SHORT COMMUNICATIONS

Similar component analysis*

ZHANG Hong[1] **, WANG Xin[1], LI Junwei[2] and CAO Xianguang[1]

(1. Image Center, Astronautics School, BeiHang University, Beijing 100083, China; 2. The 207th Institute of the Second Academy of China Aerospace Science Industry Corporation, Beijing 100854, China)

Received March 6, 2006; revised April 29, 2006

Abstract A new unsupervised feature extraction method called similar component analysis (SCA) is proposed in this paper. SCA method has a self-aggregation property that the data objects will move towards each other to form clusters through SCA theoretically, which can reveal the inherent pattern of similarity hidden in the dataset. The inputs of SCA are just the pairwise similarities of the dataset, which makes it easier for time series analysis due to the variable length of the time series. Our experimental results on many problems have verified the effectiveness of SCA on some engineering application.

Keywords: **clustering, feature extraction, similar component.**

Data clustering[1,2] is an old problem in pattern recognition and data mining community, since organizing observed data on groups or clusters is the first step to exploit coherent patterns and useful structures hidden in the dataset. Many data clustering methods have been proposed in the past several decades. However, some of them are only effective for some specific cluster shapes. For example, k-means[2] is suitable for the dataset having round clusters, while Gaussian mixture mode[3] is more effective to discover the elliptical clusters. Another fact that hinders the applications of the traditional clustering methods is that most of them are based on the dataset itself, which means that all the training data should be supplied as inputs. However, in some fields such as time series analysis, the data objects may have different dimensionalities, and it is hard for those traditional clustering methods to cluster this kind of data. For instance, in k-means, the mean of different clusters is computed in each iteration step, but for a set of time series of different lengths, we do not know what their mean is.

In recent years, a new clustering algorithm called spectral clustering[4,5] was proposed based on graph theory. The main idea of spectral clustering is firstly to embed the data points to some space in which the clusters are more "obvious", then perform a classical clustering algorithm, such as k-means[5]. In such a way, impressively good results can be obtained on the dataset with arbitrarily shaped clusters, where traditional clustering approaches would fail. However, the computation of affinity matrix in spectral clustering is also dependent on the dataset itself.

Inspired by spectral clustering, a novel feature extraction method called similar component analysis is proposed (SCA) in this paper. The features extracted by SCA are called similar components analysis. It is proved theoretically that if the clusters hidden in the dataset are non-overlapped, then the similar components can make the data from the same class self-aggregated. The similar components of the data in the same class will have the same value, while the Similar Components of the data in different classes will have different values. Moreover, in this case, we only need the pairwise similarities of the dataset as our inputs, which makes our method easily be generalized to some cluster problems difficult to be solved by the traditional method. Like time series clustering, there have been many methods to measure the similarity of time series data no matter how long the time series are[6—8]. Our experimental results verify the effectiveness of our proposed approach.

1 Similar component analysis

1.1 Overview of the algorithm

Generally we can quantify the associations among

* Supported by the State Key Laboratory of Object Identity (Grant No. 51476010105HK0101)

** To whom correspondence should be addressed. E-mail: dmrzhang@buaa.edu.cn data objects by a similarity metric, such as the Euclidean distance and Mahalanobis distance. Our SCA algorithm starts with a similarity matrix $\boldsymbol{S}=(s_{ij})$ with $s_{ij}=s_{ji}\geqslant 0$. The s_{ij} measures the similarity between data i and data j. $\boldsymbol{S}$ is row-normalized such that $\sum_{j=1}^{N} s_{ij}=1$. Thus, SCA simply performs eigenvalue decomposition to $\boldsymbol{S}$ as follows:

$$\boldsymbol{S}\boldsymbol{v}=\lambda\boldsymbol{v}. \qquad (1)$$

Then the k eigenvectors corresponding to the largest k eigenvalues (also called the k dominant eigenvectors) are called the first k similar components of the dataset. The procedure of SCA is shown as follows:

Step 1: Input the similarity metric d, number of components k.

Step 2: Construct the similarity matrix $\boldsymbol{S}=(s_{ij})=(d(i,j))$.

Step 3: Normalize the rows of $\boldsymbol{S}$ to make $\sum_{j=1}^{N} s_{ij}=1$.

Step 4: Do eigenvalue decomposition on $\boldsymbol{S}$.

Step 5: Output the k dominant eigenvectors as the first k similar component.

1.2 Self-aggregation property of SCA: the ideal case

In this subsection, the ideal case will be analyzed when there is no overlap among the clusters; and the k-dimensional space spanned by the first k similar components is found to have an interesting self-aggregation property.

Proposition 1 (self-aggregation). When there are K non-overlapped clusters in the dataset, the first k similar components of the dataset are all piecewise-constant, assuming the data objects within the same cluster are indexed consecutively. And in the space spanned by the first k similar components, all data of the same cluster self-aggregate to a single point.

Proof. The term "non-overlapped" is used to refer to the case where $s_{ij}=0$ if data i and data j belong to different clusters. Therefore, $\boldsymbol{S}$ can be written as

$$\boldsymbol{S}=\mathrm{diag}(\boldsymbol{S}_1, \boldsymbol{S}_2, \cdots, \boldsymbol{S}_k). \qquad (2)$$

Here, $\boldsymbol{S}$ is a block-diagonal matrix, because the data within the same cluster are assumed to have been indexed consecutively. Since the rows of $\boldsymbol{S}$ have been normalized, from the theorem of Perron-Frobenius[9] that 1 is the largest eigenvalue of $\boldsymbol{S}$, it can be easily verified that the column vector $\boldsymbol{z}_i=(0, 0, \cdots, \boldsymbol{e}_i, \cdots, 0, 0)\in R^N$, where $i\in\{1, 2, \cdots, K\}$ is the eigenvector of $\boldsymbol{S}$ corresponding to eigenvalue $\lambda_{\max}=1$. Here, $\boldsymbol{e}_i=(1, 1, \cdots, 1)$ is a row vector with its dimensionality identical to the size of the i-th cluster. And it can be easily deduced that for any K real numbers $(a_1, a_2, \cdots, a_K)$, $\boldsymbol{z}=\sum_{i=1}^{K} a_i\boldsymbol{z}_i$ is also an eigenvector of $\boldsymbol{S}$ with the eigenvalue $\lambda_{\max}=1$. Clearly, all the elements within the same cluster will have the same value in z, which makes z a K-step function. That is to say, in the space spanned by the first K similar components, all the data objects belonging to the same cluster will self-aggregate to the same point.

Proposition 1 also answers why the features extracted by SCA are called similar components analysis.

1.3 Connection to kernel PCA: extension to testing data

SCA has been introduced and analyzed in Section 1.1 and 1.2. However, as a feature extraction method, SCA still cannot extract features from testing data. In this subsection, the connection between SCA and kernel principal component analysis will be analyzed, and then SCA can be extended to test the data[10].

It is well known that principal component analysis (PCA) is a statistical technique that can extract the most informative k-dimensional output vector $\boldsymbol{y}$ from an input vector $\boldsymbol{x}$ of d-dimension ($d\gg k$). Scholkopf et al.[10] generalized the traditional PCA to the nonlinear case and proposed the kernel principal component analysis (KPCA) method.

The KPCA method first maps the original dataset in R^d to some high dimensional feature space F by some nonlinear mapping Φ. $\Phi=[\Phi(x_1), \Phi(x_2), \cdots, \Phi(x_M)]$ is the mapped data matrix. The main idea of KPCA is to perform PCA in this mapped feature space. By assuming that the data objects have already been centered in the feature space and the non-centered case can be referred to Ref. [10], the following equations can be used to compute the projections of the training and testing data on the

k-th kernel principal component

$$\boldsymbol{P}^k_{x_i} = \sum_{j=1}^{M} \alpha_j^k (\Phi(x_i)\Phi(x_j)) = \sum_{j}^{M} \alpha_j^k K_{ij}, \quad (3)$$

$$\boldsymbol{P}^k_{y} = \sum_{j=1}^{M} \alpha_j^k (\Phi(x_i)\Phi(x_j)). \quad (4)$$

Therefore, the projections of the training and testing data can be computed by doing eigenvalue decompositions to K. The kernel matrix is an inner-product matrix, and the inner-product of two data objects is usually used for measuring the similarity between them. If we extend this idea and let the entries of the kernel matrix $\boldsymbol{K}_e$ be some arbitrary similarity metric which can measure the pairwise similarity of the corresponding data objects, then the kernel matrix $\boldsymbol{K}_e$ will become the similarity matrix in SCA. As analyzed in Proposition 1, the corresponding eigenvalues of the similar components are all 1. In this case, Eq. (3) becomes

$$\boldsymbol{P}^k_{x_i} = (\boldsymbol{K}_e \alpha^k)_i = 1 \circ \alpha_i^k = \alpha_i^k, \quad (5)$$

which is equivalent to SCA. Thus, SCA is equivalent to non-centralized KPCA, and Eq. (4) can be used to compute the similarity components of the out-of-sample data.

1.4 Maximum within cluster association in similar component space: the general case

The ideal case of SCA is analyzed in Section 1.2 and treated as a non-centralized KPCA in Section 1.3. In this subsection, SCA will be analyzed in a general case from a KPCA viewpoint.

Proposition 2 (maximum within cluster association). If we view the similarity matrix $\boldsymbol{S}$ as an inner product matrix in some Hilbert space from the KPCA view point, then the within cluster association will be maximized in the space spanned by the first K similar components of the dataset.

Proof. In KPCA view, the similarity matrix can be treated as an inner product matrix. Φ will be used to denote the data matrix after nonlinear mapping, and the similarity matrix can be rewritten as $\boldsymbol{S} = \Phi^{\mathrm{T}}\Phi$. $\Phi_i = [\Phi(x)_{i1}, \cdots, \Phi(x)_{is_i}]$ represents the data from class i (with size s_i). Assuming that the data are indexed consecutively, that is $\Phi = [\Phi_1, \cdots, \Phi_K]$, it can be easily proved that the order of data will not affect the final results, where K is the number of classes. Then the within-class scatter matrix class k is:

$$\boldsymbol{C}_i = \frac{1}{s_i}\sum_{j=1}^{s_i}(\Phi(x)_{ij} - \boldsymbol{m}_i)(\Phi(x)_{ij} - \boldsymbol{m}_i)^{\mathrm{T}}, \quad (6)$$

where $\boldsymbol{m}_i = \Phi_i \boldsymbol{e}_i / s_i$ is the mean vector of class k, and $\boldsymbol{e}_i$ is a vector of dimension s_i with all elements being one. So with some simple algebra inferences, we can get

$$\boldsymbol{C}_i = \frac{1}{s_i}\Phi_i\left(\boldsymbol{I}_i - \frac{\boldsymbol{e}_i\boldsymbol{e}_i^{\mathrm{T}}}{s_i}\right)\Phi_i^{\mathrm{T}}, \quad (7)$$

where $\boldsymbol{I}_i$ is the identity matrix of order s_i. Then the total within-class scatter matrix can be defined as $\boldsymbol{C} = \sum_{i=1}^{K} s_i \boldsymbol{C}_i$. It is well known that a common principal for data analysis is to minimize the trace of $\boldsymbol{C}$, that is

$$\begin{aligned} \min J &= \mathrm{trace}(\boldsymbol{C}) \\ &= \sum_{i=1}^{K}\left(\mathrm{trace}(\Phi_i\Phi_i^{\mathrm{T}}) - \mathrm{trace}\left(\frac{\boldsymbol{e}_i^{\mathrm{T}}}{\sqrt{\boldsymbol{s}_i}}\Phi_i^{\mathrm{T}}\Phi_i\frac{\boldsymbol{e}_i}{\sqrt{\boldsymbol{s}_i}}\right)\right). \end{aligned}$$

In this way, the data in the same class is distributed as tightly as possible[2]. Defining the block-diagonal matrix $\boldsymbol{Q} = \mathrm{diag}(\boldsymbol{e}_1/\sqrt{\boldsymbol{s}_1}, \cdots, \boldsymbol{e}_1/\sqrt{\boldsymbol{s}_K})$, then

$$J = \mathrm{trace}(\Phi\Phi^{\mathrm{T}}) - \mathrm{trace}(\boldsymbol{Q}^{\mathrm{T}}\Phi^{\mathrm{T}}\Phi\boldsymbol{Q}). \quad (8)$$

Now if the constraint of $\boldsymbol{Q}$ is relaxed to $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q} = \boldsymbol{I}$, then optimizer (8) becomes

$$\max_{\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q} = \boldsymbol{I}} \mathrm{trace}(\boldsymbol{Q}^{\mathrm{T}}\Phi^{\mathrm{T}}\Phi\boldsymbol{Q}). \quad (9)$$

From the theorem of generalized Rayleigh-Ritz[9], it can be easily derived that the solution of (9) is $\boldsymbol{Q}^* = [\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_K]\boldsymbol{R}$, where $\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_K$ are the K dominant eigenvectors of $\Phi^{\mathrm{T}}\Phi$ and $\boldsymbol{R}$ is an arbitrary orthogonal matrix. Recalling that $\boldsymbol{S} = \Phi^{\mathrm{T}}\Phi$, so the conclusion can be drawn that the within cluster association will be maximized in the space spanned by the first K similar components of the dataset.

From Proposition 1 and Proposition 2 we can infer that the cluster structure hidden in the dataset will become clear in the space spanned by the first K similar components of the dataset. In the next section we will provide a set of experiments to show the effectiveness of SCA.

2 Experimental results

2.1 2D synthetic data clustering

The synthetic data set with 200 samples is depicted in Fig. 1. The traditional k-means, using coordinates of sample points as features, partitions the data set into two clusters as shown in Fig. 1 (a). The result is obviously not satisfying. As a comparison, our SCA method is also employed for this task using the Gaussian function

$$s_{ij} = \exp^{-\frac{\| x_i - x_j \|^2}{2\sigma^2}}$$

as the similarity measure with some proper σ (determined experimentally), since Gaussian function-based similarity has been successfully applied in many spectral based clustering methods. After the similar components have been extracted, we will perform k-means to cluster them and the result is shown in Fig. 1(b), from which we can clearly see that SCA outperforms k-means.

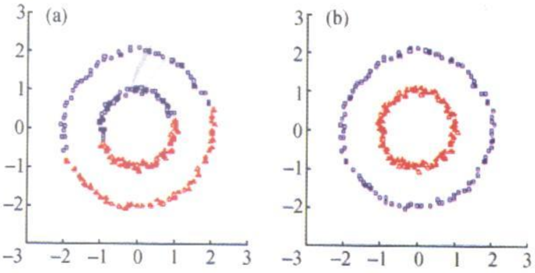

Fig. 1. 2D synthetic data clustering. (a) Clustering results by k-means; (b) clustering results by SCA+ k-means.

2.2 Time series clustering

A real EEG (Electroencephalogram) dataset which is extracted from the 2nd Wadsworth BCI dataset in BCI2003 competition is used in our experiments[11]. According to Ref. [11], the data objects can be generated from three classes: the EEG signals evoked by flashes containing targets, the EEG signals evoked by flashes adjacent to targets, and other EEG signals. All the data objects have an equal length of 144. We randomly choose 50 EEG signals from each class and use the Euclidean distance and the BP metric[12] to measure the pairwise distances of the time series. Then these distances will be transformed to represent the pairwise similarities by a Gaussian function with some proper variance.

The clustering accuracies[8] achieved from hierarchical agglomerative clustering (HAC)[2] methods and our approach (i.e. SCA + k-means) are compared, and the final cluster number is set to be 3 manually. The final results are given in Table 1. “HACC”, “HACS” and “HACA” are used to represent the complete-linkage, single-linkage and average-linkage hierarchical agglomerative clustering methods, respectively. From Table 1 we can see clearly the advantage of SCA.

Table 1. Clustering results on EEG dataset

	HACC	HACS	HACA	SCA
Euclidean	0.4778	0.3556	0.3556	0.5222
BP	0.4556	0.3556	0.4222	0.5444

2.3 Color image segmentation

Now SCA method is applied to color image segmentation problems. The RGB (Red, Green, Blue) values and the spatial coordinates of a pixel are used as its features, thus a pixel is represented by a five tuple (r, g, b, x, y). The similarity of two pixels is given by a Gaussian function. The segmentation results can be seen in Fig. 2, where Fig. 2(a), 2(b) are the original images and Fig. 2(c), 2(d) are the segmented images.

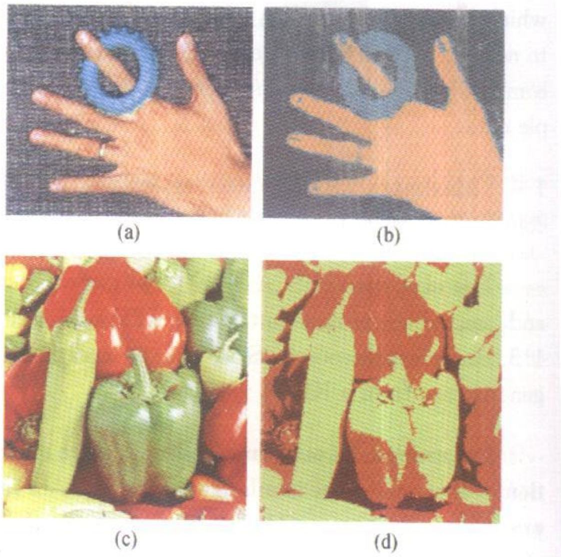

Fig. 2. Color image segmentation.

2.4 Face recognition

The ORL database[13] is selected as our experimental dataset. The recognition accuracy is tested with different numbers of training samples. t (t=2, 3, 4) images of each subject were randomly selected for training, and the remaining 10-k images of each subject for testing. The Gaussian function is used to measure the similarity between two faces. The recognition results can be seen in Table 2, and the recognition accuracy achieved by the traditional PCA and kernel PCA method is given for comparison.

Table 2. Recognition accuracy on ORL dataset

t	PCA	KPCA	SCA
2	0.7225	0.7713	0.7756
3	0.7996	0.8511	0.8516
4	0.8492	0.8990	0.8967

Since the KPCA is a popular method to face recognition[14], we can see that the recognition results obtained by SCA can approximate the KPCA recognition accuracy.

3 Conclusions and discussion

In this paper, a novel feature extraction method called similar component analysis (SCA) has been proposed. The SCA method has a self-aggregation property that the data objects will move towards each other to form clusters through SCA theoretically. It has been found that the inherent pattern of similarity hides in the dataset. The inputs of SCA are just the pairwise similarities of the dataset, which makes time series analysis easier due to the variable length of the time series. Several experiments have been presented and the advantages of our method can be seen easily. Although we can apply the Gaussian function, there are still some unanswered questions such as how to choose a proper similarity metric, which will be discussed in the subsequent paper.

References

1 Jain A. K. and Dubes R. C. Algorithms for Clustering Data, 1st ed. Englewood Cliffs NJ: Prentice Hall, 1988, 1—200.

2 Duda R. O., Hart P. E. and Stork D. G. Pattern Classification, 2nd ed. New York: John Wiley & Sons, 2001. 512—540.

3 Bilms J. A. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report, International Computer Science Institute, Berkeley CA, 1998, 1—13.

4 Shi J. and Malik J. Normalized cuts and image segmentation. IEEE Trans. on PAMI, 2000, 22(8): 888—905.

5 Ng A. Y., Jordan M. I. and Weiss Y. On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems 14 (ed. Dietterich T. G., Becker S. and Ghahramani Z.), Cambridge, MA: MIT Press, 2002, 857—864.

6 Li C. A Bayesian approach to temporal data clustering using the hidden Markov model methodology. Ph. D. thesis of Vanderbilt University, 2000, 1—223.

7 Smyth P. Clustering sequences with hidden Markov models. In: Advances in Neural Information Processing Systems 9 (ed. Mozer M., Jordan M. and Petsche T.), Cambridge, MA: MIT Press, 1997, 648—654.

8 Zhong S. and Ghosh J. A unified framework for model-based clustering. Journal of Machine Learning Research, 2003, 4(12): 1001—1037.

9 Weisstein E. W. Perron-Frobenius theorem. http://mathworld.wolfram.com/Perron-Frobenius Theorem.html. [2006-4-27]

10 Scholkopf B., Smola A. and Muler K. Nonlinear component analysis as a kernel eigenvalue problem. Technical Report, Max-Plank-Institut, 1996, 1—18.

11 Lin Z. and Zhang C. Enhancing classification by perceptual characteristic for the p300 speller paradigm. In: Proceedings of the 2nd International IEEE EMBS Conference on Neural Engineering, Virginia, USA, March 16—19, 2005, 574—576.

12 Panuccio A., Bicego M. and Murino V. A hidden Markov model-based approach to sequential data clustering. In: Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, Windsor, Canada, August 6—9, 2002, 734—742.

13 Samaria F. and Harter A. Parameterisation of a stochastic model for human face identification. In: Proceedings of the Second IEEE Workshop on Applications of Computer Vision, Florida, USA, Dec. 19—23, 1994, 138—142.

14 Yang M. Kernel eigenfaces vs. kernel fisherfaces: face recognition using kernel methods. In: Proc. IEEE International Conference on Automatic Face and Gesture Recognition, Washington DC, USA. May 20—21, 2002, 215—220.