# Simultaneous Clustering and Ensemble

**Zhiqiang Tao,[1] Hongfu Liu,[1] Yun Fu[1,2]**
[1]Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, USA
[2]College of Computer and Information Science, Northeastern University, Boston, MA 02115, USA

## Abstract

Ensemble Clustering (EC) has gained a great deal of attention throughout the fields of data mining and machine learning, since it emerged as an effective and robust clustering framework. Typically, EC methods try to fuse multiple basic partitions (BPs) into a consensus one, of which each BP is obtained by performing traditional clustering method on the same dataset. One promising direction for ensemble clustering is to derive pairwise similarity from BPs, and then transform it as a graph partition problem. However, these graph-based methods may suffer from an *information loss* when computing the similarity between data points, because they only utilize the categorical data provided by multiple BPs, yet neglect rich information from raw features. This problem can badly undermine the underlying cluster structure in the original feature space, and thus degrade the clustering performance. In light of this, we propose a novel Simultaneous Clustering and Ensemble (SCE) framework to alleviate such detrimental effect, which employs the similarity matrix from raw features to enhance the co-association matrix summarized by multiple BPs. Two neat closed-form solutions given by eigenvalue decomposition are provided for SCE. Experiments conducted on 16 real-world datasets demonstrate the effectiveness of the proposed SCE over the traditional clustering and state-of-the-art ensemble clustering methods. Moreover, several impact factors that may affect our method are also explored extensively.

## Introduction

Ensemble Clustering (EC) (Strehl and Ghosh 2003; Fred and Jain 2005), also known as consensus clustering, emerges as an effective and robust alternative to the traditional clustering method. It aims to fuse multiple basic partitions (BPs) into a consensus one, where each BP is obtained by performing traditional clustering method on the same dataset. Tremendous efforts have been made on this area (Vega-Pons and Ruiz-Shulcloper 2011). One promising direction is to derive pairwise similarity from BPs and then transform ensemble clustering as a graph partition problem (Fred and Jain 2005; Liu et al. 2015; Zhou et al. 2015). These graph-based methods usually summarize BPs into a co-association matrix, which actually calculates the co-occurrence of instances belonging to the same cluster.

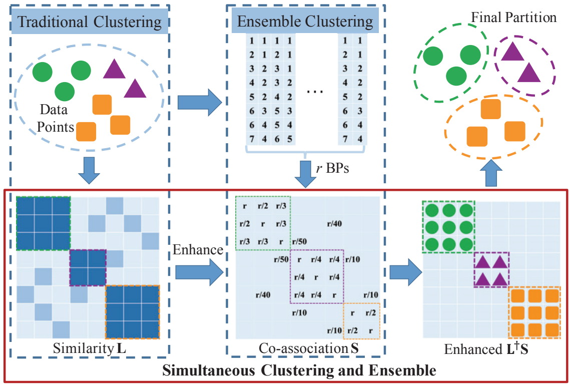


Figure 1: Traditional clustering is directly performed with raw features, while the input of EC is only the multiple BPs. However, the information loss of BPs may undermine the cluster structure in the original feature space. To alleviate this problem, SCE employs the similarity matrix to enhance the co-association matrix obtained by BPs.

Clearly, co-association matrix is the key factor for EC methods to conduct graph partitioning for the final consensus partition (Liu et al. 2015). However, it only computes the similarity between data points with the categorical data provided by BPs, yet neglects the rich information from raw features. Thus, the co-association matrix may suffer from a problem of *information loss*, which can badly undermine the clustering structure existing in the original feature space. Moreover, since BPs generally partition data with diverse cluster numbers (larger than the true cluster number) (Fred and Jain 2005; Wu et al. 2015), the co-association matrix may "dilute" the pairwise similarity between data points inevitably. Due to these problems, some EC methods even perform worse than the traditional clustering algorithms, as we observe in the empirical evidence.

In this paper, we propose a novel Simultaneous Clustering and Ensemble (SCE) framework to address the above challenges. As shown in Fig. 1, the similarity matrix from raw features and the co-association matrix derived by BPs are jointly involved for the clustering task. Different from the existing EC work, whose input is only multiple BPs, we reuse the feature information from original data to alleviate the information loss problem. Moreover, compared with several state-of-the-art EC methods, SCE can further improve the clustering performance, even under the scenario that the traditional clustering with raw features cannot achieve a satisfactory result. This indicates SCE is not just a trivial combination of similarity and co-association matrix, but an effect way to enhance the cluster structure existing in the co-association matrix. The contribution of this paper are summarized in threefold:

- A general SCE framework is proposed to perform ensemble clustering by simultaneously utilizing the raw feature information and basic partitions.
- Two neat closed-form solutions are provided for SCE. Specifically, we first give a linear solution by integrating information from two sources with a trade-off parameter and then approximate it as a non-parameter version, which is more robust and effective.
- To demonstrate the significant advantages over the traditional clustering and several state-of-the-art ensemble clustering methods, extensive experiments on 16 real-world datasets are conducted. Besides, some important impact factors are thoroughly explored as well.

# Related Work

In this section, we give a brief related work on ensemble clustering along two directions.

Co-association matrix summarizes the information of basic partitions by counting how often two instances co-occur in the same cluster, which can be regarded as a similarity matrix. By this means, any graph partition methods can be directly conducted on the co-association matrix for the final consensus partition. (Fred and Jain 2005) employed agglomerative hierarchical clustering on the co-association matrix. Graph-based Consensus Clustering (GCC) developed three graph based algorithms (Strehl and Ghosh 2003), Cluster-based Similarity Partitioning Algorithm (CSPA), HyperGraph Partitioning Algorithm (HGPA) and Meta-CLustering Algorithm (MCLA) and returned the best result according to a normalized mutual information measurement. Most recently, two interesting co-association matrix based methods emerged as Robust Consensus Clustering (RCE) (Zhou et al. 2015) and Spectral Ensemble Clustering (SEC) (Liu et al. 2015), respectively. RCE applied low-rank constraint on the co-association matrix for a robust representation, while SEC ran spectral clustering on the co-association matrix, linked it to a weighted K-means clustering for high efficiency and provided another interpretation of ensemble clustering in a utility way. Other representative methods in this direction include cumulative voting consensus (Ayad and Kamel 2008), weighted consensus clustering (Domeniconi and Al-Razgan 2009), matrix completion (Yi et al. 2012), infinite ensemble (Liu et al. 2016) and robust spectral ensemble clustering (Tao et al. 2016).

Another category of ensemble clustering is the utility-based method, which designs the utility function to measure the similarity between the basic partitions and the consensus one. By maximizing the utility function, ensemble clustering can be solved as a combinatorial optimization problem. For example, a Quadratic Mutual Information based objective function was proposed for ensemble clustering, and elegantly solved by K-means clustering (Topchy, Jain, and Punch 2003; 2005), based on the Category Utility Function (Mirkin 2001). K-means-based Consensus Clustering (KCC) was proposed as a theoretic framework in (Wu et al. 2015), which provided the sufficient and necessary condition for KCC utility functions to exactly map the ensemble clustering to a K-means problem with theoretical supports.

# Methodology

In this section, we first introduce some basic knowledge for ensemble clustering, then elaborate SCE with two closed-form solutions, and finally give a discussion on our method.

## Preliminary

Let $\mathcal{X} = \{x_1, x_2, \cdots, x_n\}$ be a set of $n$ data points belonging to $K$ crisp clusters, denoted as $\mathcal{C} = \{C_1, \cdots, C_k\}$, where $C_k \bigcap C_{k'} = \emptyset$, $\forall k \neq k'$, and $\bigcup_{k=1}^{K} C_k = \mathcal{X}$. Given $r$ basic partitions represented as $\Pi = \{\pi_1, \pi_2, \cdots, \pi_r\}$, each of which partitions $\mathcal{X}$ into $K_i$ clusters, and maps each data point to a cluster label ranged from 1 to $K_i$. The goal of ensemble cluster is to find an optimal consensus partition $\pi$ based on the input BPs $\Pi$. It is, in essence, a combinatorial optimization problem. As mentioned before, EC methods can be roughly generalized into two groups. For the utility function based one, it aims to find the consensus partition sharing the maximum utility function value with basic partitions, which has the following formulation:

$$\max_{\pi} \sum_{i=1}^{r} w_i U(\pi, \pi_i), \tag{1}$$

where $U$ is a utility function that measures the similarity between two partitions, and $w_i \in [0, 1]$ is the weight for each partition, with $\sum_{i=1}^{r} w_i = 1$. The well-known Categorical Utility Function (Mirkin 2001) can be calculated as follow:

$$U_c(\pi, \pi_i) = \sum_{k=1}^{K} p_{k+} \sum_{j=1}^{K_i} (\frac{p_{kj}^{(i)}}{p_{k+}})^2 - \sum_{j=1}^{K_i} (p_{+j}^{(i)})^2, \tag{2}$$

where $p_{kj}^{(i)}$ is the joint probability of one instance simultaneously belonging to $C_k$ and $C_j^i$. Here, $C_k$ is the $k$-th cluster in final partition $\pi$, and $C_j^i$ is the $j$-th cluster in $\pi_i$. $p_{k+}$ and $p_{+j}$ are the cluster portion of $\pi$ and $\pi_i$, respectively.

Another category is to summarize the information from basic partitions into a co-association matrix $\mathbf{S}$ (Fred and Jain 2005), which measures the times of two instances occurring in the same cluster as:

$$\mathbf{S}(x, y) = \sum_{i=1}^{r} \delta(\pi_i(x), \pi_i(y)), \delta(a, b) = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{if } a \neq b \end{cases}.$$

Based on the co-association matrix, the traditional graph partition methods can be used for the consensus clustering.

## Simultaneous Clustering and Ensemble

Given a set of BPs $\Pi$ and data matrix $\mathcal{X}$, our SCE framework is formulated as:

$$J_{ensemble}(\Pi) + \alpha J_{similarity}(\mathcal{X}), \tag{3}$$

where $J_{ensemble}(\Pi)$ represents ensemble clustering based on multiple BPs, $J_{similarity}(\mathcal{X})$ denotes traditional clustering preformed on raw features and $\alpha$ is a trade-off parameter balancing these two terms. By using Eq. (3), basic partitions and raw features are jointly involved for the clustering task.

Note that, although we can generate BPs from the similarity matrix and do clustering under a common EC framework, these BPs may still suffer from the information loss problem. However, considering that the similarity matrix represents the membership between data points in the original feature space, the proposed SCE employs it to obtain an enhanced co-association matrix for the final clustering.

In the following, we take the $U_c$ defined in Eq. (2) as the utility function in $J_{ensemble}(\Pi)$ and employ spectral clustering for $J_{similarity}(\mathcal{X})$. Thus, the objective of the Linear solution to our SCE (LSCE) is introduced as:

$$\min_{\pi} -\sum_{i=1}^{r} U_c(\pi, \pi_i) + \alpha \text{tr}(\mathbf{H}^{\text{T}}\mathbf{L}\mathbf{H}), \tag{4}$$

where tr$(\cdot)$ is the trace of a matrix, $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is the normalized Laplacian matrix, in which $\mathbf{A}$ is the affinity matrix followed the definition in (Ng, Jordan, and Weiss 2001), and $\mathbf{D}$ is the degree matrix which is diagonal with the $l$-th element being the sum of $l$-th row in $\mathbf{A}$; and $\mathbf{H} \in \mathbb{R}^{n \times K}$ is the scaled cluster membership matrix of $\pi$, which can be calculated as

$$\mathbf{H}_{lk} = \begin{cases} 1/\sqrt{|C_k|}, & \text{if } x_l \in C_k \text{ in } \pi \\ 0, & \text{otherwise} \end{cases}. \tag{5}$$

Obviously, the above objective function is difficult to optimize because (1) it is non-convex and (2) these two terms in objective function are in element-formulation and matrix-formulation, respectively. Fortunately, the following theorem is able to transform the optimization problem in Eq. (4) into an eigenvector decomposition problem.

**Theorem 1.** *Given $r$ BPs and the Laplacian matrix $\mathbf{L}$, we have:*

$$\begin{aligned} &\min_{\pi} -\sum_{i=1}^{r} U_c(\pi, \pi_i) + \alpha \text{tr}(\mathbf{H}^{\text{T}}\mathbf{L}\mathbf{H}) \\ &\Leftrightarrow \min_{\mathbf{H}} \ \text{tr}(\mathbf{H}^{\text{T}}(\alpha\mathbf{L} - \mathbf{S})\mathbf{H}), \end{aligned} \tag{6}$$

*where $\mathbf{S}$ is the co-association matrix derived from $r$ BPs*[1].

**Remark 1.** *In the left side of Eq. (6), we use the utility function to measure the similarity between two partitions while on the right side the co-association matrix is derived as a new representation for the similarity between two instances. This indicates that these two kinds of similarity in partition-level and instance-level can be convertible.*

***Algorithm 1.** Simultaneous Clustering and Ensemble*

```
Input: $\mathcal{X}$, data points $\{x_1, x_2, \cdots, x_n\}$,
       $\Pi$, basic partitions $\{\pi_1, \pi_2, \cdots, \pi_r\}$,
       $K$, the number of clusters.
Output: final partition $\pi$.
  Build the normalized Laplacian matrix $\mathbf{L}$ with $\mathcal{X}$;
  Calculate the co-association matrix $\mathbf{S}$ based on $\Pi$;
  Set $\mathbf{H}$ as the smallest $K$ eigenvectors of $\alpha\mathbf{L} - \mathbf{S}$
    for LSCE, or the $K$ largest ones of $\mathbf{L}^{\dagger}\mathbf{S}$ for NSCE;
  Run K-means on $\mathbf{H}$ to obtain the final partition $\pi$.
```

Based on Theorem 1, we can solve Eq. (4) by running K-means on the the smallest $K$ eigenvectors of $\alpha\mathbf{L} - \mathbf{S}$, as following (Zha et al. 2002; Dhillon, Guan, and Kulis 2004).

## A Non-parameter Solution

Although the above solution gives a neat mathematical way, the parameter $\alpha$ is difficult to set in practice. In the following, based on the solution in Eq. (6), we demonstrate a non-parameter solution. Taking a close look at Eq. (6), it can be roughly approximated as two trace terms:

$$\min_{\mathbf{H}} \ \text{tr}(\mathbf{H}^{\text{T}}\mathbf{L}\mathbf{H}) \text{ and } \max_{\mathbf{H}} \ \text{tr}(\mathbf{H}^{\text{T}}\mathbf{S}\mathbf{H}). \tag{7}$$

Thus, we could propose a Non-parameter version of SCE (NSCE) as an approximation to LSCE as:

$$\max_{\mathbf{H}} \frac{\text{tr}(\mathbf{H}^{\text{T}}\mathbf{S}\mathbf{H})}{\text{tr}(\mathbf{H}^{\text{T}}\mathbf{L}\mathbf{H})}. \tag{8}$$

To facilitate the solution for NSCE, we further approximate Eq. (8) as the final objective function of NSCE:

$$\max_{\mathbf{H}} \text{tr}((\mathbf{H}^{\text{T}}\mathbf{L}\mathbf{H})^{\dagger}(\mathbf{H}^{\text{T}}\mathbf{S}\mathbf{H})), \tag{9}$$

where $(\cdot)^{\dagger}$ indicates the generalized inverse of a matrix. A closed-form solution of NSCE could be given by the following theorem.

**Theorem 2.** *Given a co-association matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ and a normalized Laplacian matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$, the optimal solution $\mathbf{H} \in \mathbb{R}^{n \times K}$ for NCSE of Eq. (9) is composed by the largest $K$ eigenvectors of $\mathbf{L}^{\dagger}\mathbf{S}$.*

**Remark 2.** *Compared with our linear model, our non-parameter solution is also an eigenvector decomposition problem. The difference is that in our linear model, the decomposition matrix is the linear combination with the normalized Laplacian matrix $\mathbf{L}$ and the co-association matrix $\mathbf{S}$; while in the non-parameter solution, $\mathbf{L}^{\dagger}$ is employed to modify the co-association matrix $\mathbf{S}$.*

## Discussion

As shown by Algorithm 1, the most computing cost of our method goes to perform eigenvalue decomposition. For LSCE, it performs on $\alpha\mathbf{L} - \mathbf{S}$, and its time complexity is $\mathcal{O}(n^3)$; for NSCE, $\mathbf{L}^{\dagger}$ needs to be computed in advance, and it also has a complexity of $\mathcal{O}(n^3)$. Thus, our algorithms roughly have a similar time complexity to the traditional spectral clustering.

[1] The proof of all the theorems can be found in the supplementary material.

Table 1: Datasets Details

| Dataset | #Instance | #Feature | #Class | Density | Type |
|---|---|---|---|---|---|
| *cranmed* | 4663 | 41681 | 2 | 0.0014 | text |
| *hitech* | 2301 | 126321 | 6 | 0.0012 | text |
| *k1a* | 2340 | 13879 | 20 | 0.0068 | text |
| *la1* | 3204 | 31472 | 6 | 0.0048 | text |
| *la2* | 3075 | 31472 | 6 | 0.0047 | text |
| *mm* | 2521 | 126373 | 2 | 0.0015 | text |
| *ohscal* | 11162 | 11465 | 10 | 0.0053 | text |
| *reviews* | 4069 | 126354 | 5 | 0.0053 | text |
| *sports* | 8580 | 126355 | 7 | 0.0010 | text |
| *tr12* | 313 | 5804 | 8 | 0.0471 | text |
| *wap* | 1560 | 8460 | 20 | 0.0167 | text |
| *amazon* | 958 | 800 | 10 | 0.1215 | image |
| *ImageNet* | 7341 | 4096 | 5 | 0.1623 | image |
| *pendigits* | 10992 | 16 | 10 | 0.8717 | image |
| *USPS* | 9298 | 256 | 10 | 0.2456 | image |
| *webcam* | 295 | 800 | 10 | 0.1289 | image |

Both LSCE and NSCE enjoy a closed-form solution, thus the convergence of our algorithm is guaranteed. The final result can be directly obtained without a process of iterative optimization. Moreover, we can further speed up our method by using some off-the-shelf fast spectral clustering algorithms, such as (Yan, Huang, and Jordan 2009) and (Chen and Cai 2011).

## Experimental Results

In this section, we first evaluate the clustering performance of our approach (LSCE and NSCE) on numerous real-world datasets, and then explore the factors that may affect our method from three different views.

### Experimental Setup

*Experimental Datasets*. The testbed in our experiment mainly consists of 11 benchmark datasets selected from CLUTO[2], which is a dataset repository for document clustering (Zhao and Karypis 2002). In addition, five widely used datasets from other sources, including *OFFICE* dataset[3] (*amazon* and *webcam*), *ImageNet*[4], *pendigits*[5], and *USPS* (Cai, Wang, and He 2009), are also employed to evaluate the performance of our method on image clustering. Thus, we totally use 16 real-world datasets with types of text and image in this paper. More details are shown in Table 1.

*Validation Criteria*. All the datasets in Table 1 are provided with ground truth labels, thus we can validate our model quantitatively with some external measures, such as *Average Clustering Accuracy* (*ACC*) and *Normalized Mutual Information* (*NMI*), which are two well-known clustering criterion and introduced below. *ACC* is defined as the fraction of resulted labels given by a clustering method that match with ground-truth labels (Shao et al. 2015), and *NMI* (Cover and Thomas 1991) measures the mutual information entropy between the inferred partition and the ground-truth. Both *NMI* and *ACC* range from $0$ to $1$, and a higher value indicates better clustering performance. Besides, to ensure the statistical significance of the comparison result, the $p$-value is also calculated with $t$-test.

*Clustering Tools*. Typically, there are two common strategies for basic partition generation, *i.e.*, Random Parameter Selection (RPS) and Random Feature Selection (RFS). Here, following (Wu et al. 2015), we employ RPS to generate $\Pi$ of $r = 100$ BPs as default input for all the EC methods. Each BP is obtained by performing MATLAB *kmeans* function with a randomly selected cluster number from $[K, \sqrt{n}]$. Since some datasets we used suffer from a high feature dimension, we run *kmeans* with cosine similarity for all the datasets to speed up the process of BPs generation.

*Compared Methods*. We compare our method with three traditional clustering methods and four sate-of-the-art EC methods, respectively. To show the performance of feature information, K-means, spectral clustering (Ng, Jordan, and Weiss 2001), and Integrated K-means-Laplacian clustering (IKL) (Wang, Ding, and Li 2009) are implemented with MATLAB as three baseline methods. On the other side, four powerful EC methods, *i.e.*, Graph-based Consensus Clustering (GCC) (Strehl and Ghosh 2003), K-means-based Consensus Clustering (KCC) (Wu et al. 2015), Spectral Ensemble Clustering (SEC) (Liu et al. 2015), and Robust Consensus Ensemble (RCE) (Zhou et al. 2015) are used as the compared EC methods to demonstrate the effectiveness of our algorithm. These four methods are directly run by the authors' codes, and fed with the same BPs as ours. In addition, we use $\alpha = 1$ in our LSCE model as the default setting. We test each method 50 times and report the average result.

### Clustering Performance

Table 2 and Table 3 summarize the clustering performance of our SCE and other methods by *ACC* and *NMI*, respectively. Overall, our approach (LSCE and NSCE) outperforms all the compared methods on 16 datasets except *la2*. Specifically, as shown in Table 2 (Table 3), LSCE achieves the best *ACC* (*NMI*) on seven (nine) out of sixteen datasets, and six (four) second best among the remainder; NSEC is the top performer by *ACC* (*NMI*) on ten (seven) out of sixteen datasets, and three (eight) second best on the others. Note that, although NSCE is an approximation of the LSCE, its effectiveness can be clearly observed on *tr12*, *pendigits*, and *ohscal*, where the improvements over the best compared method are around $7\%$, $4\%$, and $3\%$ in Table 3, respectively. Moreover, our method enjoys a low *standard deviation* (*std*), nearly 0, for most cases, showing the proposed SCE as a steady clustering method. This is mainly because our method enjoys a closed-form solution, and jointly utilizes the similarity and co-association matrix. In contrast, although SEC achieves a high performance by running spectral clustering on the co-association matrix, it still suffers a *std* about $2\%$ on average, which implies the feature information can make ensemble clustering more stable.

In Table 2 and Table 3, based on input information, the compared methods are divided into two groups: (1) the baseline methods, which directly do clustering on data points; (2) EC methods, which utilize the cluster results from different

[2] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download

[3] https://www.eecs.berkeley.edu/~jhoffman/domainadapt/

[4] http://www.cs.dartmouth.edu/~chenfang/

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

Table 2: Clustering performance on 16 real-world datasets by ACC (%)

| Datasets | Ours | | Ensemble Clustering Methods | | | | Baseline Methods | | |
|---|---|---|---|---|---|---|---|---|---|
| | LSCE | NSCE | GCC | KCC | SEC | RCE | K-means | Spectral | IKL |
| *cranmed* | **98.81±0.00** | *98.77±0.00* | 90.95±0.00 | 71.00±18.41 | *98.77±0.00* | 98.35±0.00 | 69.12±0.64 | 78.90±0.00 | 66.43±0.00 |
| *hitech*$_{**}$ | **49.16±0.03** | *49.02±0.00* | 39.33±0.00 | 42.36±3.19 | 48.03±1.52 | 48.06±0.00 | 32.57±1.41 | 26.29±0.00 | 45.55±0.98 |
| *k1a*$_{*}$ | 43.80±1.51 | **44.42±1.19** | 41.21±0.10 | *44.36±3.30* | 41.14±2.04 | 40.53±0.00 | 38.11±3.07 | 43.26±2.27 | 43.16±1.27 |
| *la1*$_{**}$ | *54.06±0.00* | **54.24±0.00** | 44.44±0.00 | 48.47±4.76 | 51.98±2.63 | 48.42±0.00 | 35.05±1.88 | 29.34±0.00 | 41.17±0.00 |
| *la2* | *50.02±0.00* | **50.50±0.00** | 46.96±0.00 | 46.43±5.13 | 45.91±1.81 | 49.31±0.00 | 33.18±1.55 | 29.29±0.52 | 39.02±0.00 |
| *mm*$_{**}$ | **89.33±0.00** | **89.33±0.00** | 80.64±0.00 | 82.77±7.98 | 87.98±0.00 | 53.31±0.00 | 64.32±6.46 | 55.10±0.00 | 79.49±0.00 |
| *ohscal*$_{**}$ | *44.88±0.00* | **44.92±0.00** | 40.45±0.00 | 34.41±2.96 | 40.67±2.32 | N/A | 29.72±3.01 | 41.05±1.53 | 38.58±0.38 |
| *reviews*$_{**}$ | **66.90±0.00** | 62.69±0.00 | 57.83±0.00 | 62.77±5.58 | 57.65±3.64 | *65.81±0.00* | 47.89±3.81 | 34.42±0.06 | 65.54±0.10 |
| *sports*$_{**}$ | **52.76±0.00** | 50.10±0.00 | *50.54±0.00* | 45.94±6.62 | 44.33±3.17 | N/A | 38.91±3.01 | 39.80±0.00 | 33.29±0.00 |
| *tr12*$_{**}$ | *60.38±0.00* | **66.12±0.08** | 55.59±0.00 | 56.03±3.90 | 58.85±2.07 | 50.16±0.00 | 28.20±1.32 | 28.85±0.30 | 28.49±0.18 |
| *wap*$_{**}$ | 42.01±0.95 | **42.28±1.20** | 42.04±0.13 | *42.08±2.99* | 38.76±2.67 | 40.88±0.00 | 37.02±2.97 | 38.73±2.40 | 40.43±1.34 |
| *amazon*$_{*}$ | *44.68±0.00* | **44.88±0.04** | 43.95±0.00 | 41.30±2.30 | 41.30±1.80 | 41.44±0.00 | 30.00±2.77 | 32.03±1.44 | 30.62±1.47 |
| *ImageNet*$_{**}$ | **82.18±0.01** | *81.75±0.00* | 72.40±0.01 | 74.40±4.59 | 78.86±5.78 | N/A | 71.22±2.98 | 73.48±0.02 | 76.04±0.00 |
| *pendigits*$_{**}$ | 70.92±1.76 | **74.57**±3.37 | 73.06±0.00 | 64.67±6.42 | 67.67±7.48 | N/A | *74.50±4.03* | 70.88±3.66 | 73.31±1.45 |
| *USPS*$_{**}$ | **73.19±1.77** | 64.83±0.00 | 59.59±0.01 | 58.13±6.71 | 67.53±4.04 | N/A | 67.25±0.24 | 68.95±2.95 | *71.34±0.02* |
| *webcam*$_{*}$ | *54.98±0.78* | **55.25±0.79** | 49.83±0.00 | 49.91±2.62 | 50.82±2.77 | 49.08±0.00 | 38.44±3.19 | 45.08±2.37 | 49.39±1.48 |

The top *ACC* value is highlighted by red bold font and the second best by blue italic; * (**) indicates statistically (extremely) significant.
N/A means the failure due to out of memory with 32 GB.

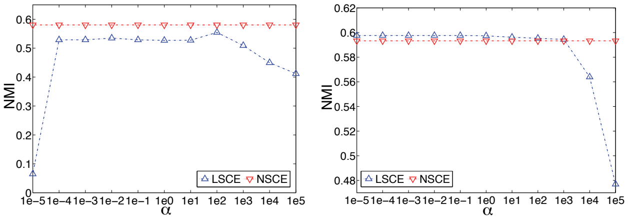
BPs. As excepted, EC methods perform better than the baselines for most of scenarios, which again demonstrates the superiority of ensemble. However, sometimes the EC methods cannot hold their advantage. For example, compared with spectral clustering, KCC loses its power on *ohscal* and *pendigits* by *ACC*. For another example, IKL outperforms GCC on *hitech*, *k1a*, *wap* and *USPS* by *NMI*, respectively. This is mainly due to the fact that existing EC methods may suffer from the information loss of multiple BPs and neglect the membership between data points in original feature space. Therefore, it motivates us to propose the SCE framework to fill in this gap. Another important observation is that, even under the cases when clustering performance of raw features is ineffective, such as *hitech*, *mm* and *tr12*, our approach still exceeds the other EC methods, which shows SCE as an effective way to utilize the information from raw features and BPs, but not a trivial combination.

To further validate our method, we employ $t$-test to compute the $p$-value between our approach and the top compared method on each dataset. We label the dataset with "$*$" to indicate the significant level ($p < 0.05$) and "$**$" to extremely significant ($p < 0.01$). As can be seen, we outperform the best compared method with a statistically significant level, only except *cranmed* and *la2*, which shows the effectiveness of our algorithm again. In summary, Table 2 and Table 3 demonstrate the superiority of our SCE over other methods from two clustering criteria and a statistical view.

## Exploration of Impact Factors

In this subsection, we will explore the effect of the trade-off parameter $\alpha$ in LSCE, BPs number $r$, and one alternative BPs generation (RFS) on the clustering performance of our method, respectively.

**Trade-off parameter**. Recall the trade-off parameter $\alpha$ in Eq. (3). It essentially balances the cluster label consensus and feature similarity in the objective function of LSCE. To explore the impact of $\alpha$ on final clustering performance, we vary it from $1e-5$ to $1e+5$, and test LSCE on *tr12* and *ImageNet*, respectively. As shown by Fig. 2, benefiting from the robust nature of ensemble clustering, LSCE keeps a stable and satisfied performance with the range of $1e-4$ to $1e+2$. However, as $\alpha$ increases vastly, LSCE will "degrade" as the spectral clustering algorithm, and its performance may drop sharply. This implies that our LSCE is insensitive with a relatively small $\alpha$.



(a) *tr12* (b) *ImageNet*

Figure 2: Impact of $\alpha$ to LSCE.

**The Number of BPs**. Fig. 3 (a&b) depicts the *ACC* variation of our methods in term of BPs number ($r$) on two datasets (*i.e.*, *la1* and *mm*), where $r$ varies from 10 to 90 with an interval of 10. Since we generate $\Pi$ (100 BPs) on each dataset as default in our experiment, here, for each $r \leq 90$, we randomly select $r$ BPs from $\Pi$ as the input for EC methods. We repeat the sampling process 100 times, and report the average testing result. For a better view, we only compare with *KCC* and *SEC* on different BPs number, as they generally outperform other compared methods.

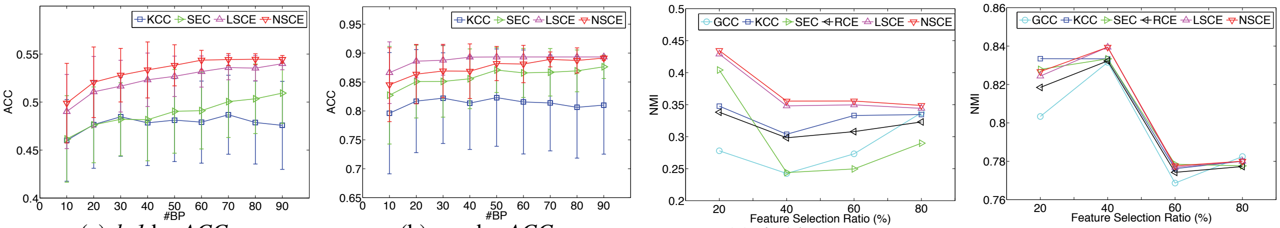
We show the *std* of *ACC* by using error bar in Fig. 3 (a&b). Note that, the *std* here is different from the ones we have shown in Table 2 and Table 3, because they are produced by different reasons: the *std* here is due to the variation of input BPs, while the other is caused by different initialization in the clustering process. Thus, the *std* generated here reveals the robustness of a EC method to the number of BPs. As

Table 3: Clustering performance on 16 real-world datasets by NMI (%)

| Datasets | Ours | | Ensemble Clustering Methods | | | | Baseline Methods | | |
|---|---|---|---|---|---|---|---|---|---|
| | LSCE | NSCE | GCC | KCC | SEC | RCE | K-means | Spectral | IKL |
| *cranmed* | **90.61±0.00** | *90.36±0.00* | 58.95±0.00 | 38.08±33.30 | *90.36±0.00* | 88.18±0.00 | 27.00±0.72 | 38.75±0.00 | 24.78±0.00 |
| *hitech*$_*$ | **34.24±0.05** | *34.10±0.00* | 28.35±0.00 | 27.84±1.71 | 33.00±2.37 | 27.67±0.00 | 17.52±2.44 | 2.73±0.17 | 29.31±0.32 |
| *k1a*$_{**}$ | *57.81±0.71* | **58.22±0.56** | 49.12±0.08 | 56.98±1.32 | 53.61±1.08 | 53.73±0.00 | 43.08±2.76 | 51.44±0.88 | 52.28±0.70 |
| *la1*$_*$ | *35.33±0.00* | **35.41±0.00** | 29.21±0.00 | 30.33±2.87 | 33.16±2.43 | 30.52±0.00 | 15.39±1.58 | 1.93±0.00 | 27.99±0.00 |
| *la2* | 32.79±0.00 | *32.90±0.00* | **33.58±0.00** | 28.28±3.78 | 26.61±1.38 | 31.49±0.00 | 17.24±1.66 | 1.70±0.34 | 26.50±0.00 |
| *mm*$_{**}$ | **51.71±0.00** | **51.71±0.00** | 29.08±0.00 | 40.96±2.67 | 49.19±0.00 | 0.00±0.00 | 16.41±0.98 | 0.65±0.00 | 30.30±0.00 |
| *ohscal*$_*$ | **34.07±0.00** | *33.22±0.00* | 27.81±0.00 | 24.53±2.53 | 30.49±1.51 | N/A | 21.25±2.92 | 30.12±0.75 | 29.61±0.22 |
| *reviews*$_{**}$ | **57.58±0.00** | *53.73±0.00* | 42.46±0.01 | 50.13±4.84 | 48.28±3.38 | 42.70±0.00 | 25.40±5.08 | 4.59±0.16 | 39.34±0.18 |
| *sports*$_{**}$ | **53.17±0.00** | *51.53±0.00* | 47.70±0.00 | 41.51±5.83 | 44.02±3.45 | N/A | 22.42±3.70 | 1.47±0.00 | 28.06±0.00 |
| *tr12*$_{**}$ | *52.69±0.00* | **58.05±0.10** | 49.03±0.00 | 51.21±3.32 | 51.39±2.07 | 45.72±0.00 | 8.90±1.68 | 7.35±0.67 | 9.47±0.23 |
| *wap*$_*$ | *57.92±0.49* | **58.00±0.47** | 49.81±0.10 | 56.40±1.32 | 54.10±1.20 | 54.97±0.00 | 42.55±2.24 | 49.10±1.31 | 51.85±0.91 |
| *amazon*$_*$ | **34.57±0.00** | *34.45±0.01* | 32.82±0.00 | 33.11±1.27 | 34.37±1.15 | 31.26±0.00 | 30.84±1.23 | 29.68±0.41 | 30.65±0.63 |
| *ImageNet*$_{**}$ | **59.73±0.01** | *59.33±0.00* | 49.35±0.01 | 52.50±3.89 | 57.84±2.62 | N/A | 45.27±2.23 | 45.96±0.03 | 50.01±0.00 |
| *pendigits*$_{**}$ | 70.17±0.96 | **74.29±1.77** | *70.58±0.00* | 69.47±3.82 | 70.21±3.71 | N/A | 68.94±0.55 | 66.02±1.05 | 69.53±0.41 |
| *USPS*$_{**}$ | **68.63±0.33** | 64.41±0.00 | 62.84±0.01 | 64.51±4.40 | *65.25±1.79* | N/A | 61.41±0.15 | 65.05±1.33 | 64.71±0.03 |
| *webcam*$_*$ | 50.86±0.57 | **51.37±0.55** | 50.13±0.00 | *51.02±1.52* | 49.39±0.73 | 49.67±0.00 | 41.43±3.78 | 44.49±1.43 | 49.30±1.21 |

The top *NMI* value is highlighted by red bold font and the second best by blue italic; * (**) indicates statistically (extremely) significant.

N/A means the failure due to out of memory with 32 GB.



(a) *la1* by *ACC* (b) *mm* by *ACC* (c) *la1* by *NMI* (d) *mm* by *NMI*

Figure 3: (a&b) Impact of BPs number ($r$) to ensemble clustering, where $r$ varies from 10 to 90. (c&d) Impact of RFS to ensemble clustering, where the selection ratio is set as 20%, 40%, 60%, and 80%, respectively.

can be seen, there are two important observations. First, as $r$ increases, the performance of all the EC methods generally goes up, which shows the diversity is a key factor to EC methods (Iam-on et al. 2011; Wu et al. 2015). Second, all the methods suffer from a large *std* when $r$ is small. However, compared to KCC and SEC, our methods trend to be stable after $r > 60$, which indicates LSCE and NSCE are more robust to the number of basic partitions.

**BP Generation Strategy**. Finally, we test our SCE framework under another common BP generation strategy, named Random Feature Selection (RFS). RFS generates multiple BPs by utilizing the randomly selected partial features from the original feature dimensions, according to a certain feature selection ratio. In details, for each dataset, we run K-means 100 times with cluster number $K$ to generate the basic partitions ($\Pi$) under the RFS strategy. Besides, we vary the ratio as 20%, 40%, 60%, and 80%, to further explore its impact on the clustering performance. We compare our method with all the compared EC methods on each ratio. As shown by Fig. 3 (c&d), we achieve the best performance.

One may note that, all the EC methods generally have a better performance at the ratio of 20% or 40%. This is mainly because a relatively low feature selection ratio can generate diverse BPs, which is important to the success of ensemble clustering (Wu et al. 2015; Iam-on et al. 2011). By using RFS, we can significantly boost the clustering performance on some cases. For example, the best RFS-based result of NSCE improves the *NMI* over 30% from its RPS-based one ($NMI = 51.71\%$) on *mm*, where the similar situation appears to other methods. We conjecture that there exists some noises in the raw feature of *mm* (Zhao and Fu 2015), and thus the quality of BPs may degrade by using RPS. To sum up, RFS is an alternative BP generation strategy for our method, and it can improve the performance when the input feature suffers from noises.

## Conclusion

In this paper, we proposed a novel SCE framework by reusing raw features to handle the problem of information loss for ensemble clustering. The similarity matrix obtained from original data was employed to enhance the cluster structure of the co-association matrix derived by input BPs. Two algorithms were put forward to solve this problem with closed-form solutions. Experiments on 16 real-world datasets were conducted to demonstrate the effectiveness of the proposed algorithms over several traditional clustering and state-of-the-art ensemble clustering methods. Moreover, three impact factors were explored extensively.

# Acknowledgment

This work is supported in part by the NSF IIS award 1651902, ONR award N00014-12-1-1028, ONR Young Investigator Award N00014-14-1-0484, and U.S. Army Research Office Young Investigator Award W911NF-14-1-0218.

# References

Ayad, H. G., and Kamel, M. S. 2008. Cumulative voting consensus method for partitions with variable number of clusters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(1):160–173.

Cai, D.; Wang, X.; and He, X. 2009. Probabilistic dyadic data analysis with local and global consistency. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 105–112.

Chen, X., and Cai, D. 2011. Large scale spectral clustering with landmark-based representation. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Cover, T. M., and Thomas, J. A. 1991. *Elements of Information Theory*. Wiley-Interscience.

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

Domeniconi, C., and Al-Razgan, M. 2009. Weighted cluster ensembles: Methods and analysis. *ACM Transactions on Knowledge Discovery from Data* 2(4):17:1–17:40.

Fred, A. L. N., and Jain, A. K. 2005. Combining multiple clusterings using evidence accumulation. *IEEE Transaction on Pattern Analysis Machine Intelligence* 27(6):835–850.

Iam-on, N.; Boongoen, T.; Garrett, S. M.; and Price, C. J. 2011. A link-based approach to the cluster ensemble problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(12):2396–2409.

Liu, H.; Liu, T.; Wu, J.; Tao, D.; and Fu, Y. 2015. Spectral ensemble clustering. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 715–724.

Liu, H.; Shao, M.; Li, S.; and Fu, Y. 2016. Infinite ensemble for image clustering. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1745–1754.

Mirkin, B. 2001. Reinterpreting the category utility function. *Machine Learning* 45(2):219–228.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *Advances In Neural Information Processing Systems*, 849–856.

Shao, M.; Li, S.; Ding, Z.; and Fu, Y. 2015. Deep linear coding for fast graph clustering. In *Proceedings of International Joint Conference on Artificial Intelligence*.

Strehl, A., and Ghosh, J. 2003. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3:583–617.

Tao, Z.; Liu, H.; Li, S.; and Fu, Y. 2016. Robust spectral ensemble clustering. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 367–376.

Topchy, A. P.; Jain, A. K.; and Punch, W. F. 2003. Combining multiple weak clusterings. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, 331–338.

Topchy, A. P.; Jain, A. K.; and Punch, W. F. 2005. Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(12):1866–1881.

Vega-Pons, S., and Ruiz-Shulcloper, J. 2011. A survey of clustering ensemble algorithms. *IJPRAI* 25(3):337–372.

Wang, F.; Ding, C. H. Q.; and Li, T. 2009. Integrated kl (k-means - laplacian) clustering: A new clustering approach by combining attribute data and pairwise relations. In *Proceedings of the SIAM International Conference on Data Mining*, 38–48.

Wu, J.; Liu, H.; Xiong, H.; Cao, J.; and Chen, J. 2015. K-means-based consensus clustering: A unified view. *IEEE Transactions on Knowledge and Data Engineering* 27(1):155–169.

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.

Yi, J.; Yang, T.; Jin, R.; Jain, A. K.; and Mahdavi, M. 2012. Robust ensemble clustering by matrix completion. In *Proceedings of the 12th IEEE International Conference on Data Mining*, 1176–1181.

Zha, H.; He, X.; Ding, C.; Gu, M.; and Simon, H. D. 2002. Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems 14*. 1057–1064.

Zhao, H., and Fu, Y. 2015. Dual-regularized multi-view outlier detection. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Zhao, Y., and Karypis, G. 2002. Criterion functions for document clustering: Experiments and analysis. Technical report.

Zhou, P.; Du, L.; Wang, H.; Shi, L.; and Shen, Y. 2015. Learning a robust consensus matrix for clustering ensemble via kullback-leibler divergence minimization. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 4112–4118.