Subspace Kernel Spectral Clustering: SKSC

Chieh Wu, Armin Moharrer

February 18, 2021

1 Introduction

One of the most important problems in unsupervised learning is clustering. The most well-known techniques for this task are k-means and Gaussian mixture model, which have been applied to many applications successfully. However, they have the disadvantage that they make strong assumptions about the shape of the clusters. For example, in cases where the clusters are not linearly separable, these methods perform poorly.

A more robust approach is spectral clustering [1]. Spectral clustering does not make any prior assumption on the shape of the clusters. Instead, the input data are represented as a graph: each vertex shows a point in the data set, and a weight is assigned to each edge in the graph, which indicates the relation between the corresponding two points. The basic idea is to partition the graph, such that, the total sum of weights corresponding to the edges between the clusters is minimized [2]. The relaxation of this problem reduces to finding the spectral embedding. This is done by finding the most dominant eigenvectors of a Laplacian matrix, and then, we find the predicted labels by running the $k-$means algorithm on this spectral embedding [3].

Spectral clustering has the disadvantage that it is adversely sensitive to presence of noisy dimensions [4]. To deal with this problem, it has been proposed to project the original data on a low-dimensional subspace, and then, running the spectral clustering algorithm on the low-dimensional data [5]. This presents two main challenges. First, a projector matrix has to be found in order to project the data on a low-dimensional linear subspace. For identification purposes the columns of this projector matrix should be orthonormal. This adds a non-convex constraint, which makes the resulting optimization problem hard. Moreover, we also have to find the dimension of the lower dimensional subspace. Niu, Dy, and Jordan propose an algorithm which dynamically increases the rank of the low-dimensional subspace, and iteratively finds the spectral labeling and projector matrix [5]. This method is undesirably slow, as it takes a long time to converge. In addition, at each iteration the optimization problem for finding the projector matrix is non-convex, which means that for the gradient-based method the global optimum is not guaranteed.

Our main contribution are that first, we do not treat the non-convex constraint explicitly, but when when finding the projector matrix we do it in such

a way that the constraint is met. Second, we propose a new formulation for the problem, in which the dimension of the lower subspace is found automatically. We run our algorithm against k-means, Gaussian Mixture Model, and spectral clustering on both synthetic and real datasets, and show that our method has a superior performance.

2 Dimensionality Reduction and Spectral Clustering

In this section we initially explain the general formulation of spectral clustering in more detail, then we explain the idea of dimension reduction of [5], and introduce our formulation.

2.1 Spectral Clsutering

Assume that N, $d-$dimensional data points $x_i \in \mathbb{R}^d, i \in \{1, \dots, N\}$ are given, we represent data as a graph $G = \{V, E\}$, where $V = \{v_1, \dots, v_N\}$ is the set of vertices, such that, a vertex v_i shows the point x_i. E is the set of edges, which shows the correlation between data. Each edge e_{ij} is associated with a similarity measure $k_{ij} \geq 0$. We represented the similarities as a matrix $K \in \mathbb{R}^{N \times N}$, which is usually given by a kernel function, e.g.,

$$k_{ij} = K(x_i, x_j).$$

Now, the goal is to cluster the data, the vertices, such that the total sum of weights between the clusters, which shows the correlation between them, is minimized. In general, this problem is hard because there are $O(2^N)$ ways to cluster the data.

The relaxation of this problem is written as:

$$\min_{U^\top U = I} \operatorname{Tr}(U^\top L U),$$

where $L \in \mathbb{R}^{N \times N}$ is a Laplacian matrix, $U \in \mathbb{R}^{N \times k}$ is the spectral embedding, and k is the number of clusters. There are many ways to define the Laplacian matrix. One spectral clustering method called the normalized cuts algorithm defines it as [6]:

$$L = I - D^{-\frac{1}{2}} K D^{-\frac{1}{2}},$$

where D is a diagonal matrix called the degree matrix, and its diagonal elements are:

$$d_{ii} = \sum_{j=1}^{N} k_{ij}.$$

After finding the spectral embedding U we find the labels by running $k-$means algorithm on it.

2.2 Dimensionality Reduction Spectral Clustering

As we mentioned in the introduction, in order to deal with the unfavorable effects of noisy dimensions, we project the data on a low-dimensional linear subspace. We follow the work of Niu, Dy, and Jordan [5]. We introduce a projector matrix $W \in \mathbb{R}^{d\times q}$, then the similarities between points are computed by applying the kernel function on the projected data, i.e, $k_{ij} = K(W^\top x_i, W^\top x_j)$. In this paper we only consider the linear kernel, so the K matrix is given as $XWW^\top X^\top$. For better identification purposes we require the columns of W, which can be viewed as the atoms of the linear subspace, to be orthonormal to each other.

We formulate the problem in such a way that the dimension of the lower space is found automatically. That is besides finding the optimum spectral embedding, we also find the lowest dimensional subspace possible. We do this by minimizing the rank of the matrix $WW^\top$. Putting it all together we formulate the problem as:

$$\text{Minimize } \operatorname{Rank}\left(WW^\top\right) - \lambda \operatorname{Tr}(U^\top HXWW^\top X^\top HU) \tag{1a}$$

$$\text{Subj. to } W^\top W = I, \tag{1b}$$

$$U^\top U = I, \tag{1c}$$

where $H \in \mathbb{R}^{N\times N}$, is the centering matrix, and λ is a tuning parameter. In the next section we explain how this parameter is set.

3 Optimization Algorithm

In this section we introduce our SKPC algorithm for solving the proposed formulation of spectral clustering (1).

3.1 The Relaxation of Rank

Problem (1) is computationally hard to solve because of the rank function in the objective. Therefore, we replace it with the log-det function, which is a good heuristic for rank function [7]. Now, the problem can be written as:

$$\min \ \operatorname{logdet}\left(A + \sigma I\right) - \lambda \operatorname{Tr}(U^\top HXAX^\top HU), \tag{2a}$$

where we have represented $WW^\top$ by A matrix, and we have omitted the non-convex constraints; however, as we will see later in this section we find W and U in such a way that the constraint is met. Also, here we are adding σI in the argument of log-det to make sure that the matrix is positive definite and avoid a trivial minus infinity answer. Note that the log-det function is concave function, so (2) is not a convex optimization problem.

3.2 Our Algorithm: SKPC

Now we describe our proposed SKPC method for solving Problem (2). First, we initialize U and A, such that, they are feasible. Then, we iteratively solve

Algorithm 1 SKPC

input: X, σ, W^0, U^0, ITERS, λ^0
for $t \in \{1, \ldots, \text{ITERS}\}$ **do**
  Find U_{t+1} by setting its columns to the eigenvectors of $HXA^kX^\top H$ corresponding to the k smallest eigenvalues.
  Find $A_{t+1} = W_{t+1}W_{t+1}^\top$, where W_{t+1} is found by setting its columns to the eigenvectors of Φ^t corresponding to the non-positive eigenvalues.
  Update $\lambda^{k+1} = \frac{\text{logdet}\left(A_{t+1}+\sigma I\right)}{\text{Tr}({U_{t+1}}^\top HXA_{k+1}X^\top HU_{t+1})}$.
end for
Find the labels $f \in \{0,1\}^N$, by running $k-$means algorithm on the rows of U matrix.

for U and A, keeping one variable constant at each step. More specifically an iteration t of the algorithm is as follows:

- The problem with respect to U_t is given as:

$$\min_U \text{Tr}\left(U^\top HXA_{t-1}X^\top HU\right),$$

  this is simply solved by setting the columns of U equal to the k smallest eigenvectors of $HXA_{t-1}X^\top H$. It is immediately seen that $U_t^\top U_t = I$.

- When solving for A we further approximate the log-det term by its first-degree Taylor expansion around its current value, A_{t-1}. Also we factor A as $WW^\top$. This results in the following formulation:

$$\min \text{Tr}\left(W_t^\top \Phi_t W_t\right),$$

  where

$$\Phi_t = (A_{t-1} + \sigma I)^{-1} - \lambda_t X^\top HU_tU_t^\top HX.$$

  As a result, W_t is found by setting its columns to those eigenvectors of Φ_t, which correspond to its negative eigenvalues. We see that by finding W_t in this way the orthonormality constraint is satisfied. Moreover, the dimension of the lower subspace q equals to the number of negative eigenvalues of Φ_t.

- We set the tuning parameter at each iteration to $\frac{\text{logdet}\left(A_t+\sigma I\right)}{\text{Tr}(U_t^\top HXAX^\top HU_t)}$, this is to make sure that both of the terms in (2a), have roughly the same magnitude, so that none of them is dominating the other.

After a pre-specified number of iterations, we find a spectral embedding U and a projector matrix W, then we find the labels by running $k-means$ algorithm on U, treating each row of it as a datapoint.

4 Experiments

In this section we present our experimental results on both synthetic and real data. For each experiment we run our algorithm against $k-$means, Gaussian mixture model, and spectral clustering. In the experiments we use the normalized mutual information between the predicted labels and ground-truth labels to measure the accuracy. More specifically it is defined as

$$NMI = \frac{I(X,Y)}{H(Y)},$$

where X is the predicted labels, Y is the ground-truth labels, $I(X,Y)$ is the mutual information and $H(Y)$ is the entropy of Y.

4.1 Synthetic Data

Gaussian Mixture Data: This dataset comprises of 60 samples, with 4 dimensions. The first 2 dimensions are generated from a mixture of two Gaussian distributions, and the other 2 dimensions are noise. The number of clusters k is 2. The results is shown in Figure 1: this figure shows the 2 dimensions, which contain the noise-free data.

Moon-shaped Clusters: This dataset is consisted of 200 samples and 5 dimensions, such that, 2 of dimensions contain information, and other 3 are noise. The non-noise dimensions of data comprise two moon-shaped clusters, see Figure 2. As we see the clusters are not linearly separable, so as we see in Figure 2, Gaussian mixture model and $k-means$ perform poorly. Spectral clustering performs better, but our algorithm beats it because it projects the data on a lower dimensional subspace and suppress the disrupting effects of noisy dimensions.

4.2 Real Data

Facial Identity: This dataset comprises 624 images of human faces. Each datapoint has 27 features. The goal is to cluster the dataset based on the identity of humans. The measured mutual information is given in Figure 3a.

Facial Poses: This is the same dataset as the Facial Identity dataset, but the goal is to cluster the data to 4 different poses. This is much harder problem, so as we see in Figurepose, all of the algorithms perform poorly. However, our algorithm is considerably more accurate than others.

Breast Cancer: This is the Breast Cancer Dataset form UCI Repository. It comprises 286 samples with 9 features. The number of clusters k is 2. We see the results in Figure 3c.

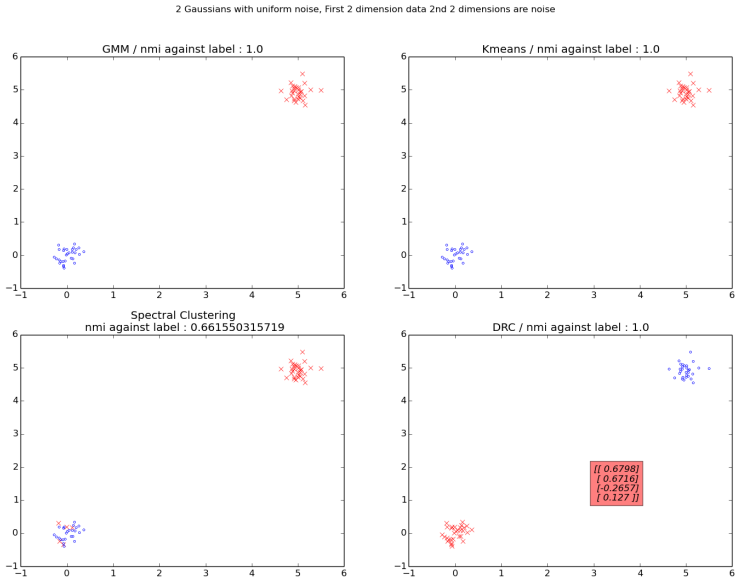


Figure 1: The figure shows the predicted clusters by the algorithms for the synthetic data generated form a mixture of two Gaussian distributions. As we see the clusters are linearly separable, and all of the methods have a decent performance, as expected.

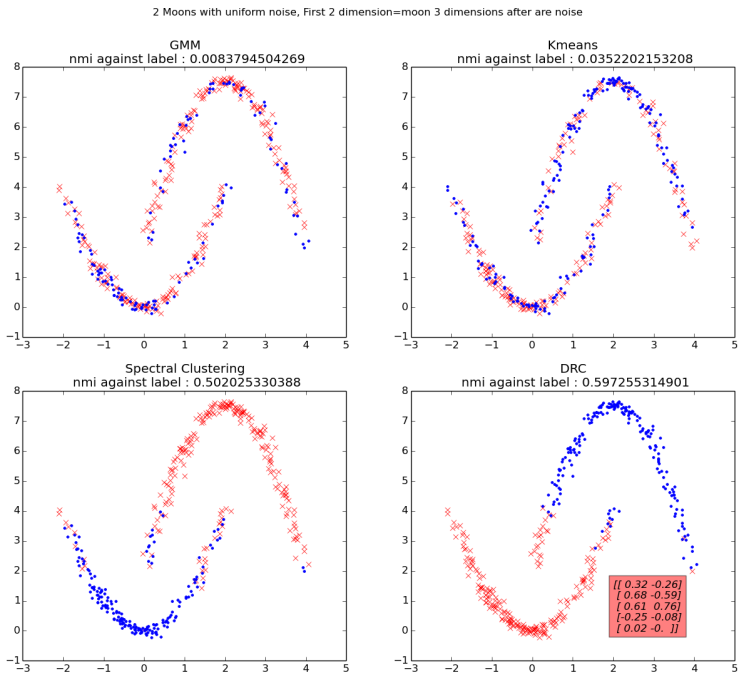
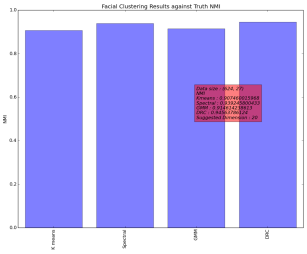
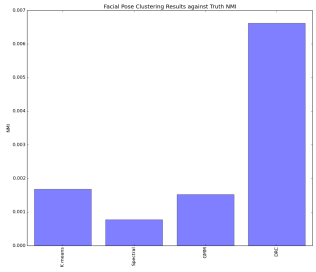


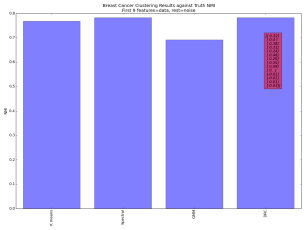
Figure 2: The figure shows the predicted clusters by the algorithms for the synthetic moon-shaped data. As we see the clusters are not linearly separable, so $k-$means and Gaussian mixture model perform poorly. Spectral clustering also gives less accurate results compared to our method, because it is sensitive to noisy dimensions.



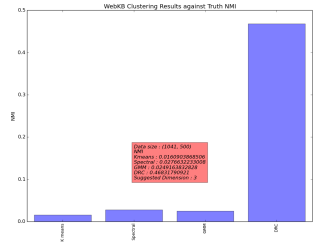
(a) The figure shows the facial clustering results based on the identity.



(b) The figure shows the facial clustering results based on the pose.



(c) The figure shows the results for breast cancer dataset. This is a binary clustering problem.



(d) The figure shows the results of clustering for the Web KB dataset.

Web KB: This data comprises of 1041 webpages from 4 universities. Each webpage has 500 features. The number of clusters k is 4, corresponding to the 4 universities. The result is seen in Figure 3d.

References

[1] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[2] Marina Meila and Jianbo Shi. Learning segmentation by random walks. In *NIPS*, volume 14, 2000.

[3] Zhihua Zhang, Michael I Jordan, et al. Multiway spectral clustering: A margin-based perspective. *Statistical Science*, 23(3):383–403, 2008.

[4] Francis R Bach and Michael I Jordan. Learning spectral clustering, with application to speech separation. *Journal of Machine Learning Research*, 7(Oct):1963–2001, 2006.

[5] Donglin Niu, Jennifer G Dy, and Michael I Jordan. Dimensionality reduction for spectral clustering. In *AISTATS*, pages 552–560, 2011.

[6] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[7] Maryam Fazel, Haitham Hindi, and Stephen P Boyd. Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2156–2162. IEEE, 2003.