On-line Clustering: K-means and Spectral Clustering

{ Mengzhe Huang, Shenghe Xu, Liyang Sun, Siyu Zhou and Shuaiyu Liang }

NYU TANDON SCHOOL OF ENGINEERING

Objectives

We investigate two popular clustering algorithms:

1. K-means;
2. Spectral clustering.

Propose an **on-line spectral clustering** algorithm, combined with on-line k-means method [1].

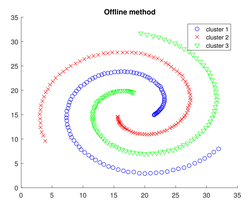
On-line Algorithms

On-line algorithms [1] aim to **handle** the situation of **large, even endless, data stream**. Also, they intend to **efficiently solve the curse of dimensionality** existing in traditional off-line clustering methods, such as clustering for images and multimedia from the web.

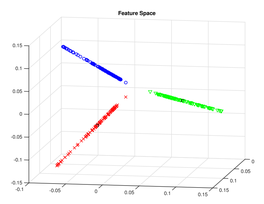
Spectral Clustering

We can view spectral clustering as a **map** from original data points to an eigen-space or a feature space.

Fig. 3 shows 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 3: Spectral clustering of spiral dataset

References

[1] E. Liberty, R. Sriharsha, and M. Sviridenko. An algorithm for online k-means clustering. 2016.

[2] H. Chang and D.-Y. Yeung. Robust path-based spectral clustering. *Pattern Recognition*, 2008.

On-line Spectral Clustering

Given a train set of points $S = \{s_1, \cdots, s_n\}$ in $\mathbb{R}^l$, we 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 1 Labeling the data and train the neural network

1: Construct the affinity matrix $A \in \mathbb{R}^{n\times n}$, the degree matrix $D \in \mathbb{R}^{n\times n}$ and the Laplacian matrix $L = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$.
2: Compute the k largest eigenvectors of L as $x_1, x_2, \cdots, x_k$. 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., $X = \left[v_1^T, \cdots, v_n^T\right]^T$, where $v_i = \phi(s_i)$.
3: Using the inputs s_i and the labels v_i, we can train a neural network $\widetilde{\phi}$ that approximates the mapping ϕ.

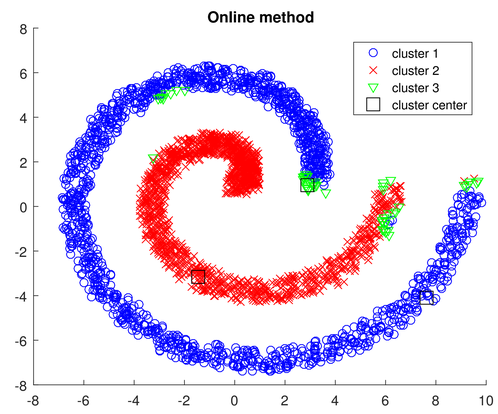
Given the trained neural network $\widetilde{\phi}$, $\mathcal{Y}$ is a stream of data. k' is the initial number of clusters, and $k' \le k$. γ is a parameter inversely proportional to the probability that new clusters are created.

Algorithm 2 On-line spectral clustering algorithm

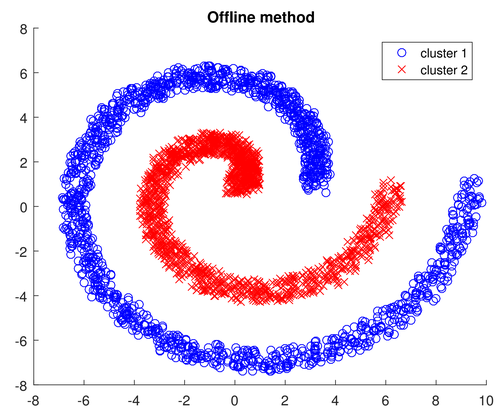
input: $\mathcal{Y}$, k', γ.
Choose first k' distinct vectors, $y_1, \cdots, y_{k'}$ in $\mathcal{Y}$, and use the learned kernel mapping $z_i = \widetilde{\phi}(y_i)$ for $i = 1, \cdots, k'$. Define the initial cluster centers $C \leftarrow z_i$ for $i = 1, \cdots, k'$.
$w^* \leftarrow \min_{z,z'\in C} \frac{\|z - z'\|^2}{2}$; $r \leftarrow 1$; $q_1 \leftarrow 0$; $f_1 = w^*/k'$.
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 $q_r \ge k'$ **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

Empirical Evaluation 1

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



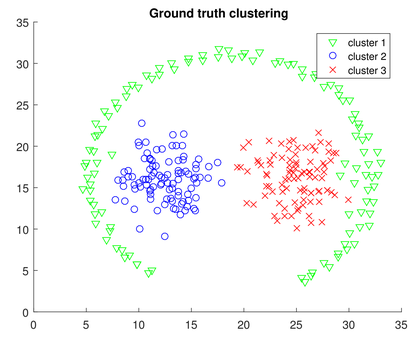
(a) result of on-line spectral clustering



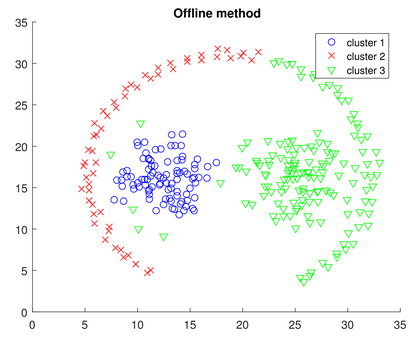
(b) result of off-line spectral clustering

Figure 1: Clustering result of toy examples

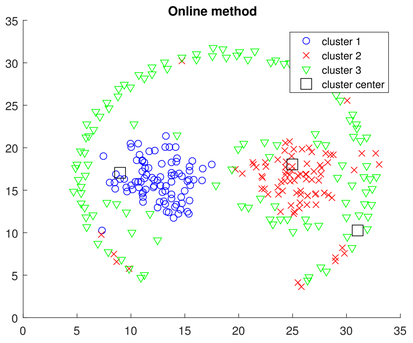
Empirical Evaluation 2 [2]



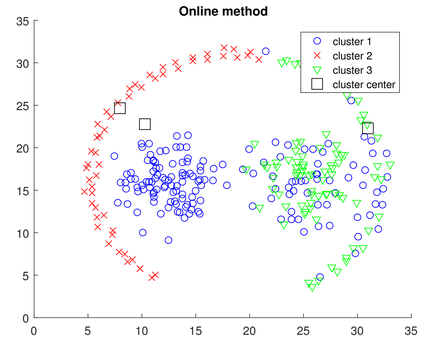
(a) ground-truth clustering



(b) result of off-line spectral clustering



(c) best performance of on-line method



(d) average result of on-line method