University of Caen Normandy

Internship report

Hierarchical Clustering of a Mixture Model

student : Abdelmounim
el-korchi

Professor : M. Faicel
Chamroukhi
Professor : M. Thien Pham

1er 2021 - May 25

Résumé

[1] this chapter we propose an efficient algorithm for reducing a large mixture of Gaussians into a smaller mixture while still preserv- ing the component structure of the original model ; this is achieved by clustering (grouping) the components. The method minimizes a new, easily computed distance measure between two Gaussian mixtures that can be motivated from a suitable stochastic model and the iterations of the algorithm use only the model parameters, avoiding the need for explicit resampling of datapoints. We demon- strate the method by performing hierarchical clustering of handwritten digits.

I introduction

The Gaussian mixture model (MoG) is a flexible and powerful parametric frame- work for unsupervised data grouping. Mixture models, however, are often involved in other learning processes whose goals extend beyond simple density estimation to hierarchical clustering, grouping of discrete categories or model simplification. In many such situations we need to group the Gaussians components and re-represent each group by a new single Gaussian density. This grouping results in a compact representation of the original mixture of many Gaussians that respects the original component structure in the sense that no original component is split in the reduced representation. We can view the problem of Gaussian component clustering as gen- eral data-point clustering with side information that points belonging to the same original Gaussian component should end up in the same final cluster. Several algo- rithms that perform clustering of data points given such constraints were recently proposed [11, 5, 12]. In this study we extend these approaches to model-based rather than datapoint based settings. Of course, one could always generate data by sampling from the model, enforcing the constraint that any two samples generated by the same mixture component must end up in the same final cluster. We show that if we already have a parametric representation of the constraint via the MoG density, there is no need for an explicit sampling phase to generate representative datapoints and their associated constraints. In other situations we want to collapse a MoG into a mixture of fewer components in order to reduce computation complexity. One example is statistical inference in switching dynamic linear models, where performing exact inference with a MoG prior causes the number of Gaussian components representing the current belief to grow exponentially in time. One common solution to this problem is grouping the Gaussians according to their common history in recent timesteps and collapsing Gaussians grouped together into a single Gaussian [1]. Such a reduction, however, is not based on the parameters of the Gaussians. Other instances in which collapsing MoGs is relevant are variants of particle filtering [10], non-parametric belief propa- gation [7] and fault detection in dynamical systems [3]. A straight-forward solution for these situations is first to produce samples from the original MoG and then to apply the EM algorithm to learn a reduced model ; however this is computationally inefficient and does not preserve the component structure of the original mixture.

II The Clustering Algorithm

We assume that we are given a mixture density f composed of k d-dimensional Gaussian components : k

$$f(y) = \sum_{i=1}^{k} \alpha_i N(y, \mu_i \Sigma_i) = \sum_{i=1}^{k} \alpha_i f_i(y) \tag{1}$$

where :

$$f_i(y) = \frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{Det(\Sigma_i)}} \exp\left(-\frac{1}{2}\, {}^t(y-\mu_i)\Sigma_i^{-1}(y-\mu_i)\right), \quad y \in R^d. \tag{2}$$

$$y = {}^t(y_1, \cdots, y_d) \in R^d \quad \mu_i = {}^t(\mu_i^1, \cdots, \mu_i^d) \in R^d \quad \Sigma_i = (\sigma_{j,j'}^i,\ j,\ j' = 1, \cdots, d) \in M(d,\ d) \tag{3}$$

We want to cluster the components of f into a reduced mixture of m < k components.
we denote the set of all (d-dimensional) Gaussian mixture models with at most m components by MoG(m), one way to formalize the goal of clustering is to say that we wish to find the element g of MoG(m) “closest” to f under some distance measure

$$\widehat{g} = argmin_g KL(f\|g) = argmax_g \int flog(g) \tag{4}$$

where KL is kullback-Leibler divergence . This criterion leads to an intractable optimization problem; there is not even a closed-form expres-sion for the KL-divergence between two MoGs let alone an analytic minimizer of its second argument. Furthermore, minimizing a KL-based criterion does not pre- serving the original component structure of f. Instead, we introduce the following new distance measure between but to find g in MoG(m) which verified (4). This leads to an intractable optimization problem, Instead, we introduce the following new distance measure between

$$f(y) = \sum_{i=1}^{k} \alpha_i f_i(y) \quad and \quad g(y) = \sum_{i=1}^{k} \beta_i g_i(y) \tag{5}$$

$$d(f,g) = \sum_{i=1}^{k} \alpha_i min_{j=1}^{m} KL(f_i\|g_j) \tag{6}$$

this distance can be analytically computed. In particular, for example between two Gaussian distributions $N(y, \mu_1\Sigma_1) \quad and \quad N(y, \mu_2\Sigma_2)$ which is given by :

$$d(N(y,\mu_1\Sigma_1) \quad , \quad N(y,\mu_2\Sigma_2)) = \frac{1}{2}(log(\frac{|\sigma_2|}{|\sigma_1|} + tr(\Sigma_2^{-1}\Sigma_1) + (\mu_2-\mu_1)\Sigma_2^{-1}(\mu_2-\mu_1) - d) \tag{7}$$

Under this distance, the optimal reduced MOG representation $\widehat{g}$ is the solution to the minimization of (6) over $\widehat{g} = argmin_g d(f,g)$
we prove that the optimal density $\widehat{g}$ is a MoG obtained from grouping the components of f into clusters and collapsing all Gaussians within a cluster into a single Gaussian, to obtain a locally optimal solution.
Denote the set of all the m^k mappings from $\{1,...,k\} \quad to \quad \{1,...,m\}$ by S. For each $\pi \in S$ et g $\in$MoG(m) definie :

$$d(f,g,\pi) = \sum_{i=1}^{k} \alpha_i KL(f_i\|g_{\pi(i)}) \tag{8}$$

for $g \in MoG(m)$, we associte a matching function $\pi^g \in S$:

$$\pi^g(i) = argmin_{j=1}^{m}(KL(f_i\|g_j)) \quad j = 1,....,k \tag{9}$$

One can easily verify that :

$$d(f,g) = d(f,g,\pi^g) = min_{\pi\in S} d(f,g,\pi) \tag{10}$$

$$\pi^g(i) = argmin_g min_{\pi\in S}(d(f,g,\pi)) \tag{11}$$

Unfortunately, we can not solve (11) analytically,Instead we can use alternating minimization to obtain a local minimum.
we define the following functions :

$$f_j^{\pi} = \frac{\sum_{i\in\pi^{-1}(j)} \alpha_i f_i}{\sum_{i\in\pi^{-1}(j)} \alpha_i} \tag{12}$$

$$\mu'_j = 1/\beta_j \sum_{i\in\pi^{-1}(j)} \alpha_i\mu_i \quad \Sigma'_j = 1/\beta_j \sum_{i\in\pi^{-1}(j)} \alpha_i(\Sigma_i + \|\mu_i - \mu'_j\|^2) \tag{13}$$

where

$$\beta_j = \sum_{i \in \pi^{-1}(j)} \alpha_i \tag{14}$$

let

$$g_j^{\pi} = N(\mu_j', \Sigma_j') = argmin_g(KL(f_i^{\pi} \| g)) = argmin_g(d(f_i^{\pi}, g)) \tag{15}$$

finally the function we are looking for is :

$$g^{\pi} = \sum_{j=1}^{m} \beta_j g_j^{\pi} \tag{16}$$

$$\begin{array}{lcll} \pi^g & = & \arg\min_{\pi} d(f, g, \pi) & (\textbf{REGROUP}) \\ g^{\pi} & = & \arg\min_{g} d(f, g, \pi) & (\textbf{REFIT}) \end{array}$$

III Hierarchical Clustering algorithm of a Mixture Model

...

Algorithm 1: Hierarchical Clustering algorithm of a Mixture Model

Input : k (the number of clusters of F function)
m (the number of clusters of G function)
X (data)

initialization : g^{π}
π^g

while $\sum_{i=1}^{k} \alpha_i KL(f_i \| g_{\pi(i)}) \quad no \quad change$ **do**

Update : REGROUP

$$\pi^g(i) = argmin_{j=1}^{m}(KL(f_i \| g_j)) \quad i = 1, ..., k$$

Update :REFIT

$$\beta_j = \sum_{i \in \pi^{-1}(j)} \alpha_i \quad j = 1, ..., m$$
$$\mu_j' = 1/\beta_j \sum_{i \in \pi^{-1}(j)} \alpha_i \mu_i \quad j = 1, ..., m$$
$$\Sigma_j' = 1/\beta_j \sum_{i \in \pi^{-1}(j)} \alpha_i (\Sigma_i + \|\mu_i - \mu_j'\|^2) \quad j = 1, ..., m$$
$$g_j^{\pi}(y) = \frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{Det(\Sigma_j')}} \exp\left(-\frac{1}{2}\, {}^t(y - \mu_j') \Sigma_j'^{-1} (y - \mu_j')\right), \quad y \in R^d, \quad j = 1, ..., m$$
$$g^{\pi} = \sum_{j=1}^{m} \beta_j g_j^{\pi}$$

end

Result: *a set of m cluster*

IV performing hierarchical clustering of data from GMM and handwritten digits

In this section, in the first hand we will test the performance of the HCMM algorithm on a dataset that follows the Gaussian mixture model law , we see that the algorithm merge together the Gaussian clusters in a hierarchical way which is the case, in fact the algorithm works in the following way it merge all the Gaussians clusters compared to the distance KL, the precision of the algorithm on a data set follows a Gaussine law and better than on any data set.

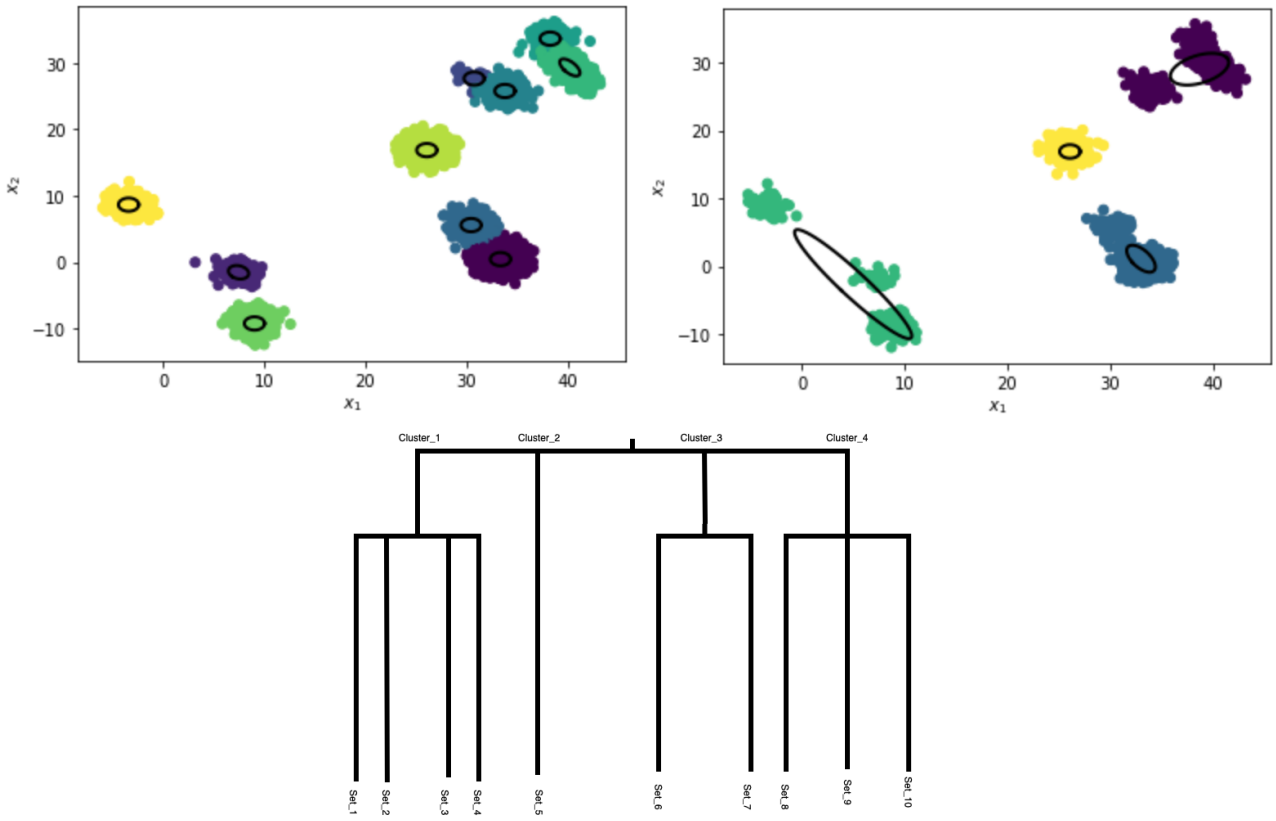

FIGURE 1 – Dendrogram of HCMM for K = 10 m = 4

IV.I experimental results on data from Gaussian mixture model.

We test the algorithm on different datasets that all follows the Gaussian mixture model law here are the results

TABLE 1 – HCMM on GMM-data

Data	Gaussian mixture models	Hierarchical Clustering of a Mixture Model
The number of mixture components :	K = 4	k = 4 and M = 2
The number of mixture components :	K = 10	K = 10 and M = 4
The number of mixture components :	K = 150	K = 150 and M = 3
The number of mixture components :	K = 40	K = 40 and M = 10

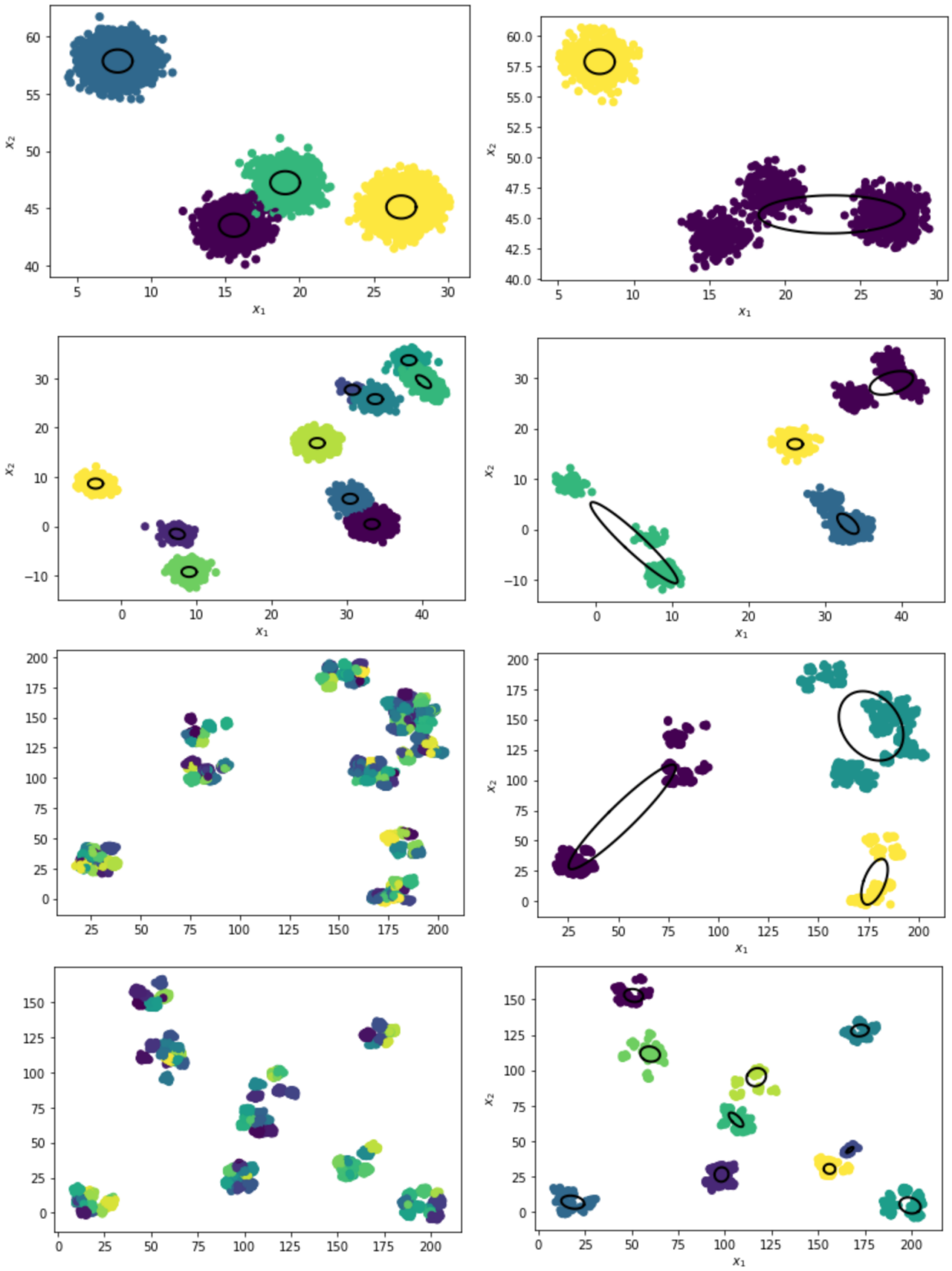

FIGURE 2 – HCMM for GMM data

IV.II experimental results on data handwritten digits.

in the second paragraph we evaluate the performance of our semi-supervised clustering al- gorithm and compare it to EM clustering approach that does not respect the original component structure. We have applied both methods to clustering handwritten digits images.
In each case, a set of objects is organized in predefined categories. For each category c we learn from a labeled training set a Gaussian distribution f(x|c). A prior distribution over the categories p(c) can be also extracted from the labeled training set. The goal is to cluster the objects into a small number of clusters (fewer than the number of class labels). The EM algorithm approach is to apply an unsupervised clustering to entire collection of original objects, ignoring their class labels. Alternatively we can utilize the given categorization as side-information in order to obtain an improved reduced clustering which also respects the original labels, thus inducing a hierarchical structure. the following image represents an extract of the results of HCMM we see that it separates the figures under two sets :

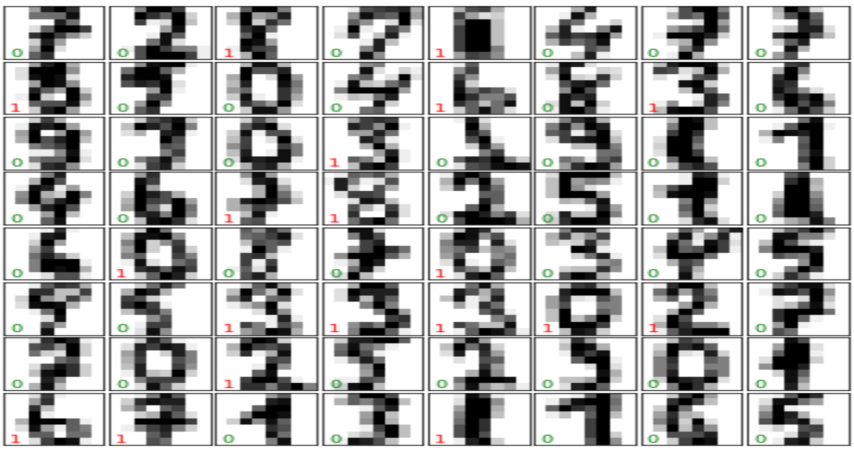

Figure 3 – results of HCMM for 64 first images

Our first experiment used a database of handwritten digits. Each example is repre-sented by a 8×8 grayscale pixel. image we use 1437 images for training and 360 images for testing . In the next step we want to divide the digits into two natural clusters, while taking into account their original 10-way structure. We applied our semi-supervised algorithm to reduce the mix- ture of 10 Gaussians into a mixture of two Gaussians. The minimal distance (2) is obtained when the ten digits are divided into the two groups [1 2 3 5 7 8 9] and [0, 4 ,6].

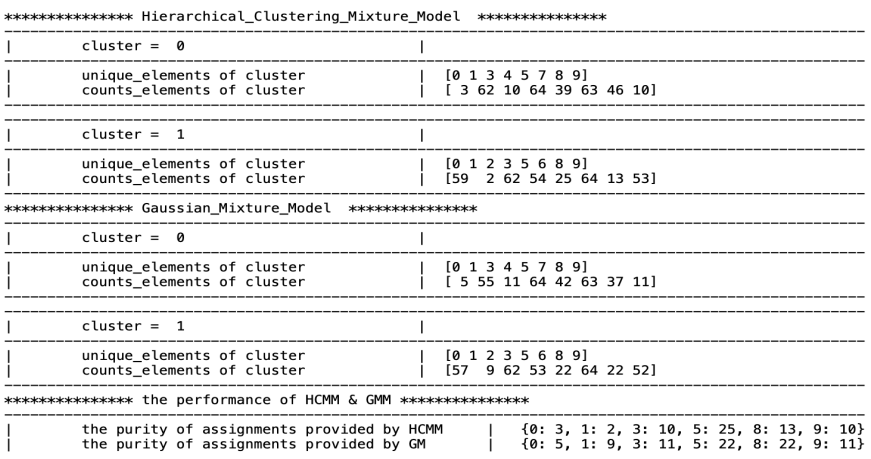

FIGURE 4 – results of HCMM for 64 first images

Table 1 : Clustering results showing the purity of a 2-cluster reduced model learned from a training set of handwritten digits in 10 original classes. For each true label, the percentage of cases (from an unseen test set) falling into each of the two re- duced classes is shown. The top two lines show the purity of assignments provided by our clustering algorithm ; the bottom two lines show assignments from a flat unsupervised fitting of a two component mixture. Table 1 (top) presents, for each digit, the percentage of images that were affiliated with each of the two clusters. Alternatively we can apply a standard EM algorithm to learn by maximum likelihood a flat mixture of 2 Gaussians directly from the 7000 training examples, without utilizing their class labels. Table 1 (bottom) shows the results of such an unsupervised clustering, evaluated on the same test set. Although the likelihood of the unsupervised mixture model was significantly better than the semi-supervised model, both on train and test data-sets it is obvious that the purity of the clusters it learns is much worse since it is not preserving the hierarchical class structure. Comparing the top and bottom of Table 1, we can see that using the side information we obtain a clustering of the digit data-base which is much more correlated with categorization of the set into ten digits than the unsupervised procedure.