Affinity Aggregation for Spectral Clustering

Hsin-Chien Huang[1,2] Yung-Yu Chuang[1,3] Chu-Song Chen[2,3]
[1] Dept. of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.
[2] Institute of Information Science, Academia Sinica, Taipei, Taiwan.
[3] Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan.
E-mail: sean@cmlab.csie.ntu.edu.tw, cyy@csie.ntu.edu.tw, song@iis.sinica.edu.tw

Abstract

Spectral clustering makes use of spectral-graph structure of an affinity matrix to partition data into disjoint meaningful groups. Because of its elegance, efficiency and good performance, spectral clustering has become one of the most popular clustering methods. Traditional spectral clustering assumes a single affinity matrix. However, in many applications, there could be multiple potentially useful features and thereby multiple affinity matrices. To apply spectral clustering for these cases, a possible way is to aggregate the affinity matrices into a single one. Unfortunately, affinity measures constructed from different features could have different characteristics. Careless aggregation might make even worse clustering performance. This paper proposes an affinity aggregation spectral clustering (AASC) algorithm which extends spectral clustering to a setting with multiple affinities available. AASC seeks for an optimal combination of affinity matrices so that it is more immune to ineffective affinities and irrelevant features. This enables the construction of similarity or distance-metric measures for clustering less crucial. Experiments show that AASC is effective in simultaneous clustering and feature fusion, thus enhancing the performance of spectral clustering by employing multiple affinities.

1. Introduction

Clustering is an important unsupervised learning method for dividing data into a set of disjoint subsets with high intra-cluster similarity and low inter-cluster similarity. It has been addressed in many contexts and widely used for computer vision, pattern recognition, and multimedia analysis. Among many clustering algorithms proposed before, spectral clustering (SC) is one of the best. It often outperforms other methods by transforming data points into another space in which their cluster properties are enhanced.

The success of spectral clustering algorithms depends heavily on the choice of the metric [3]. However, spectral clustering has no built-in mechanism for discovering good metrics for better clustering results. Therefore, it is often necessary to use other feature selection or feature weighting methods as a precursor before invoking spectral clustering. The problem is aggravated for many real-world clustering problems in which there are multiple potentially useful cues. For example, for face clustering, cues such as hair appearance, global positioning information and clothing appearance have been used for boosting the performance [1]. For clustering images, a variety of visual descriptors have been proposed for colors, textures and bag-of-word models. Each type of the descriptors defines an affinity matrix in association with its similarity metric. For such applications, to apply spectral clustering, it is often necessary to aggregate similarity measures from different features into a single affinity matrix by feature selection or feature fusion. Otherwise, performance of spectral clustering could degrade dramatically in the presence of irrelevant, ineffective or unreliable features.

This paper proposes *affinity aggregation spectral clustering* (AASC), a method for aggregating affinity matrices for spectral clustering. The proposed method shares similar ideas with multiple kernel learning (MKL) [23, 35, 8] that aggregates several kernels to construct a better one. We propose a framework for learning the similarity matrix of spectral clustering, which attempts to make spectral clustering more robust by alleviating the impact of unreliable and irrelevant features. However, the method is different from MKL in the following two aspects: (1) our method is unsupervised, *i.e*. no labels are available for data; and (2) the affinity matrix composed by the pairwise similarity between data is not necessarily positive semi-definite. We only assume that affinity matrices are symmetric.

The rest of the paper is organized as follows. Section 2 discusses related work. In Section 3, we briefly review spectral clustering. Section 4 introduces our affinity aggregation spectral clustering algorithm. Experiments are described in Section 5. Finally, Section 6 concludes the paper with directions for future work.

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

2. Related work

Multi-view and multi-kernel learning (MVL; MKL) were both initiated from machine learning. Though they are somewhat relevant, there is not a coherent view of them even in the machine learning society. MVL seeks to employ multiple "independent" clues (*e.g.*, bi-lingual information, different modalities); MKL, however, combines multiple base kernels to create a "unified" kernel for learning, where these kernels are not necessarily independent like the "views" in MVL. Similarly, both were originally studied for (semi-)supervised learning, and have been extended to unsupervised setting recently [33, 6, 14, 15, 17, 16]. Multi-view clustering [20, 34, 15, 16], as an extension of MVL, assumes inherently that the views are uncorrelated. Our affinity-aggregation approach is extended from MKL without such assumptions. More importantly, unlike MKL fuses multiple kernels into a single one, most multi-view clustering methods cannot provide an explicit form of the learned kernel/affinity. They thus suffer from the out-of-sample problem (*i.e.*, re-training is needed to produce the clustering result for the new data). Both characteristics make our method generally more applicable.

Despite most studies on learning an aggregated similarity measure focused on supervised learning (such as MKL), studies on combining multiple cues for clustering can be found in several applicational or theoretical frameworks. For example, to improve the performance of clustering, several cues or repulsive constraints could be used. In the domain of face clustering, Song and Leung [27] used contextual information, such as clothes and time, to boost the performance. They formulated the problem as a constrained optimization problem in which some cues are used for the objective function and the others serve as constraints. Anguelov *et al*. [1] constructed a graph based on face and clothing cues. However, these approaches might suffer from two problems. First, the constrained optimization problem or the constructed graph becomes complex and difficult to solve when many constrains are applied. Second, the decisions on what appropriate cues to be included in the objective function and how to impose the constrains to the optimization are still critical. MKL has been extended to unsupervised settings by some researches recently. Lin *et al*. [17] introduced a MKL-DR framework that incorporates MKL into the training process of dimensionality reduction (DR) methods. It works with multiple base kernels, each of which is created based on a specific kind of data descriptor, and fuses the descriptors in the domain of kernel matrices. Zhao *et al*. [33] proposed a multiple kernel clustering (MKC) algorithm that finds the maximum-margin hyperplane, cluster labelling, and the optimal kernel simultaneously. However, MKC requires explicit evaluation in the feature space and the formulation leads to a non-convex integer optimization problem that is difficult to solve.

There are still only few studies on improving spectral clustering by fusing multiple affinities to a single affinity. To overcome existing spectral clustering algorithms with 2-way relationships, Karydis *et al*. [14] developed a data-modeling scheme and a tag-aware spectral clustering procedure that uses tensors (high-dimensional arrays) to store the multi-graph structures for capturing the personalized aspects of similarity. First, the Laplacian tensor was constructed by stacking affinity matrices. Then, they used tensor factorization to extract spectral features from the Laplacian tensor. Finally, data were clustered by spectral features which preserve the 3-way relationships among inherent dimensions of the data. Cai *et al*. [6] proposed the multi-modal spectral clustering (MMSC) algorithm to learn a commonly shared graph-Laplacian matrix by unifying different modals (or image features). MMSC integrates heterogeneous features on unlabeled images and unsegmented images. An optimization framework is designed to find an averaged clustering result with smaller training error. One main limitation of MMSC is that it suffers from the out-of-sample problem: MMSC does not find an explicit representation to aggregate the affinities; thereby, when the input data is out of the training samples, re-training is needed to produce the clustering result for the new data. Our previous work, multiple-affinity spectral clustering [12], also dealt with the spectral clustering problem with multiple affinity measures. However, its convergence is not guaranteed when the method is implemented by finding generalized eigenvectors in each iteration. In addition, its performance is not as good as the proposed AASC.

3. Spectral Clustering

Spectral clustering is originated from spectral graph theory. Given n data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ and some pairwise affinity w_{ij} that is symmetric and non-negative, measuring the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, spectral clustering aims to divide these data into C clusters by finding n indicators $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n$ which satisfies

$$\min_{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n} \sum_{i,j} w_{ij} ||\mathbf{f}_i - \mathbf{f}_j||^2. \quad (1)$$

Let $\mathbf{W}$ be the $n \times n$ matrix constituted of the affinities w_{ij}, and $\mathbf{D}$ be the diagonal matrix with its i-th diagonal element being the sum of i-th row of $\mathbf{W}$, *i.e.* $\mathbf{D}_{ii} = w_{i1} + w_{i2} + \ldots + w_{in}$. Spectral clustering solves Equation 1 by finding the smallest eigenvalues and their corresponding eigenvectors of the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Since the smallest eigenvalue λ_1 of $\mathbf{L}$ is always 0 which corresponds to the trivial solution of the constant-one eigenvector $\mathbf{1}$, the solution of spectral clustering is constructed by the eigenvectors corresponding to the next C smallest eigenvalues, $\lambda_2, \lambda_3, \ldots, \lambda_{C+1}$. After stacking these C eigenvectors into

a $n \times C$ matrix, the i-th row of the stacked matrix corresponds to the indicator $\mathbf{f}_i$ for $\mathbf{x}_i$. The above method is called the unnormalized spectral clustering.

Shi and Malik [25] proposed a normalized spectral clustering algorithm, in which the indicators are constructed by finding the eigenvectors v of the generalized eigenproblem $\mathbf{L}v = \lambda \mathbf{D}v$. In normalized spectral clustering, when minimizing Equation 1, the constraint employed becomes $\mathbf{f}^T\mathbf{D}\mathbf{f} = 1$ instead of $\mathbf{f}^T\mathbf{f} = 1$. It is equivalent to minimizing $\mathbf{g}^T(\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2})\mathbf{g}$ when transforming the variables with $\mathbf{g} = \mathbf{D}^{1/2}\mathbf{f}$ and constrained by $\mathbf{g}^T\mathbf{g} = 1$. Many studies [25, 3] have shown that normalized spectral clustering performs considerably better than unnormalized spectral clustering for various problems.

In practice, spectral clustering often serves as a preprocessing step of other clustering algorithms such as k-means. The main trick of spectral clustering is to transform the representations of the data points $\mathbf{x}_i$ into the indicator space in which the cluster characteristics become more prominent. Because cluster properties are enhanced in this new representation space, even simple clustering algorithms, such as k-means clustering, have no difficulty on distinguishing clusters. Main reasons for spectral clustering's success include: (1) it does not make any assumptions on the form of the clusters (as opposed to k-means, where the clusters are always convex sets); and (2) it can be implemented efficiently even for large data sets as long as the affinity matrix is sparse. However, one of its limitations is that choosing a good affinity measure is not trivial for the application. For real-world clustering problems, the affinities w_{ij} could be obtained in multiple ways. They could be determined with different types of extracted features, or be constructed by reproducible kernels when $\mathbf{x}_i$ are vectors in some Euclidean space. We show how to find a weighted combination of the affinities so that a better similarity measure can be learned for spectral clustering in an unsupervised fashion.

4. Affinity aggregation spectral clustering

Assume that there are m affinity matrices $\mathbf{W}_k (k = 1 \dots m)$ available. The k-th matrix's ij-th element $w_{ij;k}$ represents the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ when measuring with the k-th type of affinity metric. Since the affinities $w_{ij;k}$ are non-negative, we can denote $w_{ij;k} = s^2_{ij;k}$ to reflect this nature. As mentioned, the goal is to find a proper weight assignment to these affinities. Let $\mathbf{v} = [v_1, v_2, \cdots, v_k]^T$ be a weight vector in association with these affinities. The k-th weighted affinity can be denoted as $\sigma_{ij;k} = v_k s_{ij;k}$. We can then formulate the AASC problem as

$$\begin{aligned}
&\min_{\substack{\mathbf{f}_1,\dots,\mathbf{f}_n \\ v_1,\dots,v_m}} \sum_k \sum_{i,j} \sigma^2_{ij;k} ||\mathbf{f}_i - \mathbf{f}_j||^2 \\
&= \min_{\substack{\mathbf{f}_1,\dots,\mathbf{f}_n \\ v_1,\dots,v_m}} \sum_k \sum_{i,j} v_k^2 w_{ij;k} ||\mathbf{f}_i - \mathbf{f}_j||^2 \\
&= \min_{\substack{\mathbf{f}_1,\dots,\mathbf{f}_n \\ v_1,\dots,v_m}} \sum_k v_k^2 \mathbf{f}^T(\mathbf{D}_k - \mathbf{W}_k)\mathbf{f} \\
&\equiv \min_{v_1,\dots,v_m} \sum_k \beta_k v_k^2 \qquad (2)
\end{aligned}$$

where $\mathbf{D}_k - \mathbf{W}_k$ is the Laplacian matrix associated with the k-th affinity metric, and

$$\beta_k = \mathbf{f}^T(\mathbf{D}_k - \mathbf{W}_k)\mathbf{f}.$$

Note that applying an affinity aggregation vector $\mathbf{v}$ makes the representation of new data easy. Hence, it avoids the out-of-sample problem of previous works such as Cai *et al*. [6]. Besides, we minimize the clustering error directly in the representation space, making the results better than finding an averaged representation of the single-affinity outputs in their approach (c.f. the experimental validation).

The objective is minimized under the constraint that weighted sum of v_k is normalized, $\sum_{k=1}^m t_k v_k = n$, where $t_k = tr(\mathbf{S}_k)$ and $\mathbf{S}_k$ is the matrix constituted of $s_{ij;k}$. This implies the trace of the aggregated affinity matrix is bounded. It is because from Cauchy-Schwartz inequality, the aggregated affinity matrix satisfies that $tr(\mathbf{W}) = \sum_{k=1}^m tr(v_k^2 \mathbf{W}_k) = \sum_{k=1}^m v_k^2 tr(\mathbf{W}_k) \geqslant \frac{1}{m}(\sum_{k=1}^m v_k tr(\mathbf{S}_k))^2 = \frac{n^2}{m}$, yielding a lower bound of the trace of $\mathbf{W}$. Without loss of generality, since the diagonal element of an affinity matrix is always set as 1, which implies $t_k = n$. The constraint $\sum_{k=1}^m t_k v_k = n$ thus becomes a simpler form, $\sum_{k=1}^m v_k = 1$.

In addition, to satisfy the normalized spectral clustering, the constraint $\mathbf{f}^T\mathbf{D}\mathbf{f} = 1$ is also required. That is,

$$1 = \mathbf{f}^T\mathbf{D}\mathbf{f} = \mathbf{f}^T(v_1^2\mathbf{D}_1 + \cdots + v_k^2\mathbf{D}_k)\mathbf{f} \equiv \sum_k \alpha_k v_k^2$$

where

$$\alpha_k = \mathbf{f}^T\mathbf{D}_k\mathbf{f}.$$

To solve the above problem, there are two sets of variables, the indicator vector $\mathbf{f}$ and the affinity aggregation weights $\mathbf{v}$. It becomes much easier to solve if we solve one set of variables at a time while fixing the other set of variables. If the weights v_k are given, the problem becomes a standard spectral clustering problem (Equation 1) and the affinities are set as $w_{ij} = \sum_k v_k^2 w_{ij;k}$. This can be done by finding the eigenvectors of the Laplacian matrix as reviewed in the previous section.

On the other hand, assume that the indicator vector $\mathbf{f}$ is given and fixed. From the definitions of α_k and β_k, it can be seen that $\gamma_k = \frac{\beta_k}{\alpha_k}$ represents the normalized spectral clustering error obtained by using a single (*i.e.*, the k-th) kernel when the indicator $\mathbf{f}$ is given. Denote $u_k = \sqrt{\alpha_k} v_k$, Equation 2 becomes $\sum_k \beta_k v_k^2 = \sum_k \gamma_k u_k^2$. In summary, the goal of AASC when $\mathbf{f}$ is fixed is conducted as

$$\min_{u_1,\ldots,u_m} \sum_k \gamma_k u_k^2$$

subject to

$$\sum_k u_k^2 = 1, \tag{3}$$

$$\sum_k \frac{u_k}{\sqrt{\alpha_k}} = 1. \tag{4}$$

It leads to a constrained optimization problem. This problem is nonconvex since it contains quadratic equality constraints in the formulation. It forms a nonlinear simultaneous equations system in the m-dimensional space. The following shows that this problem can be reduced to a 1-D search (or line-search) problem no matter m is, which is easy to solve by existing packages. By applying Lagrange multipliers λ_1 and λ_2 to the equality constraints, we have

$$J_{\lambda_1,\lambda_2} = \sum_k \gamma_k u_k^2 - \lambda_1 \left(\sum_k u_k^2 - 1 \right) - 2\lambda_2 \left(\sum_k \frac{u_k}{\sqrt{\alpha_k}} - 1 \right) \tag{5}$$

By taking its partial derivatives with respect to u_k and setting them to zero, we have

$$\frac{1}{2} \frac{\partial J_\lambda}{\partial u_k} = \gamma_k u_k - \lambda_1 u_k - \frac{\lambda_2}{\sqrt{\alpha_k}} = 0.$$

Note that the above equation is linear to u_k. Hence, u_k can be analytically represented as an explicit form:

$$u_k = \frac{\lambda_2}{\sqrt{\alpha_k}\left(\gamma_k - \lambda_1\right)}.$$

Substituting the above into Equation 4, we obtain

$$\sum_k \frac{u_k}{\sqrt{\alpha_k}} = \sum_k \frac{\lambda_2}{\left(\gamma_k - \lambda_1\right)\alpha_k} = 1$$

which implies that

$$\lambda_2 = \frac{1}{\sum_k \frac{1}{(\gamma_k - \lambda_1)\alpha_k}}. \tag{6}$$

Furthermore, from Equation 3, we obtain

$$\sum_k u_k^2 = \sum_k \frac{\lambda_2^2}{\left(\gamma_k - \lambda_1\right)^2 \alpha_k} = 1,$$

Algorithm 1 Affinity Aggregation Spectral Clustering (AASC). Given a set of n data points $\mathbf{x}_i$, a set of m affinities $\mathbf{W}_k$ and the desired number of clusters C, find a proper weight assignment v_k to affinities and cluster the data into C clusters.

procedure AASC(Data $\mathbf{x}_i$, Affinities $\mathbf{W}_k$, Number C)
  Initialize the weights as $v_k = 1/m$
  repeat
    ▷ *fix weights v_k and find indicators $\mathbf{f}_i$*
    form the aggregated affinity matrix $\mathbf{W}$ with $w_{ij} = \sum_k v_k^2 w_{ij;k}$ and the diagonal matrix $\mathbf{D}$
    find eigenvectors $\mathbf{v}_2, \ldots, \mathbf{v}_{C+1}$ of generalized eigenproblem $\mathbf{L}u = \lambda \mathbf{D}u$ corresponding to eigenvalues $\lambda_2, \ldots, \lambda_{C+1}$
    indicator $\mathbf{f}_i$ = the i-th row of $[\mathbf{v}_2 \cdots \mathbf{v}_{C+1}]$
    ▷ *fix indicators $\mathbf{f}_i$ and find weights v_k*
    let $\alpha_k = \mathbf{f}^T \mathbf{D}_k \mathbf{f}$,$\beta_k = \mathbf{f}^T(\mathbf{D}_k - \mathbf{W}_k)\mathbf{f}$,$\gamma_k = \frac{\beta_k}{\alpha_k}$
    solve a 1-D search problem of λ_1 in Equation 8
    obtain λ_2 by substituting λ_1 into Equation 6
    weight $v_k = \frac{\lambda_2}{(\gamma_k - \lambda_1)\alpha_k}$
  until convergence
  run k-means on $\mathbf{f}_1, \ldots, \mathbf{f}_n$ to cluster data into C groups
end procedure

which implies that

$$\lambda_2^2 = \frac{1}{\sum_k \frac{1}{(\gamma_k - \lambda_1)^2 \alpha_k}}. \tag{7}$$

Replacing λ_2 in Equation 7 by Equation 6, we have

$$\left(\sum_k \frac{1}{\left(\gamma_k - \lambda_1\right)\alpha_k} \right)^2 = \sum_k \frac{1}{\left(\gamma_k - \lambda_1\right)^2 \alpha_k}. \tag{8}$$

Equation 8 has only a single variable λ_1. Thus, the problem becomes a 1-D line-search problem which can be solved easily. Many approaches have been proposed to solve the 1-D optimization or equation-solving problem [2]. For example, the shuffled complex evolution (SCE-UA) method [10] has been shown to have good performance in solving the problem of irregular functions . Gradient-based approaches such as the Newton Raphson method with automatic step-size selection (such as the Armjjo step-size rule) can also solve the problem well [4].

In sum, we can solve the AASC problem (Equation 2) using a two-step iterative algorithm which alternatively finds the optimal weights v_k and the optimal indicators $\mathbf{f}_i$. More specifically, the iterations alternate between closed-form solution (eigen vectors) and 1-D search. Given the initial weights v_k, in the first step, we set the affinity as $w_{ij} = \sum_k v_k^2 w_{ij;k}$ and use standard spectral clustering to find the optimal indicator $\mathbf{f}_i$. Next, in the second step, the indicators $\mathbf{f}_i$ are fixed and we refine the weights v_k by solving a 1-D search problem in Equation 8. The convergence

is ensured since the objective in Equation 2 is minimized by solving the weights $\mathbf{v}$ under the constraints (Equation 3 and 4), and also minimized by solving indicator $\mathbf{f}$ under the constraint that $\mathbf{f}$ is orthogonal to the constant-one-vector. Thus, alternatively finding $\mathbf{v}$ and $\mathbf{f}$ keeps reducing the error, ensuring convergence of the iterative process. Algorithm 1 summarizes the proposed AASC algorithm.

5. Experiments

We have implemented and tested the proposed AASC algorithm on a variety of clustering problems including image clustering, face clustering and text clustering. This section starts by describing the procedure for calculating similarity and the adopted metrics for comparing clustering results (Section 5.1). For image clustering (Section 5.2), we used two benchmark datasets, Caltech-101 [11] and Microsoft Research Cambridge Volume 1(MSRC-v1) [30]. Two well-known face databases from ORL [24] and CMU-PIE [26] were used for face clustering (Section 5.3). As for text clustering (Section 5.4), we adopted two data sets from 20 Newsgroups and Reuters-21578. Statistics of these data sets are summarized in Table 1, including the number of instances, dimensionality of data and the number of clusters. For each set of experiments, we describe the data sets, the experimental settings, the choice of pairwise affinities, the experimental results and comparisons to other methods.

5.1. Settings and measures

We first describe how to obtain the affinity matrix for each type of feature. Given the raw data in the data set, the first step is to extract features for each instance. Each feature can be represented as a vector. These feature vectors were substituted into the Gaussian kernel to calculate pairwise distances,

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)/\sigma).$$

Assume that the minimal value of the Gaussian kernel over the data set is γ. We then obtain the corresponding σ as

$$\sigma = \min_{i,j}(-(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)/log(\gamma)).$$

and we set γ to 0.005.

For comparing clustering results, clustering measures were used to evaluate how well data are grouped in comparison with the ground truth. Clustering measures can be roughly categorized into pair-counting-based measures (*e.g*. Rand index (RI) and adjusted Rand index (ARI) [13]), set-matching-based measures (*e.g*. $\mathcal{H}$ criterion) and information-theoretic-based measures (*e.g*. mutual information and normalized mutual information (NMI) [28]). Several papers have attempted to evaluate these clustering measures. Unfortunately, there is no definite answer on which measure is the best yet. Vinh *et al*. [29] reported that some popular measures do not facilitate informative clustering comparisons because they either do not have a predetermined range or do not have a constant baseline value. For those measures, a poor clustering could yield a very high performance index, especially when there are many clusters. They suggested that ARI is a faithful measure that does not have these drawbacks. They also proposed another fair measure, adjusted mutual information (AMI). However, Wu *et al*. [32] reported that, when clustering performances are hard to distinguish, the normalized variation of mutual information, *i.e*. NMI, could still work the best. For fair comparisons, this paper uses AMI, NMI and ARI as metrics for reporting clustering performance.

Table 1. Statistics of the data sets used in the experiments. The first two data sets are adopted from Caltech-101 with seven and twenty classes. Along with the third data set MSRC-v1, these three sets were used for image clustering. For face clustering, two face databases from ORL and CMU-PIE were used. For CMU-PIE, we used the frontal images (Pose 27) with 22 different lightings. The last two are text data sets from 20 Newsgroups and Reuters-21578. For 20 Newsgroups, we randomly chose 100 instances from each class in the training set. For Reuters-21578, we used the test set of R52.

ID	Name	#instances	#dimensions	#classes
I1	*Caltech-101*	441	100,000	7
I2	*Caltech-101*	1,230	100,000	20
I3	*MSRC-v1*	210	100,000	7
F1	*ORL*	360	7,744	40
F2	*CMU-PIE*	1,496	7,744	68
T1	*20 Newsgroups*	2,000	25,753	20
T2	*Reuters-21578*	2,568	8,575	52

5.2. Image clustering

In order to compare AASC to MMSC [6], we used the same data sets Caltech-101 and MSRC-v1. For Caltech-101, we follow MMSC to choose the same 7 and 20 classes. For MSRC-v1, the same 7 classes were obtained in the same way as MMSC. As MMSC, five types of features were used, LBP [21], GIST [22], CENTRIST [31], Dog-SIFT [18], and HOG [9]. We denote SC_L, SC_G, SC_C, SC_D and SC_H as the single-affinity spectral clustering methods with five different affinity matrices derived from the above five features (LBP, GIST, CENTRIST, Dog-SIFT, and HOG), respectively. In addition, we also combined the above five affinity matrices by equal weights and denoted it as EASC. Tables 2, 3 and 4 show AMI, NMI and ARI values for different algorithms on Caltech-101 (7 classes), Caltech-101 (20 classes) and MSRC-v1. As the results show, the proposed AASC method has better performance than other methods. Note that, the performance of our single-affinity spectral clustering methods are not exactly the same with the ones listed in the MMSC paper [6] due to implemen-

Table 2. Comparisons of different methods on Caltech-101 (7 classes) in terms of AMI, NMI and ARI.

	AMI	**NMI**	**ARI**
SC_L	0.3746	0.4011	0.2631
SC_G	0.4958	0.5260	0.4289
SC_C	0.4488	0.4713	0.3280
SC_D	0.5313	0.5683	0.4051
SC_H	0.3928	0.4503	0.1768
EASC	0.6412	0.6614	0.5471
MMSC	N/A	0.6792	N/A
AASC	0.6747	0.6853	0.6692

Table 3. Comparisons of different methods on Caltech-101 (20 classes) in terms of AMI, NMI and ARI.

	AMI	**NMI**	**ARI**
SC_L	0.3673	0.4247	0.2841
SC_G	0.5100	0.5467	0.3799
SC_C	0.3356	0.4033	0.2182
SC_D	0.5718	0.5995	0.4471
SC_H	0.3967	0.4522	0.2071
EASC	0.5926	0.6210	0.4528
MMSC	N/A	0.6329	N/A
AASC	0.6202	0.6458	0.5110

Table 4. Comparisons of different methods on MSRC-v1 in terms of AMI, NMI and ARI.

	AMI	**NMI**	**ARI**
SC_L	0.4094	0.4474	0.2733
SC_G	0.5656	0.6113	0.4189
SC_C	0.5410	0.5702	0.4515
SC_D	0.5561	0.6053	0.4064
SC_H	0.4821	0.5189	0.3284
EASC	0.7428	0.7578	0.7156
MMSC	N/A	0.7745	N/A
AASC	0.7588	0.7806	0.7244

tation details. However, the performance of equal weight combination is very close. Thus, to some extent, it can be regarded as a fair comparison.

5.3. Face clustering

We have also evaluated AASC on face clustering. The face databases are from ORL and CMU-PIE. The face images are all nearly frontal; those in ORL include various facial expressions and those in CMU PIE include variable lighting conditions. All images were first normalized and cropped to 88×88 in resolution. To utilize cues from different perspectives, we extracted three different features.

1. Eigenface [5]. After performing principal component analysis, each face image was projected into the eigenspace which preserves 90% of the energy of the eigenvalues.

Table 5. Comparisons of different methods on face database ORL in terms of AMI, NMI and ARI.

	AMI	**NMI**	**ARI**
SC_e	0.5613	0.7630	0.3270
SC_g	0.6015	0.7848	0.3897
SC_l	0.5598	0.7539	0.4001
EASC	0.7377	0.8639	0.5629
AASC	0.7687	0.8820	0.6187

Table 6. Comparisons of different methods on face database CMU in terms of AMI, NMI and ARI.

	AMI	**NMI**	**ARI**
SC_e	0.8049	0.9108	0.3390
SC_g	0.7928	0.9027	0.3533
SC_l	0.7574	0.8490	0.5181
EASC	0.8347	0.9196	0.5538
AASC	0.8506	0.9310	0.5695

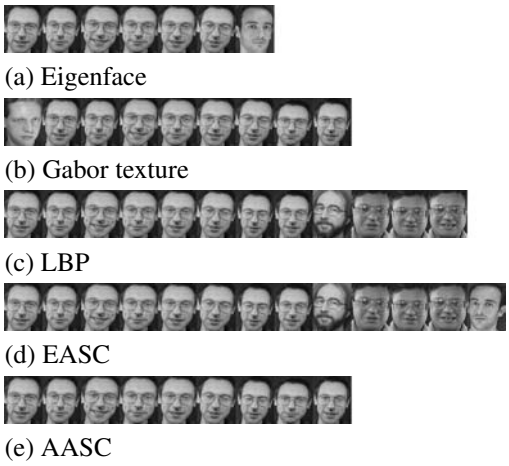
2. Gabor texture [19]. Each face image was filtered with 40 Gabor filters generated with five different scales and eight orientations.

3. Local binary pattern (LBP) [21]. We used a uniform LBP with 8 neighbors and radius 1. Thus, each face image was represented as a 256-bin histogram.

These three features are frequently used for face recognition and represent face images from different perspectives. We denote SC_e, SC_g, and SC_l as the spectral clustering methods with three different affinity matrices derived from these three features (Eigenface, Gabor texture, and LBP), respectively. Tables 5 and 6 show AMI, NMI and ARI values for different algorithms on these two face data sets. Note that faces in ORL exhibit facial expressions while CMU-PIE has more variations in illumination. Thus, Gabor is more effective in ORL and eigenface performs better for CMU-PIE. This is evident from Table 5 in which SC_g is the best among three single-affinity SCs for ORL while SC_e is the best for CMU-PIE. We also show visual clustering results in Figure 1 and Figure 2 for ORL and CMU-PIE, respectively. We can see that AASC produced better clustering results than other methods. AASC correctly grouped photos of a subject into a cluster while other methods either wrongly included photos of other subjects or left out some photos of the subject. Without knowing the characteristics of the databases, AASC successfully combined the strengths of different features and outperformed all other methods for both data sets.

5.4. Text clustering

For text clustering, we used two popular text data sets, 20 Newsgroups and Reuters-21578, downloaded from [7]. Each of them is pre-processed by four steps: *all-terms*, *no-short*, *no-stop* and *stemmed*. We use the data sets *20ng-*



(a) Eigenface

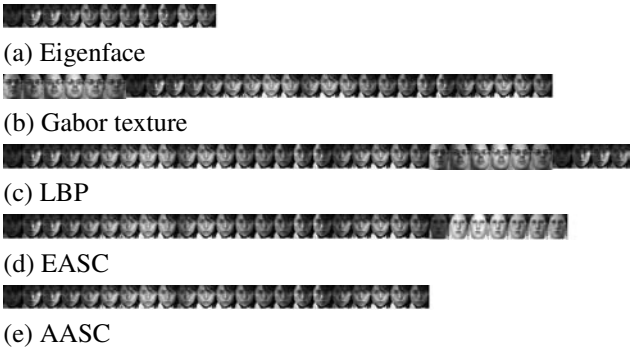
(b) Gabor texture

(c) LBP

(d) EASC

(e) AASC

Figure 1. The visual clustering performance of different methods for ORL data set. AASC correctly grouped photos of a subject into a cluster while other methods either wrongly included photos of other subjects or left out some photos of the subject.



(a) Eigenface

(b) Gabor texture

(c) LBP

(d) EASC

(e) AASC

Figure 2. The visual clustering performance of different methods for CMU data set. AASC correctly grouped photos of a subject into a cluster while other methods either wrongly included photos of other subjects or left out some photos of the subject.

train-stemmed and *r52-test-stemmed* to evaluate AASC. Let $D = \{d_1, \cdots, d_n\}$ be the set of documents and $T = \{t_1, \cdots, t_m\}$ the set of distinct words occurring in D. We denote the frequency of word $t \in T$ in the document $d \in D$ as $tf\,(d,t)$. tf-idf is a weighting scheme which weights the frequency of a word t in the document d with a factor that discounts its importance with its occurrences in the whole document collection, which is defined as

$$tf\text{-}idf\,(d,t) = tf\,(d,t) \times log(\frac{|D|}{df\,(t)}),$$

where $df\,(t)$ is the number of documents in which the word t appears. Thus, the feature vector representation of a document d is defined as

$$\overrightarrow{t_d} = \left(tf\text{-}idf\,(d,t_1), \cdots, tf\text{-}idf\,(d,t_m)\right).$$

Table 7. Comparisons of different methods on text data set 20 Newsgroups in terms of AMI, NMI and ARI.

	AMI	**NMI**	**ARI**
SC_{ed}	0.5230	0.5749	0.2868
SC_{cs}	0.5212	0.5723	0.2816
SC_{jc}	0.5237	0.5762	0.2854
SC_{pcc}	0.5152	0.5661	0.2779
EASC	0.5199	0.5690	0.2814
AASC	0.5340	0.5840	0.3003

Table 8. Comparisons of different methods on text data set Reuters-21578 in terms of AMI, NMI and ARI.

	AMI	**NMI**	**ARI**
SC_{ed}	0.3585	0.5151	0.1854
SC_{cs}	0.3519	0.5086	0.1828
SC_{jc}	0.3671	0.5208	0.1967
SC_{pcc}	0.3506	0.5085	0.1805
EASC	0.3555	0.5104	0.1861
AASC	0.3695	0.5213	0.2096

After normalizing the vectors to a unit length, we used the following four metrics to calculate the pairwise distances between two documents: Euclidean distance, Cosine similarity, Jaccard coefficient and Pearson correlation coefficient, which measure distance between feature vectors of the i-th and j-th documents $\overrightarrow{t_{d_i}}$ and $\overrightarrow{t_{d_j}}$ in different ways.

We denote as SC_{ed}, SC_{cs}, SC_{jc} and SC_{pcc} the spectral clustering with these four affinity matrices, respectively. Tables 7 and 8 show the AMI, NMI and ARI for 20 Newsgroups and Reuters-21578, respectively. Note that documents are represented with the bag-of-word model and these four affinity metrics only define different ways to measure distances. Thus, they have similar clustering capability. Nevertheless, AASC is still able to assign the weights appropriately to improve the clustering performance.

6. Conclusions

We have extended the spectral clustering algorithm to the setting where there are multiple affinities available. Our method can explore strengths of different features automatically and weight them properly. Experiments show that it effectively incorporates multiple affinities and yields better performance compared to spectral clustering with only a single affinity or naive feature fusion strategies. In addition, it outperforms the previous method such as [6], and does not suffer from the out-of-sample problem. Furthermore, it is easy to implement, since only the computation of eigenvectors and 1-D search are involved. These characteristics make it useful for real-world applications. In the future, we will work on strategies for choosing the basis kernels to calculate pairwise distances.

Acknowledgments

This work was supported in part by the National Science Council of Taiwan, R.O.C., under Grants NSC98-2221-E-001-012-MY3 and NSC100-2628-E-002-009.

References

[1] D. Anguelov, K. Lee, S. Gokturk, and B. Sumengen. Contextual identity recognition in personal photo albums. In *Proceedings of IEEE CVPR*, pages 1–7, 2007.

[2] A. Antoniou and W.-S. Lu. *Practical Optimization: Algorithms and Electrical Applications*. Springer-Verlag, 2007.

[3] F. R. Bach and M. I. Jordan. Learning spectral clustering. In *Proceedings of NIPS*, 2003.

[4] R. Baldick. *Applied Optimization: Formulation and Algorithms for Engineering Systems*. Cambridge University Press, 2009.

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):711–720, 1997.

[6] X. Cai, F. Nie, H. Huang, and F. Kamangar. Heterogeneous image feature integration via multi-modal spectral clustering. In *Proceedings of IEEE CVPR*, pages 1977–1984, 2011.

[7] A. Cardoso-Cachopo. Datasets for single-label text categorization.

[8] C. Cortes, M. Mohri, and A. Rostamizadeh. Learning nonlinear combinations of kernels. In *Proceedings of NIPS*, pages 396–404, 2009.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE CVPR*, pages 886–893, 2005.

[10] Q. Y. Duan, V. K. Gupta, and S. Sorooshian. Shuffled complex evolution approach for effective and efficient global minimization. *J. Optim. Theory Appl.*, 76:501–521, 1993.

[11] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Workshop on Generative-Model Based Vision*, 2004.

[12] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen. Multi-affinity spectral clustering. In *Proceedings of IEEE ICASSP 2012*, pages 2089–2092, March 2012.

[13] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[14] I. Karydis, A. Nanopoulos, H.-H. Gabriel, and M. Spiliopoulou. Tag-aware spectral clustering of music items. In *Proceedings of ISMIR 2009*, pages 159–164, 2009.

[15] A. Kumar and H. D. III. A co-training approach for multiview spectral clustering. In *Proceedings of ICML*, pages 393–400, 2011.

[16] A. Kumar, P. Rai, and H. D. III. Co-regularized multi-view spectral clustering. In *Proceedings of NIPS*, pages 1413–1421, 2011.

[17] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh. Multiple kernel learning for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(6):1147–1160, 2011.

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[19] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, 1996.

[20] D. Niu, J. G. Dy, and M. I. Jordan. Multiple non-redundant spectral clustering views. In *Proceedings of ICML*, pages 831–838, 2010.

[21] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002.

[22] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42:145–175, 2001.

[23] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. In *Proceedings of ICML*, pages 775–782, 2007.

[24] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of WACV*, pages 138–142, 1994.

[25] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[26] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(12):1615–1618, 2003.

[27] Y. Song and T. Leung. Context-aided human recognition: Clustering. In *Proceedings of ECCV*, pages III: 382–395, 2006.

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

[29] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of ICML*, pages 1073–1080, 2009.

[30] J. Winn. Locus: Learning object classes with unsupervised segmentation. In *Proceedings of ICCV 2005*, pages 756–763, 2005.

[31] J. Wu and J. M. Rehg. Where am I: Place instance and category recognition using spatial PACT. In *Proceedings of IEEE CVPR*, 2008.

[32] J. Wu, H. Xiong, and J. Chen. Adapting the right measures for k-means clustering. In *Proceedings of SIGKDD*, pages 877–886, 2009.

[33] B. Zhao, J. Kwok, and C. Zhang. Multiple kernel clustering. In *Proceedings of SDM*, pages 638–649, 2009.

[34] D. Zhou and C. J. C. Burges. Spectral clustering and transductive learning with multiple views. In *Proceedings of ICML*, pages 1159–1166, 2007.

[35] A. Zien and C. S. Ong. Multiclass multiple kernel learning. In *Proceedings of ICML*, pages 1191–1198, 2007.