Automated Hierarchical Mixtures of Probabilistic Principal Component Analyzers

Ting Su TSU@ECE.NEU.EDU
Jennifer G. Dy JDY@ECE.NEU.EDU
Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, 02115 USA

Abstract

Many clustering algorithms fail when dealing with high dimensional data. Principal component analysis (PCA) is a popular dimensionality reduction algorithm. However, it assumes a single multivariate Gaussian model, which provides a global linear projection of the data. Mixture of probabilistic principal component analyzers (PPCA) provides a better model to the clustering paradigm. It provides a local linear PCA projection for each multivariate Gaussian cluster component. We extend this model to build hierarchical mixtures of PPCA. Hierarchical clustering provides a flexible representation showing relationships among clusters in various perceptual levels. We introduce an automated hierarchical mixture of PPCA algorithm, which utilizes the integrated classification likelihood as a criterion for splitting and stopping the addition of hierarchical levels. An automated approach requires automated methods for initialization, determining the number of principal component dimensions, and determining when to split clusters. We address each of these in the paper. This automated approach results in a coarse to fine local component model with varying projections and with different number of dimensions for each cluster.

1. Introduction

Dimension reduction is an important problem. First, usually not all the features are useful for producing a desired clustering. Some features are redundant, some may be irrelevant. Second, dimension reduction can save storage and time when dealing with data sets with huge number of features. Dimension reduction for unsupervised clustering is difficult, because we do not have class labels.

There are two main approaches to reduce dimensions: feature selection and feature transformation. Feature selection algorithms select the hopefully best feature subset that discovers "natural" groupings from data (Dy & Brodley, 2000) (M. H. Law, 2002) (Mitra et al., 2002). Feature transformation methods transform data from the original d-dimensional feature space to a new q-dimensional ($q < d$) feature space. Principal component analysis (PCA) (Jolliffe, 1986) is one of the most popular methods for feature transformation. However, PCA is limited since it only defines a single global projection of the data. For complex data, different clusters may need different projection directions; hence, a mixture of local PCA models is desirable. In fact, a hierarchical mixtures of models is even better, because it can provide a coarse-to-fine structure and give more flexibility. In this paper, we introduce an automated algorithm that generates a hierarchical mixtures of models.

Mixture of Probabilistic PCA (PPCA) models (Tipping & Bishop, 1999a) is an extension to the Probabilistic PCA model (Tipping & Bishop, 1999b), which can determine the principal sub-space of the data through maximum-likelihood estimation of the parameters in a Gaussian latent variable model. Moreover, it can be extended to a hierarchical representation as shown in (Bishop & Tipping, 1998). One can also look at other mixture models, such as mixture of factor analyzers (FA) (Ghahramani & Hinton, 1997), and mixture of independent component analyzers(ICA) (Roberts & Penny, 2001). Mixtures of PPCA is merely a special case of mixtures of FA, where PPCA

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004. Copyright 2004 by the authors.

assumes an isotropic covariance matrix for noise while FA assumes a diagonal covariance matrix for noise.

Previous work on hierarchical mixtures of models include building an interactive environment for visualization (Bishop & Tipping, 1998) (Tino & Nabney, 2002). While the human-driven nature of their algorithms is good for visualization, it may make the algorithms expensive and slow, and can produce varying results depending on the user. Moreover, the number of retained principal dimensions in a visualization algorithm is limited to either one, two or three.

In this paper, we introduce an automated hierarchical algorithm. As such, our algorithm allows the flexibility on deciding the number of retained dimensions. Different clusters can have potentially different dimensionalities, thus varying the dimensionality for each cluster may lead to better performance. An automated approach requires automated methods for initialization, determining the number of principal component dimensions, and determining when to split/merge clusters. We address each of these in Section 3.

In section 2, we review the PPCA model, mixture of PPCA, and hierarchical mixtures of PPCA. In section 3, we describe our automated hierarchical algorithm. We, then, report our experimental results in section 4. Finally, we present our conclusions and directions for future work in Section 5.

2. Review on Probabilistic Principal Component Analyzers (PPCA)

This section reviews the theory of PPCA, mixture of PPCA, and hierarchical mixtures of PPCA.

2.1. PPCA

Conventional PCA seeks a q-dimensional ($q < d$) linear projection that best represents the data in a least-squares sense. Consider a data set D of observed d-dimensional vector $D = \{\mathbf{t}_n\}$, where $n \in 1, ..., N$. we first compute the sample covariance matrix:

$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{t}_n - \mu)(\mathbf{t}_n - \mu)^T$, where $\mu = \frac{1}{N}\sum_{n=1}^{N}\mathbf{t}_n$.

Then, the q principal axes $\mathbf{u}_j$ are given by the q dominant eigenvectors (i.e. those with the q largest eigenvalues). The projected value of data $\mathbf{t}_n$ is given by $\mathbf{x}_n = \mathbf{U}_q^T(\mathbf{t}_n - \mu)$, where $\mathbf{U}_q = (\mathbf{u}_1, \ldots, \mathbf{u}_q)$. It can be shown that PCA finds the linear projection that maximizes the variance in the projected space.

Conventional PCA does not define a probability model. PCA can be reformulated as a maximum likelihood solution to a latent variable model (Tipping & Bishop, 1999b). Let $\mathbf{x}$ be a q-dimensional latent variable. The observed variable $\mathbf{t}$ is then defined as a linear transformation of $\mathbf{x}$ with additional noise ϵ: $\mathbf{t} = \mathbf{W}\mathbf{x} + \mu + \epsilon$, Here $\mathbf{W}$ is a $d \times q$ linear transformation matrix, μ is a d-dimension vector that allows $\mathbf{t}$ to have a non-zero mean. Both the latent variable $\mathbf{x}$ and noise ϵ are assumed to be isotropic Gaussian: $p(\mathbf{x}) \sim \mathcal{N}(0, I_q)$ and $p(\epsilon) \sim \mathcal{N}(0, \sigma^2 I_d)$.

Then, the distribution of $\mathbf{t}$ is also Gaussian :

$$p(\mathbf{t}) \sim \mathcal{N}(\mu, \mathbf{W}\mathbf{W}^\mathbf{T} + \sigma^2 I_d) \tag{1}$$

One can compute the maximum-likelihood estimator for the parameters μ, $\mathbf{W}$ and σ^2 from data set D.

The log-likelihood under this model is: $\mathcal{L} = \sum_{n=1}^{N}\log[p(\mathbf{t}_n)]$.

The maximum-likelihood estimates for these parameters are:

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{t}_n \tag{2}$$

$$\sigma^2_{ML} = \frac{1}{d-q}\sum_{i=q+1}^{d}\lambda_i \tag{3}$$

$$\mathbf{W}_{ML} = \mathbf{U}_q(\Lambda_q - \sigma^2_{ML}\mathbf{I})^{1/2}\mathbf{R} \tag{4}$$

where $\lambda_{q+1}, \ldots, \lambda_d$ are the smallest eigenvalues of sample covariance matrix $\mathbf{S}$, the q columns in the $d \times q$ orthogonal matrix $\mathbf{U}_q$ are the q dominant eigenvectors of $\mathbf{S}$, diagonal matrix Λ_q contains the corresponding q largest eigenvalues, and $\mathbf{R}$ is an arbitrary $q \times q$ orthogonal matrix. We set $\mathbf{R} = \mathbf{I}$ in our experiments.

2.2. Mixture of PPCA

Clustering using finite mixture models is a well-known method(McLachlan & Peel, 2000). In this model, one assumes that data is generated from a mixture of component density functions, in which each component density function $p(\mathbf{t}|i)$ represents a cluster. Now that PPCA is defined as a probabilistic model, we can model each mixture component with a single PPCA distribution (Tipping & Bishop, 1999a). The probability density of the observed variable, $\mathbf{t}$, is expressed by:

$$p(\mathbf{t}) = \sum_{i=1}^{k_0}\pi_i p(\mathbf{t}|\mu_i, \sigma_i^2, \mathbf{W}_i) \tag{5}$$

where $p(\mathbf{t}|\mu_i, \sigma_i^2, \mathbf{W}_i)$ denotes a PPCA density function for component i, k_0 is the number of components, and π_i is the mixing proportion of the mixture component i (subject to the constraints: $\pi_i \geq 0$ and

$\sum_{i=1}^{k_0} \pi_i = 1$). The log-likelihood of the observed data is then given by:

$$\mathcal{L} = \sum_{n=1}^{N} \log\{\sum_{i=1}^{k_0} \pi_i p(\mathbf{t}_n|\mu_i, \sigma_i^2, \mathbf{W}_i)\} \qquad (6)$$

It is difficult to optimize (6), so we use the Expectation-Maximization (EM) (Dempster et al., 1977) algorithm to find a local maximum of (6). If we hypothesize a set of indicator variables z_{ni} (also known as "missing data") specifying which model is responsible for generating each data point t_n, then the log-likelihood of the complete-data is given by:

$$\mathcal{L}_c = \sum_{n=1}^{N} \sum_{i=1}^{k_0} z_{ni} \log\{\pi_i p(\mathbf{t}_n|\mu_i, \sigma_i^2, \mathbf{W}_i)\} \qquad (7)$$

We apply the EM algorithm to compute the maximum-likelihood estimation for parameters π_i, μ_i, σ_i^2 and $\mathbf{W}_i$ as follows:
E-step:
The posterior probability of data $\mathbf{t}_n$ belonging to component i, R_{ni} is given by:

$$R_{ni} = E[z_{ni}] = \frac{\pi_i p(\mathbf{t}_n|\mu_i, \sigma_i^2, \mathbf{W}_i)}{p(\mathbf{t}_n)} \qquad (8)$$

where $E[\cdot]$ is the expected value operator.

M-step:

$$\tilde{\pi}_i = \frac{1}{N} \sum_{n=1}^{N} R_{ni} \qquad (9)$$

$$\tilde{\mu}_i = \frac{\sum_{n=1}^{N} R_{ni}\mathbf{t}_n}{\sum_{n=1}^{N} R_{ni}} \qquad (10)$$

To update σ_i^2 and $\mathbf{W}_i$, we first compute the weighted sample covariance matrices, given by:

$$\mathbf{S}_i = \frac{\sum_{n=1}^{N} R_{ni}(\mathbf{t}_n - \tilde{\mu}_i)(\mathbf{t}_n - \tilde{\mu}_i)^T}{\sum_{n=1}^{N} R_{ni}} \qquad (11)$$

then apply (3) and (4). If d is large, one should use an alternative EM approach proposed in (Tipping & Bishop, 1999b) to update σ_i^2 and $\mathbf{W}_i$ for speed up.

2.3. Hierarchical mixtures of PPCA

One can extend the mixture of PPCA models to a hierarchical mixture models (Bishop & Tipping, 1998). Consider an example of extending a two-level mixture models to a three-level mixture models. Suppose each PPCA component i in the second level is extended to a group g_i of PPCA components in the third level, the probability density can be expressed as:

$$p(\mathbf{t}) = \sum_{i=1}^{k_0} \pi_i \sum_{j \in g_i} \pi_{j|i} p(\mathbf{t}|\mu_{i,j}, \sigma_{i,j}^2, \mathbf{W}_{i,j}), \qquad (12)$$

where $p(\mathbf{t}|\mu_{i,j}, \sigma_{i,j}^2, \mathbf{W}_{i,j})$ denotes a single PPCA component, $\pi_{j|i}$ denotes the mixing proportion (subject to the constraints $\pi_{j|i} \geq 0$ and $\sum_j \pi_{j|i} = 1$). If "missing data" at the second level z_{ni} are known, then the corresponding log-likelihood is given by:

$$\sum_{n=1}^{N} \sum_{i=1}^{k_0} z_{ni} \log\{\pi_i \sum_{j \in g_i} \pi_{j|i} p(\mathbf{t}_n|\mu_{i,j}, \sigma_{i,j}^2, \mathbf{W}_{i,j})\} \quad (13)$$

To maximize the expectation of (13) with respect to z_{ni}, we use an EM algorithm again. This has a similar form as the EM algorithm discussed in 2.2, except that in the E-step, the posterior probability of data $\mathbf{t}_n$ belonging to component (i, j) is given by:

$$R_{ni,j} = R_{ni} R_{nj|i} \qquad (14)$$

and

$$R_{nj|i} = \frac{\pi_{j|i} p(\mathbf{t}_n|\mu_{i,j}, \sigma_{i,j}^2, \mathbf{W}_{i,j})}{\sum_{k \in g_i} \pi_{k|i} p(\mathbf{t}|\mu_{i,k}, \sigma_{i,k}^2, \mathbf{W}_{i,k})} \qquad (15)$$

We can recursively apply the above approach to generate a hierarchy of mixtures of PPCA with any number of levels.

3. Our automated hierarchical algorithm

To build our hierarchy, we employ a divisive approach. This way, we start from a coarse data representation and then get a more and more fine data representation until a stopping criterion is reached. One can also build a hierarchy with an agglomerative algorithm. However, to make an agglomerative approach practical in terms of time efficiency, it requires $O(K_{max}^2)$ memory space, where K_{max} is the number of clusters in the initial partitions (Fraley & Raftery, 1998), and in the worst case, agglomerative methods may start at $K_{max} = N$, where N is the number of data points. Another advantage of a divisive approach is that it can be parallelized easily.

There are several issues that need to be addressed for an automated divisive hierarchical algorithm.

3.1. Building the hierarchy and determining when to split clusters

We build the hierarchy as follows: in each level, we perform an order identification test on each cluster to see if it should be split into two children clusters in the next level. In particular, we apply a hierarchical mixtures of PPCA model described in section 2.3, with the number of children clusters in g_i as two. We then compare the parent model with its two children based on a criterion evaluation measure. If the two children outperforms the parent model, we replace the parent with its two off-springs in the next level, otherwise, we copy the parent down unchanged into the next level, and we will not consider to split that cluster in any lower level. We repeat this process until either all the single PPCA clusters in the last level cannot be split into two or the number of clusters reaches K_{max}.

This leads us to the question of "which criterion should we use for splitting?" This question is similar to deciding the number of clusters in a mixture model, which is a difficult task that has not been completely resolved. There are many ways for accessing mixture order, readers can refer to (McLachlan & Peel, 2000) chapter 6 for a review. One cannot simply use the maximum likelihood criterion to decide the number of clusters, because this will lead to a final level where each data point is a cluster (which is a case of overfitting). Some form of regularization (such as penalty methods) is needed. Here, we utilize the integrated classification likelihood (ICL) criterion proposed by (Biernacki et al., 2000):

$$ICL = \mathcal{L}_c - m \log(N)/2 \tag{16}$$

$\mathcal{L}_c$ is the complete-data log-likelihood as defined in equation (7), m is the number of free parameters to be estimated, and N is the number of data points. Note that m varies with k. (Biernacki et al., 2000) estimated z_{ni} in equation (7) by one if $argmax_j R_{nj} = i$ and zero otherwise (a MAP (maximum a posterior) estimate), whereas (McLachlan & Peel, 2000) estimated z_{ni} by its conditional expectation R_{ni}. We chose to replace z_{ni} by R_{ni} because this represents a "soft" [1] clustering solution to the mixture problem. Whereas a MAP estimate of z_{ni} represents a "hard" clustering solution. The criterion we used is called ICL-BIC in (McLachlan & Peel, 2000) and they showed that it outperforms other criteria such as Bayesian information criterion (BIC) (Schwarz, 1978), and Akaike's information criterion (Akaike, 1974).

ICL chooses the number of clusters to maximize Equation (16). ICL basically looks like the more familiar BIC , but instead of penalizing the observed-data log-likelihood $\mathcal{L}$, we penalize the expectation of the complete-data log-likelihood $\mathcal{L}_c$. Recall that the expectation of $\mathcal{L}_c$ is equal to $\mathcal{L} + \sum_{n=1}^{N} \sum_{i=1}^{k_0} R_{ni} \log R_{ni}$ ($\mathcal{L}$ minus the estimated entropy of the fuzzy classification matrix $((R_{ni}))$. ICL, thus, measures the observed-data log-likelihood minus the degree of cluster overlap minus the penalty for the complexity of the model parameters.

We choose ICL out of the several other ways to penalize log-likelihood for two main reasons. Firstly, note that equation (6) contains the log of a summation. One cannot compute the observed-data log-likelihood $\mathcal{L}$ for each component separately, as would be required when determining the number of clusters in the lower levels of the hierarchy. Therefore, in a hierarchical model, it is difficult to apply some popular criteria based on the observed-data log-likelihood, such as BIC, unless one chooses to forego the flexibility of "soft" assignments in a mixture model by assigning "hard" clustering in the lower levels. Secondly, previous experiments reported in (McLachlan & Peel, 2000) chapter 6 showed that ICL outperforms other criteria. (Biernacki et al., 2000) claimed that ICL appears to be more robust than BIC due to the violation of some of the mixture model assumptions, since BIC will tend to overestimate the number of clusters regardless of the cluster overlap if the true model is not in the family of the assumed models. ICL, on the other hand, as explained above penalizes overlaps between clusters.

For clarity, let us summarize the expressions we used for computing ICL for a single PPCA component and for its two offsprings:

ICL for component i is given by:

$$\sum_{n=1}^{N} R_{ni} \log\{\pi_i p(\mathbf{t}_n | \mu_i, \sigma_i^2, \mathbf{W}_i)\} - \frac{m_1 \log(N)}{2} \tag{17}$$

ICL for the two children of component i takes the form:

$$\sum_{n=1}^{N} \sum_{j=1}^{2} R_{ni,j} \log\{\pi_{i,j} p(\mathbf{t}_n | \mu_{i,j}, \sigma_{i,j}^2, \mathbf{W}_{i,j})\} - \frac{m_2 \log(N)}{2} \tag{18}$$

where $R_{ni,j}$ is defined in (14), similarly $\pi_{i,j} = \pi_i \pi_{j|i}$, m_1 and m_2 denote the number of free parameters for component i and its two children components respectively. Our approach for cluster order identification is similar to the one used by X-means, a hierarchical K-means algorithm (Pelleg & Moore, 2000). The two main differences between our approach and their approach are: we use ICL rather than the BIC criterion, and we apply a mixture of component models that assumes soft membership.

[1] "Soft" clustering means each data point can belong to all clusters with some probability of membership, whereas "hard" clustering means that each data point can belong to only one cluster.

Variants and Extensions Instead of splitting the clusters into one or two at every level, one can extend the approach described above by considering splitting to $k =$ one, two, three, or more clusters at every level and apply ICL to pick the best k. One can also build the hierarchy by merging clusters: start at the lowest level by performing a flat clustering and ICL to determine the number of clusters, and then merge the clusters which lead to the largest likelihood, until all the clusters are merged into one component.

3.2. Determining the number of principal dimensions to be retained at each level

Recently some researchers presented methods for choosing the intrinsic dimension of the data set for mixtures of PPCA models (Bishop, 1999)(Bishop, 1998)(Minka, 2000). Those methods are for density estimation. The number of dimensions picked by those methods may be far more than one would use for feature reduction, and hence may not be appropriate for dimension reduction (Minka, 2000). We introduce a simple and fast method analogous to the dimension reduction technique for conventional PCA. For component i, the dimension q_i to be retained in the corresponding sub-components in the next level is the smallest q_i which allows the mean-square-error to be smaller than a threshold, say 10%. In our experiment, we let $q_i > 1$, then q_i is given by:

$$q_i = argmin_{1<q<d}\left(\frac{\sum_{j=1}^{q} \lambda_j}{trace(S_i)} > 90\%\right) \qquad (19)$$

where S_i is defined in equation (11), λ_j are the eigenvalues of S_i. In this way, each cluster component can have potentially different dimensionality, thus providing more flexibility.

3.3. Initialization

It is well known that the EM algorithm may converge to local minima. Different initial parameter values can lead to quite different estimates. Here, we apply 20 random starts to initialize the parameters. The details for initializing the parameters on the top level are as follows: Given the number of clusters k, we select random k data points as the initial centroids, then assign all the data points t_n to the nearest seed, then we compute the corresponding posterior probability R_{ni} by putting $R_{ni} = 1$ if t_n belong to centroid i and $R_{ni} = 0$ otherwise. Start from initial values of R_{ni}, where $n = 1, \ldots, N$, and $i = 1, \ldots, k$. Then, we apply the M-step to update π_i, μ_i, σ_i and W_i for each i. We perform the EM algorithm until convergence initialized with the above process 20 times, and pick the one set of parameters which provided the largest likelihood. The initialization of the model parameters for the lower levels is similar to that presented above, except that we compute the initial posterior probability $R_{ni,j}$ corresponding to sub-component j by setting $R_{ni,j} = R_{ni}$ if t_n belongs to centroid (i, j) and $R_{ni,j} = 0$ otherwise.

3.4. Avoiding the spurious clusters

A common problem with the expectation maximization of mixture models is dealing with "spurious clusters". A fitted component with very small mixing proportion π_i or singular covariance matrix may lead to a relatively large local maximum, but indeed this component is not useful in practice and should be considered as a "spurious cluster".

Since we are working with a dimension reduction algorithm, we need to deal with the singularity of the sample covariance matrices in the projected space. The determinant of a matrix is equal to the product of its eigenvalues. In our algorithm, if the qth largest eigenvalue of a cluster's sample covariance matrix is less than a small number (default $1e - 5$), then we consider that cluster as a spurious cluster.

4. Experiments

In the following experiments, we 1) investigate whether mixtures of PPCA results in better clustering compared to conventional PCA plus EM of multivariate Gaussian mixtures, 2) examine the flexibility of a hierarchical model, and 3) validate the appropriateness of the clusters discovered at each level.

4.1. Data sets

We test our algorithm on three synthetic data sets (toy, oil and chart) and six real data sets. Table 1 summarizes the data set characteristics. Toy and oil data sets are obtained from (Tipping, 1998) and were used in (Bishop & Tipping, 1998) for data visualization. All the other data sets are either from the UCI Machine Learning Repository (Merz et al., 1996) or (Bay, 1999).

4.2. Evaluation Criteria

Since we know the true class labels, we can measure the clustering quality by using measures such as normalized mutual information (Strehl & Ghosh, 2002) and Fowlkes-Mallows index (Fowlkes & Mallows, 1983). We report results for both criteria because no evidence show one is better than the other. These two criteria measure the agreement between the labeled classes and the estimated clusters. Both criteria are in the range

Table 1. Data set descriptions

Data set	# of Instance	# of Features	# of Classes
Toy	300	3	3
Oil	1000	12	3
Chart	600	60	6
Glass	214	9	6
Wine	178	13	3
Optical Digits	5620	64	10
Satellite Image	6435	36	6
Segmentation	2310	19	7
Letter	5000	16	26

$[0, 1]$ and bigger value means better agreement. However, note that a labeled class is not necessarily unimodal, and if our algorithm finds this multi-modality, the value of both criteria will become smaller.

- Normalized Mutual Information(NMI)
  Mutual information $MI(X, Y)$ is a measure of the amount of information shared between two distributions. We normalize it by defining $NMI(X, Y) = MI(X, Y)/\sqrt{H(x)H(Y)}$, where $H(x)$ and $H(Y)$ denote the entropy of X and Y. Let n_{ij} be the number of instances that are in class i as well as in cluster j. Let $n_{i.}$ be the number of instances in class i and $n_{.j}$ be the number of instances in cluster j. Given G classes and K clusters, NMI is given by:

$$NMI = \frac{\sum_{i=1}^{G}\sum_{j=1}^{K} n_{ij}\log(\frac{Nn_{ij}}{n_{i.}n_{.j}})}{\sqrt{(\sum_{i=1}^{G} n_{i.}\log\frac{n_{i.}}{N})(\sum_{j=1}^{K} n_{.j}\log\frac{n_{.j}}{N})}} \tag{20}$$

- Fowlkes-Mallows index (FM index)
  The Fowlkes-Mallows index is the geometric mean of two probabilities: the probability that two random instances are in the same cluster given they are in the same class, and the probability that two random instances are in the same class given they are in the same cluster. Using the same notation as presented above, the FM-index is given by:

$$FM = \frac{\sum_{i=1}^{G}\sum_{j=1}^{K}\begin{pmatrix} n_{ij} \\ 2 \end{pmatrix}}{\sqrt{\sum_{i=1}^{G}\begin{pmatrix} n_{i.} \\ 2 \end{pmatrix}\sum_{j=1}^{K}\begin{pmatrix} n_{.j} \\ 2 \end{pmatrix}}} \tag{21}$$

4.3. Experimental Results

Since conventional PCA is one of most popular methods for feature reduction, we compare our algorithm with PCA and EM hierarchical clustering. To remove the effect of other factors, we use the same criterion (ICL) to decide the number of clusters, the same initialization method, and also equation (19) to determine the retained dimensions in our implementation for PCA+EM algorithm. Note that the number of retained dimensions for PCA+EM is fixed for all clusters in all levels. Table 2 shows the results for the lowest level of PCA + hierarchical EM and for automated hierarchical mixtures of PPCA. We present the results with their NMI, FM, and the number of clusters in the lowest level, K. (We set K_{max} to be twice the number of labeled classes.) We include the number of dimensions, q, retained by conventional PCA in PCA+EM. We do not provide a q column for Auto-PPCA because we can not simply provide a single q value for Auto-PPCA, since each cluster in the hierarchy has different number of retained dimensions.

Table 2. The Results for PCA and PPCA

	PCA + EM				Auto-PPCA		
Data	NMI	FM	q	K	NMI	FM	K
toy	.761	.773	2	2	.966	.987	3
oil	.709	.736	5	7	.763	.777	7
chart	.700	.596	9	8	.469	.474	2
glass	.391	.406	4	6	.407	.547	6
opt. digits	.777	.666	21	19	.777	.690	12
sat. image	.589	.504	4	12	.511	.525	10
seg.	.449	.373	4	11	.412	.412	5
letter	.467	.187	9	30	.513	.226	40
wine	.369	.633	2	2	.299	.417	5
wine -3-8	.478	.627	8	2	.623	.722	4

Interestingly, the mixture of PPCA approach is not always better than PCA+EM. Mixtures of PPCA performed better than PCA+EM in terms of NMI and FM on most small datasets (toy, oil, and glass), which are well modeled by mixtures of Gaussians. PPCA with fewer clusters has a comparable performance with EM + PCA on the large data sets (optical digits, satellite image, segment), except for the letter data, where PPCA performed better. Finally, mixtures of PPCA are worse than PCA+EM on the chart and wine data. Upon closer inspection on the chart data, we observe that the first level of mixture of PPCA grouped the data with cluster one isolating class two, and grouped the rest into cluster two. ICL cannot split cluster two into more sub-clusters because it is highly overlapping.

On the wine data, we noticed that if on the first level mixture of PPCA projects the data to three dimensions and eight dimensions on the second level, hierarchical PPCA results in much better clusterings (as shown on the last row of the Table 2). This indicates that better results can be obtained if we have a better method for determining the number of dimensions in each level than just retaining 90% of the information. We will investigate this further in future work.

A hierarchical mixture of PPCA provides a flexible representation of the data. We obtain different dimensions q for each level and different local projections for each cluster. In fact, due to this allowed flexibility (three on the first level and eight dimensions on the second level) for the wine data, we are able to attain better clustering results than PCA+EM (when $q = 2$ and $q = 8$ as shown on the last two rows of Table 2).

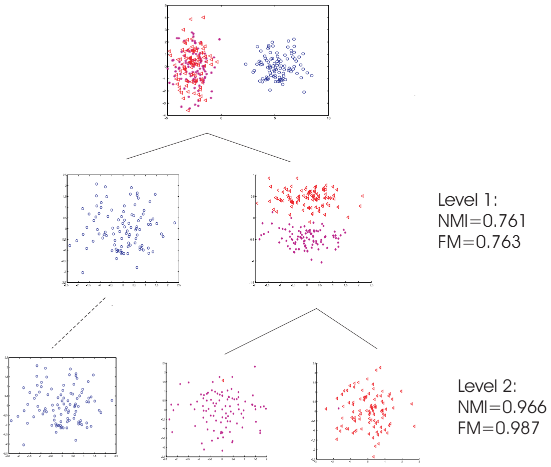

Figure 1. The hierarchy from the toy data set. The data are generated from a mixture of three Gaussians. Two of the clusters are closely spaced as shown in the top figure. Our algorithm was able to split the data into three clusters.

We examine the clustering results for each level and check whether ICL splits up clusters, which does not look like uni-modal Gaussians. Figures 1 and 2 provide a hierarchical visualization of the results for the toy data set and the satellite image data. Due to space limitations, we display the results for one synthetic data and one real data. These are representative of the results for the other data sets. Note that the dimensions for each cluster obtained by our automated approach maybe more than two, and that each data point belongs to all clusters with some probability. To plot the results in two-dimensions, we plot each data point using two leading posterior mean projection components. To prevent confusion, we only plot each data point to the cluster to which it has the largest posterior probability. To show the labeled classes in the scatterplots, we display each class with a different symbol and color.

The toy data set is generated from a mixture of three Gaussians. Two of the clusters are highly overlapped, while the third is well separated from the first two. As shown in Figure 1, our algorithm was able to find the hierarchy and the number of clusters almost perfectly. It discovered two clusters in level one. Then, based on ICL, it splits one of the clusters into two sub-clusters in level two.

The satellite data consists of four digital images of the same scene in 36 different spectral bands. For visualization purpose here, we applied our algorithm on 20% of random subsamples of the original 4435 data points (Note that in Table 2, we ran our algorithm on all the 4435 data points). Our algorithm generated four levels, we can only show the first three levels (again due to space). Again, our approach discovered reasonable clusters. Note that the evaluation criteria, both NMI and FM, decreased from level two to level three since our approach detected the multi-modality of some labeled classes (Recall that a class can actually be multi-modal). Previous study on this image data also indicated the multi-modality of some labeled classes (Bishop & Tipping, 1998).

These figures demonstrate that our automated approach was able to find reasonable clusters and that ICL broke multi-modal clusters appropriately.

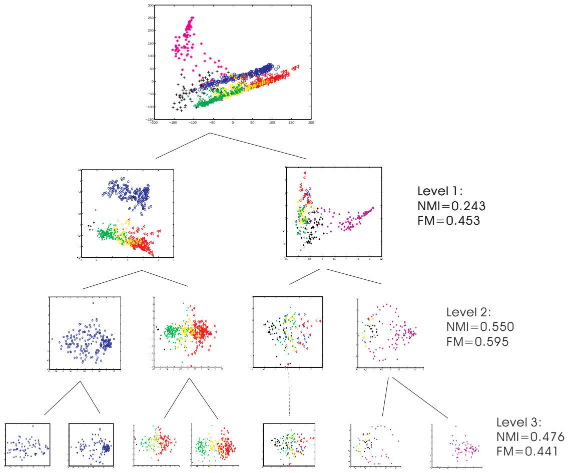

Figure 2. The hierarchy from the satellite image data

5. Conclusions

We have developed an automated hierarchical mixture of principal component analyzers algorithm. To initialize, we apply twenty random restarts, to determine the number of retained dimensions, we keep 90% of the information, and to determine the number of clusters, we utilize the integrated classification likelihood criterion (ICL). Our experimental results show that we were able to obtain reasonable clusters, and that ICL was able to split multi-modal clusters if there is enough separation between those clusters.

Without dimension reduction, EM of a mixture of Gaussians on the original data fails on high-dimensional data.

The additional flexibility offered by an automated hierarchical mixture of PPCA, which allows the algorithm to represent each cluster with a different dimension and a different local PCA projection, enabled it to find better solutions than a global PCA followed by EM. This was well demonstrated by the results on wine data where hierarchical PPCA utilized three dimensions on the first level and eight dimensions on the second level.

Hierarchical clustering provides a flexible representation showing relationships among clusters in various perceptual levels. It results in a coarse to fine local component model with varying projections and with different number of dimensions for each cluster. The automated hierarchical mixtures of PPCA presented here can easily be extended to mixtures of factor analyzers.

Another direction for future work is to investigate how the number of dimensions chosen affects the choice for the number of clusters and vice versa.

Acknowledgments

This research is supported by NSF Grant $No.\ IIS - 0347532$.

References

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

Bay, S. D. (1999). The UCI KDD archive.

Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 719–725.

Bishop, C. (1998). Bayesian PCA. *Neural Information Processing Systems* (pp. 382–388).

Bishop, C. (1999). Variational principal components. *Proceedings of the Ninth International Conf. on Articial Neural Networks* (pp. 509–514).

Bishop, C., & Tipping, M. (1998). A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*, 281–293.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal statistical Society, Series B*, *39*, 1–38.

Dy, J. G., & Brodley, C. E. (2000). Feature subset selection and order identification for unsupervised learning. *Proceedings of the 17th International Conf. on Machine Learning* (pp. 247–254). Morgan Kaufmann, San Francisco, CA.

Fowlkes, E. B., & Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *J. American Statistical Association*, *78*, 553–569.

Fraley, C., & Raftery, A. (1998). How many clusters? which clustering method? - answers via model-based cluster analysis. *The Computer Journal*, *41*, 578–588.

Ghahramani, Z., & Hinton, G. (1997). *the EM algorithm for mixtures of factor analyzers* (Technical Report). University of Toronto.

Jolliffe, I. (1986). *Principal component analysis*. New York: Spring-Verlag.

M. H. Law, M. Figueiredo, A. K. J. (2002). Feature selection in mixture-based clustering. *Advances in Neural Information Processing Systems* (pp. 609–616).

McLachlan, G., & Peel, D. (2000). *Finite mixture models*. New York: Wiley.

Merz, C. J., Murphy, P., & Aha, D. (1996). UCI repository of machine learning databases.

Minka, T. P. (2000). Automatic choice of dimensionality for PCA. *NIPS* (pp. 598–604).

Mitra, P., Murthy, C., & Pal, S. (2002). Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 301–312.

Pelleg, D., & Moore, A. (2000). X-means: Extending k-means with efficient estimation of the number of clusters. *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 727–734).

Roberts, S. J., & Penny, W. D. (2001). Mixtures of independent component analysers. *International conf. on Artificial Neural Networks* (pp. 527–534).

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*, 461–464.

Strehl, A., & Ghosh, J. (2002). Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research*, *3*, 583–617.

Tino, P., & Nabney, I. (2002). Hierarchical GTM: Constructing localized nonlinear projection manifolds in a principled way. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 639–656.

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

Tipping, M., & Bishop, C. (1999b). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, series B*, *61*, 611–622.

Tipping, M. E. (1998). PHIVIS visualization software.