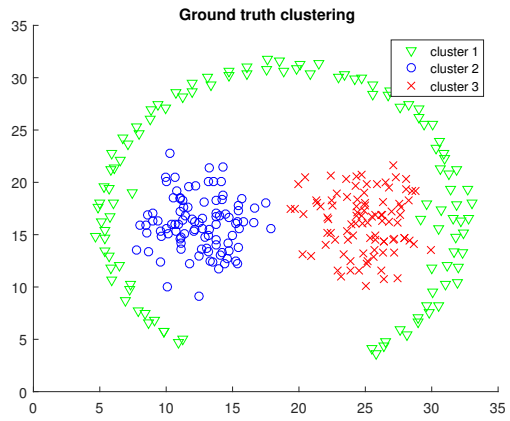
(a) result of on-line spectral clustering

(b) result of off-line spectral clustering

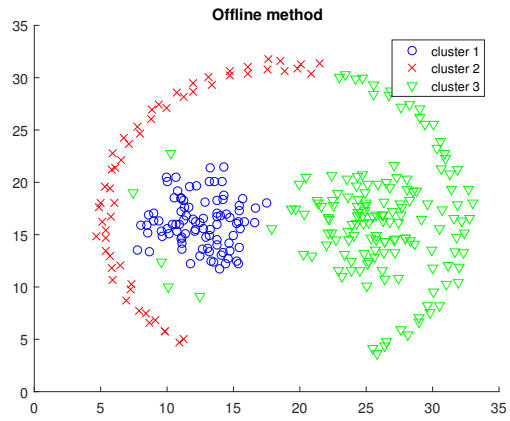
Figure 4: Clustering result of Spiral dataset

3.5 Spectral Clustering: Path-based Dataset [11]

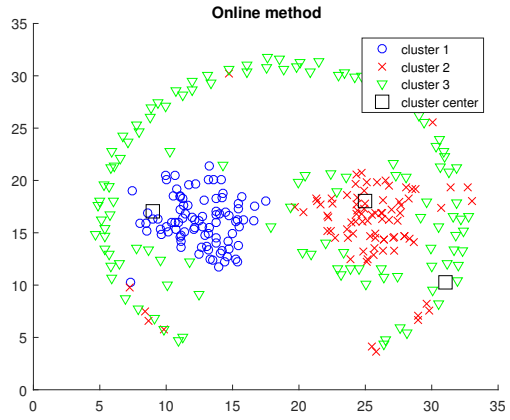
The result on this dataset is unexpected, since the on-line algorithm occasionally out-performs the off-line one for the first time.



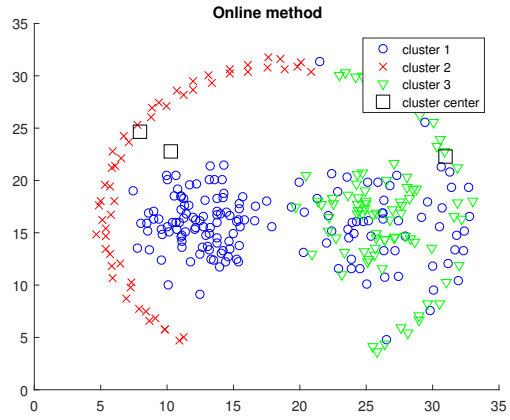
(a) ground-truth clustering



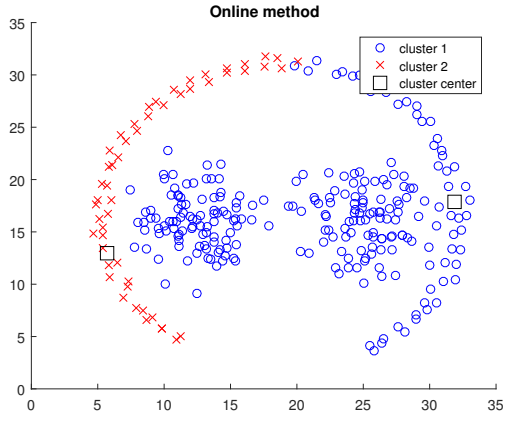
(b) result of off-line spectral clustering



(c) best performance of on-line spectral clustering



(d) average result of on-line spectral clustering



(e) bad performance of on-line spectral clustering

Figure 5: Clustering results of Path-based dataset

4 Conclusion and future work

In this paper we investigated both on-line and off-line clustering methods and the relationship between them. A novel method combining neural network and on-line spectral clustering is proposed. According to our experiment on several Datasets, our method achieves similar performance as off-line clustering methods. Hopefully, with the help of deep neural network we can achieve even better performance on high dimensional datasets. We leave this for the future work.

References

[1] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[2] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[3] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, Society for Industrial and Applied Mathematics, 2007.

[4] A. Aggarwal, A. Deshpande, and R. Kannan, "Adaptive sampling for k-means clustering," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 15–28, Springer, 2009.

[5] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii, "Scalable k-means++," *Proceedings of the VLDB Endowment*, vol. 5, no. 7, pp. 622–633, 2012.

[6] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "A local search approximation algorithm for k-means clustering," in *Proceedings of the eighteenth annual symposium on Computational geometry*, pp. 10–18, ACM, 2002.

[7] S. Yoo, H. Huang, and S. P. Kasiviswanathan, "Streaming spectral clustering," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pp. 637–648, May 2016.

[8] E. Liberty, R. Sriharsha, and M. Sviridenko, "An algorithm for online k-means clustering," in *Proceedings of the Eighteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pp. 81–89, 2016.

[9] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: Spectral clustering and normalized cuts," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 551–556, 2004.

[10] P. F. et al, "Clustering datasets," 2015. Available at https://cs.joensuu.fi/sipu/datasets/.

[11] H. Chang and D.-Y. Yeung, "Robust path-based spectral clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 191 – 203, 2008.