GOMES: A GROUP-AWARE MULTI-VIEW FUSION APPROACH TOWARDS REAL-WORLD IMAGE CLUSTERING

Zhe Xue[†]*, Guorong Li*[†]*, Shuhui Wang*[‡]*, Chunjie Zhang*[†]*, Weigang Zhang*[♯]*, Qingming Huang*[†‡]

[†]Key Lab of Big Data Mining and Knowledge Management, University of Chinese Academy of Sciences
[‡]Key Lab. of Intell. Info. Process., Inst. of Comput. Tech., Chinese Academy of Sciences
[♯]School of Computer Science and Technology, Harbin Institute of Technology at Weihai
{zhe.xue, guorong.li, chunjie.zhang, weigang.zhang}@vipl.ict.ac.cn, {wangshuhui, qmhuang}@ict.ac.cn

ABSTRACT

Different features describe different views of visual appearance, multi-view based methods can integrate the information contained in each view and improve the image clustering performance. Most of the existing methods assume that the importance of one type of feature is the same to all the data. However, the visual appearance of images are different, so the description abilities of different features vary with different images. To solve this problem, we propose a group-aware multi-view fusion approach. Images are partitioned into groups which consist of several images sharing similar visual appearance. We assign different weights to evaluate the pairwise similarity between different groups. Then the clustering results and the fusion weights are learned by an iterative optimization procedure. Experimental results indicate that our approach achieves promising clustering performance compared with the existing methods.

Index Terms— multi-view learning, group-aware fusion, image clustering

1. INTRODUCTION

With the development of digital equipments and the Internet technology, it is more convenient for people to share their multimedia data through the Internet. Unsupervised image categorization can automatically discover categories and meaningful hierarchical structures from a collection of images. It facilitates better organization of the Web multimedia data and diversified online applications, and has drawn considerable attention [1, 2].

However, the uncontrolled appearance variation in real-world images brought by different light and viewing conditions would make it difficult to derive the true semantic correlation using single visual descriptor. A practical solution is to integrate different visual descriptors to generate a more robust and accurate representation. Multiple kernel learning (MKL) tries to learn a unified kernel by combining multiple kernels together to exploit different visual features, and some clustering methods [3, 4] are proposed based on MKL. Constructing a graph for each type of feature (view), several graph based multi-view learning (MVL) methods are proposed. Some of them [5, 6] obtain the clustering results by enforcing the cluster structures of each view consensus. Besides, some of them [7, 8] jointly learn the optimal combination of each single view graph and the clustering results. Since the affinity matrix is not required to be positive semi-definite, the graph based MVL methods obtain a wider applicability than MKL and achieve good performance.

It should be noted that these methods basically adopt a globally uniform similarity measure over the whole data space. But for the real-world images, the visual distribution is complicated and different images have different visual appearance. It is difficult to describe the similarity accurately by using a globally uniform measure. Instead of the global fusion methods, some local fusion methods are proposed in supervised learning [9, 10]. The weights of features are learned according to the visual appearance of images, which generates more accurate descriptions for images. However, the complex distribution of real-world images has not drawn enough attention in previous multi-view clustering methods, which consequently limits their performance in clustering real-world images.

To address this issue, we propose GOMES, a **G**r**O**up-aware **M**ulti-vi**E**w fu**S**ion approach towards real-world image clustering. The key point of GOMES is that images are partitioned into *groups* with more compact visual cohesiveness, and the images within a group share the same fusion weights. Compared with the global fusion methods, our group-aware fusion approach provides a more flexible fusion strategy and more accurately measures the similarity among images. The framework of GOMES is shown in Fig.1.

First we extract multiple features from images and con-

This work was supported in part by National Basic Research Program of China (973 Program): 2012CB316400 and 2015CB351802, National Natural Science Foundation of China: 61303153, 61303154, 61025011, 61202322, 61332016, 61390511 and 61303160, 863 program of China: 2014AA015202, and China Postdoctoral Science Foundation: 2014T70111 and 2014T70126.

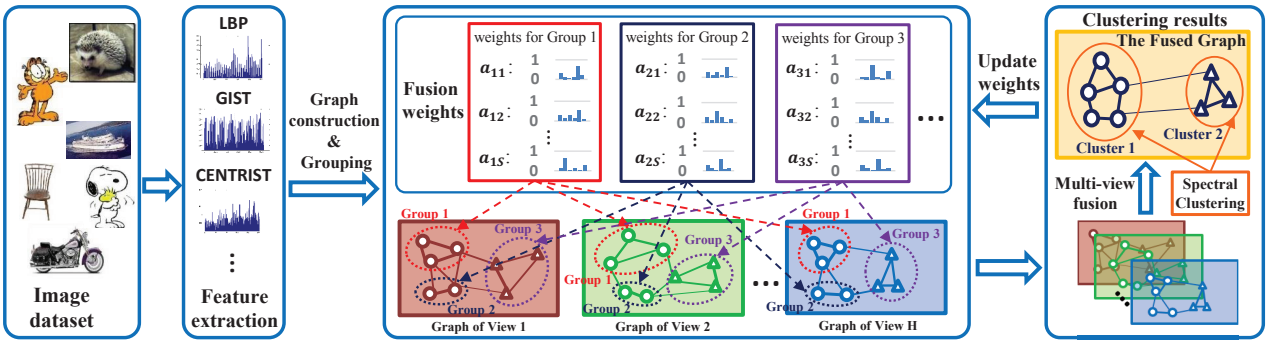

Fig. 1. The framework of GOMES.

struct a graph for each feature (view) to represent the multi-view data. Next, the images are partitioned into groups, and the fused graph is constructed by the proposed group-aware fusion strategy: the images belonging to different groups are assigned with different fusion weights. Finally, the fusion weights are learned by solving an optimization problem, and the clustering results are obtained by conducting spectral clustering on the fused graph. An iterative optimization process is proposed to jointly learn the fusion weights and the clustering results. In summary, our main contributions are:

- A group-aware fusion approach is proposed for multi-view clustering. A group can be recognized as the intermediated representation between image categories and individual images, which contains several images with similar visual appearance. By introducing group structure, the weights can vary with the visual appearance of different images and more accurate similarities between images can be obtained.
- Two reasonable and effective criterions are designed for learning the fusion weights. An iterative optimization algorithm is proposed to jointly learn the spectral clustering results and fusion weights.

2. RELATED WORK

Multiple kernel learning (MKL) and multi-view learning (MVL) are two related works that can exploit multiple features to generate a more effective representation. MKL aims to combine multiple kernels to create an optimal kernel [4, 11]. Lin *et al.* [4] generalize the framework of MKL for dimensionality reduction, which provides convenience of using multiple types of image features. Centered kernel alignment is employed in [11] to unify the two tasks of clustering and MKL into a coherent optimization problem.

To infer the clustering results for multi-view data, MVL assumes that the underlying clustering would assign corresponding samples in each view to the same cluster. Some works [5, 6, 12] learn the underlying clustering structure from multiple views by regularizing the embedding of each view towards a common consensus. Besides, some methods [7, 8] first construct a graph for each view, and then fuse each graph into a better one. By assigning weights for each graph according to their importance, their methods are more immune to the ineffective views.

Nevertheless, the above methods basically adopt globally uniform similarity measure for the whole input space. Due to the complicated distribution of data, the importance of different views for discrimination may vary with different data. Considering this point, some local fusion methods are proposed in [10, 13, 14] and achieve better performance. GS-MKL [10] constructs a set of groups, in which the images share similar visual property. Then fusion weights are learned for each group to deal with the intra-class diversity and inter-class correlation for image classification. For unsupervised tasks, a local fusion based multiple kernel clustering method is proposed in [14]. Each cluster is learned with a localized kernel for similarity evaluation. However, the sparsity of the fusion weights can not be controlled explicitly which limits the fusion ability. In addition, the kernels are learned to discriminate each cluster from the rest of samples. Different from this method, we adopt a pairwise strategy to learn the fusion weights between any two groups, which can better obtain the weights according to the distribution properties of samples.

3. GOMES

The adopted spectral clustering method is reviewed in Section 3.1. Section 3.2 presents the proposed group-aware multi-view fusion approach which aims to fuse each single view graph into a fused graph. Section 3.3 introduces how to partition the samples into groups. Finally, the method of fusion weights learning is provided in Section 3.4.

3.1. Non-negative Spectral Clustering

Although the traditional spectral clustering method is widely applied, it still has some disadvantages. The spectral solution has mix signs, which may severely deviate from the true solution. Moreover, it usually needs to resort to other clustering

methods to obtain the clustering labels. To solve these problems, Ding *et al.* [15] propose non-negative spectral clustering (NNSC) which improves the clustering performance and obtains the clustering label directly. Given the affinity matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$, $\mathbf{D}$ is a diagonal matrix with the diagonal elements are defined as $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. Then the NNSC is formulated as follows,

$$\begin{aligned} &\max_{\mathbf{F}} Tr(\mathbf{F}^{\mathrm{T}}\mathbf{W}\mathbf{F}) \\ &s.t.\ \mathbf{F}^{\mathrm{T}}\mathbf{D}\mathbf{F} = \mathbf{I},\ \ \mathbf{F} \geq 0 \end{aligned} \tag{1}$$

where $Tr(\mathbf{M})$ is the trace of matrix $\mathbf{M}$, $\mathbf{F} \in \mathbb{R}^{N\times Y}$ is the cluster indicator matrix and Y is the number of clusters. The cluster label of each sample is obtained by the column number of the maximum value of each row in $\mathbf{F}$. Problem (1) can be solved by the following update rule whose correctness and convergence has been proved rigorously,

$$\mathbf{F}_{ij} \leftarrow \mathbf{F}_{ij}\sqrt{\frac{(\mathbf{W}\mathbf{F})_{ij}}{(\mathbf{D}\mathbf{F}\mathbf{F}^{\mathrm{T}}\mathbf{W}\mathbf{F})_{ij}}} \tag{2}$$

3.2. Group-aware Multi-view Fusion

Given N image samples with H views, we construct a graph for each view and obtain H graphs $G^h = (\mathcal{V}, \mathcal{E}^h)$, $h \in \{1, 2, ..., H\}$. $\mathcal{V} = \{v_1, ..., v_N\}$ is the vertex set and $\mathcal{E}^h$ is the edge set. Each vertex v_i in $\mathcal{V}$ represents an image, and all the graphs share the same vertex set $\mathcal{V}$. $\mathbf{W}^h \in \mathbb{R}^{N\times N}$ denotes the affinity matrix of G^h, and its entries are w^h_{ij}, which represents the similarity between images i and j in view h.

Our objective is to fuse all the single view graphs $\{G^h\}_{h=1}^H$ into a fused graph $G^t = (\mathcal{V}, \mathcal{E}^t)$. G^t shares the same vertex set $\mathcal{V}$ with the single view graphs $\{G^h\}_{h=1}^H$. We only need to determine the affinity matrix $\mathbf{W}^t$ and the following group-aware fusion approach is proposed. The images are first partitioned into S non-overlapping groups: $Z = \{Z_1, Z_2, ..., Z_S\}$ where Z_i is a set of images that belong to the i-th group. We introduce weight $a^h_{ij} \in [0, 1]$ to represent the importance of view h when describing the similarity between the images belonging to group Z_i and Z_j. These weights can be denoted by H matrices $\mathbf{A}^h = \{a^h_{ij}, 1 \leq i, j \leq S\}, h \in \{1, 2, ..., H\}$ and we have $a^h_{ij} = a^h_{ji}, \forall i, j \in \{1, 2, ...S\}, h \in \{1, 2, ..., H\}$. Then the similarity between two vertices $p \in Z_i$ and $q \in Z_j$ on G^t is calculated by $w^t_{pq} = \sum_{h=1}^H a^h_{ij} w^h_{pq}$.

We assume that the images belong to C classes. After G^t is constructed, we conduct NNSC on G^t to partition the images into C clusters and the *class label* of each image is learned. Thus we obtain the clustering results of images.

3.3. Grouping Strategy

To partition images into groups so that the images in the same group share similar visual appearance without any prior knowledge, clustering methods are feasible. Although many clustering methods can be used here, which one is the most optimal is not the focus of this paper. We adopt NNSC for its convenience and effectiveness. By conducting NNSC on G^t, each obtained image cluster is treated as a group. Because we have no prior knowledge about the number of groups S, it is empirically identified. As a group is a subset of an image category, the number of groups should not be less than the number of classes C. However, if the group number is set to the maximum value N (the number of total images), the computation cost is too expensive and clustering performance could also be affected by this over-segmentation of image groups. So the appropriate value should be between the two bounds, i.e., $C < S < N$.

Since G^t is updated according to the fusion weights during learning, some of the groups are unreliable and we adjust the groups in each iteration. We first determine the *group label* of each group, i.e., the class label of that group by a vote from the class labels of the images within it. Then the image whose class label is different from its group label, is assigned to the other group, which belongs to the same class and contains its nearest element.

3.4. Weights Learning

We want to assign the fusion weights to each view according to their importance during multi-view fusion. The views generating more accurate descriptions for two groups should be assigned with higher weights. So how to evaluate the accuracy of descriptions generated by each view is the critical problem. We propose two criterions for evaluation: the *consensus* criterion and the *discrimination* criterion.

The *consensus* criterion is based on the assumption that a sample in different views would be assigned to the same cluster with high probability. This criterion is commonly adopted in multi-view global fusion methods [7, 8]. However, how to use this criterion to identify the reliable views between any two groups needs to be designed. Here, we consider that the view which generates clustering results closer to the results obtained by the fused graph, is more reliable. The following cost function is proposed to measure the disagreement between the clustering results generated by G^t and $\{G^h\}_{h=1}^H$:

$$Con(h, i, j) = \frac{\left\|\mathbf{F}^t_i {\mathbf{F}^t_j}^{\mathrm{T}} - \mathbf{F}^h_i {\mathbf{F}^h_j}^{\mathrm{T}}\right\|^2}{\sum_{h=1}^H \left\|\mathbf{F}^t_i {\mathbf{F}^t_j}^{\mathrm{T}} - \mathbf{F}^h_i {\mathbf{F}^h_j}^{\mathrm{T}}\right\|^2} \tag{3}$$

where $\|\cdot\|$ is the frobenius norm. $\mathbf{F}^t \in \mathbb{R}^{N\times C}$ and $\mathbf{F}^h \in \mathbb{R}^{N\times C}$ are the cluster indicator matrices which are obtained by conducting NNSC on the fused graph G^t and the h-th view graph G^h respectively. $\mathbf{F}^t_i$ ($\mathbf{F}^h_i$) is the cluster indicator matrix of images belonging to group Z_i, which is composed of the related rows of $\mathbf{F}^t$ ($\mathbf{F}^h$). The numerator of equation (3) is the difference of the clustering results between data within group Z_i and Z_j measured by G^t and G^h, and the denominator is

used for normalization so that its range is $[0, 1]$. The view h with smaller value of Con is more important and would obtain higher weights.

The *discrimination* criterion is to select the appropriate views that would separate the samples belonging to different categories and aggregate samples in the same category. For groups belonging to different categories, we are more inclined to select the views which generate lower similarities. While for groups belonging to the same class, we select the views yielding higher similarities. One can design several cost functions to achieve this criterion and we do not focus on it here. We adopt a direct and convenient method to define the cost function. First we calculate the similarity between two groups Z_i and Z_j measured by each view:

$$Sim(h, i, j) = \sum_{p \in Z_i} \sum_{q \in Z_j} w_{pq}^h, \quad h = 1, 2, ..., H \tag{4}$$

We use $B(Z_i)$ to denote the class label of Z_i. Then the most discriminative view $d \in \{1, 2, ..., H\}$ is selected according to the relationship between the class label of two groups:

$$d = \begin{cases} \arg\max\limits_h \ Sim(h, i, j) & if \ B(Z_i) = \ B(Z_j) \\ \arg\min\limits_h \ Sim(h, i, j) & otherwise \end{cases} \tag{5}$$

Then the cost function $Dis(h, i, j)$ for the two groups Z_i and Z_j is constructed as follows:

$$Dis(h, i, j) = \begin{cases} 0 & if \ h = d \\ 1 & otherwise \end{cases} \tag{6}$$

where we can see that it is more inclined to select the view which provides the similarity that is more consistent with the relationship between the class label of two groups.

Finally, we integrate the above two criterions to learn weights a_{ij}^h and the optimization problem is formulized as:

$$\begin{aligned} &\min \sum_{i=1}^{S} \sum_{j=i}^{S} \sum_{h=1}^{H} (a_{ij}^h)^r \left[\beta Con(h, i, j) + (1 - \beta) Dis(h, i, j)\right] \\ &s.t. \quad \sum_{h=1}^{H} a_{ij}^h = 1, \ 1 \le i \le j \le S \\ &\qquad a_{ij}^h \ge 0, \ 1 \le i \le j \le S, \ h = 1, ..., H \end{aligned} \tag{7}$$

where the parameter $\beta \in [0, 1]$ provides a tradeoff between the two criterions. $r \in [1, \infty)$ is the parameter to control the sparseness of the solution. When $r = 1$, a completely sparse solution emerges, and only one view is selected. While a more smooth solution can be obtained as the value of r become larger ($r > 1$). Problem (7) could be solved by lagrangian multiplier method. The details of GOMES learning procedure are provided in Algorithm 1.

Algorithm 1: The GOMES learning procedure

Input: $\{\mathbf{W}^h\}_{h=1}^H$, β, r, S, C, $iter$
Output: $\mathbf{F}^t$, $\{\mathbf{A}^h\}_{h=1}^H$

Initialize:
  Initialize $\{a_{ij}^h\}_{h=1}^H = \frac{1}{H}, \forall 1 \le i, j \le S$ and G^t;
  Initialize groups Z;
  Initialize $\{\mathbf{F}^h\}_{h=1}^H$ and $\mathbf{F}^t$;
Iterative update variables:
for $i = 1$ *to* $iter$ **do**
  Given $\mathbf{F}^t$, $\{\mathbf{F}^h\}_{h=1}^H$ and Z, update $\{\mathbf{A}^h\}_{h=1}^H$ and G^t;
  Given G^t, update $\mathbf{F}^t$;
  Given $\mathbf{F}^t$, update Z;

4. EXPERIMENTS

4.1. Baseline methods

To evaluate the performance of the proposed multi-view clustering method, we compare it with several baselines:

- **Single view spectral clustering** (SC(#)): conduct spectral clustering [16] on single view graph.
- **Equally combining affinity matrices spectral clustering** (EASC): equally fuses each single view graph adn conduct spectral clustering on this fused graph.
- **Multi-modal spectral clustering** (MMSC): a multi-view spectral clustering method [6]. We report its performance as listed in [7, 12].
- **Affinity aggregation spectral clustering** (AASC): a spectral clustering method which simultaneously learns the fusion weights and the clustering results [7].
- **Multi-feature spectral clustering with minimax optimization** (MSCMO): a multi-view clustering method based on minimax optimization [12].

4.2. Datasets and experimental settings

We compare our method with the baseline methods to evaluate the clustering performance. To be fair, we select several datasets that are adopted in MMSC, AASC and MSCMO as our datasets: Caltech-101 [17], Microsoft Research Cambridge Volume 1 (MSRC) [18] and Oxford Flowers [19]. For Caltech-101, we follow MMSC and AASC to choose the same 7 and 20 classes as two datasets (Caltech-7 and Caltech-20). For MSRC, the same 7 classes are obtained in the same way as MMSC and AASC. We extract the same 5 types of features for Caltech-101 and MSRC datasets as MMSC and AASC: LBP [20], GIST [21], CENTRIST [22], Dog-SIFT [23] and HOG [24]. Let {SC(1),...,SC(5)} denote spectral

Table 1. Selected parameters on each dataset.

	Caltech-7	Caltech-20	MSRC	Oxford Flowers
r	1.5	2	2	2.5
β	0.55	0.5	0.45	0.35
S	77	120	35	136

clustering with each type of feature respectively. Oxford Flowers dataset is composed of 17 flower classes, with 80 images for each class. As MSCMO, color, shape, and texture features are adopted for describing each image. Let SC(1), SC(2) and SC(3) denote spectral clustering with each type of feature respectively. As adopted in AASC, we evaluate the clustering performance using three measures: adjusted mutual information (AMI), normalized mutual information (NMI) and adjusted Rand index (ARI). For Caltech-101, MSRC and Oxford Flowers, we follow the same way to construct affinity graphs for each view as AASC and MSCMO respectively.

For Caltech-7, Caltech-20 and MSRC, the performance of MMSC and AASC are reported using the results listed in [7]. We realize MSCMO and report its performance. For Oxford flowers, the performance of MMSC and MSCMO are reported as they are listed in [12]. We implement AASC and report its performance using the code which is provided by the authors.

There are several parameters that need to be determined in GOMES. A simple method is searching on the whole parameter space, but this is very time consuming. Instead, we first set $r = 2$ to keep certain smoothness of weights and set $\beta = 0.5$ to treat each criterion equally important. Then searching the number of groups S from $\{2 \times C, 3 \times C, ..., 15 \times C\}$, where C is the number of image classes. After that, we fix S and obtain β and r by a grid-search strategy from $\{0, 0.1, 0.2, ..., 1\}$ and $\{1, 1.5, 2, 2.5, ..., 10, 20, 30, ..., 50\}$ respectively. The selected parameters in our experiments for each dataset are listed in Table 1. In our experiments, the convergence of GOMES basically occurs less in 10 iterations (the clustering results are not changed), so $iter$ is set to 10. Since the the clustering methods contain some random initialization settings, the experiments are repeated ten times and the averaged performance is reported.

4.3. Experimental results and analysis

The experimental results are shown in Table 2. Since the implementation details are different, the performance of single view spectral clustering method SC(#) are not exactly the same with the ones listed in AASC [7] and MSCMO [12]. But the results of equal weight combination EASC are very close which guarantees the fairness of the comparison to some extent. All of the multi-view clustering methods EASC, MMSC, AASC, MSCMO and GOMES obtain better results than the single view method SC(#), which indicates the power of multi-view clustering. The complementary information contained in multi-view data can be used to generate a more accurate and robust description than any single view, which helps improving the clustering performance.

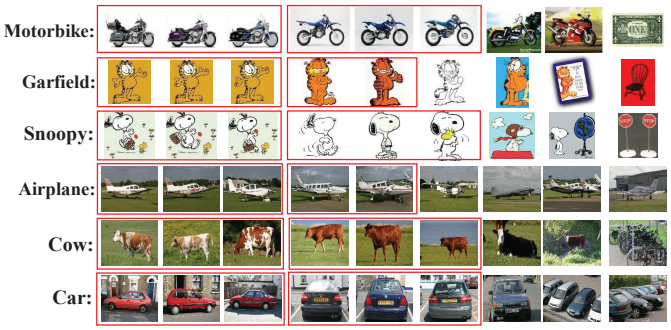

Fig. 2. Some clustering results from Caltech-7 and MSRC. We pick some images from the same cluster and put them in one row. The images belonging to the same group are put in the red box.

By minimizing the difference between the clustering results of each view, MMSC obtains better results than EASC. Nevertheless, EASC and MMSC treat each view equally important and their fusion abilities are limited. Considering different importance of different views, AASC provides a more effective fusion strategy and achieves better performance than MMSC. By minimizing the pairwise disagreement between any two views, MSCMO learns a consensus embedding from multiple views. However, it can not handle well the impacts of the views with poorer performance on the learned embedding. Although MSCMO achieves better performance than MMSC and AASC on Caltech-7 and Oxford flowers, it does not perform well on Caltech-20 and MSRC.

GOMES achieves the best performance on each dataset compared with all the baseline methods. The proposed group-aware fusion approach provides a more flexible strategy to learn the fusion weights than the other methods. GOMES can capture the different visual properties of images and learn appropriate weights for different groups of images, which generates a more discriminative description and achieves better clustering performance.

Fig.2 illustrates some clustering results of GOMES on Caltech-7 and MSRC. We put the images which belong to the same cluster into the same row. The images within the same group are put in a red box. We can see that images in the same group always share similar visual properties: some share similar background (Garfield and Snoopy), some share similar viewpoints (Airplane and Car) and some share similar sub-categories (Motorbike and Cow). Although one image category always consists of several images with different visual appearance, GOMES can cope with the complicated visual distribution and reveal the images sharing similar visual appearances just as we expected. The appropriate fusion weights are learned according to the visual appearance and a more accurate description can be generated compared with baseline methods.

Table 2. Performance comparison on different datasets.

Method	Caltech-101 (7 classes)			Caltech-101 (20 classes)			MSRC			Oxford Flowers		
	AMI	NMI	ARI	AMI	NMI	ARI	AMI	NMI	ARI	AMI	NMI	ARI
SC(1)	0.4583	0.4781	0.4020	0.4241	0.4743	0.2861	0.4466	0.4976	0.3546	0.3403	0.3658	0.1753
SC(2)	0.5601	0.5813	0.4448	0.5335	0.5651	0.3644	0.4888	0.5673	0.3478	0.3782	0.4121	0.1976
SC(3)	0.5112	0.5296	0.4416	0.5105	0.5427	0.3276	0.4909	0.5847	0.3078	0.1438	0.2049	0.0538
SC(4)	0.5397	0.5693	0.4372	0.4655	0.5048	0.2881	0.4554	0.5008	0.3529	–	–	–
SC(5)	0.4629	0.4869	0.3540	0.5101	0.5529	0.3529	0.5033	0.5404	0.4040	–	–	–
EASC	0.6355	0.6544	0.5551	0.5880	0.6220	0.4421	0.7332	0.7540	0.6585	0.3896	0.4145	0.2170
MMSC	N/A	0.6792	N/A	N/A	0.6329	N/A	N/A	0.7745	N/A	N/A	0.4270	N/A
AASC	0.6747	0.6853	0.6692	0.6202	0.6458	0.5110	0.7588	0.7806	0.7244	0.4031	0.4291	0.2363
MSCMO	0.6825	0.6922	0.6428	0.5965	0.6331	0.4164	0.6890	0.7166	0.6116	N/A	0.4840	N/A
GOMES	**0.7365**	**0.7456**	**0.6896**	**0.6852**	**0.7044**	**0.5713**	**0.8694**	**0.8770**	**0.8578**	**0.4870**	**0.5069**	**0.3351**

5. CONCLUSION

In this paper, we propose GOMES, which learns multi-view fusion weights in a more appropriate fashion for real-world images. The consensus and discrimination criterions are designed to evaluate the importance of different views, and an iterative optimization algorithm is proposed to learn the clustering results and fusion weights simultaneously. Experiments on several real-world image datasets indicate that GOMES improves the clustering performance compared with baseline methods, which verifies the effectiveness of GOMES.

6. REFERENCES

[1] Gunhee Kim, Christos Faloutsos, and Martial Hebert, "Unsupervised modeling of object categories using link analysis techniques," in *CVPR*, 2008.

[2] Tinne Tuytelaars, Christoph H Lampert, Matthew B Blaschko, and Wray Buntine, "Unsupervised object discovery: A comparison," *IJCV*, vol. 88, no. 2, pp. 284–302, 2010.

[3] Bin Zhao, James T Kwok, and Changshui Zhang, "Multiple kernel clustering.," in *SDM*, 2009.

[4] Yen-Yu Lin, Tyng-Luh Liu, and Chiou-Shann Fuh, "Multiple kernel learning for dimensionality reduction," *TPAMI*, vol. 33, no. 6, pp. 1147–1160, 2011.

[5] Abhishek Kumar, Piyush Rai, and Hal Daumé III, "Co-regularized multi-view spectral clustering.," in *NIPS*, 2011, pp. 1413–1421.

[6] Xiao Cai, Feiping Nie, Heng Huang, and Farhad Kamangar, "Heterogeneous image feature integration via multi-modal spectral clustering," in *CVPR*, 2011.

[7] Hsin-Chien Huang, Yung-Yu Chuang, and Chu-Song Chen, "Affinity aggregation for spectral clustering," in *CVPR*, 2012.

[8] Grigorios Tzortzis and Aristidis Likas, "Kernel-based weighted multi-view clustering.," in *ICDM*, 2012.

[9] Andrea Frome, Yoram Singer, Fei Sha, and Jitendra Malik, "Learning globally-consistent local distance functions for shape-based image retrieval and classification," in *ICCV*, 2007.

[10] Jingjing Yang, Yonghong Tian, Ling-Yu Duan, Tiejun Huang, and Wen Gao, "Group-sensitive multiple kernel learning for object recognition," *TIP*, vol. 21, no. 5, pp. 2838–2852, 2012.

[11] Yanting Lu, Liantao Wang, Jianfeng Lu, Jingyu Yang, and Chunhua Shen, "Multiple kernel clustering based on centered kernel alignment," *Pattern Recognition*, 2014.

[12] Hongxing Wang, Chaoqun Weng, and Junsong Yuan, "Multi-feature spectral clustering with minimax optimization," in *CVPR*, 2014.

[13] Yina Han, Kunde Yang, Yuanliang Ma, and Guizhong Liu, "Localized multiple kernel learning via sample-wise alternating optimization," *Cybernetics, IEEE Transactions on*, vol. 44, no. 1, pp. 137–148, 2014.

[14] Lujiang Zhang and Xiaohui Hu, "Locally adaptive multiple kernel clustering," *Neurocomputing*, vol. 137, pp. 192–197, 2014.

[15] Chris Ding, Tao Li, and Michael I Jordan, "Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding," in *ICDM*, 2008.

[16] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al., "On spectral clustering: Analysis and an algorithm," *NIPS*, 2002.

[17] Li Fei-Fei, Rob Fergus, and Pietro Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *CVIU*, vol. 106, no. 1, pp. 59–70, 2007.

[18] John Winn and Nebojsa Jojic, "Locus: Learning object classes with unsupervised segmentation," in *ICCV*, 2005.

[19] M-E Nilsback and Andrew Zisserman, "A visual vocabulary for flower classification," in *CVPR*, 2006.

[20] Timo Ojala, Matti Pietikainen, and Topi Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *TPAMI*, vol. 24, no. 7, pp. 971–987, 2002.

[21] Aude Oliva and Antonio Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *IJCV*, vol. 42, no. 3, pp. 145–175, 2001.

[22] Jianxin Wu and Jim M Rehg, "Centrist: A visual descriptor for scene categorization," *TPAMI*, vol. 33, no. 8, pp. 1489–1501, 2011.

[23] David G Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[24] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.