Hierarchical Clustering for Datamining

Anna Szymkowiak, Jan Larsen, Lars Kai Hansen
Informatics and Mathematical Modeling Richard Petersens Plads, Build. 321,
Technical University of Denmark, DK-2800 Kongens Lyngby, Denmark,
Web: http://eivind.imm.dtu.dk, Email: asz,jl,lkhansen@imm.dtu.dk

Abstract. This paper presents hierarchical probabilistic clustering methods for unsupervised and supervised learning in datamining applications. The probabilistic clustering is based on the previously suggested Generalizable Gaussian Mixture model. A soft version of the Generalizable Gaussian Mixture model is also discussed. The proposed hierarchical scheme is agglomerative and based on a $\mathcal{L}_2$ distance metric. Unsupervised and supervised schemes are successfully tested on artificially data and for segmention of e-mails.

1 Introduction

Hierarchical methods for unsupervised and supervised datamining give multilevel description of data. It is relevant for many applications related to information extraction, retrieval navigation and organization, see e.g., [1, 2]. Many different approaches to hierarchical analysis from divisive to agglomerative clustering have been suggested and recent developments include [3, 4, 5, 6, 7]. We focus on agglomerative probabilistic clustering from Gaussian density mixtures. The probabilistic scheme enables automatic detection of the final hierarchy level. In order to provide a meaningful description of the clusters we suggest two interpretation techniques: 1) listing of prototypical data examples from the cluster, and 2) listing of typical features associated with the cluster. The Generalizable Gaussian Mixture model (GGM) and the Soft Generalizable Gaussian mixture model (SGGM) are addressed for supervised and unsupervised learning. Learning from combined sets of labeled and unlabeled data [8, 9] is relevant in many practical applications due to the fact that labeled examples are hard and/or expensive to obtain, e.g., in document categorization. This paper, however, does not discuss such aspects. The GGM and SGGM models estimate parameters of the Gaussian clusters with a modified EM procedure from two disjoint sets of observations that ensures high generalization ability. The optimum number of clusters in the mixture is determined automatically by minimizing the generalization error [10].

This paper focuses on applications to textmining [8, 10, 11, 12, 13, 14, 15, 16] 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.

2 The Generalizable Gaussian Mixture Model

The first step in our approach for probabilistic clustering is a flexible and universal Gaussian mixture density model, the generalizable Gaussian mixture model (GGM) [10, 17, 18], which

models the density for d-dimensional feature vectors by:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} P(k)p(\boldsymbol{x}|k), \;\; p(\boldsymbol{x}|k) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_k|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right) \qquad (1)$$

where $p(\boldsymbol{x}|k)$ are the component Gaussians mixed with the non-negative proportions $P(k)$, $\sum_{k=1}^{K} P(k)$. Each component k is described by the mean vector $\boldsymbol{\mu}_k$ and the covariance matrix $\boldsymbol{\Sigma}_k$. Parameters are estimated with an iterative modified EM algorithm [10] where means are estimated on one data set, covariances on an independent set, and $P(k)$ on the combined set. This prevents notorious overfitting problems with the standard approach [19]. The optimum number of clusters/components is chosen by minimizing an approximation of the generalization error; the AIC criterion, which is the negative log-likelihood plus two times the number of parameters.

For unsupervised learning parameters are estimated from a training set of feature vectors $\mathcal{D} = \{\boldsymbol{x}_n; n = 1, 2 \ldots, N\}$, where N is the number of samples. In supervised learning for classification from a data set of features and class labels $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}$, where $y_n \in \{1, 2, \ldots, C\}$ we adapt one Gaussian mixture, $p(\boldsymbol{x}|y)$, for each class separately and classify by Bayes optimal rule by maximizing $p(y|\boldsymbol{x}) = p(\boldsymbol{x}|y)P(y)/\sum_{y=1}^{C} p(\boldsymbol{x}|y)P(y)$ (under 1/0 loss). This approach is also referred to as mixture discriminant analysis [20].

The GGM can be implemented using either hard or soft assignments of data to components in each EM iteration step. In the hard GMM approach each data example is assigned to a cluster by selecting highest $p(k|\boldsymbol{x}_n) = p(\boldsymbol{x}_n|k)P(k)/p(\boldsymbol{x}_n)$. Means and covariances are estimated by classical empirical estimates from data assigned to each component. In the soft version (SGGM) e.g., the means are estimated as weighted means $\boldsymbol{\mu}_k = \sum_n p(k|\boldsymbol{x}_n) \cdot \boldsymbol{x}_n / \sum_n p(k|\boldsymbol{x}_n)$.

Experiments with the hard/soft versions gave the following conclusions. Per iteration the algorithms are almost identical, however, SGGM requires typically more iteration to converge, which is defined by no changes in assignment of examples to clusters. Learning curve[1] experiments indicate that hard GGM has slightly better generalization performance for small N while similar behavior for large N - in particular if clusters are well separated.

3 Hierarchical Clustering

In the suggested agglomerative clustering scheme we start by K clusters at level $j = 1$ as given by the optimized GGM model of $p(\boldsymbol{x})$, which in the case of supervised learning is $p(\boldsymbol{x}) = \sum_{y=1}^{C} \sum_{k=1}^{K_y} p(\boldsymbol{x}|k, y)P(k)P(y)$, where K_y is the optimal number of components for class y. 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 1 cluster at the final level, $j = 2K - 1$. Let $p_j(\boldsymbol{x}|k)$ be the density for the k'th cluster at level j and $P_j(k)$ as its mixing proportion, i.e., the density model at level j is $p(\boldsymbol{x}) = \sum_{k=1}^{K-j+1} P_j(k)p_j(\boldsymbol{x}|k)$. If clusters k and m at level j are merged into ℓ at level $j + 1$ then

$$p_{j+1}(\boldsymbol{x}|\ell) = \frac{p_j(\boldsymbol{x}|k) \cdot P_j(k) + p_j(\boldsymbol{x}|m) \cdot P_j(m)}{P_j(k) + P_j(m)}, \; P_{j+1}(\ell) = P_j(k) + P_j(m) \qquad (2)$$

The natural distance measure between the cluster densities is the Kullback-Leibler (KL) divergence [19], since it reflects dissimilarity between the densities in the probabilistic space. The drawback is that KL only obtains an analytical expression for the first level in the

[1] Generalization error as as function of number of examples.

hierarchy while distances for the subsequently levels have to be approximated [17, 18]. Another approach is to base distance measure on the $\mathcal{L}_2$ norm for the densities [21], i.e., $D(k,m) = \int \left(p_j(\boldsymbol{x}|k) - p_j(\boldsymbol{x}|m)\right)^2 dx$ where k and m index two different clusters. Due to Minkowksi's inequality $D(k,m)$ is a distance measure. 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 k and m at level j, respectively. The density of cluster k is given by: $p_j(\boldsymbol{x}|k) = \sum_{i\in\mathcal{I}_\alpha} \alpha_i p(\boldsymbol{x}|i)$, $\alpha_i = P(i)/\sum_{i\in\mathcal{I}_\alpha} P(i)$ 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 [21] the Gaussian integral $\int p(\boldsymbol{x}|i)p(\boldsymbol{x}|\ell)\, dx = G(\boldsymbol{\mu}_i - \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_\ell)$, where $G(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-d/2} \cdot |\boldsymbol{\Sigma}|^{1/2} \cdot \exp(-\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}/2)$. Define the vectors $\boldsymbol{\alpha} = \{\alpha_i\}$, $\boldsymbol{\beta} = \{\beta_i\}$ of dimension K and the $K \times K$ symmetric matrix $\boldsymbol{G} = \{G_{i\ell}\}$ with $G_{i\ell} = G(\boldsymbol{\mu}_i - \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_\ell)$, then the distance can be then written as $D(k,m) = (\boldsymbol{\alpha} - \boldsymbol{\beta})^\top \boldsymbol{G}(\boldsymbol{\alpha} - \boldsymbol{\beta})$. Figure 1 illustrates the hierarchical clustering for Gaussian distributed toy data.

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 $p_j(k|\boldsymbol{x}) > \rho$ 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.

Interpretation of clusters is done by generating likely examples from the cluster, see further [17]. For the first level in the hierarchy where distributions are Gaussian this is done by drawing examples from a super-eliptical region around the mean value, i.e., $(\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k) < \textit{const}$. For clusters at higher levels in the hierarchy samples are drawn from each Gaussian cluster with proportions specified by $P(k)$.

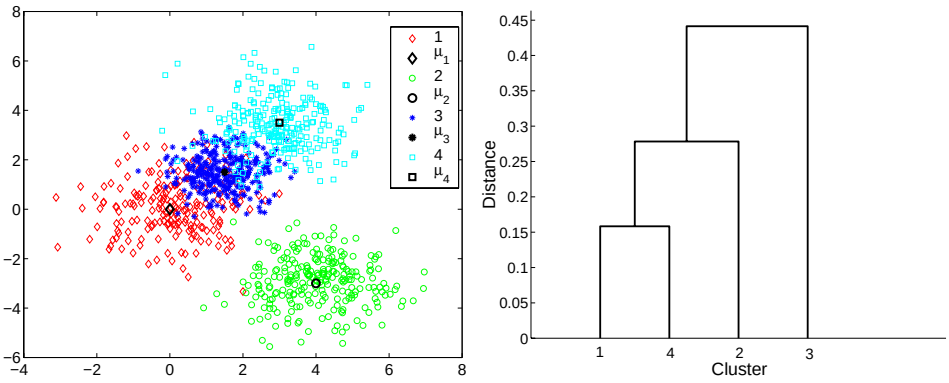

Figure 1: Hierarchical clustering example. Left panel is a scatter plot of the data. Clusters 1,2 and 4 have wide distributions while 3 has a narrow one. Since the distance is based on the shape of the distribution and not only its mean location, clusters 1 and 4 are much closer than any of these to cluster 3. Right panel presents the dendrogram.

4 Experiments

The hierarchical clustering is illustrated for segmentation of e-mails. Define 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. After suitable preprocessing[2] the term-document matrix contains 1405 (702 for training and 703 for testing) e-mail documents, and the term-vector 7798 words. The emails where annotated into the categories: *conference*, *job* and *spam*. It is possible to model

[2]Words which are too likely or too unlikely are removed. Further only word stems are kept.

directly from this matrix [8, 15], however we deploy Latent Semantic Indexing (LSI) [22] which operates from 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 value 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. We found that a 5 dimensional subspace provides good performance using SGGM.

A typical result of running supervised learning is depicted in Figure 2. Using supervised learning provides a better resemblance with the correct categories at the level in the hierarchy as compared with unsupervised learning. However, since labeled examples often are lacking or few the hierarchy provides a good multilevel description of the data with associated interpretations. Finding typical features as described on page 3 and back-projecting into original term-space provides keywords for each cluster as given in Table 1.

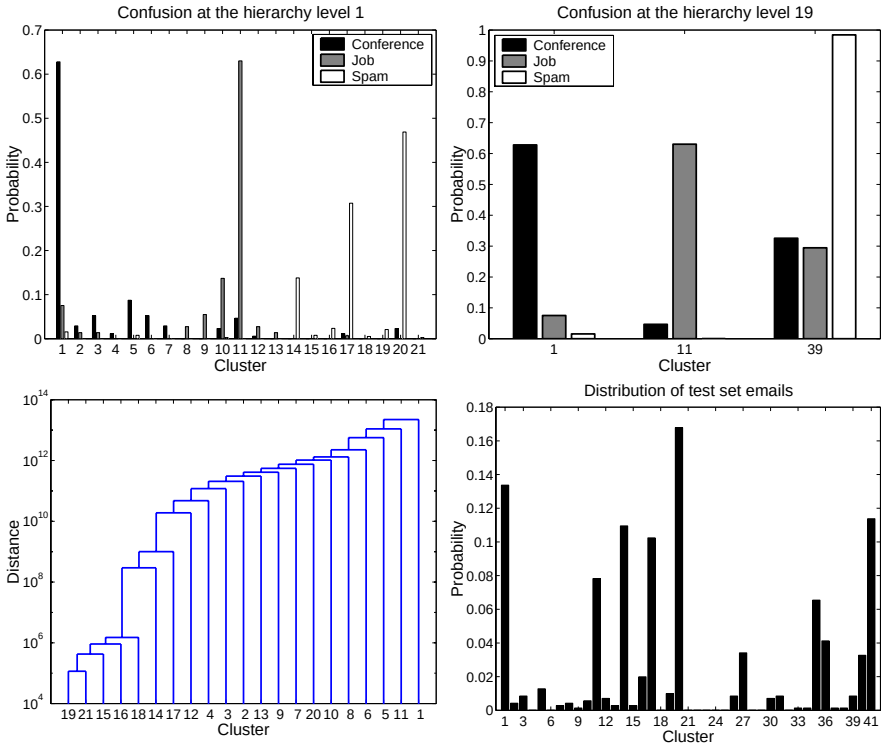

Figure 2: Supervised hierarchical clustering. Upper rows show the confusion of clusters with the annotated email labels on the training set at the first level and the level where 3 clusters remains, corresponding to the three categories *conference*, *job* and *spam*. At level 1 clusters 1,11,17,20 have big resemblance with the categories. In particular *spam* are distributed among 3 clusters. At level 19 there is a high resemblance with the categories and the average probability of erroneous category on the test set is 0.71. The lower left panel shows the dendrogram associated with the clustering. The lower right panel shows the histogram of cluster assignments for test data, cf. page 3. Clearly some samples obtain a reliable description at the first level (1–21) in the hierarchy, whereas others are reliable at a higher level (22–41).

5 Conclusions

This paper presented a probabilistic agglomerative hierarchical clustering algorithm based on the generalizable Gaussian mixture model and a $\mathcal{L}_2$ metric in probabilty density space. This leads to a simple algorithm which can be used both for supervised and unsupervised learning. In addition, the probabilistic scheme allows for automatic cluster and hierarchy level assignment for unseen data and further a natural technique for interpretation of the clusters

Table 1: Keywords for supervised learning

1	research,university,conference	8	neural,model	15	click,remove,hottest,action
2	university,neural,research	9	university,interest,computetion	16	free,adult,remove,call
3	research,creativity,model	10	research,position,application	17	website,adult,creativity,click
4	website,information	11	science,position,fax	18	website,click,remove
5	information,program,computation	12	position,fax,website	19	free,call,remove,creativity
6	research,science,computer,call	13	research,position,application	20	mac
7	website,creativity	14	free,adult,call,website	21	adult,government

1	research,university,conference	11	science,position,fax	39	free,website,cal l,creativity

via prototype examples and features. The algorithm was successfully applied to segmentation of emails.

References

[1] J. Carbonell, Y. Yang and W. Cohen, Special Isssue of Machine Learning on Information Retriceal Introduction, *Machine Learning* **39**, (2000) 99–101.

[2] D. Freitag, Machine Learning for Information Extraction in Informal Domains, *Machine Learning* **39**, (2000) 169–202.

[3] C.M. Bishop and M.E. Tipping, A Hierarchical Latent Variable Model for Data Visualisation, *IEEE T-PAMI* **3**, 20 (1998) 281–293.

[4] C. Fraley, Algorithms for Model-Based Hierarchical Clustering, *SIAM J. Sci. Comput.* **20**, 1 (1998) 279–281.

[5] 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.

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

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

[8] K. Nigam, A.K. McCallum, S. Thrun, and T. Mitchell Text Classification from Labeled and Unlabeled Documents using EM, *Machine Learning*, **39** 2–3 (2000) 103–134.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] E.M. Voorhees, Implementing Agglomerative Hierarchic Clustering Ulgorithms for Use in Document Retrieval, *Inf. Proc. & Man.* **22** 6 (1986) 465–476.

[15] 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.

[16] A.S. Weigend, E.D. Wiener and J.O. Pedersen Exploiting Hierarchy in Text Categorization, *Information Retrieval*, **1** (1999) 193–216.

[17] 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).

[18] 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, Italy, 2000, pp. 106–125.

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

[20] T. Hastie and R. Tibshirani Discriminant Analysis by Gaussian Mixtures, *Jour. Royal Stat. Society - Series B*, **58** 1 (1996) 155–176.

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

[22] 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.*, **41** (1990) 391–407.