Probabilistic Hierarchical Clustering with Labeled and Unlabeled Data

J. Larsen, A. Szymkowiak, L.K. Hansen

Informatics and Mathematical Modeling, Technical University of Denmark
Richard Petersens Plads, Build. 321, DK-2800 Kongens Lyngby, Denmark
Web: http://eivind.imm.dtu.dk, Emails: jl,asz,lkh@imm.dtu.dk

***Abstract**. This paper presents hierarchical probabilistic clustering methods for unsupervised and supervised learning in datamining applications, where supervised learning is performed using both labeled and unlabeled examples. The probabilistic clustering is based on the previously suggested Generalizable Gaussian Mixture model and is extended using a modified Expectation Maximization procedure for learning with both unlabeled and labeled examples. The proposed hierarchical scheme is agglomerative and based on probabilistic similarity measures. Here, we compare a $\mathcal{L}_2$ dissimilarity measure, error confusion similarity, and accumulated posterior cluster probability measure. The unsupervised and supervised schemes are successfully tested on artificially data and for e-mails segmentation.*

1 Introduction

Hierarchical methods for unsupervised and supervised datamining provide multilevel description of data, which is relevant for many applications related to information extraction, retrieval navigation and organization of information, see e.g., [4, 7]. Many different approaches to hierarchical analysis from divisive to agglomerative clustering schemes have been suggested, and recent developments include [3, 6, 16, 20, 24]. In this paper we focus on agglomerative probabilistic clustering from Gaussian density mixtures based on earlier work [14, 15, 19] but extended by suggesting and comparing various similarity measures in connection with cluster merging. An advantage of using the probabilistic clustering scheme is automatic detection of the final hierarchy level for new data not used for training. In order to provide a meaningful description of the clusters we suggest two interpretation techniques: listing of prototypical data examples from the cluster, and listing of typical features associated with the cluster.

The generalizable Gaussian mixture model (GGM) [8] and the soft generalizable Gaussian mixture model (SGGM) [19] are basic model for supervised and unsupervised learning. We extend this framework to supervised learning from combined sets of labeled and unlabeled data [9, 17, 18] and present a modified version of the approach in [17] called the unsupervised/supervised generalizable Gaussian mixture model (USGGM). Supervised learning from combined sets is relevant in many practical applications due to the fact that labeled examples are hard and/or expensive to obtain, for instance in document categorization or medical applications. The models estimate parameters of the Gaussian clusters with a modified EM procedure from two disjoint data sets to prevent notorious infinite overfit problems and ensuring good generalization ability. The optimum number of clusters in the mixture is determined automatically by minimizing an estimate of the generalization error [8].

This paper focuses on applications to textmining [8, 11, 12, 13, 18, 22, 21, 23] with the objective of categorizing text according to topic, spotting new topics or providing short, easy and understandable interpretation of larger text blocks – in a broader sense to create intelligent search engines and to provide understanding of documents or content of webpages like Yahoo's ontologies.

In Section 2, various GGM models for supervised and unsupervised learning are discussed, in particular we introduce the USGGM algorithm. The hierarchical clustering scheme is discussed in section 3 and introduces three similarity measures for cluster merging. Finally, Section 4 provide numerical experiments for segmentation of e-mails.

2 The Generalizable Gaussian Mixture Model

The first step in our approach for probabilistic clustering is a flexible and universal extension of Gaussian mixture density model, the generalizable Gaussian mixture model [8, 14, 15, 19] with the aim of supervised learning from unlabeled and labeled data. Define $\boldsymbol{x}$ as the d-dimensional input feature vector and the associated output, $y \in \{1, 2, \cdots, C\}$, of class labels, assuming C mutually exclusive classes. The joint input/output density is modeled as the Gaussian

This research is supported by the Danish Research Councils through Distributed Multimedia Technologies and Applications program within Center for Multimedia and the Signal and Image Processing for Telemedicine (SITE) program.

mixture in [17][1]

$$p(y,\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} P(y|k)p(\boldsymbol{x}|k)P(k) \tag{1}$$

$$p(\boldsymbol{x}|k) = \tag{2}$$
$$\frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_k|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right)$$

where K is the number of components, $p(\boldsymbol{x}|k)$ are the component Gaussians mixed with the non-negative priors $P(k)$, $\sum_{k=1}^{K} P(k) = 1$ and the class-cluster posteriors $P(y|k)$, $\sum_{y=1}^{C} P(y|k) = 1$. The k'th Gaussian component is described by the mean vector $\boldsymbol{\mu}_k$ and the covariance matrix $\boldsymbol{\Sigma}_k$. $\boldsymbol{\theta}$ is the vector of all model parameters, i.e., $\boldsymbol{\theta} \equiv \{P(y|k), \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, P(k) : \forall k, y\}$. Since the Gaussian mixture is an universal approximator, the model Eq. (1) is rather flexible. One restriction, however, is that the joint input/output for each components is assumed to factorize, i.e., $p(y,\boldsymbol{x}|k) = P(y|k)p(\boldsymbol{x}|k)$.

The input density associated with Eq. (1) is given by

$$p(\boldsymbol{x}|\boldsymbol{\theta}_u) = \sum_{y=1}^{C} p(y,\boldsymbol{x}) = \sum_{k=1}^{K} p(\boldsymbol{x}|k)P(k), \tag{3}$$

where $\boldsymbol{\theta}_u \equiv \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, P(k) : \forall k, y\}$. Assuming a 0/1 loss function the optimal Bayes classification rule is $\widehat{y} = \max_y P(y|\boldsymbol{x})$ where[2]

$$P(y|\boldsymbol{x}) = \frac{p(y,\boldsymbol{x})}{p(\boldsymbol{x})} = \sum_{k=1}^{K} P(y|k)P(k|\boldsymbol{x}) \tag{4}$$

with $P(k|\boldsymbol{x}) = p(\boldsymbol{x}|k)P(k)/p(\boldsymbol{x})$.

Define the data set of unlabeled examples $\mathcal{D}_u = \{\boldsymbol{x}_n; n = 1, 2, \cdots, N_u\}$ and a set of labeled examples $\mathcal{D}_l = \{\boldsymbol{x}_n, y_n; n = 1, 2, \cdots, N_l\}$. The objective is to estimate $\boldsymbol{\theta}$ from the combined set $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$ with $N = N_l + N_u$ examples ensuring high generalizability. If no labeled data are available we can merely perform unsupervised learning of $\boldsymbol{\theta}_u$, however, if a number of labeled data are available, estimation from both data sets is possible as $p(y|\boldsymbol{x})$ and $p(\boldsymbol{x})$ share the model parameters $\boldsymbol{\theta}_u$ [9]. The negative log-likelihood for the data sets, which are assumed to consist of independent examples, is given by

$$\begin{aligned} L = & -\log p(\mathcal{D}|\boldsymbol{\theta}) \qquad (5)\\ & - \sum_{n\in\mathcal{D}_l} \log \sum_{k=1}^{K} P(y_n|k)p(\boldsymbol{x}_n|k)P(k) \\ & -\lambda \sum_{n\in\mathcal{D}_u} \log \sum_{k=1}^{K} p(\boldsymbol{x}_n|k)P(k) \end{aligned}$$

where $0 \leq \lambda \leq 1$ is a discount factor. If the model is unbiased (realizable), the estimation $\boldsymbol{\theta}_u$ from either labeled or unlabeled data will result in identical optimal setting and thus $\lambda = 1$ is optimal. On the other hand, in the typical case of a biased mode, it is advantageous to discount the influence of unlabeled data [9, 18].

```
Initialization
 1. Choose values for $K$ and $0 \leq \lambda \leq 1$.
 2. Let $\boldsymbol{i}$ be $K$ different randomly selected indices
    from $\{1, 2, \cdots, N\}$, and set $\boldsymbol{\mu}_k = \boldsymbol{x}_{\boldsymbol{i}_k}$.
 3. Let $\boldsymbol{\Sigma}_0 = N^{-1}\sum_{n\in\mathcal{D}}(\boldsymbol{x}_n - \boldsymbol{\mu}_0)(\boldsymbol{x}_n - \boldsymbol{\mu}_0)^\top$,
    where $\boldsymbol{\mu}_0 = N^{-1}\sum_{n\in\mathcal{D}}\boldsymbol{x}_n$, and set $\forall k : \boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_0$.
 4. Set $\forall k : P(k) = 1/K$.
 5. Compute class prior probabilities: $P(y) = N_l^{-1}\sum_{n\in\mathcal{D}_l}\delta(y_n - y)$, where $\delta(z) = 1$ if
    $z = 0$, and zero otherwise. Set $\forall k : P(y|k) = P(y)$.
 6. Select a split ratio $0 < \gamma < 1$. Split the unlabeled data set into disjoint sets as $\mathcal{D}_u = \mathcal{D}_{u,1} \cup \mathcal{D}_{u,2}$,
    with $|\mathcal{D}_{u,1}| = [\gamma N_u]$ and $|\mathcal{D}_{u,2}| = N_u - |\mathcal{D}_{u,1}|$. Do similar splitting for the
    labeled data set $\mathcal{D}_l = \mathcal{D}_{l,1} \cup \mathcal{D}_{l,2}$.
Repeat until convergence
 1. Compute posterior component probabilities:
    $p(k|\boldsymbol{x}_n) = p(\boldsymbol{x}_n|k)P(k)/\sum_k p(\boldsymbol{x}_n|k)P(k)$,
    for all $n \in \mathcal{D}_u$, and for all $n \in \mathcal{D}_l$,
    $p(k|y_n, \boldsymbol{x}_n) = \frac{P(y_n|k)p(\boldsymbol{x}_n|k)P(k)}{\sum_k P(y_n|k)p(\boldsymbol{x}_n|k)P(k)}$.
 2. For all $k$ update means
    $\boldsymbol{\mu}_k = \frac{\sum_{n\in\mathcal{D}_{l,1}} \boldsymbol{x}_n P(k|y_n,\boldsymbol{x}_n) + \lambda \sum_{n\in\mathcal{D}_{u,1}} \boldsymbol{x}_n P(k|\boldsymbol{x}_n)}{\sum_{n\in\mathcal{D}_{l,1}} P(k|y_n,\boldsymbol{x}_n) + \lambda \sum_{n\in\mathcal{D}_{u,1}} P(k|\boldsymbol{x}_n)}$
 3. For all $k$ update covariance matrices
    $\boldsymbol{\Sigma}_k = \frac{\sum_{n\in\mathcal{D}_{l,2}} \boldsymbol{S}_{kn} P(k|y_n,\boldsymbol{x}_n) + \lambda \sum_{n\in\mathcal{D}_{u,2}} \boldsymbol{S}_{kn} P(k|\boldsymbol{x}_n)}{\sum_{n\in\mathcal{D}_{l,2}} P(k|y_n,\boldsymbol{x}_n) + \lambda \sum_{n\in\mathcal{D}_{u,2}} P(k|\boldsymbol{x}_n)}$
    where $\boldsymbol{S}_{kn} = (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top$. Perform
    a regularization of $\boldsymbol{\Sigma}_k$ (see text).
 4. For all $k$ update cluster priors
    $P(k) = \frac{\sum_{n\in\mathcal{D}_l} P(k|y_n,\boldsymbol{x}_n) + \lambda \sum_{n\in\mathcal{D}_u} P(k|\boldsymbol{x}_n)}{N_l + \lambda N_u}$
 5. For all $k$ update class cluster posteriors
    $P(y|k) = \frac{\sum_{n\in\mathcal{D}_l} \delta(y - y_n) P(k|y_n,\boldsymbol{x}_n)}{\sum_{n\in\mathcal{D}_l} P(k|y_n,\boldsymbol{x}_n)}$
```

Figure 1: The USGGM algorithm.

[1] In [17] referred to as the generalized mixture model.

[2] The dependence on $\boldsymbol{\theta}$ is omitted.

2.1 The USGGM Algorithm

The model parameters are estimated with an iterative modified EM algorithm [8], where means and covariance matrices are estimated from independent data sets, and $P(y|k)$, $P(k)$ from the combined set. This approach prevents overfitting problems with the standard approach [2]. It is designated the generalizable Gaussian mixture model with labeled and unlabeled data (USGGM) and may be viewed as an extension of the EM-I algorithm suggested in [17]. The GGM can be implemented using either hard or soft assignments of data to components in each EM iteration step. In the hard GGM approach each data example is assigned to a cluster by selecting highest $P(k|\boldsymbol{x})$. Means and covariances are estimated by classical empirical estimates from data assigned to each component. In the soft version (SGGM) [19] means and covariances are estimated as weighted quantities, e.g., $\boldsymbol{\mu}_k = \sum_n p(k|\boldsymbol{x}_n)\boldsymbol{x}_n / \sum_n p(k|\boldsymbol{x}_n)$. GGM provides a biased estimate, which gives better results for small data sets [19], however, in general the soft version is preferred. The USGGM algorithm is summarized in Fig. 1 and is based on the soft approach. The main iteration loop is aborted[3] when no change in example cluster assignment is noticed. Labeled examples are assigned to clusters $k_n = \arg\max_k P(k|y_n, \boldsymbol{x}_n)$, $n \in \mathcal{D}_l$, and unlabeled to $k_n = \arg\max_k P(k|\boldsymbol{x}_n)$, $n \in \mathcal{D}_u$. In contrast to EM algorithms there is no guarantee that each iteration leads to improved likelihood, however, practical experience indicates that the updating scheme is sufficiently robust. Potential poor conditioned covariance matrices for clusters where few examples are assigned is avoided by regularizing towards the overall input covariance matrix $\boldsymbol{\Sigma}_0$ (defined in Fig. 1) as $\boldsymbol{\Sigma}_k \leftarrow \boldsymbol{\Sigma}_k + \alpha\boldsymbol{\Sigma}_0$. α is selected as the smallest positive number, which ensures that the resulting condition number is smaller than $1/(d \cdot \epsilon)$, where ϵ is the floating point machine precision.

Essential algorithm parameters are the number of components K and the weighting factor λ. In principle, these parameters should be chosen as to maximize generalization performance. One method is to pick K and λ so that the cross-validation estimate of the classification error is minimized. A less computational cumbersome method is to select K based on the AIC estimate of the generalization error [1, 8, 19], which is the negative log-likelihood plus the number of parameters in the model, $K(d(d+1)/2+C)-1$. The only remaining algorithm parameter to determine is the split ratio γ, which in principle also should be selected to achieve high generalization performance. Practical simulations show that $\gamma = 0.5$ is a proper choice in most cases.

2.2 Unsupervised GGM Model

If only input data are available one has to perform unsupervised learning. In this case the object of modeling is the input density Eq. 3, which can be trained using the SGGM algorithm[4] [19].

2.3 Supervised GGM Model

Clearly USGGM can be used in the case of no unlabeled examples. Another choice is to use separate GGM models for the class conditional input densities, i.e., $p(\boldsymbol{x}|y) = \sum_{k=1}^{K_y} p(\boldsymbol{x}|y,k)P(k|y)$ with $p(\boldsymbol{x}|y,k)$ defined by Eq. (2) and where K_y is the number of components. Using Bayes optimal rule and assuming a 1/0 loss function, classification is done by maximizing $p(y|\boldsymbol{x}) = p(\boldsymbol{x}|y)P(y)/\sum_{y=1}^{C} p(\boldsymbol{x}|y)P(y)$. The approach is also referred to as mixture discriminant analysis [10] and seems more flexible than the model in Eq. (1). However, it does not use discriminative training, i.e., minimizing the classification error or negative log-likelihood $L = -\sum_n \log p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are model parameters. Modeling instead $p(\boldsymbol{x}|y)$ will provide reasonable estimates of $p(y|\boldsymbol{x})$ in the entire input space, whereas discriminative learning will use the data to obtain relatively better estimates of $p(y|\boldsymbol{x})$ close to the decision boundaries. The model in Eq. (1) describes the joint input-class probability $p(y,\boldsymbol{x}) = p(y|\boldsymbol{x})p(\boldsymbol{x})$ and may be interpreted as a partial discriminative estimation procedure.

3 Hierarchical Clustering

In the case of unsupervised learning, i.e., learning $p(\boldsymbol{x})$, hierarchical clustering concerns identifying a hierarchical structure of clusters in the feature space $\boldsymbol{x}$. In the suggested agglomerative clustering scheme we start by K clusters at level $j = 1$ as given by the optimized GGM model of $p(\boldsymbol{x})$. At each higher level in the hierarchy two clusters are merged based on a similarity measure between pairs of clusters. The procedure is repeated until we reach one cluster at the top level. That is, at level $j = 1$ there are K clusters, and one cluster at the final level, $j = K$.

For supervised learning one can either identify a hierarchical structure common for all classes, i.e., working from the associated input density $p(\boldsymbol{x})$, or identifying individual hierarchies for each class by working from the class conditional input densities $p(\boldsymbol{x}|y)$. For the model in Eq. (1) $p(\boldsymbol{x})$ is given by Eq. (3) and

$$p(\boldsymbol{x}|y) = \frac{p(y,\boldsymbol{x})}{P(y)} = \sum_{k=1}^{K} p(\boldsymbol{x}|k)P(k|y) \qquad (6)$$

[3] Convergence criteria based on changes in the negative log-likelihood can also be formulated.

[4] The SGGM is similar to USGGM in Fig. 1 and is essentially obtained by setting $\lambda = 1$, neglecting steps 5 of the initialization and main iteration loop, and further neglecting sums over labeled data.

where $P(k|y) = P(y|k)P(k)/\sum_k P(y|k)P(k)$. Let $p_j(\boldsymbol{x}|y,k)$ be the density for the k'th cluster at level j, and $P_j(k|y)$ the mixing proportion, which in the general case both may depend on y. Further, the (class conditional) density model at level j is $p(\boldsymbol{x}|y) = \sum_{k=1}^{K-j+1} P_j(k|y)p_j(\boldsymbol{x}|y,k)$. If clusters ℓ and m at level j are merged into i at level $j+1$ then

$$p_{j+1}(\boldsymbol{x}|y,i) = \qquad (7)$$
$$\frac{p_j(\boldsymbol{x}|y,\ell)P_j(\ell|y) + p_j(\boldsymbol{x}|y,m)P_j(m|y)}{P_j(\ell|y) + P_j(m|y)},$$
$$P_{j+1}(i|y) = P_j(\ell|y) + P_j(m|y) \qquad (8)$$

3.1 Level Assignment

A unique feature of probabilistic clustering is the ability to provide optimal cluster and level assignment for new data examples, which have not been used for training. $\boldsymbol{x}$ is assigned to cluster k at level j if

$$P_j(k|y,\boldsymbol{x}) = \frac{p_j(\boldsymbol{x}|y,k)P(k|y)}{p(\boldsymbol{x}|y)} > \rho \qquad (9)$$

where the threshold ρ typically is set to 0.9. The procedure ensures that the example is assigned to a wrong cluster with probability 0.1.

3.2 Cluster Interpretation

Interpretation of clusters is done by generating likely examples from the cluster [14, 19] and displaying prototype examples and/or typical features. For the first level in the hierarchy in which distributions are Gaussian, prototype examples are identified as those who has highest density values. For clusters at higher levels in the hierarchy, prototype samples are drawn from each Gaussian cluster with proportions specified by $P(k)$ or $P(k|y)$. Typical features are in the first level found by drawing ancillary examples from a super-eliptical region around the mean value, i.e., $(\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k) <$ *const.*, and then listing associated typical features, e.g., keywords as demonstrated in Sec. 4. At higher levels we proceed as described above.

3.3 Similarity measures

Many different similarity measures may be applied in the framework of hierarchical clustering. The natural distance measure between the cluster densities is the Kullback-Leibler (KL) divergence [2], since it reflects dissimilarity between the densities in the probabilistic space. The drawback is that KL only obtains an analytic expression for the first level in the hierarchy, while distances for the subsequent levels have to be approximated [14, 15]. Consequently, we consider three different measures, which express similarity in probability space for models of $p(\boldsymbol{x})$ or $p(\boldsymbol{x}|y)$ (cf. Sec. 3) and can be computed exactly at all levels in the hierarchy[5]. Fig. 2 illustrates the hierarchical clustering for Gaussian distributed toy data.

3.3.1 $\mathcal{L}_2$ Dissimilarity Measure

The $\mathcal{L}_2$ distance for the densities [25] is defined

$$D(\ell,m) = \int (p_j(\boldsymbol{x}|\ell) - p_j(\boldsymbol{x}|m))^2\, dx \qquad (10)$$

where ℓ and m index two different clusters. Due to Minkowksi's inequality, $D(\ell,m)$ is a distance measure, which also will be referred to as dissimilarity. Let $\mathcal{I} = \{1,2,\cdots,K\}$ be the set of cluster indices and define disjoint subsets $\mathcal{I}_\alpha \cap \mathcal{I}_\beta = \emptyset$, $\mathcal{I}_\alpha \subset \mathcal{I}$ and $\mathcal{I}_\beta \subset \mathcal{I}$, where $\mathcal{I}_\alpha$, $\mathcal{I}_\beta$ contain the indices of clusters, which constitute clusters ℓ and m at level j, respectively. The density of cluster ℓ is given by: $p_j(\boldsymbol{x}|\ell) = \sum_{i\in\mathcal{I}_\alpha} \alpha_i p(\boldsymbol{x}|i)$, $\alpha_i = P(i)/\sum_{i\in\mathcal{I}_\alpha} P(i)$ if $i \in \mathcal{I}_\alpha$, and zero otherwise. $p_j(\boldsymbol{x}|m) = \sum_{i\in\mathcal{I}_\beta} \beta_i p(\boldsymbol{x}|i)$, where β_i obtains a similar definition. According to [25], the Gaussian integral is given by $\int p(\boldsymbol{x}|a)p(\boldsymbol{x}|b)\, dx = \mathcal{N}(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_a + \boldsymbol{\Sigma}_b)$, where $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma}) = (2\pi)^{-d/2} \cdot |\boldsymbol{\Sigma}|^{1/2} \cdot \exp(-\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}/2)$. Define the vectors $\boldsymbol{\alpha} = \{\alpha_i\}$, $\boldsymbol{\beta} = \{\beta_i\}$ of dimension K and the $K \times K$ symmetric matrix $\boldsymbol{G} = \{G_{ab}\}$ with $G_{ab} = \mathcal{N}(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_a + \boldsymbol{\Sigma}_b)$, then the distance can be then written as $D(\ell,m) = (\boldsymbol{\alpha} - \boldsymbol{\beta})^\top \boldsymbol{G}(\boldsymbol{\alpha} - \boldsymbol{\beta})$. It turns out (see Fig. 2) that it is important to include the prior of the component in the dissimilarity measure. The modified $\mathcal{L}_2$ is then given by $\widetilde{D}(\ell,m) = \int (p_j(\boldsymbol{x}|\ell)P_j(\ell) - p_j(\boldsymbol{x}|m)P_j(m))^2\, dx$, which easily can be computed using a modified matrix $\widetilde{G}_{ab} = P(a)P(b)G_{ab}$.

3.4 Cluster Confusion Similarity Measure

Another natural principle is based on merging clusters, which have the highest confusion. Thus, when merging two clusters, the similarity is the probability of misassignment (PMA) when drawing examples from the two clusters seperately. Let $\boldsymbol{x}$ be an example from cluster $\mathcal{C}_k$ denoted by $\boldsymbol{x} \in \mathcal{C}_k$ and let $m = \arg\max_j P(j|\boldsymbol{x})$ be the model estimate of the cluster, then the PMA for all $\ell \neq m$ is given by:

$$E(\ell,m) = P(\ell \neq m) = \qquad (11)$$
$$\int_{\mathcal{R}_m} p(\boldsymbol{x}|\ell)P(\ell)d\boldsymbol{x} + \int_{\mathcal{R}_\ell} p(\boldsymbol{x}|m)P(m)d\boldsymbol{x}$$

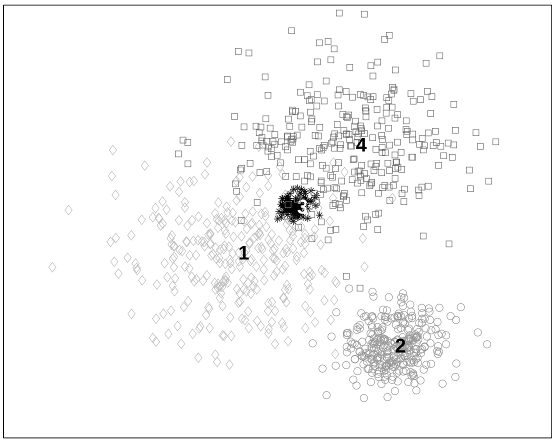
where $\mathcal{R}_m = \{\boldsymbol{x} : m = \arg\max_j P(j|\boldsymbol{x})\}$ and likewise for $\mathcal{R}_\ell$. In general, $E(\ell,m)$ can not be computed analytically, but can be approximated arbitrarily accurately by using an ancillary set of data samples drawn from the estimated model. That is, randomly select a cluster i with probability $P(i)$, draw a sample from $p(\boldsymbol{x}|i)$ and compute the estimated cluster $j = \arg\max_k P(k|\boldsymbol{x})$. Then estimate $P(\ell \neq m)$

[5] In the following sections we omit the possible dependence on y for notational convenience.

as the fraction of samples where $(i = \ell \wedge j = m)$ or $(j = \ell \wedge i = m)$.

3.5 Sample Dependent Similarity Measure

Instead of constructing a fixed hierarchy for visualization and interpretation of new data a sample dependent hierarchy can be obtained by merging a number of clusters relevant for a new data sample $\boldsymbol{x}$. The idea is based on level assignment described in Sec. 3.1. Let $P(k|\boldsymbol{x})$, $k = 1, 2, \cdots, K$, be the computed posteriors ranked in descending order and compute the accumulated posterior $A(\ell) = \sum_{k=1}^{\ell} P(k|\boldsymbol{x})$. The sample dependent cluster is then formed by merging the fundamental components $k = 1, 2, \cdots, m$ where $m = \min_\ell A(\ell) > \rho$, with e.g., $\rho = 0.9$.



Level	$\mathcal{L}_2$	modified $\mathcal{L}_2$	Error confus.
2	5={1,4} 2 3	5={1,4} 2 3	5={1,4} 2 3
3	6={1,2,4} 3	6={1,3,4} 2	6={1,3,4} 2

Figure 2: Hierarchical 2D clustering example with 4 Gaussian clusters. 1 and 4 have wide distributions, 2 more narrow, and 3 extremely peaked. The priors are $P(k) = 0.3$ for $k = 1, 2, 3$ and $P(3) = 0.1$. The table shows the construction of higher-level clusters, e.g., the $\mathcal{L}_2$ distance measure groups clusters 1 and 4 at level 2, which is due to the fact that distance is based on the shape of the distribution and not only its mean. This also applies to the other dissimilarity measures. At level 3, however, the $\mathcal{L}_2$ method absorbs cluster 4 into 5 to form cluster 6. The other methods absorbs cluster 3 at this stage. The reason is that the prior of cluster 3 is rather low, which is neglected in the $\mathcal{L}_2$ method.

4 Experiments

The hierarchical clustering is illustrated for segmentation of e-mails. Define the term-vector as a complete set of the unique words occurring in all the emails. An email histogram is the vector containing frequency of occurrence of each word from the term-vector and defines the content of the email. The term-document matrix is then the collection of histograms for all emails in the database. Suitable preprocessing of the data is required for good performance. This concerns: 1) removing words, which are too likely (stop words) or too unlikely[6]; 2) keeping only word stems; and 3) normalizing all histogram vectors to unit length[7]. After preprocessing the term-document matrix contains 1280 (640 for training and 640 for testing) e-mail documents, and the term-vector consists of 1652 words. The emails where annotated into the categories: *conference*, *job* and *spam*. It is possible to model directly from the term-document matrix, see e.g., [18, 22], however, we deploy the commonly used framework Latent Semantic Indexing (LSI) [5], which operates using a latent space of feature vectors. These are found by projecting term-vectors into a subspace spanned by the left eigenvectors associated with largest singular values of a singular value decomposition of the term-document matrix. We are currently investigating methods for automatic determination of the subspace dimension based on generalization concepts, however, in this work, the number of subspace components is obtained from an initial study of classification error on a cross-validation set. We found that a 5 dimensional subspace provides good performance. Fig. 3 presents a 3D scatter plot of the first 3 feature dimensions, viz. the largest principal components. Data seem to be well separated, however, parts of *job* and *conference* e-mails are mixed. Fig. 4 shows the performance of the USGGM algorithm, and in Fig. 5 the hierarchical representations are illustrated.

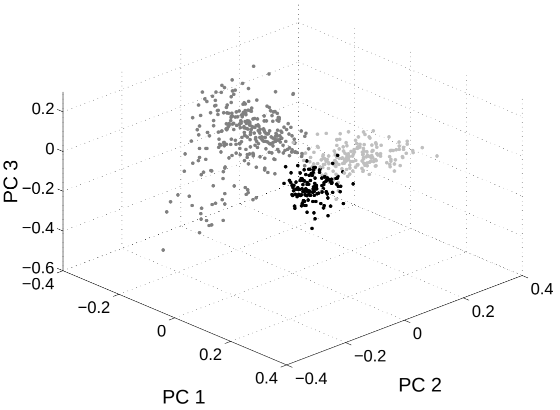


Figure 3: 3D scatter plot of the data. Three largest out of five principal components are displayed. Light grey color - *conference*, black - *job*, dark grey - *spam*. Data is well separated, however, there exists small confusion between job and conference e-mails.

[6] A threshold value for unlikely word up to approx. 100 occurrences has little influence on classification error. In the simulation the threshold was set to 40 occurrences.

[7] Another approach is to normalize the vectors to represent estimated probabilities, i.e., let vector sum to one. However, extensive experiments indicate that this approach give a feature space, which is not very appropriate for Gaussian mixture models.

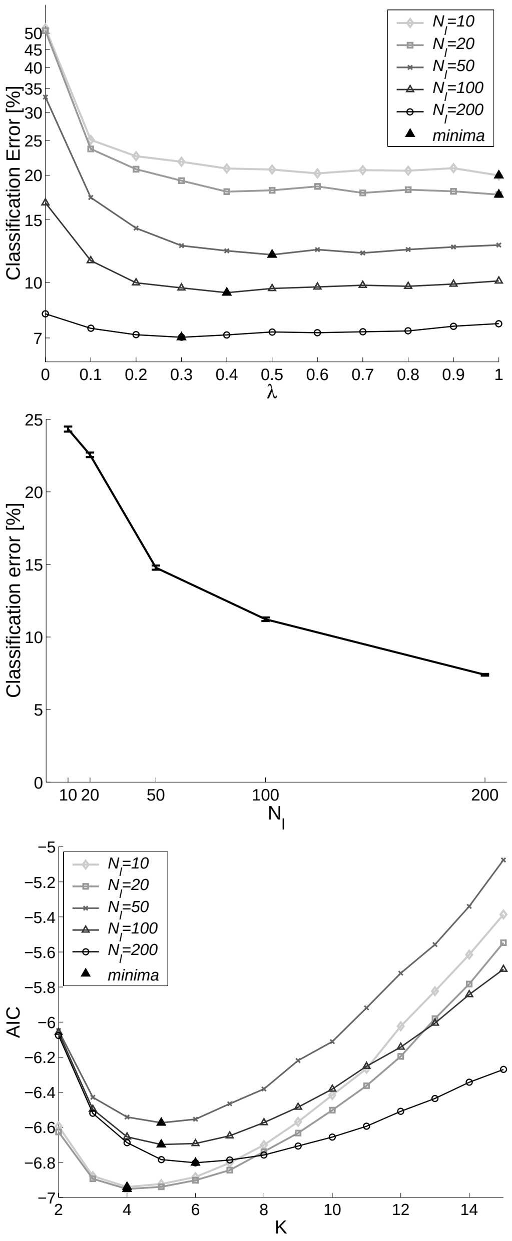


Figure 4: Average performance of the USGGM algorithm over 1000 repeated runs using $N_u = 200$ unlabeled examples and a variable number of labeled examples N_l. The algorithm parameter is set to $\gamma = 0.5$. Upper panel shows the performance as a function of the discount factor λ for unlabeled examples ($\lambda = 0$ corresponds to no unlabeled data). As expected, if few unlabeled examples are available, $N_l = 10, 20$, the optimal λ is close to one, and all available unlabeled data are fully used. As N_l increases λ decreases towards 0.3 for $N_l = 200$, indicating the reduced utility of unlabeled examples. The classification error is reduced approx. 26% using unlabeled data for $N_l = 10$, gradually decreasing to 1% for $N_l = 200$. The classification error for optimal λ as a function of N_l is shown in the middle panel. The lower panel shows number of components selected by the AIC criterion for optimal λ as described in Sec. 2.1. As N_l increase, also it is advantageous to increase the number of components.

y	k	$P(k\|y)$	**Keywords**
1	1	.7354	information, conference, call, workshop, university
	3	.0167	remove, address, call, free, business
	4	.2297	call, conference, workshop, information, submission, paper, web
	6	.0181	research, position, university, interest, computation, science
2	2	.6078	research, university, position, interest, science, computation, application, information
	6	.3922	research, position, university, interest
3	3	.6301	remove, call, address, free, day, business
	5	.3698	free, remove, call

Table 1: Keywords for the USGGM model. $y = 1$ is *conference*, $y = 2$ is *jobs* and $y = 3$ is *spam*.

Typical features as described in Sec. 3.2 and back-projecting into original term-space provides keywords for each cluster as given in Tab. 1. In Fig. 5 we choose to illustrate the hierarchies of individual class dependent densities $p(\boldsymbol{x}|y)$ using the modified $\mathcal{L}_2$ dissimilarity only. The cluster confusion measure is computational expensive if little overlap exist as many ancillary data are required. The modified $\mathcal{L}_2$ is computational inexpensive and basically treat dissimilarity as the cluster confusion, while the standard $\mathcal{L}_2$ do not incorporate priors. The *conference* class is dominated by cluster 1. This has keywords listed in Tab. 1, which are in accordance with the meaning of conference. The lower left panel shows the cluster level assignment distribution of test set emails, which are classified as conference emails cf. Sec. 3.1. Some obtain significant interpretation at level 1 (clusters 1-6), while others at a high level (cluster 9). Similar comments can be made for the *jobs* and *spam* classes.

For comparison, we further trained an unsupervised SGGM model and the results for a typical run are presented in Fig. 6. The top row illustrate the hierarchy formed by using the sample dependent, the modified $\mathcal{L}_2$ dissimilarity, and the cluster confusion similarity measures. For the sample dependent measure the numbers on top of the bars indicate the most frequent combinations of first level clusters. Clearly there is a significant resemblance among the sample dependent and the cluster confusion similarity hierarchies, e.g., higher level clusters formed by $\{1, 3\}$ and $\{2, 10\}$. However, inspection of the bottom row panels, which show the cluster confusion with the class labels, indicate that the cluster combinations of the sample dependent method is better aligned with the class labels. The modified $\mathcal{L}_2$ provides the best alignment of clusters with class labels at level 8 and is in

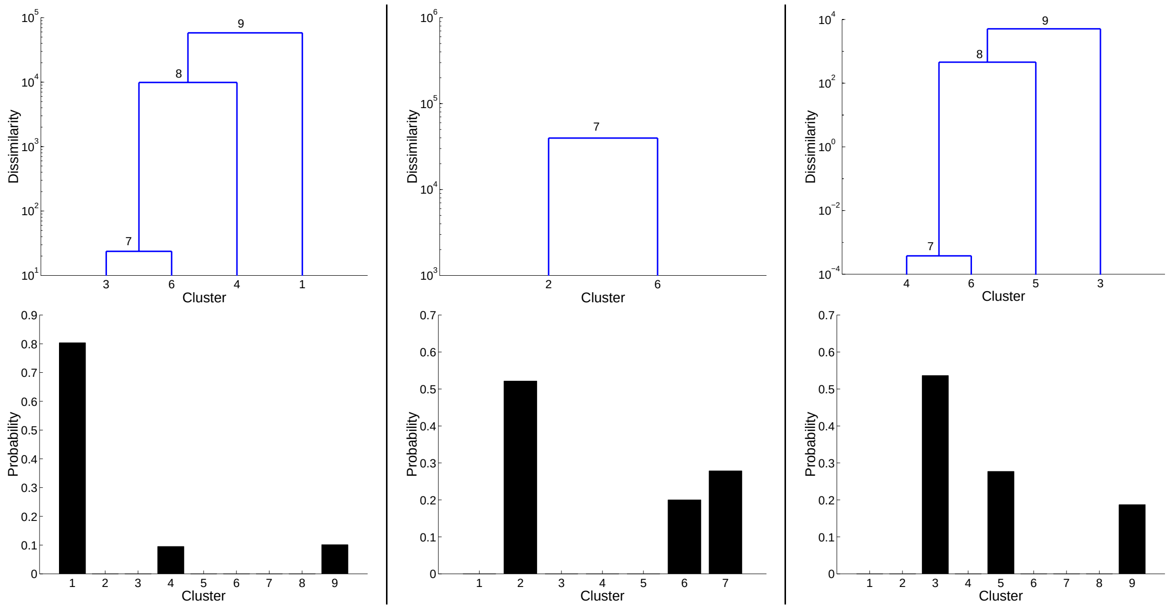
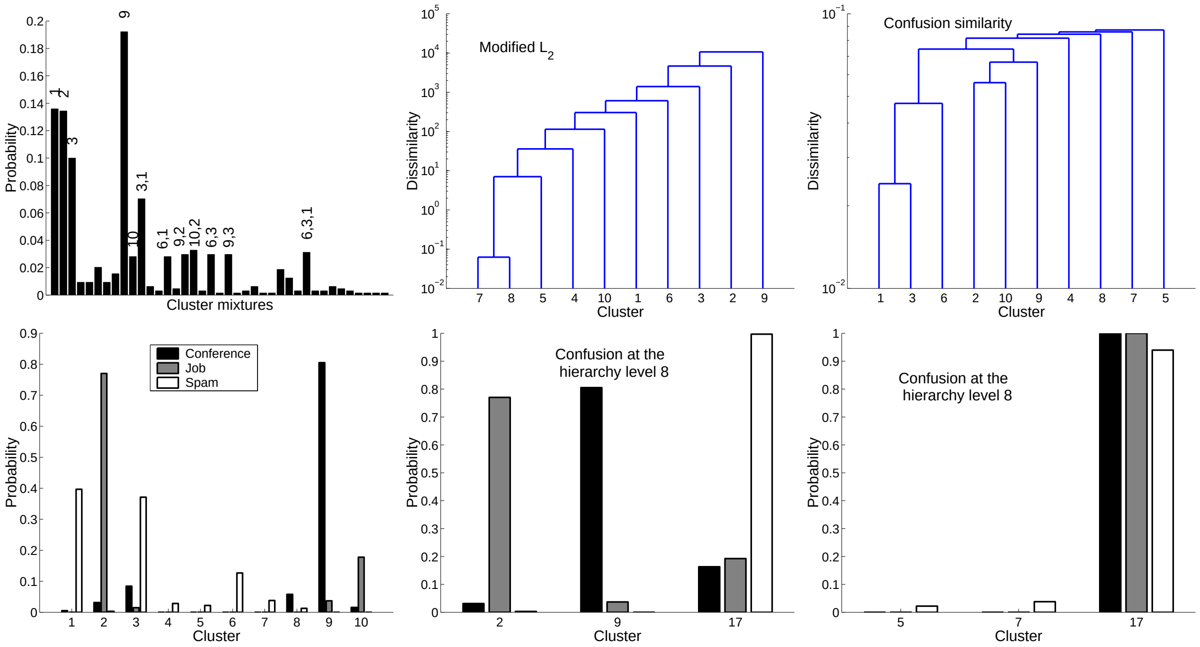


Figure 5: Hierarchical clustering using the USGGM model. Left column is class $y = 1$ *conference*, middle column $y = 2$ *jobs*, and right column is for $y = 3$ *spam*. Upper rows show the dendrogram using the modified $\mathcal{L}_2$ dissimilarity for each class, and the lower row the histogram of cluster level assignments for test data, cf. Sec 4.



1	free,remove	2	research,university,position	3	free,information,address,day
4	call,remove,address	5	free,service,site,sit,internet	6	call,remove,address,business
7	call,remove	8	free,address,state,quantity	9	information,conference,workshop
10	research,university,interest,position				

Figure 6: Unsupervised SGGM modeling of $p(\boldsymbol{x})$. Upper rows show the hierarchical structure. Left panel illustrates the sample dependent similarity measure Sec. 3.5, the middle panel the modified $\mathcal{L}_2$ dissimilarity measure, Sec. 3.3.1, and the right panel the cluster confusion measure Sec. 3.4. measure. Lower rows show the confusion of clusters with the annotated email labels at the first level in the hierarchy (left panel) and at level 8, where 3 clusters remain for the modified $\mathcal{L}_2$ dissimilarity (middle) and the cluster confusion measure (right panel). E.g., the black bars are the fraction of conference labeled test set emails ending up in a particular cluster. In addition, keywords for each cluster of the first level are also provided.

that respect superior to the other methods for the current data set. The keywords for clusters 2,10 and 9 provide perfect description of the *jobs* and *conference* emails, respectively. Keywords for the other clusters indicate that these mainly belong to the broad *spam* category.

5 Conclusions

This paper presented probabilistic agglomerative hierarchical clustering schemes based on the introduced unsupervised/supervised generalizable Gaussian mixture model (USGGM), which is an extension of [17]. The ability to learn from both labeled and unlabeled examples is important for many real world applications, e.g., text/webmining and medical decision support. The USGGM was successfully tested on a text-mining example concerning segmentation of emails.

Using a probabilistic scheme allows for automatic cluster and hierarchy level assignment for unseen data, and provides further a natural technique for an interpretation of the clusters via prototype examples and features. In addition, three different similarities measures for cluster merging were presented and compared.

References

[1] H. Akaike: "Fitting Autoregressive Models for Prediction," *Ann. of the Inst. of Stat. Math.*, vol. 21, 1969, pp. 243–247.

[2] C.M. Bishop: *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[3] C.M. Bishop and M.E. Tipping: "A Hierarchical Latent Variable Model for Data Visualisation," *IEEE T-PAMI* vol. 3, no. 20, 1998, pp. 281–293.

[4] J. Carbonell, Y. Yang and W. Cohen: "Special Isssue of Machine Learning on Information Retriceal Introduction," *Machine Learning* vol. 39, 2000, pp. 99–101.

[5] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer and R. Harshman: "Indexing by Latent Semantic Analysis," *Journ. Amer. Soc. for Inf. Science.*, vol. 41, 1990, pp. 391–407.

[6] C. Fraley: "Algorithms for Model-Based Hierarchical Clustering," *SIAM J. Sci. Comput.* vol. 20, no. 1, 1998, pp. 279–281.

[7] D. Freitag: "Machine Learning for Information Extraction in Informal Domains," *Machine Learning* vol. 39, 2000, pp. 169–202.

[8] L.K. Hansen, S. Sigurdsson, T. Kolenda, F.Å. Nielsen, U. Kjems and J. Larsen: "Modeling Text with Generalizable Gaussian Mixtures," In *Proc. of IEEE ICASSP'2000*, vol. 6, 2000, pp. 3494–3497.

[9] L.K. Hansen: "Supervised Learning with Labeled and Unlabeled Data," submitted for pulication 2001.

[10] T. Hastie and R. Tibshirani: "Discriminant Analysis by Gaussian Mixtures," *Jour. Royal Stat. Society - Series B*, vol. 58, no. 1, 1996, pp. 155–176.

[11] T. Honkela, S. Kaski, K. Lagus and T. Kohonen: "Websom — Self-organizing Maps of Document Collections, in *Proc. of Work. on Self-Organizing Maps*, Espoo, Finland, 1997.

[12] C.L. Jr. Isbell and P. Viola: "Restructuring Sparse High Dimensional Data for Effective Retrieval," in *Advances in NIPS 11*, MIT Press, 1999, pp. 480–486.

[13] T. Kolenda, L.K. Hansen and S. Sigurdsson: "Indepedent Components in Text," in *Adv. in Indep. Comp. Anal.*, Springer-Verlag, pp. 241–262, 2001.

[14] J. Larsen, L.K. Hansen, A. Szymkowiak, T. Christiansen and T. Kolenda: "Webmining: Learning from the World Wide Web," *Computational Statistics and Data Analysis*, 2001.

[15] J. Larsen, L.K. Hansen, A. Szymkowiak, T. Christiansen and T. Kolenda: "Webmining: Learning from the World Wide Web," in *Proc. of Nonlinear Methods and Data Mining*, Rome, Italy, 2000, pp. 106–125.

[16] M. Meila and D. Heckerman: "An Experimental Comparison of Several Clustering and Initialisation Methods," in *Proc. 14th Conf. on Uncert. in Art. Intel.*, Morgan Kaufmann, 1998, pp. 386–395.

[17] D.J. Miller and H.S. Uyar: "A Mixture of Experts Classifier with Learning Based on Both Labelled and Unlabelled Data," in *Advances in NIPS 9*, 1997, pp. 571–577.

[18] K. Nigam, A.K. McCallum, S. Thrun, and T. Mitchell: "Text Classification from Labeled and Unlabeled Documents using EM," *Machine Learning*, vol. 39, 2000, pp. 103–134.

[19] A. Szymkowiak, J. Larsen and L.K. Hansen: "Hierarchical Clustering for Datamining," in *Proc. 5th Int. Conf. on Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies KES'2001*, Osaka and Nara, Japan, 6–8 Sept., 2001.

[20] N. Vasconcelos and A. Lippmann: "Learning Mixture Hierarchies," in *Advances in NIPS 11*, 1999, pp. 606–612.

[21] E.M. Voorhees: "Implementing Agglomerative Hierarchic Clustering Ulgorithms for Use in Document Retrieval," *Inf. Proc. & Man.*, vol. 22, no. 6, 1986, pp. 465–476.

[22] A. Vinokourov and M. Girolami: "A Probabilistic Framework for the Hierarchic Organization and Classification of Document Collections," submitted for *Journal of Intelligent Information Systems*, 2001.

[23] A.S. Weigend, E.D. Wiener and J.O. Pedersen: "Exploiting Hierarchy in Text Categorization," *Information Retrieval*, vol. 1, 1999, pp. 193–216.

[24] C. Williams: "A MCMC Approach to Hierarchical Mixture Modelling," in *Advances in NIPS 12*, 2000, pp. 680–686.

[25] D. Xu, J.C. Principe, J. Fihser and H.-C. Wu: "A Novel Measure for Independent Component Analysis (ICA)," in *Proc. IEEE ICASSP98*, vol. 2, 1998, pp. 1161–1164.