PERGAMON

AVAILABLE AT
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Neural Networks

Neural Networks 16 (2003) 437–451

www.elsevier.com/locate/neunet

2003 Special Issue

Improved system for object detection and star/galaxy classification via local subspace analysis

Zhi-Yong Liu*, Kai-Chun Chiu, Lei Xu

Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong, People's Republic of China

Abstract

The two traditional tasks of object detection and star/galaxy classification in astronomy can be automated by neural networks because the nature of the problems is that of pattern recognition. A typical existing system can be further improved by using one of the local Principal Component Analysis (PCA) models. Our analysis in the context of object detection and star/galaxy classification reveals that local PCA is not only superior to global PCA in feature extraction, but is also superior to gaussian mixture in clustering analysis. Unlike global PCA which performs PCA for the whole data set, local PCA applies PCA individually to each cluster of data. As a result, local PCA often outperforms global PCA for data of multi-modes. Moreover, since local PCA can effectively avoid the trouble of having to specify a large number of free elements of each covariance matrix of gaussian mixture, it can give a better description of local subspace structures of each cluster when applied on high dimensional data with small sample size. In this paper, the local PCA model proposed by Xu [IEEE Trans. Neural Networks 12 (2001) 822] under the general framework of Bayesian Ying Yang (BYY) normalization learning will be adopted. Endowed with the automatic model selection ability of BYY learning, the BYY normalization learning-based local PCA model can cope with those object detection and star/galaxy classification tasks with unknown model complexity. A detailed algorithm for implementation of the local PCA model will be proposed, and experimental results using both synthetic and real astronomical data will be demonstrated.

Keywords: Star/galaxy classification; Local PCA; Gaussian mixture; BYY harmony learning; Model selection

1. Introduction

The number of sky objects contained in astronomical images are inherently huge. One typical example is the 3 terabytes POSS-II images containing objects of the order of 2 billion (Fayyad, Djorgovski, & Weir, 1996). In view of this, the tasks of object detection and star/galaxy classification, which are traditionally handled manually, need to be automated by some systematic means. In the last decade, fuelled by the development of neural networks and machine learning, various methods related to neural networks application in astronomy have been proposed in Andreon, Gargiulo, Longo, Tagliaferri, and Capuano (2000), Bertin and Arnouts (1996), Fayyad et al. (1996) and Odewahn, Stockwell, Pennington, Humphreys, and Zumach (1992).

Theoretically, object detection should precede star/galaxy classification because the former is to detect the objects out of the original image and the latter is to classify the objects identified into stars and galaxies. In implementation the two tasks can be achieved algorithmically and the whole process described in Andreon et al. (2000) is summarized in Fig. 1.

Object is regarded as some connected pixels that are brighter than a certain threshold in the original image. Currently, two main approaches have been used for object detection. The first approach detects object pixels indirectly through discarding background pixels while the second approach directly detects the object pixels. SExtractor (Bertin & Arnouts, 1996) is a typical example that uses the first approach. For this example, a background map is constructed with a weak assumption that all the objects in the plate material share the same background. After discarding the background, a thresholding and template frame based method is used for object detection. A major drawback of this approach is that the template frame parameter needs to be predefined heuristically. SExtractor can be contrasted with NExt (Neural Extractor) (Andreon et al., 2000) which belongs to the second approach. Although application of this approach does not require a predefined template frame, it is computationally

* Corresponding author.
E-mail addresses: zyliu@cse.cuhk.edu.hk (Z.-Y. Liu), kcchiu@cse.cuhk.edu.hk (K.-C. Chiu), lxu@cse.cuhk.edu.hk (L. Xu).

0893-6080/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0893-6080(03)00015-7

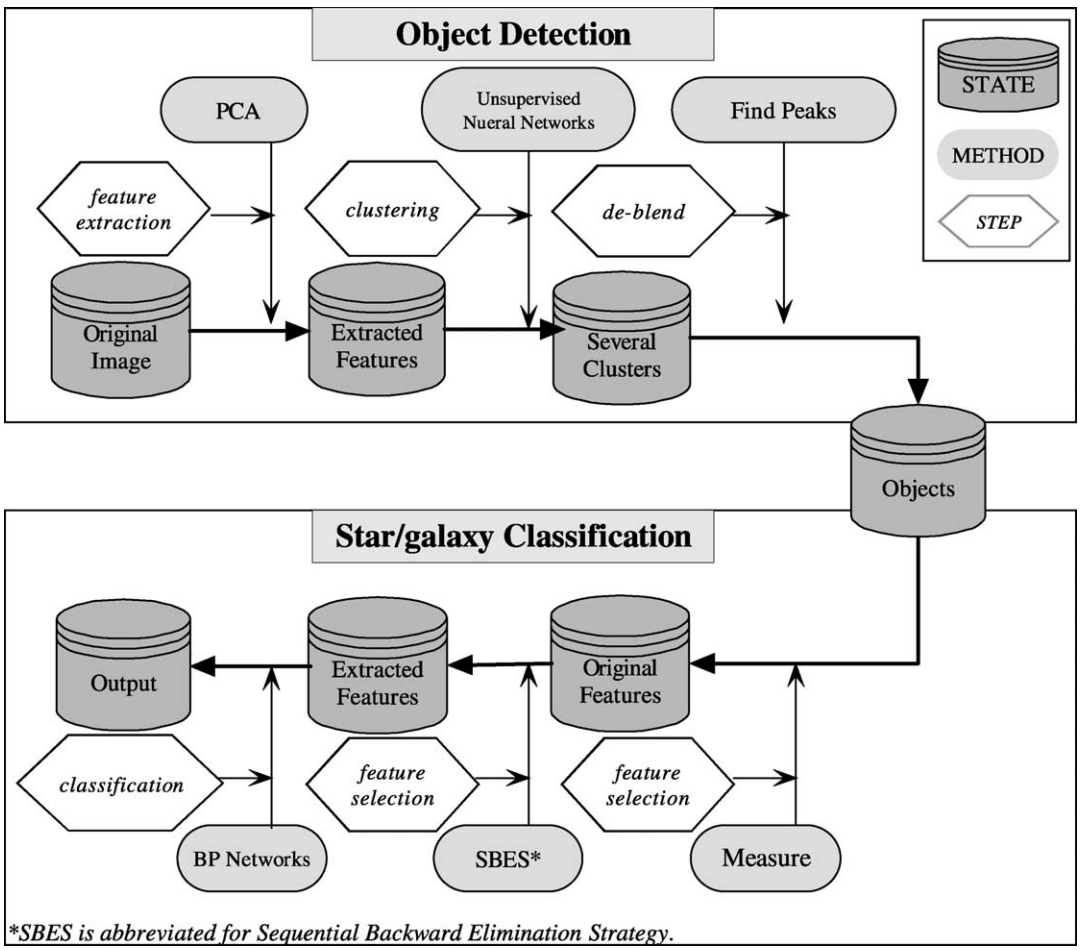

Fig. 1. Original system for implementation of object detection and star/galaxy classification.

intensive for large images as it takes each pixel, together with some neighborhood pixels, as one input.

Subsequent to the task of object detection, the task of star/galaxy classification comprises three steps, namely feature selection, feature extraction, and classification. In literature, all features of an object are used as input to neural network classifier. For instance, there are 25 features in Andreon et al. (2000) and 10 in Bertin and Arnouts (1996). However, since most of the features are correlated, the actual number of features that are useful for analysis can be much smaller. As a result, if we can make use of some feature extraction techniques to reduce redundancy between attributes, more precise results may be obtained. Although the so-called Sequential Backward Elimination Strategy (SBES) (Bishop, 1995) has been adopted in Andreon et al. (2000) to select several best features out of the original 25, some useful information contained in the discarded features will be lost as a consequence of its suboptimal nature.

Several inadequacies mentioned above can be improved via local subspace analysis. In this paper, our aim is to introduce the local Principal Component Analysis (PCA) model derived from the perspective of Bayesian Ying Yang (BYY) normalization learning (Xu, 2001a,b) and to highlight its potential application to the tasks of object detection and star/galaxy classification. BYY normalization learning is a special case of BYY harmony learning that was firstly proposed in 1995 (Xu, 1995 and 1996) and systematically developed in past years. This BYY harmony learning acts as a general statistical learning framework not only for understanding various existing statistical learning approaches, but also making model selection implemented either *automatically* during parameter learning or *subsequently after* parameter learning via a new class of model selection criteria. Also, this BYY harmony learning has motivated three types of regularization, namely a data smoothing technique that provides a new solution on the hyper-parameter in a Tikinov-like regularization (Tikhonov & Arsenin, 1977), a normalization with a new conscience de-learning mechanism that has a nature similar to the rival penalized competitive learning (Xu, Krzyzak, & Oja, 1993), and a structural regularization by imposing certain structural constraints. The details are referred to the recent papers (Xu, 2000; Xu, 2001a,b; Xu, 2002) that jointly provide a symmetrical overview on advances obtained along this direction.

The BYY normalization learning used in this paper takes the advantages of the automatic model selection ability for

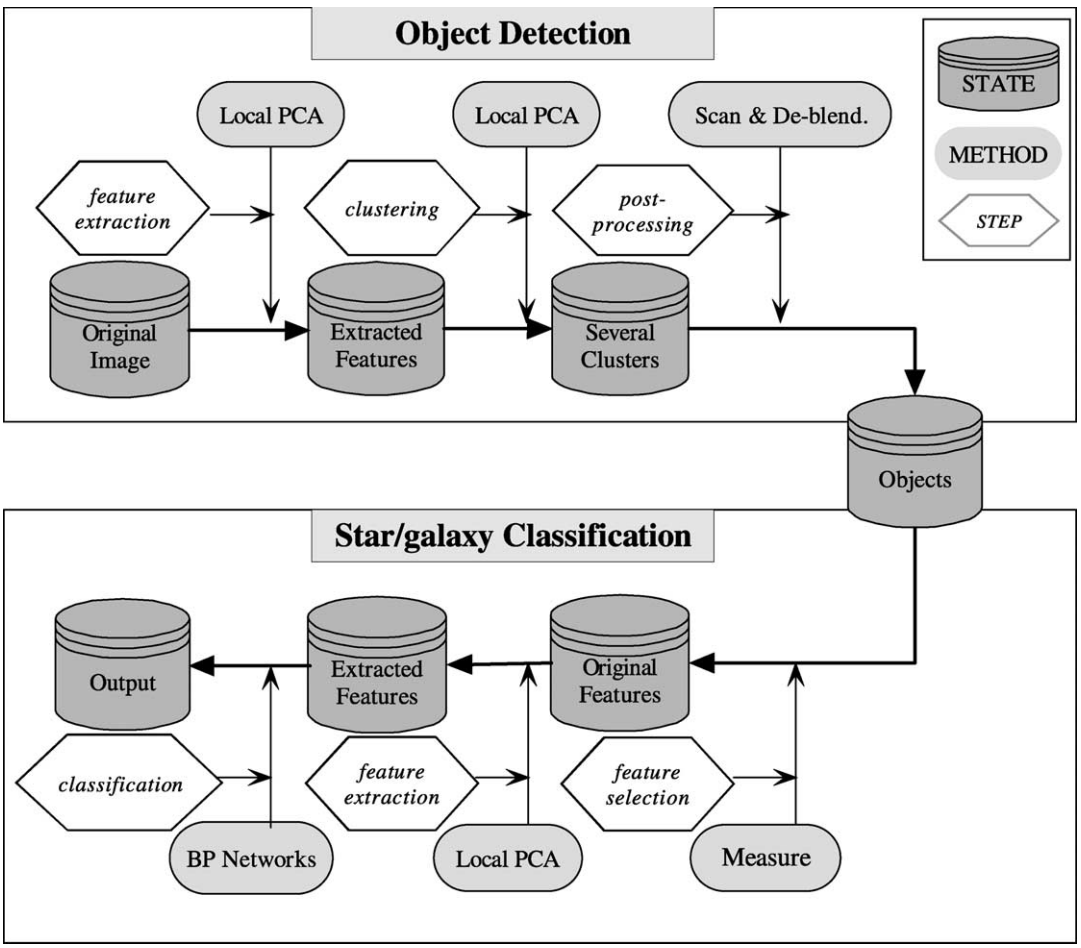

Fig. 2. Suggested System for implementation of object detection and star/galaxy classification.

deciding the number of clusters and the normalization for regularizing learning on data of high dimension and a small size. The rest of the paper is organized in the following way. Section 2 will be devoted to a discussion of the benefits of local subspace analysis in general. Section 3 gives a detailed algorithm for implementing the BYY normalization learning-based local PCA model proposed by Xu (2001b). Section 4 will highlight some beneficial properties of the algorithm that are considered useful for the two tasks discussed above. Experimental illustrations using synthetic data will be demonstrated in Section 5. In Section 6 we will present some results using the BYY normalization learning-based local PCA model with real astronomical data. Section 7 will be devoted to concluding remarks.

2. Benefits of local subspace analysis from a general perspective

In our opinion, local subspace analysis, or local PCA in this paper, is beneficial to the tasks of object detection and star/galaxy classification in two ways. First, local PCA is better than global PCA for the step of feature extraction shown in Fig. 1. Second, it can improve the step of clustering. As a consequence, we conceive the original system shown in Fig. 1 can be improved by adopting local PCA in both feature extraction and clustering. The suggested system is as shown in Fig. 2.

2.1. Local PCA vs global PCA for feature extraction

PCA is also known as *KL* transform (Jolliffe, 1986). Mathematically, it can be expressed as

$$y = A^{\mathrm{T}}x, \tag{1}$$

where x denotes the n-dimensional input vector, y denotes p-dimensional transformed vector, A denotes the $n \times p$ orthonormal transformation matrix whose columns being composed of p principal eigenvectors of the covariance matrix of x. It is well known that when data dimension is reduced from n to p, the Mean Square Error (MSE) upon reconstruction will be minimized via PCA. As a result, PCA can be employed in tasks related to dimension reduction and feature extraction that are frequently encountered in pattern recognition and image processing (Beatty & Manjunath, 1997; Chitroub, Houacine, & Sansal, 2001; Jain, 1989; Taur & Tao, 1996).

However, the performance of PCA will deteriorate for data with little or no global linearity. In such cases, it is preferable to consider some nonlinear transformation tools. For example, when the distribution of data is hardly linear in the global sense as illustrated in Fig. 3, the optimal linear descriptor which corresponds to the optimal feature for the whole data set will run out of expectation. In contrast, as shown in Fig. 4, the result can be improved to a large extent by performing PCA individually for each cluster, thus extracting three locally optimum features. Hereafter we will use the term global PCA to denote the circumstance when PCA is applied in a global sense on all data while local PCA the circumstance when PCA is performed individually on each cluster (Hinton, Dayan, & Revow, 1997; Hinton, Revow, & Dayan, 1995; Xu, 1995; Xu, 1998).

2.2. Local PCA vs gaussian mixture for clustering

Mathematically, gaussian mixture takes the following form:

$$p(x|\theta) = \sum_{i=1}^{k} \alpha_i G(x|\mu_i, \Sigma_i), \tag{2}$$

with $\alpha_i > 0$ and $\sum_{i=1}^{k} \alpha_i = 1$. $G(x|\mu_i, \Sigma_i)$ refers to a gaussian distributed random variable x with mean vector μ_i and covariance matrix Σ_i. Specifying a gaussian mixture model is nothing more than estimating the parameters k and $\{\alpha_i, \mu_i, \Sigma_i\}_{i=1}^{k}$.

From a theoretical point of view, the gaussian mixture model is already quite powerful in the sense that through the covariance matrix Σ_i it can describe each cluster in a structure of either hyper-spherical or hyper-ellipsoid. In contrast, the k-means clustering algorithm can only deal with the case that structure of every cluster is hyper-spherical, which is closely related to a special case of the gaussian mixture model, i.e. a mixture of k gaussian densities with equal proportion and equal variance $\sigma^2\mathbf{I}$ (Xu, 1997a). However, when data dimensionality is high, the gaussian mixture model is not only time-consuming to implement but also often yields poor accurate results. In particular, description of the whole data space by the covariance matrix is less preferred when some high dimensional data can be more appropriately described by a lower dimensional subspace. For the d-dimensional data, the number of free elements needs to be specified for each covariance matrix adds up to $d(d+1)/2$. When d is large and the sample size is small, it is difficult to precisely estimate all these free elements (Hinton et al., 1995; Xu, 2001b,c). Fortunately, such problems can be greatly relieved by replacing the gaussian mixture model with the local PCA model.

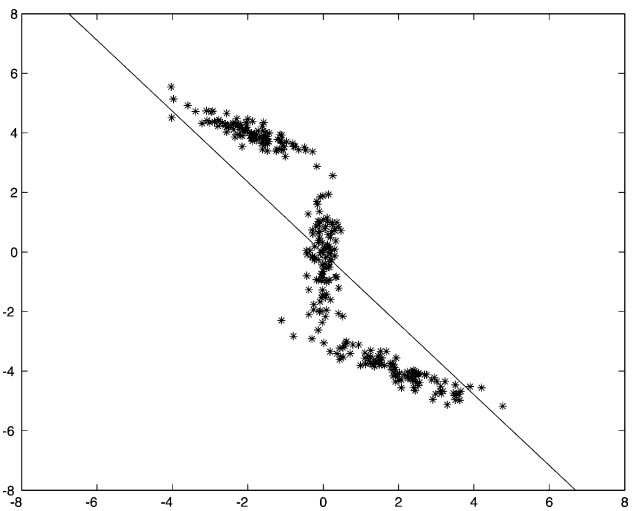

Fig. 3. Data as described by global PCA.

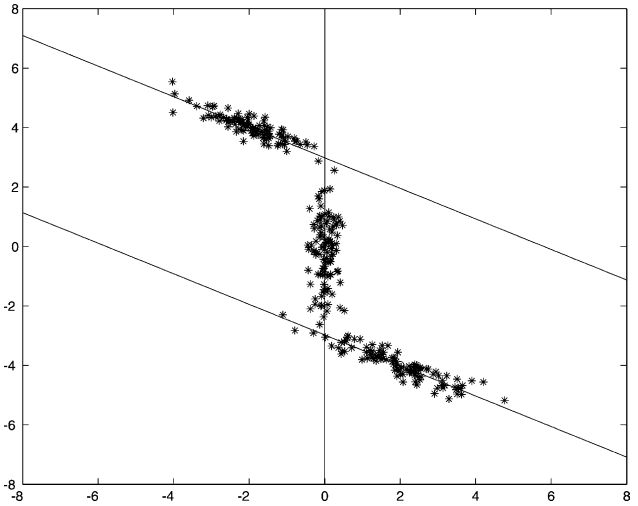

Fig. 4. Data as described by local PCA.

3. An algorithm for implementing the BYY normalization learning based local PCA model

The local PCA algorithm proposed in Kambhatla and Leen (1997) adopts a two-step approach on implementation. First, learning based on the gaussian mixture model is carried out. Subsequently, PCA is performed on the covariance matrix of each cluster. This approach is straightforward to perform, yet it makes no effort to overcome the problems of gaussian mixture mentioned above. Another approach seeks to solve the problem of a large number of free elements of the covariance matrix via constraining the covariance matrix in certain subspace structure form (Luo, Wang, & Kung, 1999; Tipping & Bishop, 1999). Nevertheless, being based on Maximum Likelihood (ML) learning, such models have no model selection ability and therefore are not able to determine the number of clusters. Although some recent works (Jain, Duin, & Mao, 2000; Olivier, Jouzel, & Matouat, 1999) achieve model selection via enumerating some cost functions, a major tradeoff for adoption is the increased

time complexity owing to re-implementation of the whole algorithm for different cluster number k.

Apparently a better model would be the one with automatic model selection ability. The word *automatic* means model complexity is determined in parallel with parameter learning. The local PCA model proposed by Xu (2001b) under the general framework of BYY normalization learning is endowed with such ability.

3.1. BYY normalization learning

As a special case of BYY harmony learning (Xu, 1995 and 1996; Xu, 2000; Xu, 2001a,b; Xu, 2002), the BYY normalization learning shares its key abilities of automatic model selection and regularization. First, the least complexity nature of the BYY harmony learning results in a winner-take-all (WTA) type competition that makes model selection. Second, the normalization provides a 'conscience' that introduces a certain degree of de-learning during the learning process such that the combination of WTA and conscience works with a nature similar to the rival penalized competitive learning (Xu et al., 1993) that is able to make clustering with the number of clusters selected automatically during learning. The detail can be referred to (Xu, 2001b; Xu, 2002).

3.2. Covariance matrix structure of gaussian mixture

We consider the structure of the covariance matrix of the gaussian mixture model in the following form (Xu, 2001b)

$$\Sigma = \sigma_0 \mathbf{I} + W \Psi W^{\mathrm{T}}, \tag{3}$$

where $W = [\phi_1, \phi_2, \ldots, \phi_m]$, $m \leq d$ with d being dimension of the observed data, $\Psi = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_m)$, with $W^{\mathrm{T}}W = \mathbf{I}$, $\sigma_j > 0$ $(j = 0, 1, 2, \ldots, m)$. Decomposition in the above form waives the need to specify $d(d+1)/2$ free parameters for each covariance matrix of gaussian mixture, especially for high-dimensional data d with $m \ll d$. In effect, the number of free elements is substantially reduced to $[m(2d - m + 1)/2]$.

It follows that the set of eigenvalues of Σ in Eq. (3) is:

$$\{\lambda_1 = \sigma_0 + \sigma_1, \ldots, \lambda_m = \sigma_0 + \sigma_m, \lambda_{m+1} = \sigma_0, \ldots, \lambda_d = \sigma_0\} \tag{4}$$

and the corresponding set of eigenvectors is:

$$\{v_1 = \phi_1, \ldots, v_m = \phi_m, v_{m+1}, \ldots, v_d\}. \tag{5}$$

Since $\{\phi_j\}_{j=1}^m$ correspond to the m largest eigenvalues, they are the m principal components of the covariance matrix. Consider a gaussian mixture model with k gaussian components. The set of parameters to be determined becomes $\Theta \cup \{k, m\}$ with $\Theta = \{\alpha_i, \mu_i, W_i, \Psi_i, \sigma_{i,0}\}_{i=1}^k$. In general, for $\theta \in \Theta$, updating via BYY normalization learning is typically (Xu, 2001b)

$$\theta^{\mathrm{new}} = \theta^{\mathrm{old}} + \eta_0 \eta_t(i) \nabla_\theta \ln[\alpha_i G(x|\mu_i, \Sigma_i)], \tag{6}$$

where Σ_i is in the decomposed form as in Eq. (3), η_0 is the learning step size, ∇_θ denotes gradient with respect to θ in the ascent direction of $\ln[\alpha_i G(x|\mu_i, \Sigma_i)]$. A key point is the coefficient $\eta(i)$ resulted from an interaction between a WTA type competition and a 'conscience' de-learning. This coefficient $\eta(i)$ not only can take a positive sign and thereby implying learning, but also can be negative and implying de-learning, which is similar to the rival penalized competitive learning (Xu et al., 1993) that is able to make clustering with the number of clusters selected automatically during learning.

Specific algorithm for learning and updating $\{\alpha_i, \mu_i\}_{i=1}^k$ can be found in Xu (2001b). To derive an algorithm for learning $\{W_i, \Psi_i, \sigma_{i,0}\}_{i=1}^k$ of Eq. (3) requires two constraints to be imposed. First, $\sigma_{i,j} > 0$, $i = 1, 2, \ldots, k$, $j = 0, 1, 2, \ldots, m_i$, where m_i denotes the number of principal components of cluster i and second, $W_i^{\mathrm{T}} W_i = \mathbf{I}$. The first constraint will ensure Σ being positive definite. The second constraint will ensure ϕ's being mutually orthonormal, thus fulfilling the definition of principal components. In implementation, the first constraint can be satisfied via a simple transformation of variables while the second constraint, being a typical constraint optimization problem, may be solved by a constraint gradient projection method which is similar to the so-called Gradient Projection Method (see, for instance, Papalambros and Wilde (2000)). Alternatively, learning on $\{W_i, \Psi_i, \sigma_{i,0}\}_{i=1}^k$ of Eq. (3) can also be made by the Stiefel manifold based algorithms proposed in Xu (2002). A comparison on different implementing algorithms will be made elsewhere.

3.3. Updating rules for the algorithm

3.3.1. Updating rules for σ_0, Ψ

Updating $\sigma_{i,0}$ and Ψ_i requires the first constraint to be satisfied. This can be achieved by writing $\sigma_{i,0} = e^{\varsigma_i}$ and $\Psi_i = \mathrm{dg}(e^{\Omega_i})$, $i = 1, 2, \ldots, k$, where Ω_i is a diagonal matrix, e^{Ω_i} and $\mathrm{dg}(\cdot)$ is defined as for follows. For any arbitrary matrices $A, B, B = e^A$ is defined as $B_{m,n} = e^{A_{m,n}}$, and $B = \mathrm{dg}(A)$ is defined as $B_{i,j} = A_{i,j}$ for $i = j$ and 0 otherwise. Instead of directly modifying each $\sigma_{i,0}$ and Ψ_i, updating can be indirectly made via ς_i and each Ω_i as shown in Table 1.

3.3.2. Updating rules for W

Updating W requires the second constraint to be satisfied and may be achieved in two steps. First, parameter learning is carried out as in Eq. (6). Second, principle components, ranked in non-ascending order of their respective eigenvalues, are projected to a direction orthogonal to the subspace spanned by their predecessors. For illustrative

Table 1
Updating rules for σ_0, Ψ

$\sigma_{i,0}$:

$$\sigma_{i,0} = e^{\varsigma_i^{\text{new}}},$$

$$\varsigma_i^{\text{new}} = \varsigma_i^{\text{old}} + 0.5\eta_0\sigma_{i,0}^{\text{old}}N_{i,ef}\,\text{Tr}((\Sigma_i^{\text{old}})^{-1}R_i),$$

Ψ_i :

$$\Psi_i^{\text{new}} = \text{dg}(e^{\Omega_i^{\text{new}}}),$$

$$\Omega_i^{\text{new}} = \Omega_i^{\text{old}} + 0.5\eta_0 N_{i,ef}\,\Psi_i^{\text{old}}\,\text{dg}(W_i^{\text{T}}(\Sigma_i^{\text{old}})^{-1}R_iW_i),$$

where $N_{i,ef} = \sum_{t=1}^{N}\eta_t(i)$ with $\eta_t(i)$ being the normalization coefficient defined in Xu (2001b),
η_0 denotes the learning step size,
$R_i = (S_i(\Sigma_i^{\text{old}})^{-1} - \mathbf{I})$,

$$S_i = \frac{1}{N_{i,ef}}\sum_{t=1}^{N}\eta_t(i)e_i, e_{i,t}^{\text{T}},$$

$$e_{i,t} = x_t - \mu_i^{\text{old}}, \qquad i = 1, 2, \ldots, k.$$

purpose, consider the simple problem shown below

$$\begin{aligned} &\max \quad f(\phi_1, \phi_2) \\ &\text{subject to} \quad h = \phi_1^{\text{T}}\phi_2 = 0, \end{aligned}$$

where ϕ_1, ϕ_2 are two column vectors. This problem can be solved in two steps. In step 1, update ϕ_i by $\phi_i^{\text{new}} = \phi_i^{\text{old}} + \eta P_i\nabla_{\phi_i}(f)$ where $\nabla_{\phi_i}(f)$ denotes the partial derivative of f with respect to ϕ_i, η is the learning rate, P_i is a projection matrix defined by $P_i = \mathbf{I} - \nabla_{\phi i}(h)(\nabla_{\phi_i}^{\text{T}}(h)\nabla_{\phi_i}(h))^{-1}\nabla_{\phi_i}^{\text{T}}(h)$, by which the vector ϕ_i is projected to a direction that is orthogonal to another one (or the others if there are more than two vectors). Taking the above problem, for example, we have $\nabla_{\phi_1}(h) = \nabla_{\phi_1}(\phi_1^{\text{T}}\phi_2) = \phi_2^{\text{old}}$. As each principal component undergoes updating individually, the principal components may not be orthogonal to each other anymore. As a result, step 2 seeks to ensure that mutual orthogonality is preserved.

Specifically, W is updated according to the two steps shown in Table 2.

4. Some beneficial properties of the BYY normalization learning-based local PCA model

In this section, we would like to highlight some of the beneficial properties of the BYY normalization learning-based local PCA model not shared by their counterparts and discuss their contribution to the tasks of object detection and star/galaxy classification.

Table 2
Updating rules for W

Step 1
$W_i^{\text{new}} = W_i^{\text{old}} + \eta_0 N_{i,ef}\tilde{\delta}W_i$,
$\tilde{\delta}W_i$ is a $d \times m_i$ matrix with jth column $\tilde{\delta}w_{i,j}$,
$\tilde{\delta}w_{i,j} = P_{i,j}\delta w_{i,j}$,
$\delta w_{i,j}$ denotes the jth column vector of δW_i,
$\delta W_i = \Psi_i^{\text{old}}(\Sigma_i^{\text{old}})^{-1}(S_i(\Sigma_i^{\text{old}})^{-1} - \mathbf{I})W_i^{\text{old}}$,
$P_{i,j} = \mathbf{I} - M_{i,j}(M_{i,j}^{\text{T}}M_{i,j})^{-1}M_{i,j}^{\text{T}}$,
$M_{i,j}$ is a $d \times (m_i - 1)$ matrix derived from W_i^{old} by omitting the jth column of W_i^{old}.

Step 2
Sort the diagonal elements of Ψ_i^{new}, together with the corresponding column vectors in W_i^{new} in non-ascending order. Then make each column vector $\phi_{i,j}^{\text{new}}$ of W_i^{new} orthonormal (denoted by $\hat{\phi}_{i,j}^{\text{new}}$) via $\hat{\phi}_{i,j}^{\text{new}} = \tilde{\phi}_{i,j}^{\text{new}}/\|\tilde{\phi}_{i,j}^{\text{new}}\|$, $j = 1, 2, \ldots, m_i$,
$\tilde{\phi}_{i,j}^{\text{new}} = (\mathbf{I} - Q_{i,j}(Q_{i,j}^{\text{T}}Q_{i,j})^{-1}Q_{i,j}^{\text{T}})\phi_{i,j}^{\text{new}}$,
$\phi_{i,j}^{\text{new}}$ denotes the jth column vector of W_i^{new},
$Q_{i,j}$ is a $d \times (j-1)$ matrix with the lth column being $\phi_{i,l}^{\text{new}}$, $l = 1, 2, \ldots, j-1$.

4.1. Selecting the number of clusters k

Model selection is important for the tasks of object detection and star/galaxy classification because model scale and complexity is unknown. For example, the number of clusters is blind before the step of clustering and thus is one of the unknowns that needs to be determined.

Conventionally, the methodology of model selection via cost function has been developed. In literature, cost functions used for the purpose of model selection include minimum description length (Barron, Rissanen, & Yu, 1998), Akaike's information criterion (Akaike, 1974), and harmony value (Xu, 2001a), etc. Model selection via the cost function approach usually involves enumeration of the objective function for different k's and choosing the k that makes the function attain minimum or maximum. However, this approach is repetitive and inefficient. In contrast, model selection can be done during parameter estimation by the normalization learning-based local PCA algorithm (Xu, 2001b).

4.2. Deciding the number of principal components m

Deciding the number of principal components for each cluster forms the core of local subspace analysis. It is another model selection problem for the local PCA model. Interpretation of the number m can be different depending on the purpose of the original task (Hinton et al., 1997; Luo et al., 1999; Tipping & Bishop, 1999). Viewed from the perspective of dimension reduction, the number of principal components could mean the dimensionality of each cluster. Moreover, it is also the number of features when similar question is discussed under the context of feature extraction. Traditionally, the ratio of preserved variance by the first m principal components as shown below is used as the criteria

for deciding what m should be

$$r = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{d} \lambda_i} = \frac{m\sigma_0 + \sum_{i=1}^{m} \sigma_i}{d\sigma_0 + \sum_{i=1}^{m} \sigma_i}. \quad (7)$$

However, this ratio increases monotonically with m. So a critical value for this ratio should be set heuristically. In contrast, in implementation the BYY normalization learning-based local PCA model, we can adopt the following cost function for selecting the best m (Xu, 1997b; Xu, 2001a; Xu, 2002):

$$\min_m J(m) = 0.5\left[m \ln(2\pi) + m + \sum_{i=1}^{m} \lambda_i + d\ln \sigma_0 \right], \quad (8)$$

which was first proposed by Xu in 1997, e.g. Eq. (16) in Xu (1997b).

4.3. Avoiding the dead unit problem

The so-called dead unit problem (Grossberg, 1987; Rumelhart & Zipser, 1985), also known as the under-utilized problem, is induced by local optimization and is frequently encountered during traditional clustering. In the extreme case, this refers to a phenomenon where one cluster occupies data belonging to two or more clusters, leaving the other clusters with no data. This problem can be successfully overcome by the BYY normalization learning-based local PCA model due to the 'conscience' (Desieno, 1988; Xu et al., 1993) ingredient of BYY normalization learning.

5. Experimental illustrations with synthetic data

Various experiments will be presented in this section to illustrate the benefits of the BYY normalization learning-based local PCA model discussed above. To facilitate interpretation, they will be based solely on synthetic data.

5.1. On the model selection ability of cluster number k

The aim of this experiment is to compare the performance of the BYY normalization learning-based local PCA with that of the ML learning-based local PCA on selecting the cluster number k. We use synthetic data generated from four distinct gaussian densities with equal a priori probability. The number of clusters k is initialized as 5 and the algorithm in Tipping and Bishop (1999) is used for implementing the ML-based local PCA. Results are shown in Figs. 5 and 6, respectively, with little circles indicating mean positions. It is clear that the BYY normalization learning-based local PCA is capable of locating the four clusters and automatically getting rid of the redundant

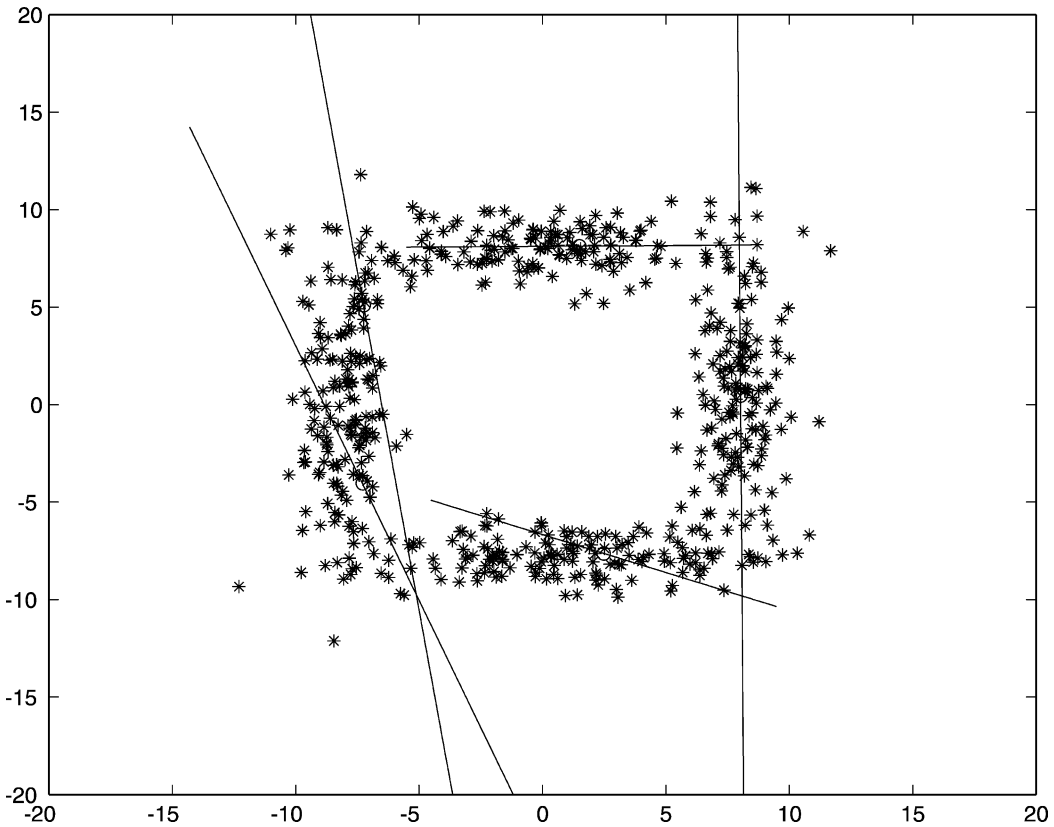

Fig. 5. Results by the ML learning-based local PCA.

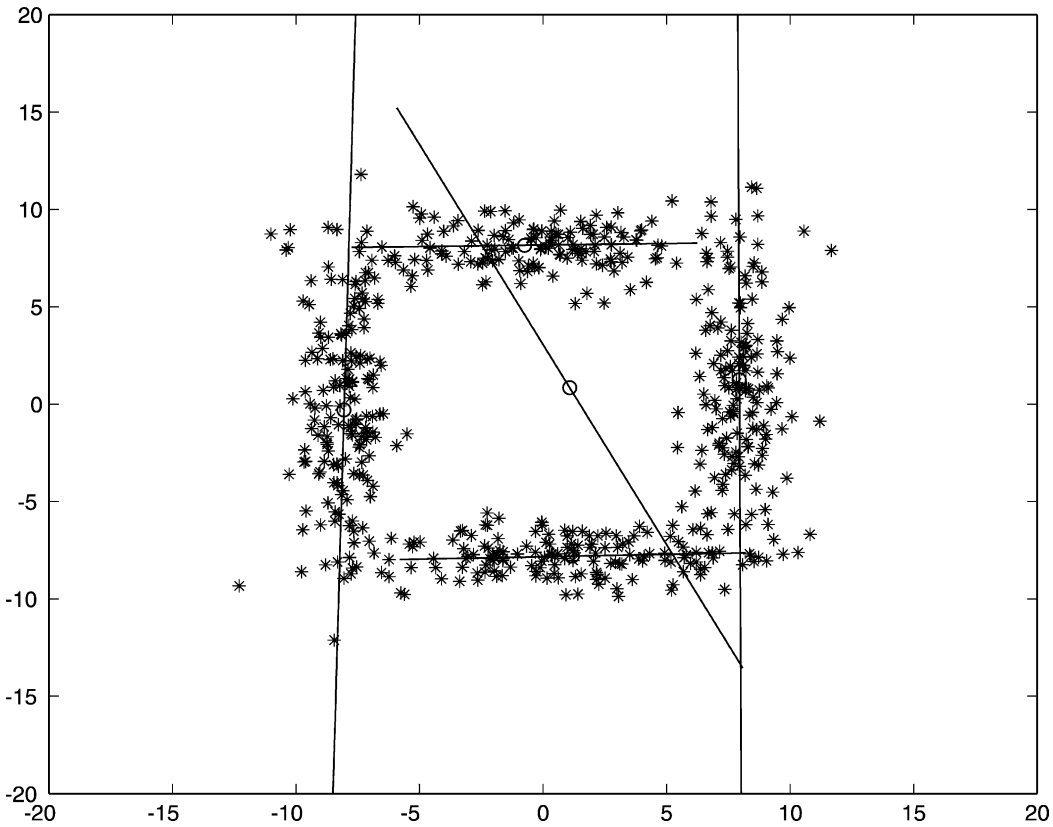

Fig. 6. Results by the BYY normalization learning-based local PCA.

cluster while its counterpart tries to use the initial five clusters to describe the original four-cluster data with a poor performance.

5.2. On exploring local subspace structure

This experiment aims to demonstrate the strength of local PCA on exploring local subspace structure of high dimensional data with small sample size. The BYY normalization learning-based local PCA algorithm together with the cost function $J(m)$ given in Eq. (8) will be used in implementation for deciding m. As pointed out in Section 2, the performance of conventional gaussian mixture using the covariance approach under similar setting is nevertheless disappointing. Thus results by gaussian mixture will be given as well for comparison. We assume the 20-dimensional data generated from three distinct gaussian densities with sample size 30, 50, and 70, respectively, with the three covariance matrices being given by $\Sigma_1 = \Sigma_2 = \Sigma_3 = \mathbf{I}_{20}$, $\Sigma_1(1,1) = \Sigma_2(1,1) = \Sigma_3(1,1) = 36$, $\Sigma_2(2,2) = \Sigma_3(2,2) = 25$, $\Sigma_3(3,3) = 16$.

Results showing the value of $J(m)$ for different number of principal components are shown in Fig. 7. Based on the criterion of selecting the number of principal components that makes $J(m)$ attain minimum, one principal component is chosen for the third cluster, two for the first cluster, and three for the second cluster. Fig. 8 shows how the local subspace structure of 20-dimensional data can be appropriately described by the six principal components.

In contrast, results obtained via decomposing the covariance matrices obtained from modeling a gaussian mixture in a conventional way is far from satisfactory since the recovered principal components do not reflect the actual local subspace structure, in terms of directions of principal components, of the clusters. Without loss of generality, only results of the third cluster is given here. The single principal component whose corresponding eigenvalue is much larger than that of the aggregate of all the other 19 is [0.23, 0.25, 0.22, 0.22, 0.23, 0.22, 0.22, 0.22, 0.22, 0.23, 0.22, 0.21, 0.22, 0.22, 0.22, 0.23, 0.22, 0.22, 0.22, 0.22]$^{\mathrm{T}}$.

The most prominent characteristic of the principal component is that all its attributes are roughly the same in magnitude. However, this result differs vastly from the fact described in Fig. 8. Intuitively, it is well known that the more parameters that needs to be learned, the less exact will be the results. For the local PCA model, the number of free elements that requires to be learned is 124, but for the gaussian mixture model the number is 638.

5.3. On avoiding the dead unit problem

The aim of this experiment is to illustrate the ability of the BYY normalization learning-based local PCA to solve the dead unit problem. For the sake of comparison, the ML

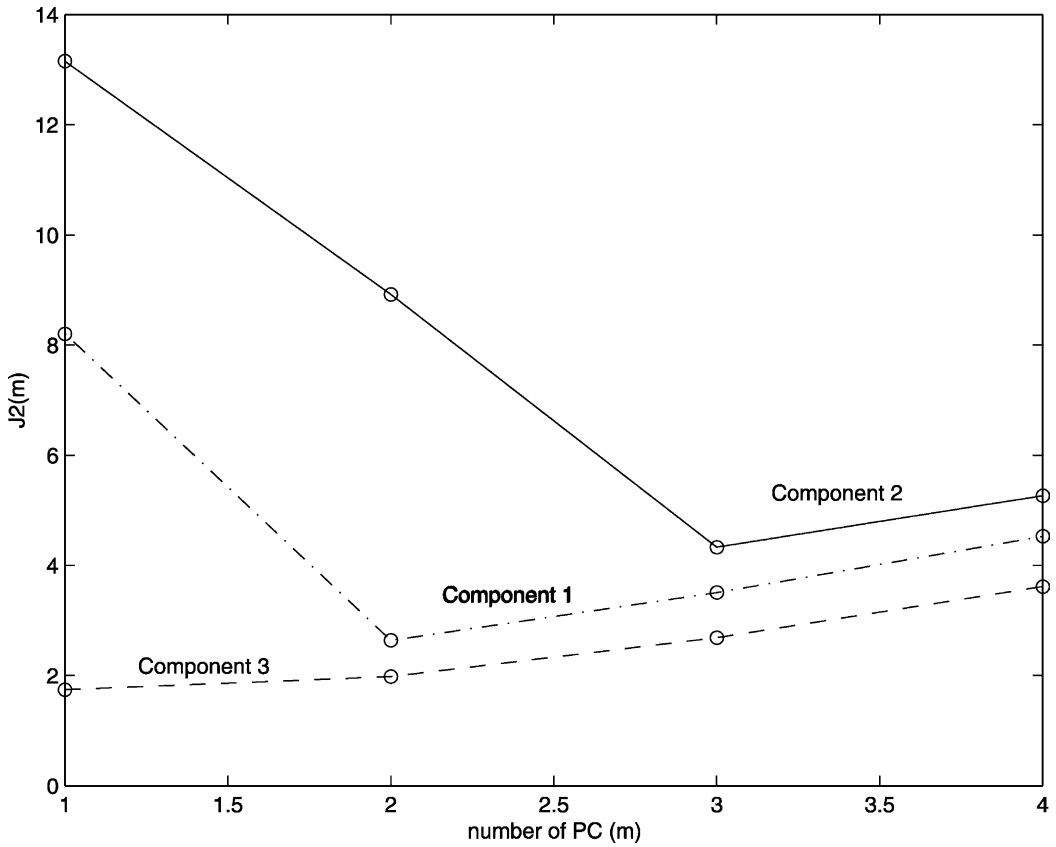

Fig. 7. Model selection of principal components by $J(m)$.

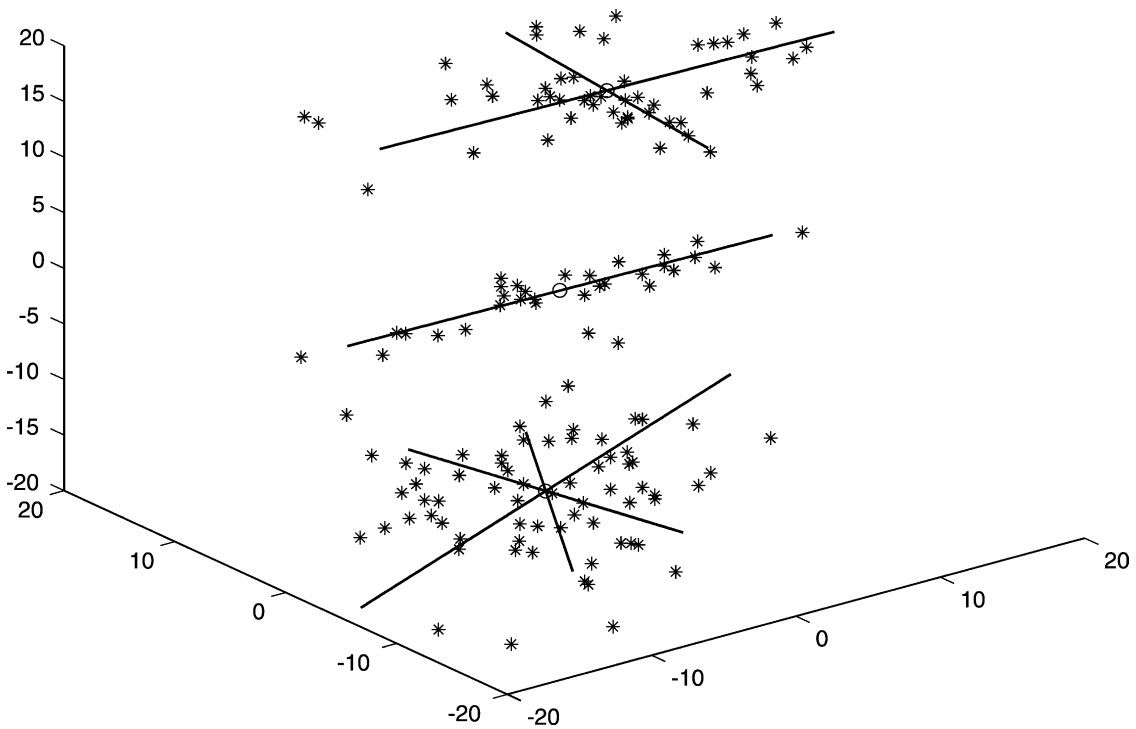

Fig. 8. Local subspace structure as described by the BYY normalization learning-based local PCA.

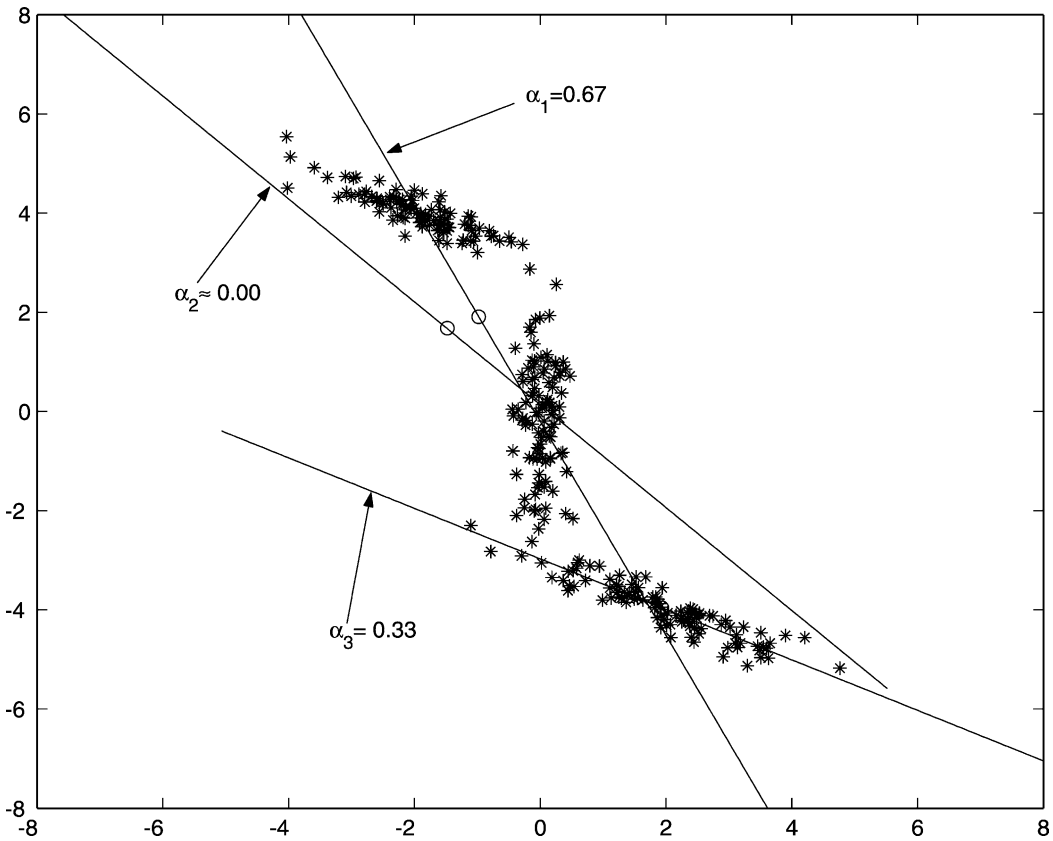

Fig. 9. Results by the ML learning-based local PCA.

learning-based local PCA will be used. Synthetic data from three distinct gaussian densities with equal a priori probability are generated. Experimental results for the ML learning-based local PCA and the BYY normalization learning-based local PCA are shown in Figs. 9 and 10, respectively. In Fig. 9, the cluster with $\alpha_1 = 0.67$ occupies data belonging to two clusters, making the one with $\alpha_2 \approx 0$ 'dead'. In contrast, $\alpha_1 = \alpha_2 = \alpha_3 = 0.33$ in Fig. 10 and the dead unit problem is successfully avoided.

6. Star/galaxy detection and classification using local PCA and astronomical data

This section is devoted to the discussion of real life application of local PCA algorithm to object detection and star/galaxy separation in astronomy. Experimental results on all those steps related to local PCA as shown in Fig. 2 will be shown.

6.1. Data considerations and preprocessing

Experiments will be primarily based on the FITS image shown in Fig. 11. It is a 2000×2000 arcsec^2 region on the North Galactic Pole and is extracted from the POSS-II F No. 443 plate.

Usually the original image is first preprocessed by dividing into $D_x D_y/n^2$ disjoint $n \times n$ blocks, where D_x and D_y denote the width of height of the original image,[1] respectively. In this paper we set $n = 6$. Consequently, one block corresponds to a 36-dimensional input vector. Since direct processing of the 36-dimensional vectors is still rather inefficient, it is more desirable to reduce data dimensionality before further analysis.

6.2. Feature extraction for object detection

Feature extraction for this step is essentially dimension reduction for the input vector. The four steps summarized below are used by the BYY normalization learning-based local PCA for feature extraction.

1. Remove the mean value μ from the original data X
2. Implement the BYY normalization learning-based local PCA algorithm
3. Coding: $Y = \sum_{y=1}^{k} P_{i|x} W_y^{\mathrm{T}}(X - \mu_y)$
4. Decoding: $\hat{X} = \sum_{y=1}^{k} P_{i|x}(W_y Y + \mu_y)\mu$

[1] The width refers to the x axis, or axis '1' in FITS image, and height refers to the y axis, or axis '2' in FITS image.

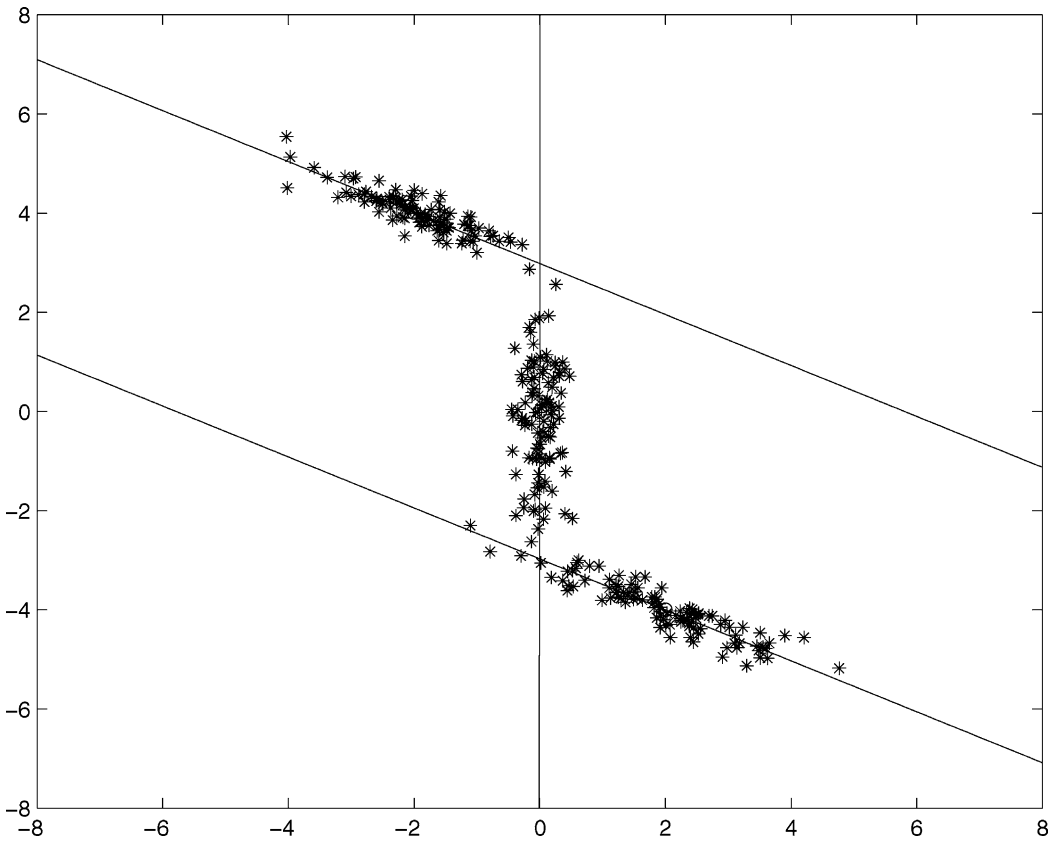

Fig. 10. Results by the BYY normalization learning-based local PCA.

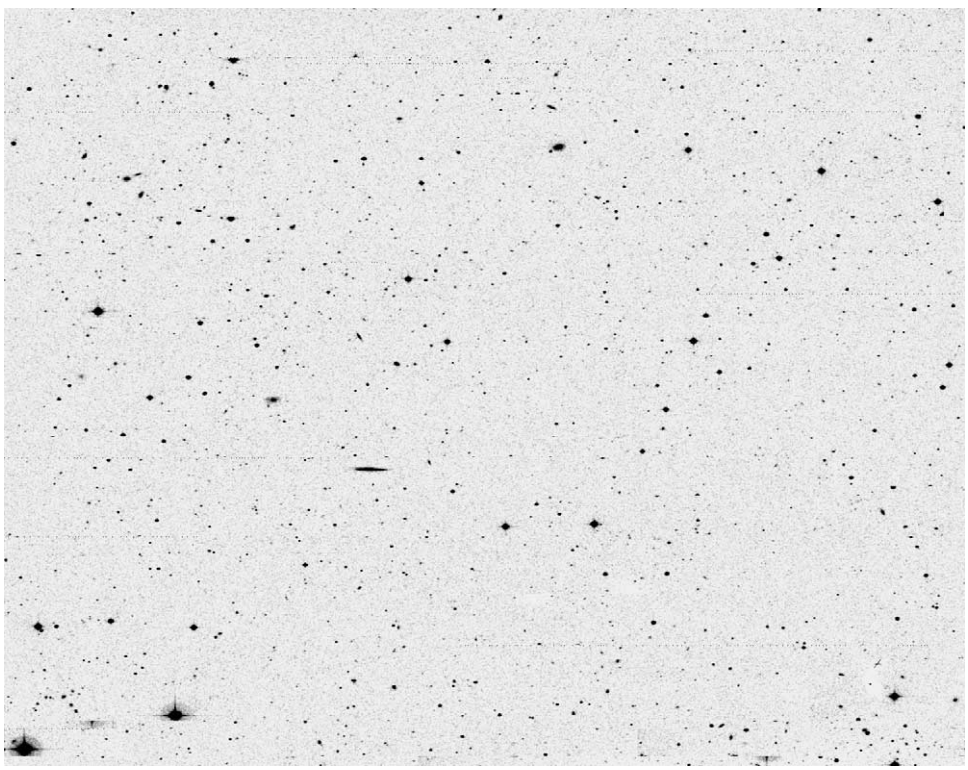

Fig. 11. Original image.

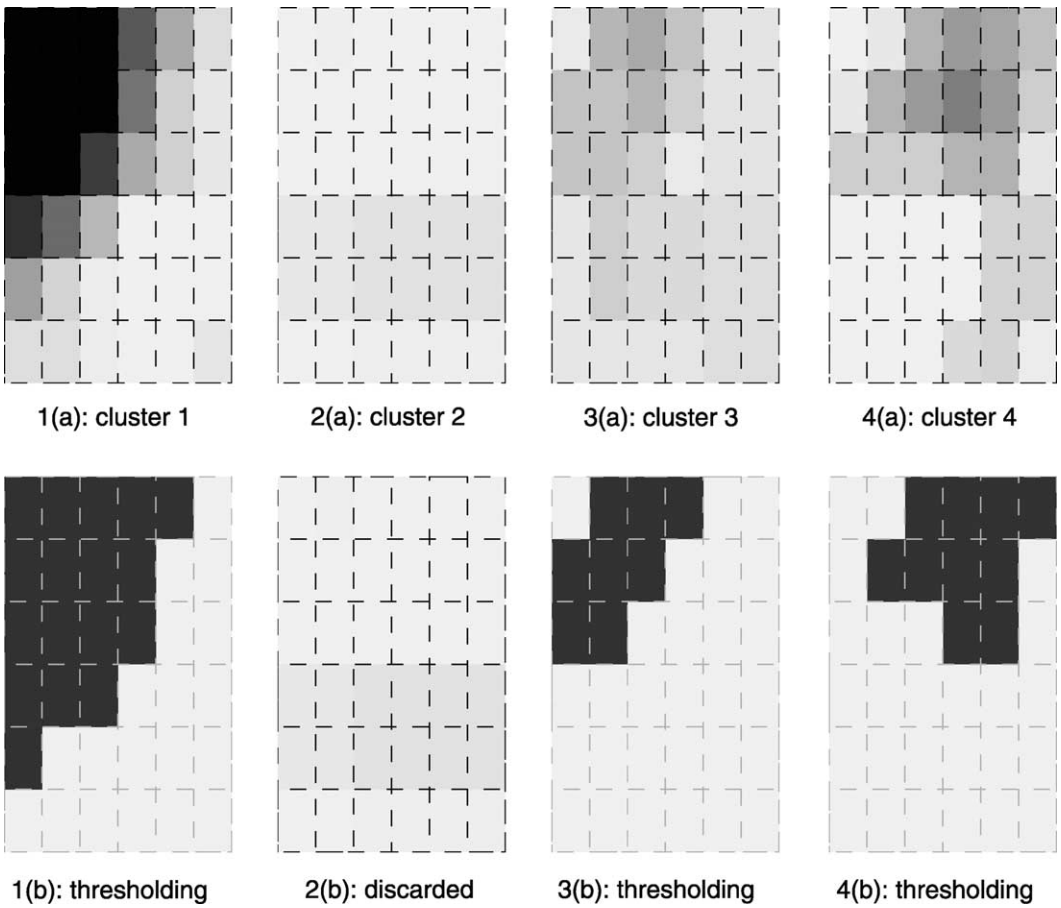

Fig. 12. Typical modules belonging to four different clusters.

where

$$P_{i|x} = \frac{\alpha_i G(x|\mu_i, \Sigma_i)}{\sum_{j=1}^{k} \alpha_j G(x|\mu_j, \Sigma_j)}$$

denotes the probability of the occurrence of ith component conditioned on sample x. Y is the extracted feature and $\hat{X}$ is the reconstructed signal.

The effectiveness of feature extraction can be measured by a performance index called Peak Signal to Noise Ratio (PSNR) of the reconstructed signal

$$\mathrm{PSNR} = 10 \log_{10}\left[\frac{S_\mathrm{p}^2}{\mathrm{MSE}}\right], \tag{9}$$

where S_p denotes the peak signal value. We use the bottom-left (300×300) region of the original image for training and the top-left (300×300) region for testing.

Table 3
List of 20 selected features for each object

Number	Features	Symbol	Number	Features	Symbol
1	Isophotal area	A	11	Ratio 2	r_2
2	Semimajor axis	a	12	Central intensity	I_0
3	Semiminor axis	b	13	Average surface brightness	S
4	Position angle	α	14	Second order moment x	$X2$
5	Object diameter	d	15	Second order moment y	$Y2$
6	Ellipticity	e	16	Second order moment xy	XY
7	Kron radius	r_k	17	Ellipse para x	C_{xx}
8	Area logarithm	c_2	18	Ellipse para y	C_{yy}
9	Peak intensity	I_p	19	Ellipse para xy	C_{xy}
10	Ratio 1	r_1	20	Elongation	E

Table 4
Comparative results of star/galaxy classification

	Star/galaxy classification using supervised BP networks with four features extracted by		
	Local PCA	Global PCA	SBES
% Correct for training data	99.3	99.5	98.4
% Correct for test data	91.6	88.7	86.9

Meanwhile, we fix the number of principal components at 4 to keep the compression rate at 1:9. Similar results by applying global PCA is also given here for comparison. The PSNR for training data is 33.1 for global PCA and 34.83 for local PCA. Using the transformation matrices obtained via training, the PSNR of global PCA on test data is 33.73 while that of local PCA is 35.29. The larger PSNR ratio indicates better performance of local PCA for feature extraction.

6.3. Clustering for object detection

Based on the extracted features, the BYY normalization learning-based local PCA model is used for clustering. First, the number of clusters as determined by the automatic model selection ability of the BYY normalization learning-based local PCA model is 4 with a priori probability of each cluster being 2.5, 87.8, 8.5, and 1.2%, respectively. Also, the number of principal components required for each cluster is found to be 3 for all four clusters. Next, the data belonging to cluster 2 consisting of almost only background pixels are discarded. The decision is based on inspection that almost all blocks containing even one or two object pixels are assigned to the other three clusters.

Considering that the other three clusters are still a mixture of object and background pixels, some segmentation method is needed to remove the background pixels. An efficient approach to remove the background pixels (Liu & Liu, 2000) is to segment the image in each cluster by some computed threshold values. The first row of Fig. 12 shows four typical modules, each belonging to a different cluster. It is obvious that not all the 16 pixels of each module pertaining to either cluster 1, 3, and 4 are object pixels. Post segmentation results are shown in the second row of Fig. 12 for comparison.

6.4. Results of feature extraction for star/galaxy classification

Assume some 20 features as discussed in Andreon et al. (2000), Bertin and Arnouts (1996) and Odewahn et al. (1992) have been selected and shown in Table 3.

We now have a 20-dimensional feature vector for each object. This high dimensional representation of an object is not only difficult to manage, but also unnecessary due to some dependencies between attributes. Thus, it is reasonable to perform feature extraction, or dimension reduction, before classification in the next step. Moreover, since the 20 features are not measured on the same unit or scale, it should be normalized first.

Model selection for both k and m is necessary for this step. As usual, with the help of the BYY normalization learning-based local PCA algorithm, we get the number of clusters 3 and number of principal components for the three clusters $3, 4, 4$, respectively. For consistency, the number of principal components of the first cluster is forced to be 4 for obtaining a four-dimensional combinational feature.

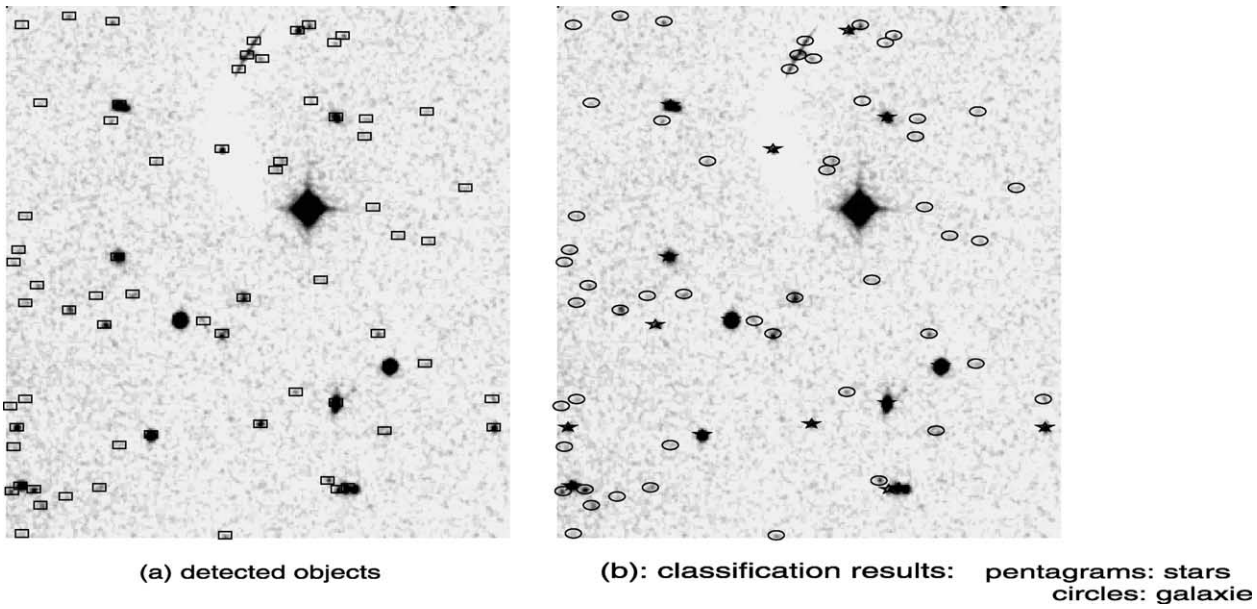

Fig. 13. Results of object detection and star/galaxy classification on the bottom-left portion of the original image.

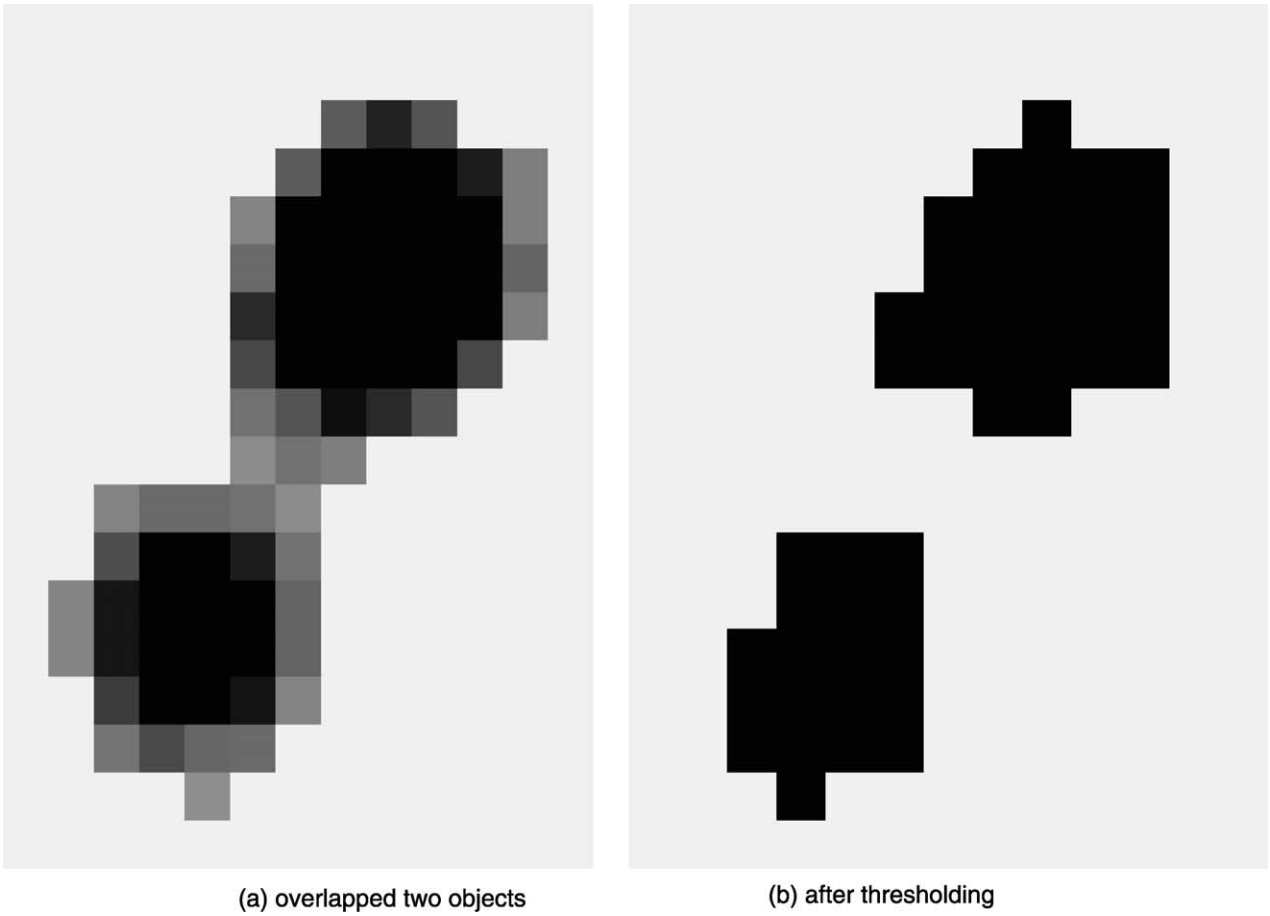

Fig. 14. Example of de-blend.

For comparing the effectiveness of different strategies for feature extraction, Table 4 shows the results of subsequent star/galaxy classification using Back-Propagation (BP) networks[2]. Graphical illustration of typical object detection and classification results can be found in Fig. 13.

6.5. A digression

Although not directly related to local PCA, an important post-processing technique called de-blend is worth mentioning. After the whole image is scanned and the neighboring pixels are connected to get pixel chains, some of them may consist of overlapped objects. De-blend using similar segmentation strategy for clustering is required before measurement and classification. The only difference is that pixels are segmented within each connected pixel chain one by one with a threshold value calculated respectively for each pixel chain (Liu & Liu, 2000). An example is shown in Fig. 14.

[2] In the experiment, we adopt the results obtained by SExtractor as the standard objects catalogue after applying some minor modifications.

7. Conclusion

In the context of object detection and star/galaxy classification, local subspace analysis, or local PCA, is preferred to global PCA for the task of feature extraction, and preferred to gaussian mixture for the task of clustering. Compared with other local PCA models, the one based on the BYY normalization learning firstly given in Xu (2001b) with an algorithm developed earlier in this paper is found to be superior mainly because of the automatic model selection ability for locating cluster number k, the ability of selecting the number of principal components m via a cost function and of avoiding of the dead unit problem. Consequently, by incorporating the BYY normalization learning-based local PCA model into the original system for object detection and star/galaxy classification, we have in effect improved the structure of the old system, as shown by the experiments.

Acknowledgements

The first author would like to thank Dr S. Andreon of *Osservatorio Astronomico di Capodimonte*, Italy for

providing the original FITS image for experiments in Section 6 and also for some beneficial discussions. Both the first and second authors would like to thank their colleagues Mr Him Tang and Miss Xuelei Hu for some helpful discussions. The authors also thank the anonymous reviewers for their helpful comments on revising the original manuscript.

The work described in this paper was fully supported by a grant from the Research Grant Council of the Hong Kong SAR (Project No: CUHK 4336/02E)

References

Akaike, H. (1974). A new look at statistical model identification. *IEEE Transactions on Automatic Control*, *19*, 716–723.

Andreon, S., Gargiulo, G., Longo, G., Tagliaferri, R., & Capuano, N. (2000). Wide field imaging. I. Applications of neural networks to object detection and star/galaxy classification. *Monthly Notices of Royal Astronomical Society*, *319*, 700–716.

Barron, A., Rissanen, J., & Yu, B. (1998). The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, *44*, 2743–2760.

Beatty, M., & Manjunath, B. S. (1997). Dimensionality reduction using multidimensional scaling for image search. *Proceedings of IEEE International Conference on Image Processing, Santa Barbara, CA*, *II*, 835–838.

Bertin, E., & Arnouts, S. (1996). SExtractor: software for source extraction. *Astronomical Astrophysics Supplement Series*, *117*, 393–404.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford: Oxford University Press.

Chitroub, S., Houacine, A., & Sansal, B. (2001). Principal component analysis of multispectral images using neural network. *Proceedings of ACS/IEEE International Conference on Computer Systems and Applications*, 89–95.

Desieno, D. (1988). Adding a conscience to competitive learning. *Proceedings of IEEE International Conference on Neural Networks*, *I*, 117–124.

Fayyad, U. M., Djorgovski, S. G., & Weir, N. (1996). *Automating the analysis and cataloging of sky*. *Advances in knowledge discovery and data mining*, AAAI Press/The MIT Press.

Grossberg, S. (1987). Competitive learning: from iterative activation to adaptive resonance. *Cognitive Science*, *11*, 23–63.

Hinton, G. E., Dayan, P., & Revow, M. (1997). Modeling the manifolds of images of handwritten digits. *IEEE Transactions on Neural Networks*, *8*, 65–74.

Hinton, G. E., Revow, M., & Dayan, P. (1995). Recognizing handwritten digits using mixtures of linear models. *Advances in Neural Information Processing System*, *7*, 1015–1022.

Jain, A. K. (1989). *Fundamentals of digital image processing*. Englewood Cliffs, NJ: Prentice Hall.

Jain, A. K., Duin, R., & Mao, J. (2000). Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 4–37.

Jolliffe, I. T. (1986). *Principal component analysis*. New York: Springer.

Kambhatla, N., & Leen, T. K. (1997). Dimension reduction by local principal component analysis. *Neural Computation*, *9*, 1493–1516.

Liu, Z. Y., & Liu, Y. J. (2000). Image extraction and segmentation in license plate recognition. *Journal of Chinese Information Processing*, *14*, 29–34.

Luo, L., Wang, Y., & Kung, S. Y. (1999). Hierarchy of probabilistic principal component subspaces for data mining. *Neural Networks for Signal Processing IX, Proceedings of the 1999 IEEE Signal Processing Society Workshop*, 497–506.

Odewahn, S. C., Stockwell, E. B., Pennington, R. L., Humphreys, R. M., & Zumach, W. A. (1992). Automated star/galaxy discrimination with neural networks. *Astronomical Journal*, *103*, 318–331.

Olivier, C., Jouzel, F., & Matouat, A. E. (1999). Choice of the number of component clusters in mixture models by information criteria. *Proceeding of Vision Interface '99*, 74–81.

Papalambros, P. Y., & Wilde, D. J. (2000). *Principles of optimal design*. Cambridge: Cambridge University Press.

Rumelhart, D. E., & Zipser, D. (1985). Feature discovery by competitive learning. *Cognitive Science*, *9*, 75–112.

Taur, J. S., & Tao, C. W. (1996). Medical image compression using principal component analysis. *Proceedings of International Conference on Image Processing*, *2*, 903–906.

Tikhonov, A. N., & Arsenin, V. Y. (1977). *Solutions of ill-posed problems*. Baltimore, MD: Winston & Sons.

Tipping, M. E., & Bishop, C. M. (1999). Mixtures of probabilistic principal component analysis. *Neural Computation*, *11*, 443–482.

Xu, L. (1995). A unified learning framework: multisets modeling learning. *Proceedings of World Congress on Neural Networks*, *I*, 35–42.

Xu, L. (1996). A unified learning scheme: Bayesian-Kullback YING-YANG machine. *Advances in Neural Information Processing Systems*, *8*, 444–450. A part of its preliminary version on *Proceedings of ICONIP95-Peking* (pp. 977–988).

Xu, L. (1997a). Bayesian Ying-Yang machine, clustering and number of clusters. *Pattern Recognition Letters*, *18*, 1167–1178.

Xu, L. (1997b). Bayesian Ying-Yang system and theory as a unified statistical learning approach (III): models and algorithms for dependence reduction, data dimension reduction, ICA and supervised learning. In K. M. Wong, et al. (Eds.), *Theoretical aspects of neural computation: A multidisciplinary perspective* (pp. 43–60). Berlin: Springer.

Xu, L. (1998). Rival penalized competitive learning, finite mixture, and multisets clustering. *Proceedings of International Joint Conference on Neural Networks*, *II*, 2525–2530.

Xu, L. (2000). Temporal BYY learning for state space approach, hidden Markov model and blind source separation. *IEEE Transactions on Signal Processing*, *48*, 2132–2144.

Xu, L. (2001a). BYY harmony learning, independent state space and generalized APT financial analyses. *IEEE Transactions on Neural Networks*, *12*, 822–849.

Xu, L. (2001b). Best harmony, unified RPCL and automated model selection for unsupervised and supervised learning on gaussian mixtures, three-layer nets and ME-RBF-SVM models. *International Journal of Neural Systems*, *11*, 43–69.

Xu, L. (2001c). An overview on unsupervised learning from data mining perspective. *WSOM Proceedings on Advances in Self-Organising Maps*, 181–210.

Xu, L. (2002). BYY harmony learning, structural RPCL, and topological self-organizing on mixture models. *Neural Networks*, *15*, 1125–1151.

Xu, L., Krzyzak, A., & Oja, E. (1993). Rival penalized competitive learning for clustering analysis, RBF net and curve detection. *IEEE Transactions on Neural Networks*, *4*, 636–649.