

---

# Degenerate Expectation-Maximization Algorithm for Local Dimension Reduction

Xiaodong Lin<sup>1</sup> and Yu Zhu<sup>2</sup>

<sup>1</sup> Statistical and Applied Mathematical Science Institute, RTP, NC, 27709 USA  
University of Cincinnati, Cincinnati, OH 45221 [linxd@samsi.info](mailto:linxd@samsi.info)

<sup>2</sup> Department of Statistics, Purdue University, West Lafayette, IN, 47907 USA  
[yuzhu@stat.purdue.edu](mailto:yuzhu@stat.purdue.edu)

**Summary.** Dimension reduction techniques based on principal component analysis (PCA) and factor analysis are commonly used in statistical data analysis. The effectiveness of these methods is limited by their global nature. Recent efforts have focused on relaxing global restrictions in order to identify subsets of data that are concentrated on lower dimensional subspaces. In this paper, we propose an adaptive local dimension reduction method, called the Degenerate Expectation-Maximization Algorithm (DEM). This method is based on the finite mixture model. We demonstrate that the DEM yields significantly better results than the local PCA (LPCA) and other related methods in a variety of synthetic and real datasets. The DEM algorithm can be used in various applications ranging from clustering to information retrieval.

*Keywords:*

Degeneration; EM algorithm; Principal Component Analysis; Local Dimension Reduction

## 1 Introduction

Dimension reduction is a commonly used technique for analyzing large datasets. Over the years, several methods have been proposed for dimension reduction, among which are Principal Component Analysis, Factor Analysis, Self Organizing Map (Kohonen 1989, 1990) and Principal Curve (Hastie and Stuetzle 1989). One important assumption for these methods is the existence of a global low dimensional structure. However, this does not hold in general. For a high dimensional data set, different subsets of the data may concentrate on different subspaces. Thus it is important to develop methods that can identify low dimensional structures locally.

As an early effort to solve this problem, Kambhatla and Leen (1997) and Archer(1999) proposed Local Principal Component Analysis (LPCA). The

high dimensional data is assumed to consist of a number of clustered subsets. Starting from an initial assignment of cluster membership to the data points, the LPCA determine a principal subspace for each cluster. Then it allocates data points, one by one, to the clusters whose principal subspaces are the closest to them. This iterative procedure continues until a stopping criterion is met. It has been shown that the LPCA has many advantages over the popular classical dimension reduction techniques.

In this paper we propose the DEM algorithm based on the finite mixture model for local dimension reduction. In a finite mixture model, each component is modelled by a probability density belonging to a parametric family  $f_j(x; \theta_j)$  and the mixture density is

$$f(x; \theta) = \sum_{j=1}^m \pi_j f_j(x; \theta_j), \quad (1)$$

where  $\pi_j > 0$  and  $\sum_{j=1}^m \pi_j = 1$ . When  $f_j(x; \theta_j)$  are multivariate Gaussian distributions with parameters  $\mu_j$  and  $\Sigma_j$  for  $1 \leq j \leq m$ , (1) is a typical Gaussian mixture model. The Gaussian mixture model has been used for model based clustering and the EM algorithm is usually used for parameter estimation. Note that the covariance matrices for the Gaussian mixture model contain the information regarding the shape of the components. When some of the covariance matrices become singular or near singular, it implies that the corresponding components are concentrated on low dimensional subspaces. Thus the finite mixture model can also be used as a device for local dimension reduction.

The existing methods for finite mixture model cannot be directly applied to achieve our goal. First, the likelihood function for the Gaussian mixture density with unequal covariance matrices is unbounded (Kiefer and Wolfowitz (1956)), so many numerical methods for computing the MLE may not work well. Hathaway (1985) and Ciuperea et. al. (2003) proposed the constrained and the penalized methods respectively to address this problem. They avoided the likelihood unboundedness problem by discarding the degenerate components, genuine or spurious. Their methods are quite unstable when the true parameters are close to the boundary of the parameter space. Secondly, in computation, when some parameter estimates get close to degeneration which causes the likelihood function to be infinity, the computing has to stop. However, other parameter estimates may not have converged yet. Therefore, the resulted parameter estimates cannot be used. The DEM algorithm described below solves these two problems by adaptively adding perturbations to the singular covariance matrices of the corresponding components. It will be demonstrated later that the DEM can also distinguish genuine degenerate components from the spurious ones and achieve the goal of local dimension reduction.

This paper is organized as follows. In Section 2, we discuss the likelihood unboundedness problem and the breakdown of the EM algorithm, then

propose the DEM algorithm to address these issues. Simulation results are presented in Section 3. Section 4 contains the conclusions and future work.

## 2 Degenerate EM algorithm (DEM)

In the Gaussian mixture model, both spurious and genuine degeneration can lead to likelihood unboundedness. Spurious degeneration occurs when a cluster with a small number of points lies on a lower dimensional subspace. Genuine degeneration, on the other hand, occurs when a large portion of the data are concentrated on a lower dimensional subspace. Traditional approaches avoid degeneracy by confining the parameters to the interior of the parameter space. In high dimensional data analysis, however, it is likely that one or more components can be genuinely degenerate. New methods are needed to distinguish between these two cases.

### 2.1 Likelihood infinity and break down point

It is known that the likelihood function of a Gaussian mixture model goes to infinity when certain parameters reach the boundary of the parameter space. In the EM algorithm, it is important to find out when the breakdown occurs. Given the observed data  $\mathbf{x}$ , at the E-step of the ( $k$ )th iteration, expected value of the complete data likelihood is

$$E_{\theta^{(k)}}(l_c(\theta)|\mathbf{x}) = \sum_{j=1}^m \sum_{i=1}^n E_{\theta^{(k)}}(Z_{ij}|\mathbf{x}) \{\log \pi_j + \log f_j(x_i; \theta_j)\}, \quad (2)$$

where  $Z_{ij}$  is the usual missing value and  $\theta^{(k)}$  is the parameter estimate after ( $k$ )th iteration. Define

$$z_{ij}^{(k)} \doteq E_{\theta^{(k)}}(Z_{ij}|\mathbf{x}). \quad (3)$$

It can be shown that  $E_{\theta^{(k)}}(l_c(\theta)|\mathbf{x})$  reaches infinity only when  $z_{ij}$  are strictly 0 or 1 for some components. Namely, when

$$z_{ij}^{(k)} = \begin{cases} 1, & \text{while } x_i \in \text{the degenerate component } j, \\ 0, & \text{else.} \end{cases} \quad (4)$$

Before this point is reached, maximizing  $E_{\theta^{(k)}}(l_c(\theta)|\mathbf{x})$  ensures the increase of the observed data log likelihood. When  $E_{\theta^{(k)}}(l_c(\theta)|\mathbf{x})$  reaches infinity, we can detect the degenerate components with singular covariance matrices.

The first goal of the DEM algorithm is to prevent the iterative procedure from breaking down. To achieve this goal, artificial perturbations are applied to the detected singular covariance matrices. This can force the covariance matrices to become non-singular, thus assures the likelihood function to be bounded. The second goal of the DEM algorithm is to discriminate between

spurious and genuine degeneration. The points trapped in spurious degenerate components should be reassigned. This is done by adjusting their corresponding  $z_{ij}$  values: the probability that the datum  $x_i$  is assigned to component  $j$ . For a spuriously degenerate component, a large perturbation needs to be applied on the covariance matrix in order to deviate the component from degeneration. Meanwhile, for a genuine degenerate components, the perturbation on the covariance matrix should be relatively small so that the degeneracy can be retained. The proper level of perturbation depends on the size of the degenerate component and its relative distance from the other components.

## 2.2 The DEM algorithm

The key steps of the proposed DEM algorithm include the identification of the degenerate components and directions and the perturbation of the singular covariance matrices. When a degenerate component, e.g., component  $j$ , is identified, its covariance matrix will be decomposed into  $D'_j \Lambda_j D_j$ , with  $\Lambda_j = \text{diag}\{\lambda_1, \dots, \lambda_d\}$ . Here,  $\lambda_1, \dots, \lambda_d$  are the sorted eigenvalues of  $\Sigma_j$  in a decreasing order and the columns of  $D_j$  consist of the corresponding eigenvectors. Should this matrix be singular, there exists an  $s$  such that when  $p \geq s$ ,  $\lambda_p = 0$ . For the identification of a nearly singular covariance matrix, the criterion can be defined as:  $\lambda_p \leq \alpha$ , where  $\alpha$  is a pre-specified threshold.

Assume an  $m$ -component Gaussian mixture model. Without loss of generality, assume further that component 1 is degenerate at a breakdown point. The observed data log likelihood and the complete data log likelihood are

$$l = \sum_{i=1}^n \log \left( \sum_{j=1}^m \pi_j f_j(x_i; \theta_j) \right), \quad (5)$$

and

$$l_c = \sum_{i=1}^n \sum_{j=1}^m Z_{ij} \{ \log \pi_j + \log f_j(x_i; \theta_j) \}, \quad (6)$$

respectively, where,

$$f_1(x_i, \theta_1) = \frac{1}{\sqrt{\lambda_1 \cdots \lambda_{s-1}}} \exp \left\{ -\frac{1}{2} (x_i - \mu_1)' \Sigma_1^- (x_i - \mu_1) \right\} \cdot \delta(\mathbf{N}\mathbf{X} = \mathbf{B}),$$

and

$$\Sigma_1^- = D' \cdot \text{diag}\{\lambda_1^{-1}, \dots, \lambda_{s-1}^{-1}, 0, \dots, 0\} \cdot D. \quad (7)$$

The  $\mathbf{N}\mathbf{X} = \mathbf{B}$  is a set of  $d - s$  linear equations representing the subspace spanned by the non-degenerate eigen-directions.  $\Sigma_1^-$  is the generalized inverse of  $\Sigma_1$  and  $\delta$  is the usual delta function.

In order to achieve the goals discussed in the previous subsection, we substitute  $\delta(\cdot)$  with a bounded regular density function. With an artificial perturbation added, the proposed substituting function  $K(x)$  is

$$K(x) = \frac{1}{\sqrt{\lambda_s \cdots \lambda_d}} \exp\left\{-\frac{1}{2}(x - \mu_1)'(\Sigma_1^* - \Sigma_1^-)(x - \mu_1)\right\}, \quad (8)$$

where

$$\Sigma_1^* = D' \Lambda^* D, \quad (9)$$

and

$$\Lambda^* = \text{diag}\{\lambda_1^{-1}, \dots, \lambda_{s-1}^{-1}, \lambda_s^{-1}, \text{cdots}, \lambda_d^{-1}\}. \quad (10)$$

For simplicity, we assume  $\lambda_s = \cdots = \lambda_d = \lambda^*$ . Clearly  $\lambda^*$  controls the level of perturbation applied to the covariance matrix, and using a proper  $\lambda^*$  value is essential for the success of the algorithm. For spurious degeneration,  $\lambda^*$  should be large so that it is difficult for the algorithm to return to the same spurious solution. For genuine degeneration, this value should be small so that the true degenerate components can be preserved. adaptiveness of  $\lambda^*$  to data

The values of  $z_{ij}^{(k)}$  determine the assignment of each data point to certain component. After the perturbation,  $\Sigma_1$  in  $\theta^{(k)}$  is replaced by  $\Sigma_1^*$ , and the corresponding  $z_{i1}^{(k)}$  becomes  $z_{i1}^*$ , where

$$z_{i1}^{(k)} = \frac{\pi_1^{(k)} f_1(x_i, \theta^{(k)})}{\sum_{j=1}^m \pi_j^{(k)} f_j(x_i, \theta^{(k)})},$$

$$z_{i1}^* = \frac{\pi_1^{(k)} f_1(x_i, \theta^*)}{\pi_1^{(k)} f_1(x_i, \theta^*) + \sum_{j=2}^m \pi_j^{(k)} f_j(x_i, \theta^{(k)})},$$

and  $\theta^*$  denotes the new parameter estimates. The  $i$  is the index for the data points in the degenerate component only. Define

$$U_1 = \sum_{i=1}^{n_1} z_{i1}^{(k)}, \quad U_1^* = \sum_{i=1}^{n_1} z_{i1}^*,$$

$$D_U = |U_1 - U_1^*|,$$

where  $n_1$  is the number of points in component 1.  $D_U$  indicates the effect of perturbation on the degenerate component. the data points belonging to Finally, the perturbation  $\lambda_{DEM}^*$  is defined as

$$\lambda_{DEM}^* = \max\{\lambda^* | D_U \leq \beta\}, \quad (11)$$

where  $\beta$  is a pre-specified threshold. Let us discuss two properties of  $D_U$  which are related to the size of the degenerate component and the relative distance between components.

1.  $D_U$  is positively associated with the size of the degenerate component. For a fixed  $\beta$ , a spurious degenerate component with small size gives a large  $\lambda_{DEM}^*$  value. After applying the perturbation with  $\lambda_{DEM}^*$ , the nearby data points will be absorbed into the degenerate component so that the DEM algorithm can divert from breakdown.

2. Let the distance between components  $j$  and  $k$  be the usual Kullback-Leibler distance between  $f_j(x|\theta_j)$  and  $f_k(x|\theta_k)$ . Then  $D_U$  depends on the relative distance between the degenerate component and the other components. When a degenerate component is far away from the others, a large  $\lambda_{DEM}^*$  is needed to reach a certain  $\beta$  level.

In practice, if a component is found to be degenerate repeatedly, the degeneration is likely to be genuine. In order for the algorithm to converge, it is necessary to have the perturbation level decrease to zero. To achieve this, every time when the same degeneration is repeated,  $\beta$  will be decreased by a certain ratio. A value  $s$  is used to count the number of reoccurrence of the same degeneration. Once this value exceeds a threshold (in the algorithm we set to 10), the DEM algorithm declares that a genuine degenerate component has been founded and stop the iteration.

The DEM algorithm is summarized in Algorithm 1. For simplicity and the clarity of presentation, we assume there exists only one degenerate component at a time. The  $\alpha$  is used as a threshold to identify the nearly degenerate components. If we care looking for complete degeneration,  $\alpha$  is set to be 0. Usually  $\alpha$  is a value specified by the user.

**Table 1.** The three components detected by DEM for Synthetic Dataset 1 and the corresponding eigenvalues.

Comp.	Corresponding Eigenvalues		
1	1.0722954	0.9034532	0.8311285
2	0.9034483	0.9023349	0
3	0.8713843	1.2013586e-03	1.6785237e-05

### 3 Experimental results

A number of simulations are performed to compare the DEM with the LPCA algorithm. We show that the DEM gives significantly better results in various datasets. In the following experiments, we set  $\beta = 1$  and  $\alpha = 0.01$ .

#### *Synthetic Dataset 1.*

This dataset is comprised of three components, one component is a three-dimensional sphere with 120 points, another is a two-dimensional plane with 80 points, and the third, a one-dimensional line with 80 points. The means of these three components are  $(0,0,0)$ ,  $(0,0,1)$ , and  $(0,0,-1)$ , respectively. We report the eigenvalues of the three component covariance matrices given by the DEM in Table (??).

---

**Algorithm 1** The Degenerate EM algorithm
 

---

- 1: For each component  $j$ , set counters  $\beta_j = \beta_0$  and  $s_j = 0$ .
- 2: Run the EM algorithm until a degenerate component is found, say the ( $l$ )th component.
- 3: Decompose  $\Sigma_l$  into  $D_l' \Lambda_l D_l$  as described in Section (2.2).
- 4: Identify zero (close to zero for  $\alpha > 0$ ) eigenvalues  $\lambda_s, \dots, \lambda_d$  for those  $\lambda \leq \alpha$ .
- 5: **while** (zero (close to zero for  $\alpha > 0$ ) eigenvalues exist) and  $s_l < 10$  **do**
- 6: Calculate:

$$\lambda_{DEM}^* = \max\{\lambda^* : D_U < \beta\}. \quad (12)$$

- 7: Modify  $\Sigma_l$  by

$$\Sigma_l^* = D' \text{Diag}\{\lambda_1^{-1}, \dots, \lambda_{s-1}^{-1}, \lambda_{DEM}^{*-1}, \dots, \lambda_{DEM}^{*-1}\} D. \quad (13)$$

- 8: Update  $z_{il}$  by setting

$$z_{il} = z_{il}^* = \frac{\pi_l f_l^*(x_i)}{\sum_{k \neq l} \pi_k f_k(x_i, \mu_k, \Sigma_k) + \pi_l f_l^*(x_i, \mu_l, \Sigma_l^*)}. \quad (14)$$

The probabilities for the point allocations  $z_{ij}$  become  $z_{ij}^*$ , where  $1 \leq i \leq n$  and  $1 \leq j \leq m$ .

- 9: Update parameters for all the components:  $j \in \{1 : m\}$  by:

$$\begin{aligned} \mu_j &= \sum_{i=1}^n z_{ij}^* x_i / \sum_{i=1}^n z_{ij}^*, \\ \Sigma_j &= \sum_{j=1}^m \sum_{i=1}^n z_{ij}^* (x_i - \mu_j)(x_i - \mu_j)' / n, \\ \pi_j &= \sum_{i=1}^n z_{ij}^* / n. \end{aligned}$$

- 10:  $\beta_l = \beta_l/5$ ;  $s_l = s_l + 1$ .

- 11: **end while**
- 

Judging from the strength of their eigenvalues, It is evident from this table that the three components have been clearly identified, with component 1 being the sphere, component 2 the plane and component 3 the line.

### *Synthetic Dataset 2.*

We further validate these results on a 50 dimensional dataset with three components. Each component contains 500 data points generated from  $\mathcal{N}(0, \Sigma_j)$ ,  $1 \leq j \leq 3$ . The component covariance matrices  $\Sigma_1$ ,  $\Sigma_2$  and  $\Sigma_3$  are of ranks 50, 30 and 20 respectively. Because all the three components are centered at the origin, distance based clustering methods such as K-means will perform poorly. Judging from the mis-allocation rate in Table (??), the LPCA does not work well either. However, the DEM algorithm has identified all the components and has achieved very low error rates.

**Table 2.** Comparison of the errors and the error rates of DEM and LPCA for Synthetic Dataset 2.

Data	Error		Error Rate	
Set	DEM	LPCA	DEM	LPCA
1	0	513.2	0	34.2 %
2	1.2	488.4	0.08 %	32.6 %
3	0.8	396.1	0.053 %	26.4 %

*Synthetic Dataset 3.*

The DEM algorithm is designed to distinguish spurious degeneration from genuine degeneration. In Figure 1, all the plots contain two components: one line and one two-dimensional plane. For the two plots in the upper panel, Plot (A) indicate the initial allocation of data points to components 1 and 2. Clearly, component 1 is degenerate and spurious, because it contains only a small portion of the genuine degenerate component. Plot (B) indicates the result of the DEM algorithm, in which we can see that the one-dimensional and two-dimensional structures have been identified. For the two plots in the lower panel, Plot (C) indicates the initial allocation of the data with a spurious degenerate component around the center of the data. After running the DEM, the genuine degenerate component as well as the two-dimensional plane have been clearly recovered, as shown in plot (D).

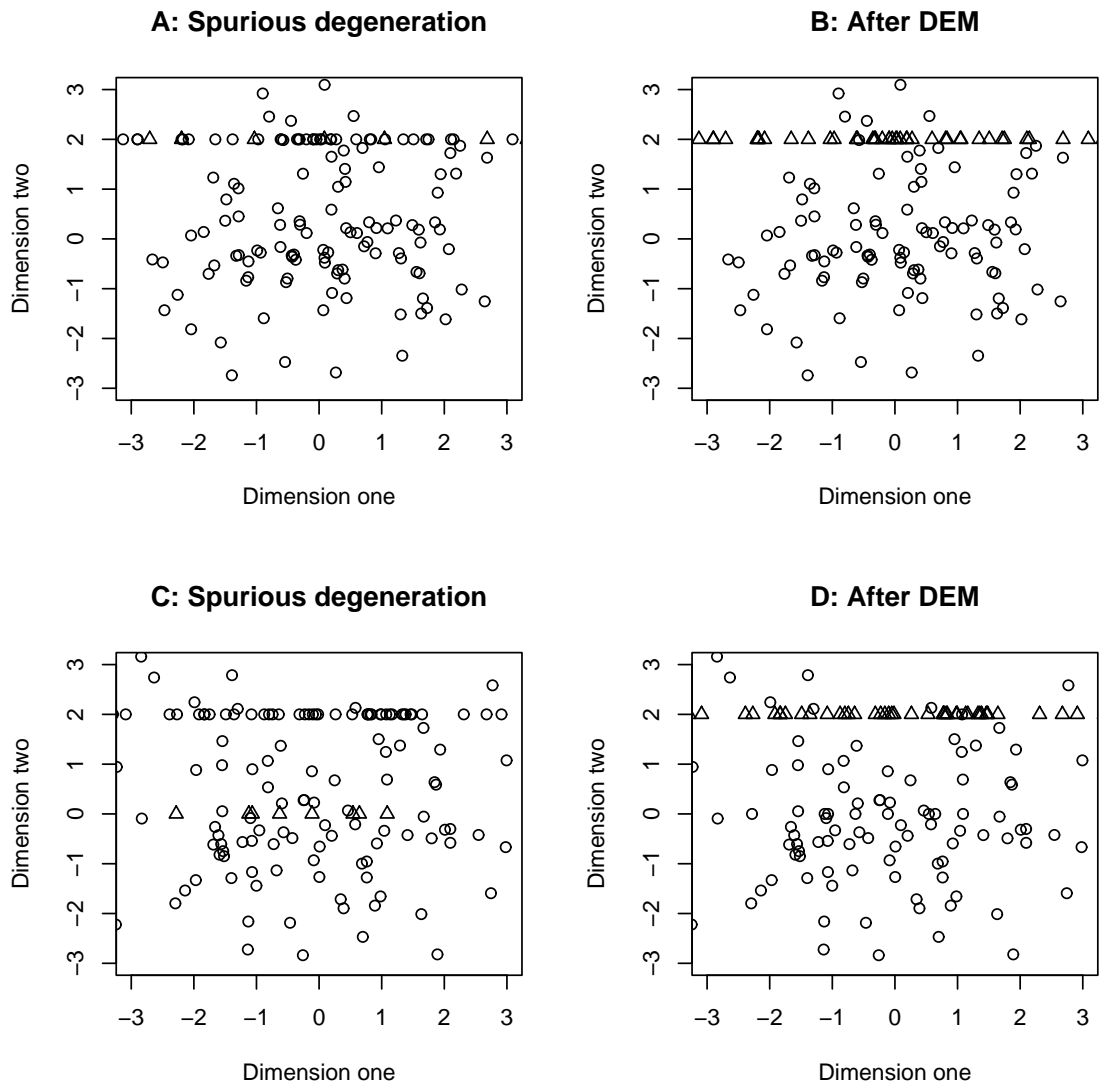
*Iris data.*

The Iris data consist of measurements of the length and width of both sepals and petals of 50 plants for each of the three types of Iris species – Setosa, Versicolor, and Virginica. The objective is to demonstrate the ability of the DEM algorithm for identifying subsets of data that are degenerate. The simulation result of the DEM is compared with that of the LPCA algorithm, with the first two eigen-directions retained.

All but 4 points (69, 71, 73 and 84th data points) are classified accurately by the DEM algorithm. In contrast, the LPCA algorithm misclassifies 58 points. The superior performance of the DEM algorithm on the Iris data is attributed to its ability to detect degeneracy effectively. In Table (??), we include the eigenvalues corresponding to the component covariance matrices respectively. Clearly there is a sharp drop between the third and the fourth eigenvalues in components 1 and 2, while the change is moderate in component 3. This implies that components 1 and 2 are in fact degenerate.

## 4 Conclusions

In high dimensional data analysis, it is often the case that subsets of data lie on different low dimensional subspaces. In this paper, we have proposed



**Fig. 1.** Dataset A and B used to demonstrate the ability of DEM in deviating from spurious degeneracies.  $\triangle$  denotes component 1 and  $\circ$  denotes component 2.

**Table 3.** Eigen-values of Iris data after running DEM

Comp.	1st Eigen	2nd Eigen	3rd Eigen	4th Eigen
1	0.23645569	0.03691873	0.02679643	0.00903326
2	0.48787394	0.07238410	0.05477608	0.00979036
3	0.69525484	0.10655123	0.05229543	0.03426585

the DEM algorithm to address this problem. The DEM enjoys superior performance compared to other methods, and it also has the desirable characteristics of identifying subspaces of different dimensions across clusters, while for other methods, such as the LPCA, the dimension is assumed to be fixed. Although the DEM algorithm is currently only a search technique, we believe that various model selection procedures can be combined with the algorithm to improve its performance in general applications. This is one of the directions we will pursue in the future. Another promising direction is to develop Bayesian procedures for simultaneous dimension reduction and clustering in high dimensional settings.

## References

1. G. Ciuperca, A. Ridolfi, and J. Idier (2003). Penalized maximum likelihood estimator for normal mixtures. *Scandinavian Journal of Statistics* 30, 45-59.
2. T. Hastie and W. Sturtzle (1989). Principal curves. *Journal of the American Statistical Association* 84, 502-516.
3. R. J. Hathaway (1985). A Constrained formulation of maximum-likelihood estimation for normal mixture distributions. *Annals of Statistics*, 13(2): 795-800.
4. Geoffrey E. Hinton and Zoubin Ghahramani (1997). Generative models for discovering sparse distributed representations. *Philosophical Transactions Royal Society B*, 352(1177-1190).
5. J. Kiefer and J. Wolfowitz (1956). Consistency of the maximum likelihood estimates in the presence of infinitely many incidental parameters. *Annals of Mathematical Statistics* 27, 887-906.
6. T. Kohonen (1989). Self-Organization and Associative Memory (3rd ed.). *Springer-Verlag*, Berlin.
7. T. Kohonen (1990). The Self-Organizing Map. *In Proceedings of the IEEE*, Volume 78, pp. 1464-1479.
8. N. Kambhatla and T. K. Leen (1997). Dimension reduction by local principal component analysis. *Neural Computation*, 9(7):1793-1516.
9. C. Archer and T. K. Leen (1999). Optimal dimension reduction and transform coding with mixture principal components. *International Joint Conference on Neural Networks (IJCNN)*, IEEE.
10. X. Lin (2003), Finite Mixture Models for Clustering, Dimension Reduction and Privacy Preserving Data Mining, *Ph.D. Thesis, Purdue University*.