MULTI-AFFINITY SPECTRAL CLUSTERING

Hsin-Chien Huang[⋆†] *Yung-Yu Chuang*[⋆] *Chu-Song Chen*[†]

[⋆]National Taiwan University [†]Academia Sinica

ABSTRACT

Spectral clustering (SC) has become one of the most popular clustering methods. Given an affinity matrix, SC explores its spectral-graph structure to partition data into disjoint meaningful groups. However, in many applications, there are multiple potentially useful features and thereby multiple affinity matrices. For applying spectral clustering to such cases, these affinity matrices must be aggregated into a single one. Unfortunately, affinity measures based on different features could have different characteristics. Some are more effective than others. We propose a multi-affinity spectral clustering (MASC) algorithm which extends the SC algorithm with multiple affinities available. By automatically adjusting the weights of affinity matrices, MASC is more immune to ineffective affinities and irrelevant features. This makes the choice of similarity or distance-metric measures for clustering less crucial. Experiments show that MASC is effective in simultaneous clustering and feature fusion, thus maintaining robustness of SC for multi-affinity clustering problems.

Index Terms— spectral clustering, affinity matrix, multiple kernel learning.

1. INTRODUCTION

Clustering is an 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 widely used for exploratory data analysis in many fields of science. Over the past decades, many clustering algorithms have been proposed. Among them, spectral clustering (SC) is one of the best algorithms [1]. It often outperforms other methods by transforming the representations of the data points into another space in which cluster properties of the data are enhanced. In addition, spectral clustering is elegant in theory, simple to implement and can be solved efficiently using standard linear algebra packages.

Most SC methods explicitly or implicitly assume a metric or a similarity structure. The success of such algorithms depends heavily on the choice of the metric [2]. Unfortunately, SC does not have any built-in mechanism to discover a good metric for better clustering results. Therefore, it is often necessary to use additional feature selection or feature weighting methods as a precursor before invoking SC. The problem is aggravated for many real-world clustering problems in which there are multiple potentially useful cues. Without proper feature selection, performance of SC can degrade dramatically in the presence of irrelevant, ineffective or unreliable features.

Similar to the recent advances in supervised learning with multiple kernels [2], the motivation of this paper is to develop SC methods that are robust to irrelevant and unreliable features. The main difference from their work is that our method is unsupervised while those methods learn affinity from training examples. The proposed method is called milti-affinity spectral clustering (MASC) which deals with the clustering problem when multiple affinity measures are given. By assuming that the aggregated affinity is a weighted combination of the given affinity measures, the problem is reduced to simultaneously finding the optimal weight assignment for affinities and cluster assignment for data points under that weight assignment. We show that appropriate weight assignment and clustering can be found by extending the multiple kernel learning paradigm to SC.

2. METHOD

2.1. Spectral clustering

SC 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 $\mathbf{x}_j$. SC aims to divide these data into c clusters by finding n indicators $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n$ ($\mathbf{f}_i \in \mathbb{R}^c$) 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., $\mathbf{D}_{ii} = w_{i1} + w_{i2} + \ldots + w_{in}$. SC 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 SC is constructed by the eigenvectors corresponding to the next c smallest eigenvalues,

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.

$\lambda_2, \lambda_3, \ldots, \lambda_{c+1}$. 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 $\mathbf{x}_i$.

In practice, SC often serves as a preprocessing step of other clustering algorithms such as k-means. The main trick of SC is to transform the representation of the data points $\mathbf{x}_i$ into the indicator space $\mathbf{f}_i \in \mathbb{R}^c$ 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 has no difficulty to detect the clusters. Main reasons for SC's success include that 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 can be implemented efficiently even for large data sets as long as the similarity graph is sparse. However, one of its limitations is that choosing a good similarity graph is not trivial. For real-world clustering problems, the affinities w_{ij} could be obtained in multiple ways. They could be determined with different types of features extracted, and could also be constructed by reproducible kernels when $\mathbf{x}_i$ are vectors in some Euclidean space. This paper shows how to find a weighted combination of the affinities so that a better similarity measure can be learned for SC in an unsupervised fashion.

2.2. Multi-affinity spectral clustering

Assume that there are m affinity matrices $\mathbf{W}_k (k = 1 \ldots m)$ 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. Since the affinities $w_{ij;k}$ are non-negative, we can denote $w_{ij;k} = s^2_{ij;k}$ to reflect this assumption. As mentioned, the goal of MASC is to find a proper weight assignment to these affinities. Let $\mathbf{v} = [v_1, v_2, \ldots, v_m]$ be a weight vector acting as selectors to these affinities. The k-th weighted affinity can be denoted as $\underline{s}_{ij;k} = v_k s_{ij;k}$. We can then formulate the MASC problem as

$$\begin{aligned} &\min_{\substack{\mathbf{f}_1,\ldots,\mathbf{f}_n \\ v_1,\ldots,v_m}} \sum_k \sum_{i,j} \underline{s}^2_{ij;k} ||\mathbf{f}_i - \mathbf{f}_j||^2 \\ = &\min_{\substack{\mathbf{f}_1,\ldots,\mathbf{f}_n \\ v_1,\ldots,v_m}} \sum_k \sum_{i,j} v_k^2 w_{ij;k} ||\mathbf{f}_i - \mathbf{f}_j||^2 \end{aligned} \tag{2}$$

under the constraints that the sum of v_k's p-norm is normalized; that is, $v_1^p + v_2^p + ... + v_M^p = 1$, $1 \le p \le 2$; and the weights are non-negative, $v_k \ge 0$. It leads to a constrained optimization problem. By applying a Lagrange multiplier λ to the equality constraint, we have

$$J_\lambda = \sum_k \sum_{i,j} v_k^2 w_{ij;k} ||\mathbf{f}_i - \mathbf{f}_j||^2 - 2\lambda \left(\sum_k v_k^p - 1 \right) \tag{3}$$

Equation 3 is complicated to optimize since we have two sets of variables, indicators $\mathbf{f}_i$ and weights v_k. However, it becomes much easier to solve if we solve one set of variables at a time while fixing the other set of variables.

Algorithm 1 Multi-Affinity Spectral Clustering (MASC). 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 MASC(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 generalized eigenvectors $\mathbf{v}_2, \ldots, \mathbf{v}_{c+1}$ for the pair of matrices $(D - W, D)$ corresponding to generalized 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*
    $\beta_k = \sum_{i,j} w_{ij;k} ||\mathbf{f}_i - \mathbf{f}_j||^2$
    weight $v_k = \frac{1}{\left[(\frac{\beta_k}{\beta_1})^{\frac{p}{2-p}} + (\frac{\beta_k}{\beta_2})^{\frac{p}{2-p}} + \cdots + (\frac{\beta_k}{\beta_m})^{\frac{p}{2-p}} \right]^{\frac{1}{p}}}$
  until convergence
  run k-means on $\mathbf{f}_1, \ldots, \mathbf{f}_n$ to cluster data into c groups
end procedure

Let's first assume that the indicators $\mathbf{f}_i$ are given and fixed. By taking its partial derivatives with respect to v_k^p and setting them to zero, we have

$$\frac{\partial J_\lambda}{\partial v_k^p} = 2p^{-1} v_k^{2-p} \left(\sum_{i,j} w_{ij;k} ||\mathbf{f}_i - \mathbf{f}_j||^2 \right) - 2\lambda = 0.$$

To simplify notations, we denote the sum as $\beta_k = \sum_{i,j} w_{ij;k} ||\mathbf{f}_i - \mathbf{f}_j||^2$. The solution becomes

$$v_k = (p\lambda)^{\frac{1}{2-p}} \beta_k^{\frac{-1}{2-p}}.$$

As λ is a dummy variable, it can be solved by considering the constraint $1 = \sum_k v_k^p$, which gives that

$$(p\lambda)^{\frac{1}{2-p}} = \left(\sum_k \beta_k^{\frac{-p}{2-p}} \right)^{-\frac{1}{p}}.$$

Hence, $v_k = \left(\sum_k \beta_k^{\frac{-p}{2-p}} \right)^{-\frac{1}{p}} \beta_k^{\frac{-1}{2-p}}$. Note that β_k is non-negative, making the constraint $v_k \ge 0$ automatically satisfied. Therefore, if the indicators $\mathbf{f}_i$ are known, the optimal solution of the weights v_k becomes

$$v_k = \frac{1}{\left[(\frac{\beta_k}{\beta_1})^{\frac{p}{2-p}} + (\frac{\beta_k}{\beta_2})^{\frac{p}{2-p}} + \cdots + (\frac{\beta_k}{\beta_m})^{\frac{p}{2-p}} \right]^{\frac{1}{p}}} \tag{4}$$

On the other hand, if the weights v_k are given, the problem becomes a standard SC problem (Equation 1) and the affinities

are set as $w_{ij} = \sum_k v_k^2 w_{ij;k}$. Thus, we can solve the MASC problem (Equation 2) using a two-step iterative algorithm which alternatively finds the optimal weights v_k and the optimal indicators $\mathbf{f}_i$. 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 SC to find the optimal indicator $\mathbf{f}_i$. Shi and Malik [3] showed that normalized SC has better performance than unnormalized SC. Thus, we adopt normalized SC to find the $\mathbf{f}_i$ by solving the generalized eigen-vector problem $(D - W)f = \lambda D f$. Next, in the second step, the indicators $\mathbf{f}_i$ are fixed and we refine the weights v_k by using the closed-form solution in Equation 4. This alternating process is repeated as long as the objective function is decreased. Algorithm 1 summarizes the MASC algorithm.

3. EXPERIMENTS

We have tested the proposed method on a number of real data sets from the UCI machine learning repository [4] (Section 3.1) and two well-known face databases from ORL [5], and CMU-PIE [6]. We set $p = 1$ in the current implementation. In this setting, Equation 4 becomes

$$v_k = \frac{1}{(\frac{\beta_k}{\beta_1}) + (\frac{\beta_k}{\beta_2}) + \cdots + (\frac{\beta_k}{\beta_m})}.$$

These data sets are summarized in Table 1. 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. We used NMI for evaluating clustering results as Wu *et al.* [7] reported that it generally works the best.

3.1. UCI repository

Fourteen data sets were selected from the UCI repository. For each set, only the extracted feature vectors are available – not the raw data. These vectors were normalized to have zero mean and unit standard deviation. They are then substituted into different types of kernel functions to obtain several sets of pairwise distances. Here, following the strategy of other multiple kernel learning approaches, we select a set of reasonable kernels that are frequently used. In our experiments, we used one polynomial kernel

$$\kappa_k(\mathbf{x}_i, \mathbf{x}_j) = (\theta + \mathbf{x}_i^T \mathbf{x}_j)^q,$$

with $\theta = 1$ and $q = 2$, and several Gaussian kernels

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

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

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

Table 1. Summary of the data sets used in the experiments. The first 14 data sets are adopted from the UCI machine learning repository and the last two face databases are from ORL and CMU-PIE. For CMU-PIE, we used the frontal images (Pose 27) with 21 different illuminations.

ID	Name	#instances	#features	#classes
R1	*Iris*	150	4	3
R2	*Wine*	178	13	3
R3	*Glass Identification*	214	9	6
R4	*Solar Flare*	323	12	6
R5	*Protein Localization Sites*	336	7	8
R6	*Libras Movement*	360	90	15
R7	*WDBC*	569	30	2
R8	*Balance Scale Weight and Distance Database*	625	4	3
R9	*Connectionist Bench(Vowel Recognition-Deterding Data)*	990	10	11
R10	*Yeast*	1,484	8	10
R11	*Letter Recognition*(A,B)	1,555	16	2
R12	*Letter Recognition*(A,B,C,D)	3,096	16	4
R13	*Abalone*	4,177	8	28
R14	*Waveform Database Generator*	5,000	21	3
F1	*ORL*	360	7,744	40
F2	*CMU-PIE-illum*	1,496	7,744	68

We vary γ over $\{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$ to obtain 7 Gaussian kernels. Finally, we normalize the values of each kernel function to the range of $[0.0001..1]$ to obtain a total of 8 affinity matrices.

We use SC_1, $\cdots$, SC_8 to denote single-affinity spectral clustering methods with the above 8 kernels (1 polynomial and 7 Gaussians) respectively. In addition, we also combined above 8 kernels by equal and random weights, denoted as EASC and RASC respectively. Since each method serves as a preprocessing step of k-means whose performance depends on the initialization, we performed 50 runs and reported the average for each method. Table 2 presents the average NMI values and the corresponding ranks for different algorithms on the 14 UCI data sets. The numbers in parentheses are the ranks of different methods for each data set. For example, MASC ranks the first with an NMI of 0.360 for the data set R3 while SC_5 ranks the eighth with an NMI of 0.316. The last two rows (*m*NMI, *m*Rank) of Table 2 display the average NMI value and the average rank for each method over the 14 data sets, respectively. MASC has an average NMI 0.455 and ranks the best of all the methods in terms of average NMI (*m*NMI). In terms of average rank (*m*Rank), MASC's average rank is 2.438, again the best of all the methods. Note that *m*NMI and *m*Rank both yield similar rankings.

3.2. Face clustering

We also evaluated MASC with face clustering. The experimented face databases are ORL and CMU-PIE. In contrast to the UCI datasets, for this application, we have access to the raw data. Thus, the first step is to extract features from the images. All images were first normalized and cropped to 88×88 pixels. To utilize cues from different perspectives, we extracted three types of features.

1. Eigenface [8]. After performing principal component analysis, each face image was projected into the eigenspace which preserves 90% of the energy.

Table 2. Comparisons of different algorithms on UCI data sets in terms of NMI.

ID	SC_1	SC_2	SC_3	SC_4	SC_5	SC_6	SC_7	SC_8	EASC	RASC	MASC
R1	0.582(11)	0.885(10)	0.914(1)	0.900(3)	0.900(3)	0.900(3)	0.900(3)	0.900(3)	0.900(3)	0.900(2)	0.900(3)
R2	0.848(11)	0.862(10)	0.878(7)	0.878(7)	0.893(3)	0.893(2)	0.893(3)	0.893(3)	0.878(7)	0.882(6)	0.905(1)
R3	0.342(5)	0.342(6)	0.347(2)	0.316(9)	0.316(8)	0.308(10)	0.308(10)	0.327(7)	0.346(3)	0.344(4)	0.360(1)
R4	0.229(8)	0.227(9)	0.225(10)	0.223(11)	0.238(4)	0.248(1)	0.236(6)	0.238(5)	0.243(2)	0.233(7)	0.243(2)
R5	0.509(11)	0.532(10)	0.544(9)	0.570(1)	0.567(2)	0.565(6)	0.565(7)	0.545(8)	0.566(3)	0.565(5)	0.566(3)
R6	0.612(9)	0.611(10)	0.599(11)	0.617(6)	0.617(5)	0.623(4)	0.612(8)	0.615(7)	0.630(2)	0.629(3)	0.645(1)
R7	0.578(10)	0.578(10)	0.584(1)	0.584(1)	0.584(1)	0.584(1)	0.584(1)	0.584(1)	0.584(1)	0.584(1)	0.584(1)
R8	0.093(8)	0.261(1)	0.193(6)	0.195(5)	0.253(2)	0.006(11)	0.014(10)	0.025(9)	0.225(4)	0.146(7)	0.253(3)
R9	0.292(11)	0.296(10)	0.322(8)	0.338(5)	0.339(3)	0.301(9)	0.330(7)	0.339(4)	0.344(2)	0.335(6)	0.358(1)
R10	0.238(10)	0.246(5)	0.262(1)	0.259(2)	0.252(3)	0.244(7)	0.238(11)	0.246(6)	0.238(9)	0.243(8)	0.252(4)
R11	0.665(10)	0.665(11)	0.677(9)	0.705(4)	0.705(4)	0.718(1)	0.718(1)	0.718(1)	0.695(7)	0.694(8)	0.705(4)
R12	0.493(11)	0.502(10)	0.504(9)	0.519(6)	0.519(5)	0.522(4)	0.523(3)	0.528(1)	0.515(7)	0.514(8)	0.527(2)
R13	0.171(7)	0.168(11)	0.169(10)	0.172(6)	0.170(9)	0.171(8)	0.180(1)	0.178(2)	0.176(3)	0.174(5)	0.176(4)
R14	0.369(2)	0.369(1)	0.367(8)	0.367(4)	0.367(4)	0.366(9)	0.365(10)	0.363(11)	0.367(4)	0.367(7)	0.367(3)
mNMI	0.430(11)	0.467(7)	0.470(6)	0.474(4)	0.480(2)	0.461(10)	0.462(9)	0.464(8)	0.479(3)	0.472(5)	0.488(1)
mRank	8.857(11)	8.143(10)	6.571(9)	5.000(5)	4.000(2)	5.429(6)	5.786(8)	4.857(4)	4.071(3)	5.500(7)	2.357(1)

Table 3. Comparison of different methods on face database ORL and CMU-PIE in terms of NMI.

	SC_e	SC_g	SC_l	EASC	RASC	MASC
F1	0.782	0.791	0.749	0.867	0.846	0.880
F2	0.919	0.914	0.816	0.920	0.918	0.928

2. Gabor texture [9]. Each face image was filtered with 40 Gabor filters generated with five different scales and eight orientations.

3. Local binary pattern (LBP) [10]. 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. After extracting these three features, each feature was treated as a vector; these vectors were substituted into the Gaussian kernel ($\gamma = 0.005$) to calculate pairwise distances. 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 3 shows NMI values for different algorithms on the 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 3 in which SC_g is the best among three single-affinity SCs for ORL while SC_e is the best for CMU-PIE. Without knowing the characteristics of the databases, MASC successfully combined the strengths of different features and outperformed all other methods for both data sets.

4. CONCLUSIONS

We extended the spectral clustering algorithm to multi-affinity setting to explore strengths of different features automatically. Experiments show that it effectively incorporates multiple affinities and yields better overall performance compared to spectral clustering with only a single affinity or naive feature fusion strategies. In addition, it is easy to implement. These characteristics make it useful for real-world applications. In the future, we will work on the application of this algorithm for bag-of-words learning.

5. REFERENCES

[1] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[2] M. I. Jordan, F. R. Bach, and F. R. Bach, "Learning spectral clustering," in *Proceedings of NIPS*, 2003.

[3] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[4] A. Frank and A. Asuncion, "UCI machine learning repository," 2010.

[5] F.S. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of WACV*, 1994.

[6] T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, 2003.

[7] J. Wu, H. Xiong, and J. Chen, "Adapting the right measures for k-means clustering," in *Proceedings of KDD*, 2009.

[8] 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.*, vol. 19, no. 7, pp. 711–720, 1997.

[9] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 837–842, 1996.

[10] 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.*, vol. 24, no. 7, pp. 971–987, 2002.