Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Spectral Clustering with Brainstorming Process for Multi-View Data

Jeong-Woo Son, Junkey Jeon, Alex Lee, Sun-Joong Kim
{jwson, jkjeon, lhjalex, kimsj}@etri.re.kr
Smart Platform Research Department, Electronics and Telecommunications Research Institute
218 Gajeong-ro, Yuseong-gu, Deajeon, Korea

Abstract

Clustering tasks often requires multiple views rather than a single view to correctly reflect diverse characteristics of the cluster boundaries. The cluster boundaries estimated using a single view are incorrect in general, and those incorrect estimation should be compensated by helps of other views. If each view is independent to other views, incorrect estimations will be mostly revised as the number of views grow. However, on the contrary, as the number of views grow it is almost impossible to avoid dependencies among views, and such dependencies often delude correct estimations. Thus, dependencies among views should be carefully considered in multi-view clustering. This paper proposes a new spectral clustering method to deal with multi-view data and dependencies among views. The proposed method is motivated by the *brainstorming process*. In the brainstorming process, an instance is regarded as an agenda to be discussed, while each view is considered as a brainstormer. Through the discussion step in the brainstorming process, a brainstormer iteratively suggests their opinions and accepts others' different opinions. To compensate the biases caused by information sharing between brainstormers with dependent opinions, those having independent opinions are more encouraged to discuss together than those with dependent opinions. The conclusion step makes a compromise by merging or concatenating all opinions. The clustering is finally done after the conclusion. Experimental results in three tasks show the effectiveness of the proposed method comparing with ordinary single and multi-view spectral clusterings.

Introduction

The spectral clustering is derived from a non-linear data embedding scheme, called the spectral embedding (Luxburg 2007). In the spectral embedding, instances are mapped onto the space spanned by eigenvectors obtained from the graph Laplacian of the data. The spectral embedding has an ability to distinguish instances sampled from non-convex clusters. This advantage makes the spectral clustering to be one of the most widely-used techniques among unsupervised machine learning techniques (Verma and Meila 2003).

The original spectral clustering has been modified and expanded to handle multi-view data (Feng et al. 2016; Xia et al. 2014). In various tasks such as video processing and text clustering, an instance is often represented with multiple aspects; for example, videos have images as well as audio features, while text documents can have several multi-lingual translations. Those multi-view representations reflect various characteristics of cluster boundaries. Several approaches have been proposed (Bickel and Scheffer 2004; Chaudhuri et al. 2009) to utilize complementary information in different views. Min-disagreement (Wang, Weng, and Yuan 2014) is a typical method that constructs a common vector for which all views agree, while information propagation (Kumar and Daumé III 2011) suggests a way to explicitly share complementary information in each view.

Clusters appeared in a real world often represented with multiple aspects, and to find them correctly, a human should use diverse viewpoints to check information from each viewpoint. multi-view data clustering aims to realize this situation. When multiple representations are constructed for data, multiple views have to be independent each other. Since they need to reflect different aspects of cluster boundaries. However, it is hard to avoid dependencies among views in real world problems, and in that case, multiple views often delude correct estimations. Thus, dependencies among views should be carefully considered in multi-view clustering.

This paper proposes a novel extension of spectral clustering for multi-view data; especially the proposed method is designed to handle data with three or more views. The proposed method forces the information of each view to be shared by using the *brainstorming process*. In the brainstorming process, views are regarded as brainstormers with different perspectives, while an instance is considered as an agenda. When an agenda and brainstormers are given, the brainstorming process starts to discuss about the agenda. Through the discussion on the agenda, a brainstormer iteratively suggests their opinions and accepts others' different opinions. The biases caused by information sharing between brainstormers with dependent opinions is compensated by encouraging to discuss brainstormers with independent opinions. While brainstormers with dependent opinions are prevented from discussions for opinion sharing. By encouraging information sharing between less dependent views, dilution of valuable information is prevented. After the discussion, we suggest two ways to draw a conclusion: merging and concatenation. Merging tries to make a compromise that maximally satisfies the all brainstormers, while concatenation just collect all the opinions.

We have evaluated the proposed method with three datasets: synthetic data with three views, Reuter multilingual data with five views, BBC data with four views. In all experiments, the proposed method achieves the best performances comparing with five ordinary spectral clustering.

Related Work

Spectral clustering is a unsupervised machine learning technique with a well-defined theoretical basis (Ng, Jordan, and Weiss 2002). Spectral clustering can reveal clusters of arbitrary forms from data by using a spectral embedding, a non-linear data embedding based on a graph Laplacian and its eigenvectors. As a way to utilize the strength of spectral clustering in various fields, there have been proposed a number of variations of spectral clustering for multi-view data (Li et al. 2015; Xia et al. 2014).

Previous works can be categorized mainly into two groups: min-disagreement and information sharing. Studies in min-disagreement aim to construct a single representation with which all views can maximally agree (Zhou and Burges 2007). For example, (Xia et al. 2010) proposed Multi-view Spectral Embedding (MSE) to merge graph Laplacians from multiple views by using a weighted summation. Since weights are estimated to minimize distances between graphs from multiple views, MSE can be considered as a min-disagreement as well. Co-Regularized Multi-view Spectral Clustering (CoRMSC) proposed by (Kumar, Rai, and Daumé III 2011) estimates unified eigenvectors for multi-view data. Unified eigenvectors are constructed by minimizing weighted distances to eigenvectors of each view. Min-disagreement is appropriate to reduce noise in each view by extracting common information in all multiple views. On the other hand, it expects that complementary information in views to be shared in constructing single representation implicitly.

Explicit preservation of complementary information can be found in information sharing. Co-EM based clustering (Bickel and Scheffer 2004) defines class labels as a latent variable for data. The latent variables are obtained by clustering data with one view in E-step. Then the latent variables are shared to the other view in M-step. Co-trained Multi-view Spectral Clustering (CMSC) (Kumar and Daumé III 2011) adopts a co-training approach which learn a classifier by sharing selected label information in multiple views. CMSC adopts indirect sharing of clustering results by using eigenvectors obtained in each view. In CMSC, a set of eigenvectors obtained from a view is passed to the other view. Each similarity matrix is mapped onto delivered eigenvectors to reflect the information from others. Then, a similarity matrix for a view is reconstructed by back-projection. Information sharing has an advantage to preserve the complementary characteristics in different views.

Most of multi-view spectral clustering methods basically assume the independence among views. However, the assumption can be easily broken in many real-world situations, since it is natural to have different dependencies among views when an instance is represented with three or more views. In this case, dependencies in views often contaminate the final result by ignoring unique information in each view.

Co-trained Multi-view Spectral Clustering

The proposed method is basically motivated from Co-trained Multi-view Spectral Clustering (CMSC) (Kumar and Daumé III 2011). Hence, this section describes CMSC briefly. CMSC adopts a co-training approach to share information among views. Let a data set $\mathcal{D}$ with n instances be given ($|\mathcal{D}| = n$). With two views v_1 and v_2, the i-th instance x_i is represented as $x_i = (x_{i,1}, x_{i,2})$. Thus, two similarity graph $K_1 \in \mathbb{R}^{n\times n}$ and $K_2 \in \mathbb{R}^{n\times n}$ are constructed with v_1 and v_2 respectively. The i-th column in K_1 represents the similarities between $x_{i,1}$ and others. The graph Laplacians L_1 and L_2 are given as $D_1^{-1/2} K_1 D_1^{-1/2}$ and $D_2^{-1/2} K_2 D_2^{-1/2}$, where D_v is a diagonal matrix with diagonal entries given by $D_{v,ii} = \sum_{j=1}^{n} K_{v,ij}$.

CMSC modifies the graph structure from one view by using the clustering result from the other view. Conceptually, the modification is performed to reduce similarities between nodes in different clusters and to amplify similarities between nodes belong to the same cluster. This modification is indirectly performed by using eigenvectors of L_1 and L_2.

The similarities described in the matrix K_1 is modified as follows. Let $U_2 \in \mathbb{R}^{n\times m}$ be the matrix obtained by collecting the first m eigenvectors of the graph Laplacian L_2, for some m much smaller than n. After that, columns of K_1 are projected into the column space of U_2, and then reconstructed back as similarity vectors using the back-projection. Similarly, K_2 is modified by the same procedure using m eigenvectors of L_1. This procedure can be simply described as

$$S_1 = \mathrm{sym}(U_2 U_2^T K_1),$$
$$S_2 = \mathrm{sym}(U_1 U_1^T K_2),$$

where S_1 and S_2 are reconstructed similarity graphs for v_1 and v_2 respectively, and $\mathrm{sym}(\cdot)$ denotes symmetrization of square matrices ($\mathrm{sym}(A) := \frac{A+A^T}{2}$). This modification process is performed iteratively; at the t-th iteration, the graph Laplacians L_1 and L_2 are constructed from S_1, S_2 at the previous step. That is,

$$S_1^{(t)} = \mathrm{sym}\left(U_2^{(t-1)} (U_2^{(t-1)})^T K_1\right),$$
$$S_2^{(t)} = \mathrm{sym}\left(U_1^{(t-1)} (U_1^{(t-1)})^T K_2\right),$$

where $S_1^{(t)}, S_2^{(t)}$ are the outputs at the t-th iteration, and $U_1^{(t)}, U_2^{(t)}$ are the matrices consisting of m eigenvectors of the graph Laplacians of $S_1^{(t)}, S_2^{(t)}$, respectively.

When there are three or more views, the reconstruction procedure just described is given as follows:

$$S_v^{(t)} = \mathrm{sym}\left(\sum_{i\neq v} \left(U_i^{(t)} (U_i^{(t-1)})^T\right) K_v\right).$$

As shown in this equation, all views have equal importance for each other, because they are assumed to be independent as usual in co-training. However, in most real-world tasks, this assumption can be easily broken and views have dependencies with different strengths.

Brainstorming Process for Spectral Clustering

A spectral clustering with the brainstorming process is designed to deal with three or more views with different dependencies. Let a data set $\mathcal{D}$ ($|\mathcal{D}| = n$) be given and each instance x_i in $\mathcal{D}$ is represented with respect to a set of views, $V = \{v_j\}_{j=1}^{|V|}$, $|V| \geq 3$. Information sharing among these views is modeled with a simple brainstorming process.

The brainstorming process is a face-to-face meeting generally held by humans to derive an elegant ideas from diverse perspectives. The proposed method adopts the central idea in the brainstorming process by regarding x_i as an agenda to be discussed, while V as brainstormers. When an agenda and brainstormers are prepared, the brainstorming process is started from the discussion.

Discussion in Brainstorming Process

For agenda x_i, all brainstormers $v_j \in V$ suggest their ideas based on their own perspective. These are the initial vector representations $x_{i,1}, x_{i,2}, \ldots, x_{i,|V|}$ of x_i. For a simple description, x_i is used to denote similarity vector between the i-th instance and others. After presenting all the opinions at the first stage, a brainstormer v_j tries to accept the others' ideas to improve his/her opinion.

In the discussion, v_j needs to pick valuable opinions from $V_{\neg j} := V \setminus \{v_j\}$. A strong dependency between v_j and $v_k \in V_{\neg j}$ means these two brainstormers have similar opinions. Thus, the information given by v_k should be meaningless to v_j, since v_k cannot change v_j. The brainstorming process in the proposed method measures the "value" of an opinion as how less it correlates with v_j.

The proposed method extracts opinions from $V_{\neg j}$ by treating it as a single virtual brainstormer. The construction of this virtual brainstormer is based on dependencies with v_j. More precisely, for $x_{i,j}$, the opinion of v_j for the agenda x_i, the opinion of the virtual brainstormer is defined as

$$x_{i,\neg j} = \sum_{k \neq j} w_{i,k} x_{i,k},$$

where $w_{i,k}$'s are parameters that need to be estimated.

By using a correlation coefficient, the weight vector $W_{i,\neg j} := (w_{i,k})_{k \neq j} \in \mathbb{R}^{|V|-1}$ is estimated to minimize the dependency between $x_{i,j}$ and $x_{i,\neg j}$; that is,

$$\begin{aligned} W^*_{i,\neg j} &= \underset{W_{i,\neg j} \in \mathbb{R}^{|V|-1}}{\arg\min} \langle x_{i,j} - \bar{x}_{i,j}, x_{i,\neg j} - \bar{x}_{i,\neg j} \rangle^2 \\ &\quad + C|W_{i,\neg j}|^2 \\ &= \underset{W_{i,\neg j} \in \mathbb{R}^{|V|-1}}{\arg\min} \langle b, AW_{i,\neg j} \rangle^2 + C|W_{i,\neg j}|^2, \end{aligned}$$

where $b = x_{i,j} - \bar{x}_{i,j}$, and $A = X_{i,\neg j} - \bar{X}_{i,\neg j}$. Here, $X_{i,\neg j} := \begin{bmatrix} x_{i,1} & \cdots & \widehat{x_{i,j}} & \cdots & x_{i,|V|} \end{bmatrix} \in \mathbb{R}^{n \times (|V|-1)}$ is the matrix formed by aggregating $x_{i,k}$'s except $x_{i,j}$, while $\bar{x}_{i,j}$, $\bar{X}_{i,\neg j}$ are sample averages of $x_{i,j}$, $X_{i,\neg j}$, respectively. The $C \geq 0$ is a user parameter for L^2-regularizer term $C|W_{i,\neg j}|^2$. We tested no-, L^1-, and L^2-regularizers for all experimental data and obtained the best performance with L^2-regularizer in all tasks. With L^2-regularizer, $W^*_{i,\neg j}$ tends to consider all views fairly and it can prevent propagation of incorrect information caused by noise.

Note that all the entries of $x_{i,j}$'s should be inside the interval $[0, 1]$, because they are similarities. The opinion $x_{i,\neg j}$ of the virtual brainstormer must have the same property. To keep this property, the above optimization problem is under the following two constraints:

$$\forall k \neq j,\ 0 \leq w_{i,k} \leq 1,$$
$$\sum_{k \neq j} w_{i,k} = 1.$$

After constructing $x_{i,\neg j}$ for all $x_i \in \mathcal{D}$ and $v_j \in V$, a pair of similarity graphs K_j and $K_{\neg j}$ can be obtained for all views. That is, for each brainstormer, corresponding virtual brainstormer who has the minimum dependency on him/her is constructed. The information propagation between a brainstormer and his/her virtual brainstormer is performed by using a method similar to that of CMSC. From $K_{\neg j}$, the graph Laplacian $L_{\neg j} = D_{\neg j}^{-1/2} K_{\neg j} D_{\neg j}^{-1/2}$ as well as its first m eigenvectors $U_{\neg j} \in \mathbb{R}^{n \times m}$ can be obtained. Then the adjusted similarity matrix is constructed as

$$S_j = \mathrm{sym}(U_{\neg j} U_{\neg j}^T K_j).$$

After the discussion is done, for each $v_j \in V$ the matrix S_j is obtained. Then the same procedure is repeated: for each $v_j \in V$, obtain the first m eigenvectors from the graph Laplacian of the matrix S_j, and then update S_j using the above equation. After iterating over this procedure several times, the conclusion is drawn from the final opinions $U_1, U_2, \ldots, U_{|V|}$.

Conclusion in Brainstorming Process

There are two ways to draw a conclusion: merging and concatenation. In general brainstorming process, after sufficiently long discussions, the chairman generates a conclusion. If there were enough discussions to share information among brainstormers, he/she tries to make a compromise for which all the brainstormers agree. This is the *merging*. The *concatenation* corresponds to the case when the chairman just pass all the opinions to the next stage.

The *merging* process is explained first. The i-th row of U_j can be considered as a non-linear transformation of x_i into the space spanned by v_j. Thus, we may consider $U_j U_j^T$ as a kind of similarity matrix. Then the compromised similarity matrix K^* can be written as

$$K^* = \sum_{j=1}^{|V|} U_j U_j^T \theta_j,$$

where $\theta_j = \mathrm{diag}(\theta_{1,j}, \cdots, \theta_{n,j}) \in \mathbb{R}^{n \times n}$ is a diagonal matrix consists of instance-wise weights for v_j, and it is estimated to minimize the total disagreement. Then the compromised opinion U^* can be obtained as

$$U^* = \underset{U \in \mathbb{R}^{n \times m}}{\arg\max}\, \mathrm{tr}(U^T L^* U), \ \text{ s.t. } U^T U = I, \tag{1}$$

where L^* is the graph Laplacian obtained from K^* and $\mathrm{tr}(\cdot)$ is the matrix trace; this is an instance of the ordinary spectral clustering with a single view data.

The weights $\Theta := (\theta_{i,j}) \in \mathbb{R}^{n\times|V|}$ are chosen to minimize the total disagreement, which is defined as $\mathrm{tr}\left(\sum_{j=1}^{|V|}(U^* - U_j)(U^* - U_j)^T\theta_j\right)$. With the same reason in the discussion step, taking this together with an L^2-regularization into consideration, the optimization problem (1) turns into:

$$\begin{aligned}
&(\Theta^*, U^*)\\
&= \underset{\Theta\in\mathbb{R}^{n\times|V|}, U\in\mathbb{R}^{n\times m}}{\arg\max} \quad \mathrm{tr}(U^T L^* U)\\
&\quad - \left(\mathrm{tr}\left(\sum_{j=1}^{|V|}(U - U_j)(U - U_j)^T\theta_j\right) + \frac{C|\Theta|^2}{2}\right)
\end{aligned}$$

with the constraints

$$U^T U = I,$$

$$\forall(i,j),\ 0 \le \theta_{i,j} \le 1, \tag{2}$$

$$\forall i,\ \sum_{j=1}^{|V|}\theta_{i,j} = 1. \tag{3}$$

This optimization problem can be solved by iterating over the following two steps after initializing $\theta_{i,j} = \frac{1}{|V|}$ for all i, j:

- Fix Θ and find U that maximizes $\mathrm{tr}(U^T L^* U)$ under the constraint $U^T U = I$.
- Fix U and find Θ that minimizes

$$\mathrm{tr}\left(\sum_{j=1}^{|V|}(U - U_j)(U - U_j)^T\theta_j\right) + \frac{C}{2}|\Theta|^2$$

  under the constraints (2) and (3).

The merging process is similar to the Co-Regularized Multi-view Spectral Clustering (CoRMSC) proposed by (Kumar, Rai, and Daumé III 2011). However, in CoRMSC, Θ^* is defined as view-wise weights not as instance-wise and is given manually.

The *concatenation* process just produce the final opinion $U^* \in \mathbb{R}^{n\times(m\times|V|)}$ by column-wise vector concatenation of $U_1, \cdots, U_{|V|}$. After constructing U^*, a clustering method such as k-means clustering is applied to determine actual clusters. All optimization problems considered in this paper are kinds of simple constrained quadratic optimizations. Thus, solutions of these optimizations can be easily obtained with ordinary quadratic programmings (Griva, Nash, and Sofer 2009).

Experiments

Experimental Setting

The evaluation of the proposed method is performed with three datasets: synthetic, Reuter multilingual, and BBC datasets. All the experiments are performed with the same setting in (Kumar and Daumé III 2011). Synthetic data is generated from two clusters. Each cluster is composed of three Gaussian components that means an instance is represented as three views. The first cluster is generated with means of $\mu_1 = (1,1)$, $\mu_2 = (1,2)$, and $\mu_3 = (1,1)$, while the second one is defined with means of $\mu_1 = (3,4)$, $\mu_2 = (2,2)$, and $\mu_3 = (3,3)$. Covariances of the first cluster are defined as

$$\Sigma_1 = \begin{pmatrix} 1.0 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 1.0 & -0.2 \\ -0.2 & 1.0 \end{pmatrix},$$

$$\Sigma_3 = \begin{pmatrix} 1.2 & 0.2 \\ 0.2 & 1.0 \end{pmatrix},$$

while those for the second cluster are

$$\Sigma_1 = \begin{pmatrix} 0.3 & 0.2 \\ 0.2 & 0.6 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 0.6 & 0.1 \\ 0.1 & 0.5 \end{pmatrix},$$

$$\Sigma_3 = \begin{pmatrix} 1.0 & 0.4 \\ 0.4 & 0.7 \end{pmatrix}.$$

For each cluster, five hundreds instances are randomly sampled ($n = 1,000$).

Table 1: Performances on the synthetic dataset

Method	ARI	MI	Hom	Com
Singleview	0.809	0.714	0.714	0.714
KernelSum	0.886	0.819	0.819	0.820
KernelProd	0.898	0.826	0.826	0.826
MSE	0.891	0.823	0.824	0.824
CMSC	0.917	0.853	0.853	0.853
CoMSC	0.899	0.817	0.817	0.817
SCB_Concat	0.933	0.877	0.877	0.877
SCB_Merge	0.913	0.852	0.852	0.852

Reuter multilingual dataset is composed of documents from six categories (Amini, Usunier, and Goutte 2009), and a document is written in English, French, German, Italian, and Spanish for the same content. Thus, the instance in Reuter multilingual data has totally five views. In experiments, we determine the clustering performances with three to five views (English, French, German + Italian + Spanish). For each cluster, two-hundred instances are randomly sampled ($n = 1,200$). The vector representation of an instance is constructed with Latent Semantic Analysis (Hofmann 1999) with 100-dimension. BBC dataset contains 2,225 news articles in five categories (Greene and Cunningham 2006). The dataset supports clustering with three and four views that are generated by segmenting documents into three and four fragments.

The proposed method is compared with five ordinary spectral clustering algorithms: **Singleview -** a spectral clustering with the most informative view, **KernelSum -** a spectral clustering with a similarity matrix based on a summation of similarities from all views, **KernelProd -** similar to KernelSum except that it uses a production to merge similarities, **MSE -** multi-view spectral embedding proposed by (Xia et al. 2010), this is a method on min-disagreement, **CMSC -** a multi-view spectral clustering described in Section , **CoMSC -** Co-Regularized Multi-view Spectral Clustering proposed by (Kumar, Rai, and Daumé III 2011), **SCB_Concat -** and **SCB_Merge -** the proposed method with the concatenation and merging for the discussion respectively.

Table 2: Performances on Reuter multilingual dataset

	Method	ARI	MI	Hom	Com	Comp. Time (sec)
Three views	Singleview	0.226 (0.011)	0.285 (0.011)	0.294 (0.011)	0.309 (0.017)	3.120
	KernelSum	0.245 (0.027)	0.302 (0.028)	0.310 (0.028)	0.331 (0.030)	3.146
	KernelProd	0.180 (0.004)	0.236 (0.003)	0.245 (0.003)	0.260 (0.003)	3.111
	MSE	0.127 (0.058)	0.194 (0.069)	0.205 (0.067)	0.280 (0.064)	3.139
	CMSC	0.159 (0.014)	0.232 (0.014)	0.242 (0.014)	0.275 (0.026)	32.834
	CoMSC	0.206 (0.034)	0.269 (0.039)	0.277 (0.038)	0.282 (0.037)	5.260
	SCB_Concat	0.307 (0.011)	0.359 (0.016)	0.367 (0.016)	0.375 (0.016)	171.050
	SCB_Merge	0.311 (0.015)	0.366 (0.015)	0.374 (0.015)	0.379 (0.015)	1227.426
Four views	Singleview	0.193 (0.017)	0.252 (0.027)	0.261 (0.027)	0.281 (0.032)	3.943
	KernelSum	0.198 (0.012)	0.264 (0.015)	0.272 (0.014)	0.294 (0.019)	3.957
	KernelProd	0.171 (0.010)	0.235 (0.015)	0.244 (0.015)	0.270 (0.017)	3.903
	MSE	0.124 (0.046)	0.191 (0.050)	0.201 (0.050)	0.253 (0.061)	4.070
	CMSC	0.172 (0.046)	0.249 (0.052)	0.258 (0.051)	0.291 (0.048)	40.415
	CoMSC	0.222 (0.013)	0.294 (0.019)	0.303 (0.019)	0.312 (0.022)	5.166
	SCB_Concat	0.302 (0.015)	0.355 (0.017)	0.363 (0.017)	0.369 (0.018)	229.773
	SCB_Merge	0.309 (0.009)	0.369 (0.015)	0.376 (0.014)	0.381 (0.016)	1386.856
Five views	Singleview	0.188 (0.020)	0.246 (0.028)	0.255 (0.028)	0.273 (0.030)	4.845
	KernelSum	0.228 (0.035)	0.296 (0.032)	0.304 (0.032)	0.324 (0.038)	4.884
	KernelProd	0.124 (0.008)	0.179 (0.008)	0.189 (0.008)	0.242 (0.005)	4.843
	MSE	0.159 (0.059)	0.233 (0.075)	0.242 (0.073)	0.282 (0.073)	4.993
	CMSC	0.183 (0.021)	0.254 (0.023)	0.263 (0.023)	0.290 (0.023)	50.613
	CoMSC	0.227 (0.008)	0.291 (0.010)	0.300 (0.010)	0.304 (0.010)	5.198
	SCB_Concat	0.303 (0.009)	0.355 (0.011)	0.363 (0.010)	0.368 (0.012)	292.555
	SCB_Merge	0.311 (0.012)	0.371 (0.013)	0.379 (0.013)	0.387 (0.015)	1560.699

The similarity matrix is constructed with Gaussian kernel. In Gaussian kernel, we used the locality preserved kernel (Zelnik-Manor and Perona 2004) for the synthetic and Reuter multilingual datasets, while we give the standard deviation 100 for BBC dataset. MSE has a user parameter the size of neighbor nodes. As addressed in (Xia et al. 2010), we set it as 9 for the synthetic dataset and 5 for both Reuter multilingual and BBC datasets. The performances of CoMSC is obtained the best performances with the view weights from 0.1 to 0.9, while the summation of weights are kept 1.0. The number of clusters is assumed to be known in all experiments, Performances are determined with four measures: adjusted rand index (ARI) (Vinh, Epps, and Bailey 2009), mutual information (MI) (Romano et al. 2014), homogeneity (Hom), and completeness (Com) (Rosenberg and Hirschberg 2007). In all experiments, the user parameter C in both discussion and conclusion steps is fixed as 1.0. Experimental results are obtained by 10 trials for all datasets. For both Reuter multilingual and BBC datasets, the computation times spent to perform ten trials are additionally represented.

Experimental Results

Table 1 shows the results on the synthetic dataset. In this experiment, methods with information propagation for multi-view data show high performances around 0.90 of ARI, while Singleview just achieves 0.809. There exist three methods whose Adjusted RI is over 0.90: CMSC, SCB_Concat, and SCB_Merge. One of the proposed method SCB_Concat achieves the best performance with respect to all four measures. Even though SCB_Merge adopts more complex method than SCB_Concat to combine vectors from views, it shows performances similar with CMSC. This dataset is defined with informative views for just two clusters. For such simple dataset, the concatenation is enough to generate a compromise representation, since merging and selection can eliminate some information still kept in each view.

The experimental results on Reuter multilingual dataset are shown in Table 2. In these experiments, each view is generated with an whole document written in different languages. Thus, three views are enough to share information on a document and there rarely exist information that can be supplemented with additional views. It results in the stable performances of experimental methods except KernelProd.

KernelProd achieves 0.180 of Adjusted RI with three views, however it shows 0.124 with five views. KernelProd constructs a similarity matrix with the productions of similarities from all views. Even though a view suggests a high similarity between a pair of certain instances, it could be decreased when it does not agree with all other views.

Unlike the result in Table 1, CoMSC achieves better performances with Reuter multilingual dataset. As we mentioned, all views contain similar information in this dataset and thus, it results in strong dependencies among views. CMSC propagates incorrect information in this case. On the other hand, CoMSC improve its performance by just reducing noise in each view. These results might lead the conclusion that a min-disagreement is somewhat better than an information propagation in a real-world data. The experimental results from the proposed method give a refutation for this conclusion.

Table 3: Performances on BBC dataset

	Method	ARI	MI	Hom	Com	Comp. Time (sec)
Three views	Singleview	0.214 (0.015)	0.307 (0.013)	0.314 (0.013)	0.328 (0.012)	2.196
	KernelSum	0.256 (0.048)	0.331 (0.051)	0.337 (0.050)	0.346 (0.053)	2.243
	KernelProd	0.587 (0.007)	0.601 (0.008)	0.605 (0.008)	0.617 (0.008)	2.267
	MSE	0.141 (0.057)	0.262 (0.074)	0.269 (0.074)	0.408 (0.096)	2.359
	CMSC	0.666 (0.045)	0.660 (0.030)	0.664 (0.030)	0.671 (0.028)	19.863
	CoMSC	0.161 (0.027)	0.304 (0.024)	0.311 (0.024)	0.395 (0.037)	3.600
	SCB_Concat	0.694 (0.001)	0.690 (0.002)	0.694 (0.002)	0.699 (0.002)	124.775
	SCB_Merge	0.711 (0.005)	0.701 (0.005)	0.704 (0.005)	0.708 (0.005)	636.651
Four views	Singleview	0.199 (0.010)	0.284 (0.014)	0.291 (0.014)	0.313 (0.016)	2.881
	KernelSum	0.171 (0.024)	0.225 (0.028)	0.233 (0.028)	0.236 (0.028)	2.935
	KernelProd	0.493 (0.005)	0.521 (0.004)	0.526 (0.004)	0.530 (0.004)	2.882
	MSE	0.091 (0.016)	0.189 (0.031)	0.198 (0.030)	0.276 (0.068)	2.972
	CMSC	0.666 (0.031)	0.658 (0.007)	0.661 (0.007)	0.667 (0.007)	25.459
	CoMSC	0.110 (0.009)	0.257 (0.021)	0.265 (0.021)	0.378 (0.012)	4.548
	SCB_Concat	0.698 (0.003)	0.693 (0.004)	0.696 (0.004)	0.702 (0.004)	156.572
	SCB_Merge	0.703 (0.003)	0.699 (0.004)	0.702 (0.004)	0.708 (0.004)	780.531

The brainstorming process in the proposed method successfully propagates information among views in this dataset. In all experiments, SCB_Concat and SCB_Merge show better performances than other methods. Especially, SCB_Merge indicates the efficiency of the merging in the conclusion by showing about 0.310 of ARI. Since the difference between performances of SCB_Merge and SCB_Concat is not significant, SCB_Concat can be an alternative for clustering with less time or computational resources.

Table 3 shows experimental results on BBC dataset. Unlike the results on Reuter multilingual dataset, BBC dataset is constructed by slicing a document into three or four fragments. Thus, each view cannot contain information sufficient to estimate entire cluster boundaries. In this situation, KernelProd shows much better performance than both of Singleview and KernelSum. KernelProd eliminates uncertain information on views and this characteristic prevents its performance to be decreased by insufficient views.

When the information is not insufficient, the proposed methods and CMSC show much higher performances than others. Furthermore, while the proposed methods and CMSC show stable performances with BBC dataset, the performance of other methods is decreased as the number of views are increased from three to four. This result also proves that the efficiency of an information propagation with insufficient views, since they adopt explicit information propagation step in common. Through all experiments, SCB_Concat and SCB_Merge are superior to both CMSC and CoMSC which give motivation to the proposed method. This result can prove the effectiveness of the proposed method based on the consideration of dependencies among multiple views.

Conclusion

This paper proposes a novel extension of spectral clustering. Normally, multi-view data is handled under the assumption of the independence among views. As a result, when there exist dependencies among views, it can prevent information sharing among independent views that contain truly valuable information for each other. To avoid this problem, the proposed method adopts the brainstorming process.

In the brainstorming process, an instance and its multiple views are regarded as an agenda and brainstormers, respectively. The complementary information in different brainstormers is shared through the discussion and the opinions are merged with the conclusion. The discussion forces a brainstormer to adopt information from others who have more different opinions. In other words, to compensate the biases caused by information sharing between brainstormers with dependent opinions, those having independent opinions are more encouraged to discuss together than those with dependent opinions. A compromise opinion is finally obtained through the conclusion. In the conclusion, we suggest two ways to merge opinions in a single representation. Concatenation is a simple and contemporary method for multi-view data, while a linear model for merging is motivated from the min-disagreement.

The evaluation of the spectral clustering with the brainstorming process is performed with three datasets for clustering: synthetic, Reuter multilingual, and BBC datasets. We compared the proposed method with six ordinary spectral clustering based on four performance measures. Through experiments, our method always achieves the best performances in all three datasets, and at least in these datasets, the proposed method shows its effectiveness on multi-view data clustering.

Acknowledgements

This work was supported by ICT R&D program of MSIP/IITP. [B0126-16-1002, Development of smart broadcast service platform based on semantic cluster to build an open-media ecosystem]

References

Amini, M.; Usunier, N.; and Goutte, C. 2009. Learning from multiple partially observed views - an application to multilingual text categorization. In *Advances in Neural Information Processing Systems 22*, 28–36.

Bickel, S., and Scheffer, T. 2004. Multi-view clustering. In *Proceedings of IEEE International Conference on Data Mining*, 19–26.

Chaudhuri, K.; S, K.; Livescu, K.; and Sridharan, K. 2009. Multi-view clustering via canonical correlation analysis. In *Proceedings of the International Conference on Machine Learning*, 129–136.

Feng, L.; Cai, L.; Liu, Y.; and Liu, S. 2016. Multi-view spectral clustering via robust local subspace learning. *Soft Computing* 1–12.

Greene, D., and Cunningham, P. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the International Conference on Machine Learning*, 377–384.

Griva, I.; Nash, S.; and Sofer, A. 2009. *Linear and Nonlinear Optimization*. Splinger.

Hofmann, T. 1999. Probabilistic latent semantic indexing. In *Proceedings of ACM SIGIR conference on Research and Development in Information Retrieval*, 50–57.

Kumar, A., and Daumé III, H. 2011. A co-training approach for multiview spectral clustering. In *Proceedings of the International Conference on Machine Learning*, 393–400.

Kumar, A.; Rai, P.; and Daumé III, H. 2011. Co-regularized multi-view spectral clustering. In *Advances in Neural Information Processing Systems 24*, 1413–1421.

Li, Y.; Nie, F.; Huang, H.; and Huang, J. 2015. Large-scale multi-view spectral clustering via bipartite graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2750–2756.

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

Ng, A.; Jordan, M.; and Weiss, Y. 2002. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 15*, 849–856.

Romano, S.; Bailey, J.; Nguyen, V.; and Verspoor, K. 2014. Standardized mutual information for clustering comparisons: One step further in adjestment for chancee. In *Proceedongs of the 31st International Conference on Machine Learning*, 1143–1151.

Rosenberg, A., and Hirschberg, J. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedongs of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 410–420.

Verma, D., and Meila, M. 2003. A comparison of spectral clustering algorithms. Technical report.

Vinh, N. X.; Epps, J.; and Bailey, J. 2009. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedongs of the 26th Annual International Conference on Machine Learning*, 1073–1080.

Wang, H.; Weng, C.; and Yuan, J. 2014. Multi-feature spectral clustering with minimax optimization. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 4106–4113.

Xia, T.; Tao, D.; Mei, T.; and Zhang, Y. 2010. Multiview spectral embedding. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 40(6):1438–1446.

Xia, R.; Pan, Y.; Du, L.; and Yin, J. 2014. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2149–2155.

Zelnik-Manor, L., and Perona, P. 2004. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17*, 1601–1608.

Zhou, D., and Burges, C. 2007. Spectral clustering and transductive learning with multiple views. In *Proceedings of the International Conference on Machine Learning*, 1159–1166.