MODELING TEXT WITH GENERALIZABLE GAUSSIAN MIXTURES

Lars Kai Hansen, Sigurdur Sigurdsson, Thomas Kolenda,
Finn Årup Nielsen, Ulrik Kjems and Jan Larsen

Department of Mathematical Modelling
Technical University of Denmark
DK-2800 Lyngby, Denmark
email: *lkhansen,siggi,thko,fn,uk,jl@imm.dtu.dk*

ABSTRACT

We apply and discuss generalizable Gaussian mixture (GGM) models for textmining. The model automatically adapts model complexity for a given text representation. We show that the generalizability of these models depends on the dimensionality of the representation and the sample size. We discuss the relation between supervised and unsupervised learning in text data. Finally, we implement a novelty detector based on the density model.

1. INTRODUCTION

Information retrieval is a very active research field which is starting to adapt advanced machine learning techniques for solving hard real world problems [17, 18]. Textmining or pattern recognition in text data is used to categorize text according to topic, to spot new topics, and in a broader sense to create more intelligent searches, e.g., by WWW search engines [12, 13, 14]. Textmining proceeds by pattern recognition based on text features, typically document summary statistics. While there are numerous high-level language models for extraction of text features, simple summary statistics are still preferred because they can be adapted automatically and continually, without costly manual intervention of language expertise. In the face of limited sets of labeled data pattern recognition algorithms typically fail to generalize in high dimensions, and there is a need for efficient and robust means for data reduction and feature extraction. To be able to generalize in high dimensions [17] choose to apply a biased architecture, the so-called naive Bayes classifier, here we will demonstrate generalizability of schemes with adaptive bias.

In [7] the Latent Semantic Indexing (LSI) approach was defined. LSI is based on a summary of the *term by document matrix*, i.e., a count of how often a given set of terms occur in the set of documents under analysis. The list of terms is adaptive and derived, e.g., by words that occur with a certain minimum frequency, in several documents, and possibly screened by a list of simple high-frequency *stop words*.

Research supported by the Danish Research Councils through the Danish Computational Neural Network Center (CONNECT), the THOR Center for Neuroinformatics, and Center for Multimedia.

In LSI term occurrence histograms are projected on a orthogonal set of "eigen-histograms" found by singular value decomposition. LSI can aid interpretation by visualizing group structure in the set of documents, typically by scatter plots of the term histograms on a reduced set of salient eigen-histograms. Another virtue of this representation is that it can be used as a dimensionality reduction scheme.

2. GENERALIZABLE GAUSSIAN MIXTURES

Our primary pattern recognition device will be the Gaussian mixture, see, e.g., [16] for a review. The Gaussian mixture density of a data vector $\boldsymbol{x}$ of dimension d, is defined as

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

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

where the component Gaussians are mixed with proportions $\sum_k P(k) = 1$, and we have defined the parameter vector $\boldsymbol{\theta}_k \equiv \{\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k\}$. The parameters are estimated from a set of examples $D = \{\boldsymbol{x}_n | n = 1, ..., N\}$. In the pattern recognition literature mixture densities are most estimated by maximum likelihood (ML), using various estimate-maximize (EM) methods [16]. The (negative log-) likelihood cost function is defined by

$$\mathcal{E}(D;\boldsymbol{\theta}) = \sum_{n=1}^{N} -\log p(\boldsymbol{x}_n|\boldsymbol{\theta}) \tag{3}$$

and is minimized by the ML parameters. The Gaussian mixture model is extremely flexible and simply minimizing the above cost function will lead to an "infinite overfit". It is easily verified that the cost function has a trivial (infinite) minimum attained by setting $\boldsymbol{\mu}_k = \boldsymbol{x}_k$ for $k = 1, \cdots, K-1$, and letting the corresponding covariances shrink to the zero matrix, while the remaining K'th Gaussian is adapted to the ML fit of the remaining $N - K + 1$ data points. This solution is optimal for the training set, but unfortunately has a generalization error roughly equal to that of the single

"background" Gaussian. To see this, let the generalization error is defined as the limit

$$\Gamma(\boldsymbol{\theta}) = \lim_{N\to\infty} \sum_{n=1}^{N} -\log p(\boldsymbol{x}_n|\boldsymbol{\theta}). \quad (4)$$

The ML mixture adapted on a finite data set has a generalization error where the singular components do not contribute because the data points assigned to them in the training set have zero measure in the test set. This instability has lead to much confusion in the literature and needs to be addressed carefully. Basically, there is no way to distinguish generalizable from non-generalizable solutions if we only consider the likelihood function. The most common remedy is to bias the component distributions so that they have a common covariance matrix, see e.g. [10]. Here we have decided to combine three approaches to ensure generalizability. First, we compute centers and covariances on different resamples of the data sets. Secondly, we make an exception rule for sparsely populated components –the covariance matrix defaults to the scaled full-sample covariance matrix. Finally we estimate the number of mixture components by choosing minimal value of the AIC-criterion [1, 9].

The algorithm is a modified EM procedure [8] and is defined as follows for a fixed number of mixture components, K.

Algorithm: Generalizable Gaussian Mixture

Initialization for K components

1. Compute the mean vector $\boldsymbol{\mu}_0 = N^{-1}\sum_n \boldsymbol{x}_n$.
2. Compute the covariance matrix of the data set: $\boldsymbol{\Sigma}_0 = N^{-1}\sum_n (\boldsymbol{x}_n - \boldsymbol{\mu}_0)(\boldsymbol{x}_n - \boldsymbol{\mu}_0)^\top$.
3. Initialize $\boldsymbol{\mu}_k \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$.
4. Initialize $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_0$.
5. Initialize $P(k) = 1/K$.

Repeat until convergence

1. Compute $p(k|\boldsymbol{x}_n)$ and assign $\boldsymbol{x}_n$ to the most likely component.
2. Split the data set in two parts[1] $D_{\boldsymbol{\mu}}$, $D_{\boldsymbol{\Sigma}}$.
3. For each k estimate $\boldsymbol{\mu}_k$ on the points in $D_{\boldsymbol{\mu}}$ assigned to component k.
4. For each k estimate $\boldsymbol{\Sigma}_k$ on the points in $D_{\boldsymbol{\Sigma}}$ assigned to component k. If the number of data points assigned to the k'th component, N_k, is less than $d+1$, then $\boldsymbol{\Sigma}_k \leftarrow (N_k\boldsymbol{\Sigma}_k + \boldsymbol{\Sigma}_0)/(N_k+1)$.
5. Estimate $P(k)$ as the frequency of assignments to component k.

2.1. Generalizable Gaussian Mixture Classifier

In pattern recognition we are interested in the joint density of patterns $\boldsymbol{x}$ and class labels c, denoted by $p(\boldsymbol{x}, c) = p(\boldsymbol{x}|c)P(c)$ where $p(\boldsymbol{x}|c)$ is the class conditioned density and $P(c)$ is the marginal class probabilities. For a labeled data set we design the classifier by adapting GGM's to each class separately. Hence, the joint density can be written

$$p(\boldsymbol{x}, c) = \sum_{k=1}^{K_c} p(\boldsymbol{x}|k)P(k|c)P(c), \quad (5)$$

where $P(k|c)$ and K_c are the component frequencies and number components found for class c.

Labels are assigned to a new data point in accordance with the optimal Bayes classification rule by selecting the maximum posterior probability $p(c|\boldsymbol{x})$.

3. GENERALIZABLE MODELS OF TEXT

To demonstrate the viability of the GGM for text modeling we apply it to a text database for which there are some results using an alternative strategy, namely the so-called naive Bayes classifier, see [17]. The primary objective of [17] is to show evidence that it is possible to enhance the learning process by mixing in unlabeled data, but these authors also present results from learning with labeled data alone. Here we will produce learning curves (generalization error as function of training set size) to compare with [17].

The available 2240 labeled documents where downloaded from the CMU WebKB repository [6]. The web pages are labeled according to the following categories: Course (24.7%), Faculty (21.6 %) Project (15.7%), Student (38.0%). A term list of 13071 words that occurred in two or more documents was defined without screening for "stop words". Term frequency histograms (i.e., normalized to unity) were computed for all documents.

Latent semantic analysis was performed and low dimensional projections ($d = 5, 20, 30$) used for modeling. Projections were selected by variance (PCA).

Learning curves for the GGM classifier were estimated by cross-validation. Data are randomly split 10 times into a test set of ($N_{\text{test}} = 1240$) and training sets of increasing sizes ($N_{\text{train}} = 100 - 1000$). Learning curves were estimated as the averaged test error as a function of d, the PCA dimension. We find a generalization cross-over as function of the dimension so that the larger dimensional representations need more samples for generalization.

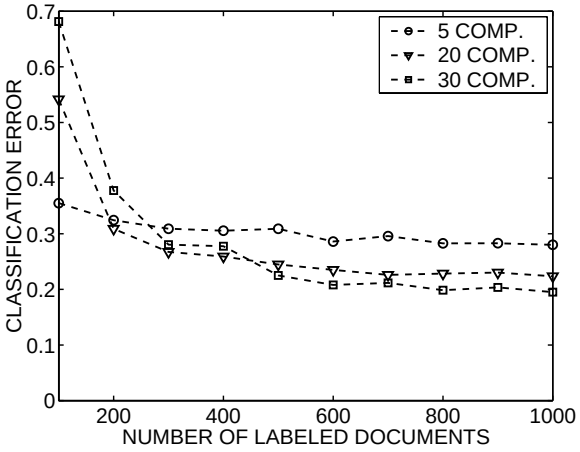
In [17] the interplay between supervised and unsupervised learning was discussed. To estimate the role of the labels for the GGM model, we have carried out a similar learning curve experiment for a "unsupervised-then-supervised" Gaussian mixture model. In this experiment we first estimate the GGM input density

$$p(\boldsymbol{x}) = \sum_k P(k)p(\boldsymbol{x}|k), \quad (6)$$

from all training data. Next we estimate the voting pattern for each component in the mixture and normalize to frequencies $P(c|k)$. The "unsupervised-then-supervised" classifier operates from the conditional probabilities

$$P(c|\boldsymbol{x}) = \frac{\sum_k P(k)P(c|k)p(\boldsymbol{x}|k)}{p(\boldsymbol{x})}. \quad (7)$$

[1] Often 50/50 splitting is used.



	Course	Faculty	Project	Student
Course†	0.92	0.03	0.05	0.02
Faculty†	0.04	0.64	0.10	0.13
Project†	0.03	0.09	0.75	0.02
Student†	0.01	0.24	0.10	0.83

Figure 1: Learning curves for supervised learning of the generalizable Gaussian mixture (GGM) classifier. The learning curves are indexed by the dimension of the input representation. The confusion matrix for $d = 30$ and $N_{\text{train}} = 1000$ († refers to the estimated class) shows that the main confusion appears among Faculty and Student groups.

The learning curve for the latter and the GGM classifier are compared in figure 2.

The proposed GGM classifier achieves classification rates and learning curves comparable to those found in [17]. The GGM model achieves this performance based on the full 13071 dimensional term-histograms showing the strength Latent Semantic Analysis representation. This allows for handling more complex textmining problems and also avoiding the selection of terms as in [17].

Learning is much less efficient for the "unsupervised-then-supervised" than the supervised GMM classifier, indicating significant class overlap in this problem.

4. NOVELTY DETECTION

When deploying a machine learning scheme in textmining we need to address the confidence problem of its predictions. Since the GGM classifier produces conditional probabilities we obtain in this way a clue to the "internal" confidence. The magnitude of the probabilities is determined by proximity of the decision boundary of the closest competing class. The overall test error rate give a clue to our confidence in the probabilities obtained from the system. However, when applied to new data the possibility exist, of course, that the new data can not in a meaningful way be assigned to any of the classes in the training data. In other words we need to address the novelty problem.

In line with recent work [2, 4, 15, 3], we will here develop a novelty detector based on the input density estimate

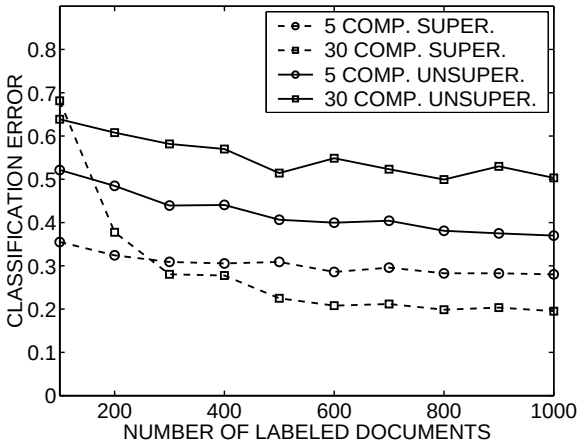


Figure 2: Learning curves for "unsupervised-then-supervised" learning and the supervised generalizable Gaussian mixture (GGM) classifier. Both sets of learning curves are indexed by the dimension of the input vector. Learning is much less efficient for the unsupervised procedure indicating significant class overlap.

available through the GMM model,

$$p(\boldsymbol{x}) = \sum_{c=1}^{C} \sum_{k=1}^{K_c} p(\boldsymbol{x}|k) P(k|c) P(c). \tag{8}$$

Since $p(\boldsymbol{x})$ is a probability density we cannot compare its values directly. However, by computing the probability $Q(t) = \text{Prob}(\boldsymbol{x} \in \mathcal{R})$ where $\mathcal{R} = \{\boldsymbol{x} : p(\boldsymbol{x}) < t\}$ for all thresholds t, we can set a threshold to reject low probability events[2]. In figure 3 we show $Q(t)$ based on training and a test set gathered from the documents above. We see that the test data are not rejected at reasonable Q-levels. The third curve in the figure shows the list of a third independent set of documents not related in an obvious way to the training and test sets. This data is declared novelty at levels below, e.g., $Q = 5\%$.

5. WEB VISUALIZATION

An objective of our multimedia aims is to create a tool that can assist the user in navigating complex multimedia web sites based on the VRML standard. The idea is to create a plug-in for the browser that generates a overview of one or more web sites At first the supervised part uses a list of labeled web pages, as typically can be found in a bookmark list ordered in folders. On the basis of the labeled pages a GGM classifier is build in order to classifies new pages into known bookmark labels or a new type (novelty

[2] $Q(t)$ can not be computed analytically, but can be obtained by ordering training/test set data according to $p(\boldsymbol{x})$ values and then forming the cumulative distribution. Another possibility is to base the calculation on large set of Monte Carlo samples obtained from $p(\boldsymbol{x})$. This also relates to the idea of highest probability density regions [5, Ch. 2.8].

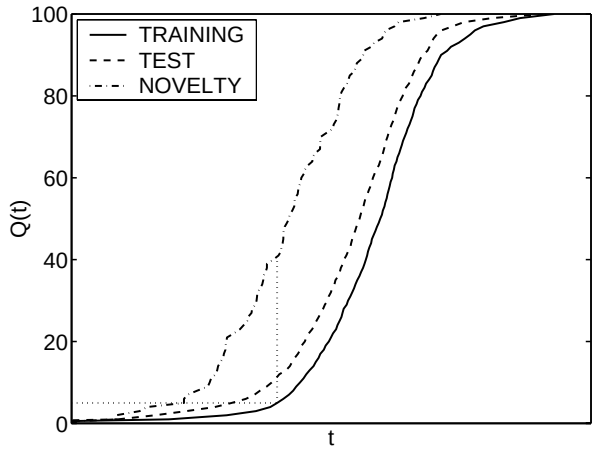


Figure 3: Novelty detection using web 173 pages from the Department group of the WebKB data set. The model has $d = 30$ dimensions and both the training and test sets contained 1120 documents. Threshold t for $p(\boldsymbol{x})$ is selected for $Q = 5\%$.

class). Using unsupervised GGM clustering of the pages in the novelty group and evaluating representative keywords for each mixture component we are able to get an overall description of the documents.

Four of the groups in the WebKB data set [6] have been used as labeled data, and are grouped in: Course, Faculty, Project and Student group. Using a fifth group (Other/Misc) of pages from the WebKB data set the, 40% of this group is detected as novelty pages, and these were clustered unsupervised into 4 new groups. In these groups the most probable patterns were back-projected into histogram space. Keywords were then defined as the most probable terms leading to interpretation of the 4 groups as: Places, Spare time, Computer systems and Multimedia.

Entering an unknown web site the user can without navigation learn its structure in relation to his/hers own list of topics, e.g., given by the "bookmarks". Documents not qualifying w.r.t. the current list of topics are represented by a list of keywords.

6. CONCLUSION

We have proposed the generalizable Gaussian mixture model for modeling of text based on the Latent Semantic Analysis representation. This approach enables us to work with very high dimensional term lists and still achieve state-of-the-art performance. For the WebKB database we found supervised learning superior to an "unsupervised-then-supervised" scheme. Novelty detection based on density estimates was used to spot odd documents.

REFERENCES

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

[2] L.D. Baker, T. Hofmann, A.K. Maccallum & Y. Yang: "A Hierarchical Probabilistic Model for Novelty Detection in Text," preprint CMU, 1999.

[3] M. Basseville & I.V. Nikiforov: *Detection of Abrupt Changes: Theory and Application*, Prentice-Hall, 1993.

[4] C.M. Bishop: "Novelty Detection and Neural Network Validation," *IEE Proceedings - Vision Image and Signal Processing*, vol. 141, no. 4, pp. 217–222, 1994.

[5] G.E.P. Box & G.C. Tiao: *Bayesian Inference in Statistical Analysis*, John Wiley & Sons, 1992.

[6] Carnegie Mellon University. `http://www.cs.cmu.edu/~textlearning`

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

[8] A.P. Dempster, N.M. Laird, D.B. Rubin: "Maximum likelihood from incomplete data via the EM algorithmm," *Jour. R. Stat. Soc. B*, vol. 39, pp. 1–38, 1977.

[9] L.K. Hansen and J. Larsen: "Unsupervised Learning and Generalization," *Proc. of the IEEE Int. Conf. on Neural Networks 1996*, vol. 1, pp. 25–30. 1996.

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

[11] T. Hofmann: "Learning the Similarity of Documents: An Information-Geometric Approach to Document Retrieval and Categorization," in T. Leen etal (eds.) *Adv. in NIPS 12*, MIT Press, to appear, 2000.

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

[13] T. Kolenda, L.H. Hansen, S. Sigurdsson: "Indepedent Components in Text," in M. Girolami (ed.) *Advances in Independent Component Analysis*, Springer-Verlag, to appear, 2000.

[14] T.K. Landauer, D. Laham & P. Foltz: "Learning Human-like Knowledge by Singular Value Decomposition: A Progress Report," *Adv. in NIPS 10*, MIT Press, pp. 45–51, 1998.

[15] A. Nairac *et al.*: "Choosing An Appropriate Model for Novelty Detection," *IEE 5th Int. Conf. on Artificial Neural Networks*, pp. 117–122, 1997.

[16] B.D. Ripley: *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.

[17] K. Nigam, A.K. McCallum, S. Thrun, T. Mitchell: "Text Classification from Labeled and Unlabeled Documents using EM," *Machine Learning*, to appear, 1999.

[18] A.S. Weigend, E.D. Wiener & J.O. Pedersen: *Exploiting Hierarchy in Text Categorization.* Working paper IS-98-22, Stern School of Business, NY Univ. `http://www.stern.nyu.edu/~aweigend/Research/Papers/TextCategorization/hierarchy.ps`