Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Large-Scale Multi-View Spectral Clustering via Bipartite Graph

Yeqing Li, Feiping Nie, Heng Huang, Junzhou Huang*
University of Texas at Arlington, Arlington, TX, 76019, USA
{yeqing.li@mavs.uta.edu, feipingnie@gmail.com, heng@uta.edu, jzhuang@uta.edu}

Abstract

In this paper, we address the problem of large-scale multi-view spectral clustering. In many real-world applications, data can be represented in various heterogeneous features or views. Different views often provide different aspects of information that are complementary to each other. Several previous methods of clustering have demonstrated that better accuracy can be achieved using integrated information of all the views than just using each view individually. One important class of such methods is multi-view spectral clustering, which is based on graph Laplacian. However, existing methods are not applicable to large-scale problem for their high computational complexity. To this end, we propose a novel large-scale multi-view spectral clustering approach based on the bipartite graph. Our method uses local manifold fusion to integrate heterogeneous features. To improve efficiency, we approximate the similarity graphs using bipartite graphs. Furthermore, we show that our method can be easily extended to handle the out-of-sample problem. Extensive experimental results on five benchmark datasets demonstrate the effectiveness and efficiency of the proposed method, where our method runs up to nearly 3000 times faster than the state-of-the-art methods.

Introduction

Clustering multi-view data is an important problem. In many real-world datasets, data are naturally represented by different features or views. This is due to the fact that data may be collected from different sources or be represented by different kind of features for different tasks. For example, documents can be written in different languages; gene can be measured by different techniques, e.g. gene expression, Single-nucleotide polymorphism (SNP), methylation; images can be described by different features like Gabor (Oliva and Torralba 2001), HoG (Dalal and Triggs 2005), GIST (Oliva and Torralba 2001), LBP (Ojala, Pietikainen, and Maenpaa 2002). Different features capture different aspects of data and can be complementary to each other. Therefore, it is critical for learning algorithm to integrate these heterogeneous features to improve its accuracy and robustness. In this paper, we focus on one specific unsupervised learning task, i.e., multi-view spectral clustering.

Recently, spectral clustering (SC) is drawing more and more attention because of its effectiveness (Shi and Malik 2000; Von Luxburg 2007; Ng et al. 2002; Zelnik-Manor and Perona 2004; Nie, Wang, and Huang 2014; Chang et al. 2015). However, the growth of the scale of data has rendered the multi-view clustering problem more challenging. None of the existing methods is applicable on large-scale multi-view data. In general, SC methods usually involve two time consuming steps. The first step is to construct the affinity graph and the second step is to compute the eigen-decomposition. The first step usually takes $O(n^2d)$ time while the second step takes $O(Kn^2)$ time, where n is the number of data points, d is the dimension of features and K is the number of clusters. Many works have been proposed to accelerate SC algorithm (Fowlkes et al. 2004), (Shinnou and Sasaki 2008), (Sakai and Imiya 2009), (Yan, Huang, and Jordan 2009), (Chen et al. 2011), (Chen and Cai 2011). These methods reply on various off-the-shelf projection or sampling methods (Bingham and Mannila 2001; Li et al. 2014; Li, Chen, and Huang 2014) to reduce the complexity of graph construction or eigen-decomposition. However, they only discuss the situation of handling single view data, which limits their usage. There are also SC methods that deal with multi-view data, such as (Kumar, Rai, and Daume 2011), (Cai et al. 2011). These methods try to model the multi-view clustering problem as solving local and global optimization among different views. Although they have achieved better accuracy than single-view SC methods, they are more computationally expensive due to the fact that they require iterations to reach consensus of different views or large-scale matrix inversion.

Another drawback of SC methods is that they usually do not provide natural extension to handle the out-of-sample problem (Nie et al. 2011; Bengio et al. 2004). To address this problem, several methods have been proposed, e.g. (Passerini, Pontil, and Frasconi 2004; Fowlkes et al. 2004; Alzate and Suykens 2010; Bengio et al. 2004; Nie et al. 2011). They either rely on approximation of eigenfuncions (Fowlkes et al. 2004; Bengio et al. 2004) or data projection such as error correcting output code (ECOC) method

*Corresponding author: Junzhou Huang. Email: jzhuang@uta.edu. This work was partially supported by NSF IIS-1423056, CMMI-1434401, CNS-1405985.

(Dietterich and Bakiri 1995; Passerini, Pontil, and Frasconi 2004) or regression model (Nie et al. 2011). None of them address the out-of-sample problem in setting involved heterogeneous features.

In this paper, we proposed a multi-view spectral clustering method that is able to deal with large-scale data. Our method is inspired by the large-scale semi-supervised learning algorithm proposed in (Liu, He, and Chang 2010). First, we generate consensus m salient points for all views. Then we construct bipartite graph between raw data points and these salient points. These generated points play an important role in capturing the manifold of the original views. Then, the graph of all the views are combined together using a local manifold fusion method. Finally, we run spectral clustering on the resulting fused graph. There are several benefits of our method: **First**, manifold fusion preserves the manifold structure of all the views; **Second**, the construction of the bipartite graph is very efficient; **Third**, by exploring the special structure of the bipartite graph, spectral analysis on it is also very efficient; **Fourth**, our method also output cluster indicator of the salient points, which enables us to handle the out-of-sample problem efficiently. Additionally, we have conducted extensive experiments on five ' benchmark data sets, which demonstrate the effectiveness and efficiency of our proposed method comparing to the state-of-the-art methods.

The remainder of this paper is organized as follows: we first introduce basic notations and concepts of spectral clustering in Section 2. In Section 3, details of our proposed large-scale multi-view spectral clustering method is presented. All the experimental results are shown in Section 4. Finally, we conclude our work in Section 5.

Background and Notations

In this section, we will briefly introduce the notations and the spectral clustering framework. Let $X = [x_1, \ldots, x_n]^T \in \mathbb{R}^{n\times d}$ denote the data matrix, where n is the number of data points and d is dimension of features. Each data point $x_i \in \mathbb{R}^d$ belongs to one of K classes $C = \{c_1, \ldots, c_K\}$. Given the whole dataset X, each data point is represented as a vertex on the affinity graph and each edge represents the affinity relation of one pair of vertexes. In practice, the k-NN graph are usually used. Specifically, x_i and x_j are connected if at least one of them is among the k nearest neighbours of the other in the given measured (usually Euclidean distance). The weight of the edge between x_i and x_j is defined as:

$$w_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|}{2\sigma^2}\right), & \text{if } x_i \text{ and } x_j \text{ are connected} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where σ is the bandwidth parameter. Note that we use Gaussian Kernel for example, this method is also applicable to other types of kernel. Thus, $W = \{w_{ij}\} \in \mathbb{R}^{n\times n}, \forall i, j \in 1, \ldots, n$ is the adjacent matrix of the graph and it is a symmetric undirected graph. Let $D \in \mathbb{R}^{n\times n}$ be the degree matrix whose i-th diagonal element is $d_{ii} = \sum_{j=1}^{n} w_{ij}$. Let L denote the normalized graph Laplacian matrix, then it is defined as:

$$L = I - D^{-1/2} W D^{-1/2} \tag{2}$$

The objective function of the normalized spectral clustering (Ng et al. 2002) is defined as:

$$\min_{G^T G = I} \operatorname{Tr}(G^T L G), \tag{3}$$

where $G \in \mathbb{R}^{n\times K}$ is the class indicator matrix of all data. The solution of G in Eq. (3) is the K smallest eigen vectors of L.

Multi-view Spectral Clustering Revisit

For multi-view data, let V be the number of views and $X^{(1)}, \ldots, X^{(V)}$ be the data matrix of each view, where $X^{(v)} \in \mathbb{R}^{n\times d^{(v)}}$ for $v \in 1 \ldots, V$ and $d^{(v)}$ is the feature dimension of the v-th view. Let $L^{(1)}, \ldots, L^{(V)} \in \mathbb{R}^{n\times n}$ denote the normalized Laplacian matrices of each view, respectively. Two important questions that are needed to be answered by multi-view approaches are how to reach consensus of the results and how to express the relationship of all the views. There are several forms for the multi-view spectral clustering (Kumar, Rai, and Daume 2011; Cai et al. 2011). We use the following form:

$$\min_{G^T G = I, a^{(v)}} J_1(G, a^{(v)}) = \sum_{v=1}^{V} (a^{(v)})^r \operatorname{Tr}(G^T L^{(v)} G), \quad \text{s.t. } \sum_{v=1}^{V} a^{(v)} = 1, a^{(v)} \geq 0, \tag{4}$$

where $a^{(v)}$ is the non-negative normalized weight factor for the v-th view and r is a scalar to control the distribution of different weights among different views. Here, we try to find a consensus result G among all the views. This unique consensus eliminates the need for computing the local results for each view and the computation cost of communicating back and forth between local results and the global result e.g. (Kumar, Rai, and Daume 2011). To further explain the inter-view relation, we rewrite Eq. (4) as:

$$\min_{G^T G = I, a^{(v)}} J_2(G, a^{(v)}) = \operatorname{Tr}(G^T L G), \quad \text{s.t. } \sum_{v=1}^{V} a^{(v)} = 1, a^{(v)} \geq 0, \tag{5}$$

where $L = \sum_{v=1}^{V} (a^{(v)})^r L^{(v)}$. Here, L can be regarded as local manifold fusion of all the views.

Equation (5) can be solve by iterative optimization techniques. However, to construct the graphs for all the views and to solve the equation is time consuming. The computational complexity is about $O(TKn^2 + \sum_{v=1} Vnd_v^2)$, where T is the number of iterations.

Methodology

In this section, we present an efficient approximation algorithm that can be applied to large-scale graph construction. Then, an efficient clustering algorithm is proposed for large-scale multi-view spectral clustering. Finally, we extended our method to handle the out-of-sample problem.

Large-Scale Graph Construction

In order to reduce the computational cost of multi-view spectral clustering, we introduce a fast approximation algorithm. The idea is to use a small set of data points $U = [U_1, \ldots, U_m] \in \mathbb{R}^{m \times d}$ to capture the manifold structure, where each u_k is called a salient point. Then a bipartite graph is constructed between the raw data points and the salient points. By utilizing the structure of the bipartite graph, the graph construction and spectral analysis can be perform very efficient.

The salient points can be chosen by random sampling from raw data points or using lightweight clustering methods such as k-means. We find that the salient points generated by k-means have stronger representation power compared to sampling ones, where fewer points are needed for the same level of performance. However, in multi-view data, different views will generate different salient points if we run k-means independently on each view, which makes manifold fusion impossible. Therefore, we generate salient points on concatenated all the features and then separate resulting points into different views. This process can generate uniform salient points for different views, which will simplified the process of clustering.

With the generated points, the k-NN graph is constructed between the raw data and the salient points. We further constrain that connections are only allowed between raw data point and salient point. This constraint results in a bipartite graph between raw data X and salient points U. And the weight of each edge is defined as

$$Z_{ij} = \frac{K(x_i, u_j)}{\sum_{k \in \Phi_i} K(x_i, u_k)}, \forall j \in \Phi_i, \tag{6}$$

where $K()$ is a given kernel function (e.g. Gaussian Kernel in Eq. (1)), $\Phi_i \subset \{1 \ldots m\}$ denotes the indexes of s nearest neighbours of x_i in U.

For the v-th view, the affinity matrix becomes $W^{(v)} = \begin{bmatrix} 0 & Z^{(v)} \\ {Z^{(v)}}^T & 0 \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}$. The degree matrix becomes $D^{(v)} = \begin{bmatrix} D_r^{(v)} & 0 \\ 0 & D_c^{(v)} \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}$, where D_r is a diagonal matrix of whose diagonal elements are row sums of Z and D_c is a diagonal matrix of whose diagonal elements are column sums of Z. Since Z is by definition row normalized, we have $D_r = I_n$, where I_n is the n by n identity matrix. The construction of the graph is extremely efficient since now we only need to consider $O(mn)$ distances. However, directly computing eigenvectors of L in Eq. (4) is still time consuming. Therefore, we need to transform the problem to utilize the structure of bipartite graph.

Multi-view Spectral Clustering Algorithm

By utilize the bipartite graph, we can obtain an algorithm that can optimize the cluster indicator of raw data points and salient points simultaneously. We name this algorithm **Multi-view Spectral Clustering (MVSC)**. We first propose our alternative optimization framework for solving Eq. (5). With all the $a^{(v)}$ are initialized to be equal, i.e. $a^{(v)} = 1/V$ for $v \in 1 \ldots V$, we solve Eq. (5) in iterations of two following steps.

First, we fix $a^{(v)}$ and then solve G, where the objective function become:

$$\min_{G^T G = I} J_2(G) = \mathrm{Tr}(G^T L G), \tag{7}$$

which is equivalent to original spectral clustering. The solution of G is obtained by compute K smallest eigenvectors of L.

Second, we fix G and then solve $a^{(v)}$. Let $h^{(v)} = \mathrm{Tr}(G^T L^{(v)} G)$, then the Eq. (4) can be rewritten as:

$$\min_{a^{(v)}} \sum_{v=1}^{V} (a^{(v)})^r h^{(v)}, \text{ s.t. } \sum_{v=1}^{V} a^{(v)} = 1, a^{(v)} \geq 0, \tag{8}$$

Thus, using method of Lagrange multiplier, Eq. (8) becomes:

$$\min_{a^{(v)}} \sum_{v=1}^{V} (a^{(v)})^r h^{(v)} - \beta(\sum_{v=1}^{V} a^{(v)} - 1), \tag{9}$$

where β is the Lagrange multiplier. With simple algebraic manipulations, we get

$$a^{(v)} = \frac{\left(r(h^{(v)}\right)^{\frac{1}{1-r}}}{\sum_{v=1}^{V} \left(r h^{(v)}\right)^{\frac{1}{1-r}}}. \tag{10}$$

The first sub-problem (Eq. (7)) tries to minimize $J_2(G) = \mathrm{Tr}(G^T L G)$, which takes $O(cn^2)$ for general case. Fortunately, we can reduce the complexity by using the following theorem.

Theorem 1. *: Solving $J_2(G) = \mathrm{Tr}(G^T L G)$ is equivalent to compute the singular vectors of Z corresponding to K largest singular values.*

Proof. Let $S^{(v)} = (D^{(v)})^{-1/2} W^{(v)} (D^{(v)})^{-1/2}$. The objective function $J_2(G)$ can be rewritten as

$$\begin{aligned} J_2(G) &= \mathrm{Tr}(G^T L G) \\ &= \mathrm{Tr}(G^T \sum_{v=1}^{V} (a^{(v)})^r L^{(v)} G) \\ &= \mathrm{Tr}(G^T \left(\sum_{v=1}^{V} (a^{(v)})^r (I - S^{(v)}) \right) G) \\ &= \mathrm{Tr}(\sum_{v=1}^{V} (a^{(v)})^r G^T G \\ &\quad - G^T \left(\sum_{v=1}^{V} (a^{(v)})^r S^{(v)} \right) G) \\ &= n \sum_{v=1}^{V} (a^{(v)})^r - \mathrm{Tr}(G^T S G), \end{aligned} \tag{11}$$

where $S = \sum_{v=1}^{V} (a^{(v)})^r S^{(v)}$. Then, minimizing J_2 with respect to G is equivalent to the following equation

$$\max_{G^T G = I} \mathrm{Tr}(G^T S G) \tag{12}$$

The solution of G is the eigenvectors corresponding to the K largest eigenvalues. We can use the structure of S to transform the problem of computing the eigenvectors of S to that of computing the eigenvectors of Z. Let $G = [G_X^T, G_U^T]^T$, where G_X, G_U are rows corresponding to raw data and salient points respectively. Therefore, the objective function in Eq. (12) becomes

$$\begin{aligned}\mathrm{Tr}(G^T S G) &= \mathrm{Tr}\left(\begin{bmatrix}G_X\\G_U\end{bmatrix}^T S \begin{bmatrix}G_X\\G_U\end{bmatrix}\right)\\ &= \mathrm{Tr}\left(\begin{bmatrix}G_X\\G_U\end{bmatrix}^T \begin{bmatrix}0 & \hat{Z}\\(\hat{Z})^T & 0\end{bmatrix} \begin{bmatrix}G_X\\G_U\end{bmatrix}\right)\\ &= \mathrm{Tr}\left(2G_X^T \hat{Z} G_U\right), \end{aligned} \tag{13}$$

where $\hat{Z} = \sum_{v=1}^{V} (a^{(v)})^r \hat{Z}^{(v)}$ and $\hat{Z}^{(v)} = (D_r^{(v)})^{-\frac{1}{2}} Z^{(v)} (D_c^{(v)})^{-\frac{1}{2}} = Z^{(v)} (D_c^{(v)})^{-\frac{1}{2}}$. Thus, solving Eq. (12) is equivalent to computing the left and right singular vectors corresponding to the K largest singular values of $\hat{Z}$

$$svd(\hat{Z}) = G_X \Sigma G_U^T, \tag{14}$$

where $svd()$ is the Singular Value Decomposition (SVD) operator, $\Sigma = diag(\sigma_1, \ldots, \sigma_K)$ and $\sigma_1 \geq \sigma_2, \ldots, \sigma_K \geq 0$ are the singular values of $\hat{Z}$. □

With Theorem 1, we can solve the whole problem very efficiently. The whole algorithm is summarized in Alg. 1.

Computational analysis. The proposed Multi-view Spectral Clustering (MVSC) consists of three stages: 1) generating salient points using k-means, 2) constructing graph Z and 3) optimization by iteratively solving the clustering problem. The first stage takes $O(t_1 nmd)$ time, where t_1 is the number of iterations for running k-means and $d = \sum_{v=1}^{V} d^{(v)}$. The second stage takes $O(nmd)$ to construct the graph Z, while constructing a normal k-NN graph of n vertexes takes $O(n^2 d)$. The third stage takes $O(t_2 nm^2)$, where t_2 is the number of iterations. Note that the optimization stage is much faster than clustering on a normal n by n graph, which takes $O(Kn^2)$ time. So the overall time complexity is approximately $O(t_1 nmd + t_2 nm^2)$. Since $m, d \ll n$, this is nearly linear to n. The computational cost is summarized in Table 1.

Table 1: Summary of computational complexity.

Stages	1 and 2	3	Total
Normal graph	$O(n^2 d)$	$O(Kn^2)$	$O(n^2 d + Kn^2)$
Bipartite graph	$O(t_1 nmd)$	$O(t_2 nm^2)$	$O(t_1 nmd + t_2 nm^2)$

Convergence analysis. The original problem Eq. (4) is not a joint convex problem of $a^{(v)}$ and G. Hence, there is no guarantee for obtaining a global solution. Since we divide the original problem into two sub-problems and each of them is convex problem. The proposed method will converge to a local solution. In all our experiments, the process always converges in less than 10 iterations.

Parameter r. Another advantage of our approach is using the parameter r, which controls the fusion weights of all the views by only one parameter. Some previous methods just simply assume equal weights (Kumar, Rai, and Daume 2011) or tuning one parameter for each view (Liu et al. 2013). The effect of r ranges from assigning equal weights to all views when $r = \infty$ to assigning all the weights to one best view when $r = 1$. By tuning r between $(1, \infty)$ we can reach a balance between all the views.

Algorithm 1 Multi-view Spectral Clustering (MVSC)

Input: Data matrix of all views $X^{(v)} \in \mathbb{R}^{n \times d^{(v)}}$ for $v \in 1 \ldots V$, Number of classes K, Number of salient points m, parameter r.
Output: Cluster labels Y of each data points, all salient points U and cluster labels of all salient points.
Generate m salient points using k-means on concatenate features;
Compute affinity matrix $Z^{(v)}$ of each view.
Compute Laplacian $L^{(v)}$ of each view;
Initialize $a^{(v)} = 1/K$;
repeat
    Compute G by using Eq. (14);
    Update $a^{(v)}$ by using Eq. (8);
until Converges.
Treat each row of G as new representation of each data point and compute the clustering labels Y by using k-means algorithm.

Out-of-sample Problem

In general, spectral clustering methods only work on the training data. Most methods do not provide clear extension to deal with out-of-sample points (a.k.a. test data). In contrast, our method can be easily extended to handle test data. Recall that when carrying out clustering on training data, we also get the feature vectors and clustering labels for the salient points. Therefore, we simple find the k nearest neighbours of test data among salient points and propagate the labels to the test data. The k-NN algorithm can be done in $O(md)$ computational cost for each data point. Hence, p test data points can be clustered in $O(pmd)$ computational cost. This computational cost is far lower than carried out k-NN on the training data ($O(pnd)$).

Experiment

In this section, we conduct several experiments to evaluate the performance of the proposed methods on five benchmarks datasets. These datasets are summarized in Table 2. All our experiments are conducted on a desktop computer with a 3.4GHz Intel Core i7 CPU and 12GB RAM, MatLab 2012a (64bit).

Data Set Description

Handwritten (HW)[1] is a dataset of handwritten digits of 0 to 9 from UCI machine learning repository (Frank, Asun-

[1] https://archive.ics.uci.edu/ml/datasets/Multiple+Features

cion, and others 2010). It consists of 2000 data points. We use all the 6 published features including 76 Fourier coefficients of the character shapes (FOU), 216 profile correlations (FAC), 64 Karhunen-love coefficients (KAR), 240 pixel averages in 2×3 windows (Pix), 47 Zernike moment (ZER) and 6 morphological (MOR) features.

Caltech-101 (Fei-Fei, Fergus, and Perona 2007) image data set consists of 101 categories of images for object recognition problem. We follow previous work (Dueck and Frey 2007) and select the widely used 7 classes, i.e. Face, Motorbikes, Dolla-Bill, Garfield, Snoopy, Stop-Sign and Windsor-Chair and get 1474 images, which we called **Caltech7 (Cal7)**. We also select a larger set named **Caltech20 (Cal20)** which contains totally 2386 images of 20 classes: Face, Leopards, Motorbikes, Binocular, Brain, Camera, Car-Side, Dolla-Bill, Ferry, Garfield, Hedgehog, Pagoda, Rhino, Snoopy, Stapler, Stop-Sign, Water-Lilly, Windsor-Chair, Wrench and Yin-yang. Five features are extracted from all the images: i.e. 48 dimension Gabor feature, 40 dimension wavelet moments (WM), 254 dimension CENTRIST feature, 1984 dimension HOG feature, 512 dimension GIST feature, and 928 dimension LBP feature.

Reuters[2] consists of documents that are written in five different languages and their translations. All the documents are categorized in to 6 classes. We use the subset that are written in English all their translations in all the other 4 languages (French, German, Spanish and Italian).

NUS-WIDE-Object (NUS) (Chua et al. July 8 10 2009) is a dataset for object recognition which consists of 30000 images in 31 classes. We use 5 features provided by the website [3], i.e. 65 dimension color Histogram (CH), 226 dimension color moments (CM), 145 dimension color correlation (CORR), 74 dimension edge distribution and 129 wavelet texture.

Animal with attributes (**AWA**)[4] is a data set of animal images. It consists of 50 kinds of animals described in 6 features. We randomly sample 80 images for each class and get 4000 images in total. All the published features are used: Color Histogram (CQ, dim 2688), Local Self-Similarity (LSS, dim 2000), Pyramid HOG (PHOG, dim 252), SIFT (dim 2000), Color SIFT (RGSIFT, dim 2000) and SURF (dim 2000).

Clutering Evaluation

In this subsection, we first evaluate the capability of the proposed multi-view clustering method on 5 datasets: HW, Caltech7, Caltech20, Reuters and NUS. We compare the proposed methods with three other state-of-art approaches as stated bellow:

Single view Spectral Clustering (**SC**): Running spectral clustering on each single view (Ng et al. 2002).

Feature Concatenation Spectral Clustering (**ConSC**): Concatenating features of all the views and run spectral clustering on the resulted feature (Kumar, Rai, and Daume 2011).

Co-regularized Spectral Clustering (**CoregSC**): one of the state-of-the-art multi-view spectral clustering method proposed in (Kumar, Rai, and Daume 2011).

Multi-Modal Spectral Clustering (**MMSC**): another recent multi-view clustering method proposed in (Cai et al. 2011).

Multi-view Spectral Clustering (**MVSC**): this is the proposed method in Alg. (1).

For fair comparison, we download the source code from the authors' website and follow their experimental setting and the parameter tuning steps in their paper. And we use Gaussian kernel for all the experiments except for the Reuters dataset, where we use linear kernel. We search the parameter r in logarithm form ($\log_{10} r$ from 0.1 to 2 with step size 0.2. We also set $m = 400$ and construct 8-nearest-neighbour graph between raw All the experiments are repeated for 10 times and average results are reported. For the experimental results, we report three metrics (Manning, Raghavan, and Schütze 2008): **mean purity**, **mean mutual information (NMI)** and **mean running time**.

Table 3: Clustering purity comparison on all data sets. "OM" means "Out-of-memory error" while running the experiment.

Data set	HW	Cal7	Cal20	Reuters	NUS
SC(1)	75.12%	79.22%	68.13%	53.10%	15.98%
SC(2)	75.44%	79.85%	68.13%	54.86%	16.13%
SC(3)	76.39%	79.36%	69.09%	56.92%	15.78%
SC(4)	73.47%	80.56%	67.01%	53.82%	16.29%
SC(5)	75.84%	80.48%	67.99%	56.79%	16.44%
SC(6)	78.89%	79.97%	66.90%	-	-
ConcatSC	59.33%	77.96%	60.33%	**56.70%**	26.81%
CoRegSC	**82.23%**	**83.71%**	**76.11%**	55.23%	**26.49%**
MMSC	75.84%	84.47%	69.04%	39.01%	OM
Proposed	**84.41%**	**84.66%**	**74.06%**	**57.73%**	**28.21%**

Table 4: Clustering NMI comparison on all data sets. "OM" means "Out-of-memory error" while running the experiment.

Data set	HW	Cal7	Cal20	Reuters	NUS
SC(1)	0.7589	0.4189	0.4842	0.3099	0.0398
SC(2)	0.7549	0.4239	0.4813	0.3033	0.0419
SC(3)	0.7556	0.4217	0.4848	0.3039	0.0403
SC(4)	0.7547	0.4220	0.4816	0.3123	0.0432
SC(5)	0.7576	0.4206	0.4830	0.3078	0.0429
SC(6)	0.7577	0.4190	0.4830	-	-
ConcatSC	0.5795	0.2734	0.3590	0.3228	0.1421
CoRegSC	**0.8358**	0.5253	**0.6107**	**0.3261**	**0.1428**
MMSC	0.7920	**0.5638**	**0.5938**	0.1335	OM
Proposed	**0.8324**	**0.5586**	0.5698	**0.3567**	**0.1493**

Table 3 and Table 4 show clustering purity and NMI respectively, while Table 5 shows the running time of all the methods. In general, the multi-view methods can achieve better results than the single view algorithms. Additionally, our proposed method MVSC constantly outperforms the single view methods and achieves comparable or even better results than the other multi-view methods. For running time

[2] https://archive.ics.uci.edu/ml/datasets.html

[3] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

[4] http://attributes.kyb.tuebingen.mpg.de/

Table 2: Summary of the multi-view datasets used in our experiments.

No.	HW	Caltech7/20	Reuters	NUS	AWA
1	Pix(240)	Gabor(48)	English(21531)	CH(65)	CQ(2688)
2	Fou(76)	WM(40)	France(24892)	CM(226)	LSS(2000)
3	Fac(216)	CENTRIST(254)	German(34251)	CORR(145)	PHOG(252)
4	ZER(47)	HOG(1984)	Italian(15506)	EDH(74)	SIFT(2000)
5	KAR(64)	GIST(512)	Spanish(11547)	WT(129)	RGSIFT(2000)
6	MOR(6)	LBP(928)	-	-	SURF(2000)
# of data	2000	1474/2386	18758	26315	4000
# of classes	20	7/20	6	31	50

Table 5: Running time comparison on all data sets (seconds). "OM" means "Out-of-memory error" while running the experiment.

Data set	HW	Cal7	Cal20	Reuters	NUS
SC(1)	1.74	10.94	29.18	556.98	852.07
SC(2)	1.54	10.30	29.27	443.92	580.36
SC(3)	1.53	10.20	29.32	422.91	478.88
SC(4)	1.53	10.16	29.15	354.68	527.62
SC(5)	1.58	10.32	29.32	307.62	633.01
SC(6)	1.53	10.24	29.33	-	-
ConcatSC	2.20	11.00	26.19	556.73	2172.90
CoRegSC	16.42	61.78	180.65	7074.17	56327.26
MMSC	6.13	27.62	80.25	14556.13	OM
Proposed	**0.84**	**1.21**	**2.26**	**135.48**	**19.34**

comparison in Table 5, the proposed method is up to several orders of magnitude faster than the baseline methods. The gap is even larger in the large datasets. The other benefits of the proposed method is low space complexity. In fact, nearly all the baseline methods raise out-of-memory exception when number of data points are more than 40,000 while the proposed method can easily handle more than 100,000 samples at once.

Out-of-sample Problem

In this subsection, we consider the out-of-sample problem. Experiments are conducted on AWA dataset. Five fold cross-validation is used and we report the mean purity, the mean NMI and the mean testing time. At each fold, $1/5$ of the data are used as in-sample clustering like that in the previous subsection and the other $4/5$ are used for the out-of-sample test. For the out-of-sample test, the data and the estimated cluster labels of the in-sample clustering are used as training data for the model. Here we compare two situations: **1)** training model with the whole raw in-sample data; **2)** training model with the generated salient points. Two kinds of models are trained in both situation: **Linear Regression (LR)** and **Nearest Neighbour (1NN)**. The corresponding models trained on the salient points are called **Salient 1 Nearest Neighbour (Sa1NN)** and **Salient Linear Regression (SaLR)** respectively. We compare the proposed method with a third baseline method Spectral Embedded Clustering (SEC) (Nie et al. 2011). Since the data have several views, we train and apply models on each view and use simple voting scheme to decide the final cluster label for each testing sample.

Table 6: Results of out-of-sample test om AWA.

Method	1NN	LR	SEC	Sa1NN	SaLR
Purity	8.13%	7.22%	7.79%	8.37%	7.31%
NMI	0.1395	0.1124	0.1252	0.1490	0.1120
Time (s)	972.53	0.97	0.99	436.45	0.94

Table 6 shows the testing performance of all the methods. The first two rows of the table are purity and NMI, respectively, while the third row shows the testing time. We can observe that the purity of the salient-point-based models are comparable or even better than the raw-data-based models. The testing time of Sa1NN is much less than the 1NN model. This is reasonable since the computational complexity of 1NN algorithm is proportional to the number of training samples. All these results demonstrate that we can achieve comparable performance using models trained only on the salient points.

Conclusion

In this paper, we propose a novel large-scale multi-view spectral clustering method based on bipartite graph, named MVSC. Given a multi-view data set with n data points, MVSC select m uniform salient points among all the views to represent the manifold structures of all the features. For each view, one sub-bipartite-graph is constructed between the raw data points and the generated salient points. We use local manifold fusion to generate a fused bipartite graph to integrate information of all the sub-graph. By exploring the structure of the bipartite graph, the clustering process can be accelerated significantly. The computational complexity is close to linear to the number of data points. For the clustering results, we not only obtain cluster labels for the training data but also cluster labels for the salient points. The later information has been used to handle the out-of-sample problem in low computational cost. Extensive experiments on five benchmark data sets demonstrate that our proposed method is up to several orders of magnitude faster than the state-of-the-art methods, while preserving the comparable or even better accuracy.

References

Alzate, C., and Suykens, J. A. 2010. Multiway spectral clustering with out-of-sample extensions through weighted

kernel pca. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32(2):335–347.

Bengio, Y.; Paiement, J.-F.; Vincent, P.; Delalleau, O.; Le Roux, N.; and Ouimet, M. 2004. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. *Advances in neural information processing systems* 16:177–184.

Bingham, E., and Mannila, H. 2001. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 245–250. ACM.

Cai, X.; Nie, F.; Huang, H.; and Kamangar, F. 2011. Heterogeneous image feature integration via multi-modal spectral clustering. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 1977–1984. IEEE.

Chang, X.; Nie, F.; Ma, Z.; and Yang, Y. 2015. A convex formulation for spectral shrunk clustering. In *AAAI*.

Chen, X., and Cai, D. 2011. Large scale spectral clustering with landmark-based representation. In *AAAI*.

Chen, W.-Y.; Song, Y.; Bai, H.; Lin, C.-J.; and Chang, E. Y. 2011. Parallel spectral clustering in distributed systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33(3):568–586.

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y.-T. July 8-10, 2009. Nus-wide: A real-world web image database from national university of singapore. In *Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09)*.

Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, 886–893. IEEE.

Dietterich, T. G., and Bakiri, G. 1995. Solving multiclass learning problems via error-correcting output codes. *arXiv preprint cs/9501101*.

Dueck, D., and Frey, B. J. 2007. Non-metric affinity propagation for unsupervised image categorization. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 1–8. IEEE.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2007. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding* 106(1):59–70.

Fowlkes, C.; Belongie, S.; Chung, F.; and Malik, J. 2004. Spectral grouping using the nystrom method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 26(2):214–225.

Frank, A.; Asuncion, A.; et al. 2010. Uci machine learning repository.

Kumar, A.; Rai, P.; and Daume, H. 2011. Co-regularized multi-view spectral clustering. In *Advances in Neural Information Processing Systems*, 1413–1421.

Li, Y.; Chen, C.; Liu, W.; and Huang, J. 2014. Subselective quantization for large-scale image search. In *AAAI*.

Li, Y.; Chen, C.; and Huang, J. 2014. Transformation-invariant collaborative sub-representation. In *Proceedings of the 22nd International Conference on Pattern Recognition*.

Liu, J.; Wang, C.; Gao, J.; and Han, J. 2013. Multi-view clustering via joint nonnegative matrix factorization. In *Proc. of SDM*, volume 13, 252–260. SIAM.

Liu, W.; He, J.; and Chang, S.-F. 2010. Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 679–686.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

Ng, A. Y.; Jordan, M. I.; Weiss, Y.; et al. 2002. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems* 2:849–856.

Nie, F.; Zeng, Z.; Tsang, I. W.; Xu, D.; and Zhang, C. 2011. Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering. *Neural Networks, IEEE Transactions on* 22(11):1796–1808.

Nie, F.; Wang, X.; and Huang, H. 2014. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 977–986. ACM.

Ojala, T.; Pietikainen, M.; and Maenpaa, T. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24(7):971–987.

Oliva, A., and Torralba, A. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision* 42(3):145–175.

Passerini, A.; Pontil, M.; and Frasconi, P. 2004. New results on error correcting output codes of kernel machines. *Neural Networks, IEEE Transactions on* 15(1):45–54.

Sakai, T., and Imiya, A. 2009. Fast spectral clustering with random projection and sampling. In *Machine Learning and Data Mining in Pattern Recognition*. Springer. 372–384.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22(8):888–905.

Shinnou, H., and Sasaki, M. 2008. Spectral clustering for a large data set by reducing the similarity matrix size. In *LREC*.

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

Yan, D.; Huang, L.; and Jordan, M. I. 2009. Fast approximate spectral clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 907–916. ACM.

Zelnik-Manor, L., and Perona, P. 2004. Self-tuning spectral clustering. In *Advances in neural information processing systems*, 1601–1608.