, , 1–18 ()

A Probabilistic Framework for the Hierarchic Organisation and Classification of Document Collections

ALEXEI VINOKOUROV AND MARK GIROLAMI

{alexei.vinokourov, mark.girolami}@paisley.ac.uk

School of Communication and Information Technologies
University of Paisley,
High Street, Paisley, PA1 2BE, Scotland, UK

Editor:

Abstract. This paper presents a probabilistic mixture modeling framework for the hierarchic organisation of document collections. It is demonstrated that the probabilistic corpus model which emerges from the automatic or unsupervised hierarchical organisation of a document collection can be further exploited to create a kernel which boosts the performance of state-of-the-art Support Vector Machine document classifiers. It is shown that the performance of such a classifier is further enhanced when employing the kernel derived from an appropriate hierarchic mixture model used for partitioning a document corpus rather than the kernel associated with a flat non-hierarchic mixture model. This has important implications for document classification when a hierarchic ordering of topics exists. This can be considered as the effective combination of documents with no topic or class labels (unlabeled data), labeled documents, and prior domain knowledge (in the form of the known hierarchic structure), in providing enhanced document classification performance.

Keywords: hierarchical probabilistic clustering, probabilistic latent semantic analysis, text categorization, support vector machines

1. Introduction

The notion of hierarchy is of great importance in areas such as, for example Pattern Recognition, Machine Learning, Artificial Intelligence and of course Information Retrieval (IR). Hierarchy embodies the principle of *divide-and-conquer* which takes a complex problem and breaks it up into a number of simpler sub-problems whose solutions, when combined, provide a solution to the original complex problem. A tree structured mixture of simple *experts* was developed in [16] to perform classification on complex problems and its performance provided enhanced results on particularly challenging pattern recognition problems [16]. A similar *mixture of experts* architecture was employed in [26] for text classification with consistently improved results reported on the particular test corpora utilized. Somewhat more recently a hierarchic structure composed of state-of-the-art Support Vector Machines (SVM) [23] [8] was employed for the classification of a large collection of web page summaries in [9].

Complex classification tasks, in many cases, benefit from the adoption of a hierarchic approach to the problem. It is then natural to consider the benefit that the

task of *unsupervised* classification or clustering can derive when set within a hierarchic framework. Indeed agglomerative approaches to the clustering of data are inherently hierarchical and agglomerative clustering has a long tradition in IR research [22]. The clustering of a document collection using a mixture model provides certain advantages over non-probabilistic clustering methods for example providing a *soft* rather than a *hard* clustering. In addition, a statistical model of the document generation process is provided and, as will be shown in this paper, the generative model can be further exploited in substantially boosting the performance of a related document classifier. Therefore this paper will focus on probabilistic methods of soft partitioning or clustering based on generative probabilistic methods of data modeling [2].

Probabilistic generative models are emerging to be particularly elegant and potentially powerful data analytic methods in a diverse number of areas [3]. The well known statistical method of principal component analysis (PCA) has also been shown to have a probabilistic basis and the principal directions of multi-variate data emerge as the maximum likelihood solutions of the associated generative model [2]. Indeed a probabilistic generative model of Latent Semantic Analysis [7] has recently been introduced in [11].

Many generative models take the form of a mixture-model such as that found in probability density estimation [1] [3]. In this case the notion of a structural hierarchy of classes (or mixture components) has not been considered in great detail, chiefly as the expressive power of a *flat* mixture model and a *hierarchic* mixture model are statistically equivalent. In other words the mixture of a mixture is a mixture. Therefore the data modeling capability of *flat* and *hierarchic* mixture models, it can be argued, are both the same. However despite this hierarchic generative models, can and have been shown to be more powerful than their *flat* equivalents in a number of significant cases [10] [25].

In [2] a hierarchic mixture of probabilistic principal component analysers was developed for the visualisation of structure in complex data. The ensuing method allowed for the hierarchy to be interactively built to aid exploratory data visualisation and analysis. Autoclass [3] the classic Bayesian mixture modeling software provides for the creation of a class hierarchy to allow parameter inheritance, and whereas algebraicly, both *flat* and *hierarchic* are equivalent an appropriate hierarchic model will provide a better fit to the data [10].

The automated hierarchic organisation of a collection of documents is an important and timely subject of study in IR [9], for example the automatic hierarchic organisation of web search results [4].

This paper presents a probabilistic mixture model with hierarchic structure for the unsupervised organisation of a collection of documents. The mixtures are based on both standard multinomial event models [18] and probabilistic latent semantic analysers (PLSA) [11] [25]. In addition to providing a hierarchic partitioned organization of a document collection the associated generative model allows the derivation of the Fisher kernel for the hierarchy. The Fisher kernel [14] engenders a similarity measure between documents based on the metric space induced by the probabilistic representation of the document class hierarchy.

Preliminary experimental results with SVM classifiers employing the derived kernels indicate that the classification performance is enhanced when a kernel - derived from an appropriate hierarchic class representation - is employed. This further extends the results originally reported in [12].

The remainder of the paper is structured as follows Section (2) describes the probabilistic mixture models employed whilst Section (3) presents the associated Fisher kernel for a hierarchic multinomial mixture model. Section (4) provides some experimental results and the final section concludes with some closing remarks.

2. The family of hierarchical probabilistic mixture models

Formally, text collections are represented by a *bag-of-words* model where a word $w \in W$, with dictionary cardinality $|W| = M$ occurs n_{dw} times in a part of a sample S representing a document $d \in D$, $|D| = N$. The sample S relates to the observed part of the data. It is known that one of the effective techniques to reduce the dimensionality of input space is to introduce unobserved variables and apply the Expectation Maximization (EM) algorithm [19] in the estimation of the associated parameters (a generative model). Usually in latent variable models hidden variables represent classes which *generate* documents or document / word pairs.

2.1. Basic framework

Hierarchical mixtures of probabilistic principal component analysers have been exploited recently in a number of works [2][24]. We develop the general framework for the multivariate, and conditionally independent, multinomial distribution which is the most appropriate for a vector space representation of text documents.

Let $\mathcal{T}$ be a network structure which we assume here to be a tree although all derivations in the sequel can be rewritten for any acyclic graphs. Each node of $\mathcal{T}$ from a layer $m-1$ represents a *cluster* c_{m-1} and if the cluster has children we assume that all data assigned to c_{m-1} are generated by a mixture of its children c_m with mixture weights $p(c_m|c_{m-1})$. The parameters $p(c_m|c_{m-1}) \equiv 0$ if cluster c_m is not a child of c_{m-1} in $\mathcal{T}$ (Fig. 1). The marginal probability of a cluster $\mathbf{c}_l = (c_1, c_2, \ldots, c_l)$ where $c_{m-1} = Pa(c_m)$ (parent of c_m), $m = 2, \ldots, l$, will be

$$p(\mathbf{c}_l) \equiv \prod_{m=1}^{l} p(c_m|c_{m-1}), \quad p(c_1|c_0) \equiv p(c_1) \tag{1}$$

Let $\{z_e^{c_m}\}$ be a set of *latent variables* for a sample element e that can be either a document d or a document-word pair (d, w). These variables indicate whether the sample element e belongs to the child c_m given the fact that it belongs to its parent c_{m-1}. In other words, $Z_e^{\mathbf{c}_l} \equiv z_e^{c_1} z_e^{c_2} ... z_e^{c_l} = 1$ if and only if e belongs to $\mathbf{c}_l$. As this kind of information is not contained in the sample these variables are sometimes called *latent variables*. The problem is then to restore the hidden or latent information from the sample. Approaches to the solution or estimations (expectations) of the latent variables are denoted as $p(c_m|c_{m-1}, e)$ and are also called *posteriors* .

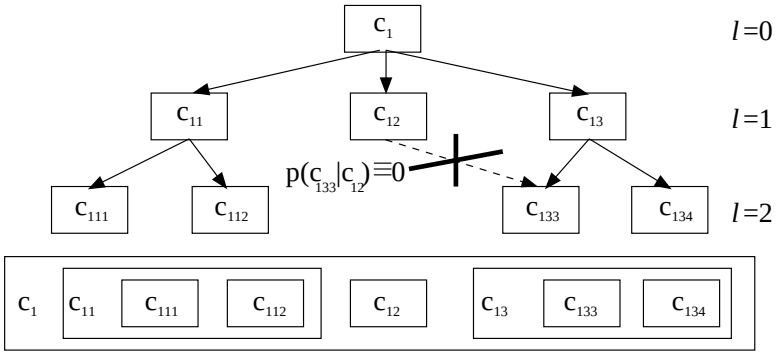

Figure 1. An example of a hierarchy (boundaries between clusters are relative as posterior probabilities form *soft* clustering.)

By definition the latent variables are independent and the expectation of $Z_e^{\mathbf{c}_l}$ can be decomposed as a mere product of expectations of all components:

$$p(\mathbf{c}_l|e) \equiv \prod_{m=1}^{l} p(c_m|c_{m-1}, e) \tag{2}$$

where $p(c_1|c_0, e) \equiv p(c_1|e)$ for convenience.

Let us consider a sample to be a number of document-word pairs generated by a mixture

$$p(d, w) = \sum_{\mathbf{c}_l} p(\mathbf{c}_l)p(d, w|\mathbf{c}_l) \tag{3}$$

It should be noted that for the case where there is only one layer in the hierarchy the model reduces to Probabilistic Latent Semantic Analysis (PLSA) [11]. Now in contrast to PLSA we may be interested in organising documents rather than modeling document-word pairs. If we restrict all pairs of the same document to the same cluster it would be equivalent to assuming a document $d = (w_1, w_2, \ldots, w_{n_d})$ to be generated by a mixture:

$$p(d) = \sum_{\mathbf{c}_l} p(\mathbf{c}_l)p(d|\mathbf{c}_l) \tag{4}$$

Referring to the symmetry of appearance of d and w variables in equations (3)-(4) the models defined by them are termed symmetrical and asymmetrical accordingly. To complete this section we cite the following statement that we will refer to later and that we leave here without proof (it follows trivially from Bayes and the independence of the tree structure).

Statement. (*Bayes formula for hierarchical mixture models*). The following holds

$$p(\mathbf{c}_l)p(d|\mathbf{c}_l) = p(\mathbf{c}_l|d)p(d) \tag{5}$$

2.2. Multinomial Asymmetric Hierarchical Analysis (MASHA)

Consider now the asymmetric model and let us instantiate the generic model (4) by the multinomial distribution

$$p(d|\mathbf{c}_l) \propto \prod_w p(w|\mathbf{c}_l)^{n_{dw}}, \quad \sum_w p(w|\mathbf{c}_l) = 1 \tag{6}$$

This yields the following EM-algorithm (Detailed derivation can be found in [25])

$$\begin{aligned} p_{new}(c_l|c_{l-1}, d) &= \frac{p(c_l|c_{l-1})p(d|\mathbf{c}_l)}{\sum_{c_l'} p(c_l'|c_{l-1})p(d|\mathbf{c}_l')} \\ p_{new}(w|\mathbf{c}_l) &= \frac{1+\sum_d n_{dw}\, p(\mathbf{c}_l|d)}{M+\sum_d \sum_w n_{dw}\, p(\mathbf{c}_l|d)} \\ p_{new}(c_l|c_{l-1}) &= \frac{1}{N}\sum_d p(c_l|c_{l-1}, d) \end{aligned}$$

In the equation for updating the class means $p(w|\mathbf{c}_l)$ we used Laplace smoothing [18] due to the sparseness of the data. The above equations define the EM l-step for layer

l and this depends on the previous $l-1$ steps. This determines the following order of calculations: first the parameters for the $l = 1$ layer are estimated using EM, the parameters are frozen, and inherited by the children as initial estimates. This reduces the number of parameters which require to be estimated by the children. For example there may be two *expert* nodes at the first layer corresponding to the topics c_{11} = MEDICINE and c_{12} = ART; clearly $p(w = Hockey \mid c_{11} = MEDICINE) \to 0$ and so the children of this *expert* do not require to estimate this particular parameter value. The subsequent layers $l = 2$ and so on up to the last layer $l = L$ are then parameterised in the same fashion. It is worthwhile stressing the difference between plain and hierarchical mixtures. Due to the *conveyor-like* computations in the hierarchical models they really are different from the flat ones although any hierarchical mixture is a mixture of mixtures and mathematically it is equivalent to a flat mixture. However the non-convex optimisation of the likelihood which is required will benefit from the combination of estimation of parameters for a number of smaller and simpler models. Indeed, for lower level mixtures some *work* is done by the upper level mixtures i.e. some subset of input data is chosen by the appropriate upper posteriors and more relevant (compact in terms of input space) data is left to be processed or inherited by the children nodes. This is why results of the hierarchical models are substantially different and, moreover, are usually somewhat better as our experiments described further confirm.

An example of (a part of) a hierarchy derived by MASHA from the `20 Newsgroups` corpus is shown in Fig. 2. Clusters or nodes of the hierarchy are represented by the directory labels along with the 10 most probable words estimated by the model. This collection consisted of 9000 documents drawn from nine Newsgroups with a dictionary size of 50,000 words being employed for this demonstration.

The authors also have developed an interactive MASHA-based demonstration tool for analysis of raw text collections. The tool provides several functions with one main function that runs EM algorithm for children of a chosen node or for subtree. A user can also manually change the hierarchy adding new randomly initialized nodes or deleting some nodes. It can be demonstrated that the tool allows to analyze the content of the collection and to reveal its underlying hierarchical structure.

2.3. Hierarchical Probabilistic Latent Semantic Analysis (HPLSA)

Like virtually most of the latent models hierarchic PLSA explicitly introduces a conditional independence assumption, namely that d and w are independent conditioned on the state of the associated latent variable. Hence we decompose the joint probability of a document-term pair as the following

$$p(d, w|\mathbf{c}_l) = p(d|\mathbf{c}_l)p(w|\mathbf{c}_l) \tag{7}$$

and then we obtain the following EM updates

$$p_{new}(c_l|c_{l-1}, d, w) = \frac{p(c_l|c_{l-1})p(d, w|\mathbf{c}_l)}{\sum_{c'_l} p(c'_l|c_{l-1})p(d, w|\mathbf{c}'_l)} \tag{8}$$

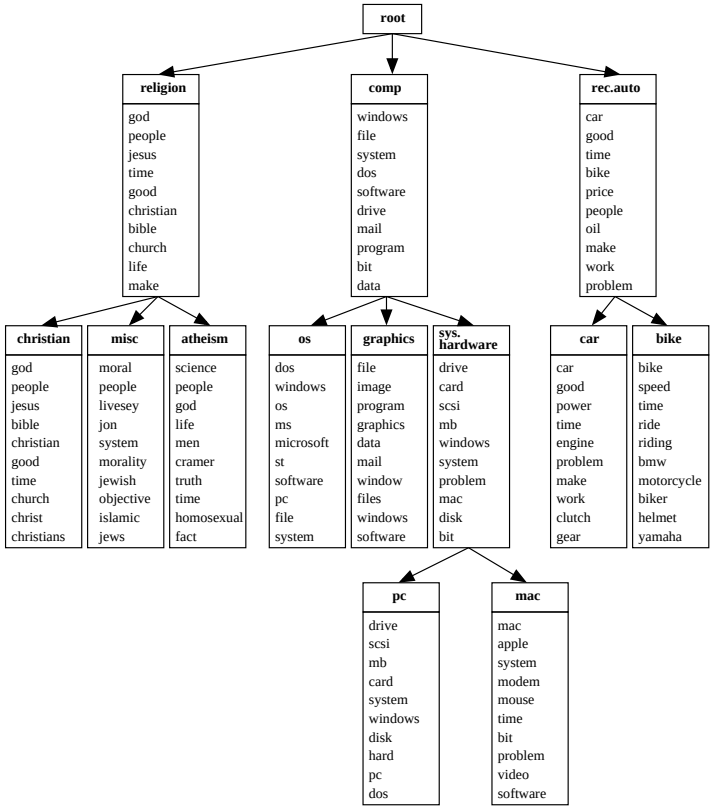

Figure 2. An example of (a part of) a hierarchy derived by MASHA from the `20 Newsgroups` collection.

$$p_{new}(d|\mathbf{c}_l) = \frac{\sum_w n_{dw} p(\mathbf{c}_l|d,w)}{\sum_d \sum_w n_{dw} p(\mathbf{c}_l|d,w)} \tag{9}$$

$$p_{new}(w|\mathbf{c}_l) = \frac{\sum_d n_{dw} p(\mathbf{c}_l|d,w)}{\sum_d \sum_w n_{dw} p(\mathbf{c}_l|d,w)} \tag{10}$$

$$p_{new}(c_l|c_{l-1}) = \frac{\sum_d \sum_w n_{dw} p(c_l|c_{l-1},d,w)}{\sum_d \sum_w n_{dw}} \tag{11}$$

We named the method Hierarchical PLSA to stress the connection with Probabilistic Latent Semantic Analysis (PLSA) [11]. As it can be seen the plain or flat PLSA is a particular case of the more general HPLSA where the hierarchy has only one layer and therefore no parameter inheritance mechanism.

3. Model selection

In contrast to other methods that could be applied for model selection we derive here a criterion specific for hierarchical models considered in this paper. Let a discrete variable $\mathcal{T}$ encode all possible true model configurations i.e. hierarchies. Then we compute the posterior probability of a model given sample S by Bayes rule $p(\mathcal{T}|S) \propto p(\mathcal{T})p(S|\mathcal{T})$ and select the model with the highest $p(\mathcal{T}|S)$. We will refer to $p(S|\mathcal{T})$ as the *marginal likelihood* (ML). Suppose all models are equal *a priori* then we are interested only in maximizing $p(S|\mathcal{T})$ that is obtained after choosing an *a priori* distribution of parameters $p(\theta_{\mathcal{T}}|\mathcal{T})$ and integrating them out:

$$p(S|\mathcal{T}) = \int p(S|\theta_{\mathcal{T}}, \mathcal{T}) \mathbf{p}(\theta_{\mathcal{T}}|\mathcal{T}) \mathbf{d}\theta_{\mathcal{T}} \tag{12}$$

where $L = p(S|\theta_{\mathcal{T}}, \mathcal{T})$ is the likelihood. For a Dirichlet prior distribution *conjugate* to multinomial families the integral in (12) cannot be computed in closed form and has to be approximated.

Assumption 1. The Cheeseman-Stutz approximation [5] is sufficient for our case.

Assumption 2. Moreover, we can disregard all terms except the marginal complete likelihood term in the Cheeseman-Stutz approximation or in other words we assume that

$$\log p(S|\mathcal{T}) \approx \log p(S'|\mathcal{T}) \tag{13}$$

So we approximate the marginal log-likelihood $\log p(S|\mathcal{T})$ through the marginal complete log-likelihood $\log p(S'|\mathcal{T})$ which can be computed analytically. The above two claims are supported by extensive experiments. The first assumption is also supported by experimental results in *[5]*.

Assumption. *Instead of complete data in S' that we do not have we can take its expectation estimation that we do have after running the EM algorithm.*

To give the result of the derivation of $\log p(S'|\mathcal{T})$ full version of which is given in Appendix A.1 we need some additional notations. Let $\mathcal{T}_l$ be the lth layer of $\mathcal{T}$ $\forall l = 1, 2, \ldots, L$; $Ch(c_m)$ be the set of children of c_m; $\wp(\mathcal{T}' \subset \mathcal{T})$ be the set of all paths starting at the root and ending at a some node in $\mathcal{T}'$, $Ds(c_m)$ be $\{\mathbf{c}_L \in \wp(\mathcal{T}_L) : c_m \in \mathbf{c}_L\}$ (descendants of c_m) and $[x]^y$ be $(x+y)!/x! = (x+1)(x+2)\ldots(x+y)$ for convenience. Then if we assume here that mixture weights (marginal cluster probabilities) $p(c_m|c_{m-1})$ are distributed uniformly and integrate them out in the expected likelihood (for the sake of brevity we consider only HPLSA model here) we obtain a Dirichlet integral so the θ_{CP}-term in the expression for the marginal likelihood is

$$\prod_{\substack{c_{m-1} \in \mathcal{T}: \\ Ch(c_{m-1}) \neq \emptyset}} \frac{\prod_{c_m \in Ch(c_{m-1})} \left(\sum_d \sum_w n_{dw} \sum_{\mathbf{c} \in Ds(c_m)} p(\mathbf{c}|d,w) \right)!}{[|Ch(c_{m-1})|]^{\sum_d \sum_w n_{dw} \sum_{\mathbf{c} \in Ds(c_{m-1})} p(\mathbf{c}|d,w)}} \tag{14}$$

In case of multinomial sampling the natural conjugate prior distribution of parameters $p(w|\mathbf{c}_L)$ is defined by a Dirichlet distribution $Dir(\{p(w|\mathbf{c}_L)\}_{w \in \mathcal{W}} \mid \{\alpha_{w|\mathbf{c}_L}\}_{w \in \mathcal{W}})$ where hyperparameters of the Dirichlet distribution are $\forall w \ \alpha_{w|\mathbf{c}_L} > 0$. Integrating out the parameters $p(w|\mathbf{c}_L)$ in the expected likelihood we obtain the θ_{WC}-term of $p(S'|\mathcal{T})$:

$$\prod_{\mathbf{c}_L \in \wp(\mathcal{T}_L)} \frac{(\alpha_{\mathbf{c}_L} - 1)!}{(N_{\mathbf{c}_L} + \alpha_{\mathbf{c}_L} - 1)!} \prod_w \frac{(N_{w\mathbf{c}_L} + \alpha_{w|\mathbf{c}_L} - 1)!}{(\alpha_{w|\mathbf{c}_L} - 1)!} \tag{15}$$

Here we denoted the *equivalent sample size* $\sum_w \alpha_{w|\mathbf{c}_L}$ as $\alpha_{\mathbf{c}_L}$, $\sum_d n_{dw} p(\mathbf{c}_L|d,w)$ as $N_{w\mathbf{c}_L}$ and $\sum_w N_{w\mathbf{c}_L}$ as $N_{\mathbf{c}_L}$. In experiments we set all hyperparameters to 1 although one could obtain a more theoretically justified choice from the expression of the individual means and variances for each random variable of the distribution [20]. For the parameters $p(d|\mathbf{c}_L)$ the derivation and result is the same up to substitutions of w by d. One should notice that (15) is similar to the result from the Bayesian networks theory (cf. [5]).

In Section 5 we will demonstrate the performance of the derived Stochastic Complexity Criterion (SCC) for model selection when applied to various document corpora.

Unlike a standard clustering of the document collection the methods presented provide a probabilistic model of the collection of documents. This model can then be employed in assessing the likelihood that a new or unseen document was generated from the hierarchy and therefore what part of the hierarchy it is most likely to fit. Another use of this emerging generative model of the document collection is the creation of a natural distance measure which can then be employed in building a subsequent classifier for new documents. The following section introduces what has been termed the Fisher kernel for the specific hierarchic models presented in the previous sections.

4. Fisher kernel

4.1. Theory

In our previous work [25] we have shown that hierarchic multinomial and asymmetric models can produce a superior clustering to that of flat models (also see [10]). This fact itself may not seem of great import in connection with end-user information retrieval applications but it has a number implications that might be quite utilitarian from this point of view. One of these implications is the possibility to build a Support Vector Machine *SVM-classifier* [23] based on the statistical models which have been discussed. The outcome of this can not be overestimated as it has been established that SVM classifiers are the best known classifiers in many areas including text classification [15] [9] [24]. To bridge a gap between unsupervised probabilistic hierarchic organisation and classification we now employ the so-called *Fisher kernel* [14].

First let us consider the average expected log-probability of a document normalized by its length. For brevity we will take only the MASHA model and for this model it is given by

$$l(d) = \sum_{\mathbf{c}_l} p(\mathbf{c}_l|d) \sum_w \hat{p}(w|d) \log p(w|\mathbf{c}_l) \tag{16}$$

where an empirical distribution of words in the document $\hat{p}(w|d) = n_{dw}/\sum_w n_{dw}$. For the moment we omitted a $p(\mathbf{c}_l)$-term as it carries no essential information about the document compared with the other term. The Fisher score $\nabla_\theta l(d)$, where ∇_θ is the gradient operator with respect to the parameters $\theta = \{p(w|\mathbf{c}_l), p(\mathbf{c}_l)\}$ then determines the direction of the steepest ascent in the average log-likelihood function whenever the Fisher information matrix $I = E\{\nabla_\theta l(d) \nabla_\theta^T l(d)\}$ maps natural gradients in the parameter space to ordinary (Euclidean) ones [14]. The Fisher kernel defined as

$$K(d_1, d_2) = \nabla_\theta^T l(d_1) I^{-1} \nabla_\theta l(d_2) \tag{17}$$

engenders a measure of similarity between any two documents d_1 and d_2. The derivation of the kernel is quite straightforward and follows [12]. Let us set $\rho(w|\mathbf{c}_l) = 2\sqrt{p(w|\mathbf{c}_l)}$, then

$$\frac{\partial l(d)}{\partial \rho(w|\mathbf{c}_l)} = \frac{\partial l(d)}{\partial p(w|\mathbf{c}_l)} \frac{\partial p(w|c_l)}{\partial \rho(w|c_l)} = \frac{p(\mathbf{c}_l|d)\hat{p}(w|d)}{p(w|\mathbf{c}_l)} \sqrt{p(w|\mathbf{c}_l)} = \frac{p(\mathbf{c}_l|d)\hat{p}(w|d)}{\sqrt{p(w|\mathbf{c}_l)}} \tag{18}$$

Similarly, $\rho(\mathbf{c}_l) = 2\sqrt{p(\mathbf{c}_l)}$. By Bayes rule (5) we have $p(d|\mathbf{c}_l) = p(\mathbf{c}_l|d)p(d)/p(\mathbf{c}_l)$ that yields

$$\frac{\partial l(d)}{\partial \rho(\mathbf{c}_l)} = \frac{\partial l(d)}{\partial p(\mathbf{c}_l)} \frac{\partial p(\mathbf{c}_l)}{\partial \rho(\mathbf{c}_l)} = \frac{1}{p(d)} p(d|\mathbf{c}_l) \sqrt{p(\mathbf{c}_l)} = \sqrt{p(\mathbf{c}_l|d)} p(\mathbf{c}_l) \tag{19}$$

Thus finally we have the result: $K(d_1, d_2) = K_1(d_1, d_2) + K_2(d_1, d_2)$ where

$$K_1(d_1, d_2) = \sum_{\mathbf{c}_l} p(\mathbf{c}_l|d_1)p(\mathbf{c_l}|\mathbf{d_2})/\mathbf{p(c_l)} \tag{20}$$

$$K_2(d_1, d_2) = \sum_{w} \hat{p}(w|d_1)\hat{p}(w|d_2) \sum_{\mathbf{c_l}} p(\mathbf{c}_l|d_1)p(\mathbf{c}_l|d_2)/p(w|\mathbf{c}_l) \tag{21}$$

In the derivation we have used a common assumption, namely that the Fisher matrix I can be approximated by an identity matrix. This result looks quite similar to [12] but in contrast to that work it is derived for asymmetric and a more general hierarchical model here.

The first part of the kernel (20) accounts for *topic similarity* of documents. Indeed, each document d can be represented by a vector of posteriors $\{p(\mathbf{c}_l|d)\}_{\mathbf{c}_l}$ where $\mathbf{c}_l$ runs through all the clusters. This representation has a number of advantages over the vector-space model. It can be shown that topic representation can resolve *polysems* i.e. words with multiple meanings and account for *synonyms*. For example, a word "president" can be used in different contexts: "president of a company" and "president of U.S.". Indeed, if both documents relate to a rare topic, they should be regarded as being much more alike. The second part of the kernel (21) without the sum over clusters is a simple dot-product of term-frequency vectors, a rather straightforward method of determining documents similarity called *cos-tf*. The sum over clusters, that is a dot-product of topic vectors weighed by the inverse word-conditional probabilities $p^{-1}(w|\mathbf{c}_l)$ adjusts cos-tf so that it becomes possible to distinguish between polysems. Indeed, suppose both documents have the term "president" but relate to different topics: "company" and "U.S.". Then a contribution of the term to the dot-product of topic vectors will be low. It would be much higher if the term were rare and the documents related to the same topic, e.g. "company" that would suggest that the documents concern a head of a company.

5. Experiments

5.1. Clustering

For an assessment of experimental results we need an evaluation measure of clustering. In all data sets that we experimented with documents were classified manually i.e. they had class labels. So we had two distributions of class variable, one given manually and another obtained by automatic clustering. Information theory has a popular criterion for the assessment of the diversity of two distributions known as mutual information (MI). This criterion that we will refer to as MIC has been suggested in [21] for unsupervised clustering and we found it rather informative. We have not chosen another criterion such as predicting occurrences of certain words in the context of a particular document popular among the natural language processing community because our primary aim is to organize raw document collections in a hierarchical manner and that is why we did not use a test set.

The mutual information between cluster and class label variables is given by

$$MI(\mathbf{c}, k) = \sum_{c_L} \sum_{\beta} p(\mathbf{c}_L, k_\beta) \log \frac{p(\mathbf{c}_L, k_\beta)}{p(\mathbf{c}_L)p(k_\beta)} \tag{22}$$

where $p(\mathbf{c}_L, k_\beta)$ is the joint probability of cluster $\mathbf{c}_L$ at the last layer of hierarchy L and labeled class β. The terms $p(\mathbf{c}_L)$ and $p(k_\beta)$ are the marginal probabilities. As an estimate of the joint distribution we used an average of posterior over a labeled class k_β.

The summary of experimental results is given in Table 1. Table 1 shows the performance of the hierarchical models along with plain ones in terms of MIC for a number of configurations. Configurations are given in the format: [# of experts] - [# of children] (# of clusters in plain model). For the plain methods the number of clusters is taken to be the same as the number of nodes at the last layer of the hierarchy in the hierarchical methods. The best result over 30 iterations for each method is taken. The (normalized) values of SCC for HPLSA are given in column 'SCC' in Table 1 besides MIC values so one could easily compare these supervised and data-driven criteria. As one can see SCC predicted correctly the best models for all text corpora.

5.2. Assessing Fisher kernel

The reported experiments were run on the `ModApt` split of the `Reuters-21578` text collection for 5 chosen classes: `earn`, `acq`, `money-fx`, `grain` and `crude` of all 90 classes that have at least one training and one test example. This subset was chosen to allow comparison with the results reported in [12]. So in total we had 9603 training and 3299 test documents with a vocabulary size of 9962 distinct terms. We obtained our results for subsamples with subsampling factors 0,05, 0,1 and 0,2 as it has been done for the PLSA-Fisher kernel in [12]. Subsamples have been chosen because it gives a smaller number of labeled documents in the training set. This may be handy as costs of labeling raw text collections are usually high. In Table 2 we give classification errors on the 5 chosen classes for SVM classifiers with different kernels and in Table 3 average classification errors along with percentage of improvements compared with other methods. The results suggest the following ascending order of performance amongst methods considered: linear, PLSA, MASA and MASHA where the last one demonstrates up to about 1.5 times better classification than the linear model and even with half the number of clusters outperforms flat PLSA. For all flat methods we have searched for the best configuration varying the number of clusters K=32,64,128 and for MASHA we tested hierarchies with up to 7 nodes at the upper level (experts) and up to 8 children of each of these nodes so we have chosen the best hierarchy which had 6 experts and 8 children of each expert. To ensure the optimal choice of a parameter C in SVM regulating a trade-off between generalization and fitting data a *cross-validation* with 1/10th of the training data randomly chosen forming a validation held-out set has been performed for each method. For all that first we were trying $C = 0.0001, 0.001, 0.01, \ldots, 1000, 10000$ values and then we were breaking each interval onto 4 parts until the current best

Table 1. The MIC-comparison of the clustering methods.

	MultiAsymm		MultiSymm		
Conf	MASHA	Plain	HPLSA	SCC	PLSA
4 Newsgroups N=4000, M=100					
5-2(10)	**0.78**	0.66	0.58	0.94	0.49
5-3(15)	**0.83**	0.67	0.42	0.41	0.45
5-4(20)	**0.86**	0.68	0.58	0.16	0.50
6-2(12)	0.29	**0.66**	0.32	0.71	0.45
6-3(18)	**0.83**	0.66	0.38	0.20	0.44
6-4(24)	0.90	0.58	**1.14**	1.00	0.49
7-2(14)	**1.61**	0.65	0.83	0.24	0.51
7-3(21)	**0.81**	0.57	0.67	0.58	0.48
7-4(28)	**0.82**	0.60	0.59	0.14	0.53
Reuters (grain,wheat,corn,ship,trade,crude) N=1977, M=100					
5-2(10)	**0.52**	0.44	0.17	0.40	0.26
5-3(15)	**0.58**	0.55	0.31	0.55	0.29
5-4(20)	**1.22**	0.64	0.48	1.00	0.33
6-2(12)	**0.67**	0.49	0.31	0.75	0.27
6-3(18)	**0.85**	0.60	0.24	0.30	0.33
6-4(24)	**1.03**	0.44	0.32	0.67	0.36
7-2(14)	**0.92**	0.53	0.24	0.00	0.30
7-3(21)	**0.75**	0.66	0.28	0.49	0.34
7-4(28)	**0.81**	0.49	0.31	0.40	0.37
WebKB N=8277, M=100					
5-2(10)	0.58	**0.62**	0.44	1.00	0.26
5-3(15)	**0.76**	0.74	0.34	0.78	0.30
5-4(20)	**0.88**	0.79	0.29	0.58	0.32
6-2(12)	0.50	**0.71**	0.33	0.53	0.27
6-3(18)	**0.84**	0.78	0.16	0.00	0.28
6-4(24)	0.60	**0.81**	0.26	0.70	0.29
7-2(14)	**0.99**	0.69	0.34	0.85	0.29
7-3(21)	0.54	**0.82**	0.29	0.69	0.30
7-4(28)	**0.89**	0.88	0.29	0.27	0.34

value stopped changing its magnitude by more than 1%. We should note that our figures for PLSA are somewhat different than those reported in [12], although the trend over the sub-sample splits is the same. This can be explained by the use of different SVM software or/and different cross-validation optimal parameter value search strategies.

6. Conclusions

In this paper we have presented a probabilistic hierarchical modeling method that provides the potential to automate the structuring of document corpora such as web collections. The results indicate 1) the superiority of the asymmetric methods over symmetric ones in terms of clustering quality; 2) the enhanced performance of the hierarchical methods over the plain ones and 3) SCC predicted the best models in terms of MIC correctly for all corpora tested.

The first conclusion may seem counter-intuitive at first glance. Indeed, as discussed in Section 2.1 the symmetric models are more complex (have more free parameters) than the asymmetric ones which can be regarded as constrained versions of the former. So the question remains why the asymmetric methods have exhibited a superior clustering performance to that provided by symmetric models? The explanation may be found in the fact that more complex models may have more sub-optimal solutions in terms of local maxima of the likelihood function in which EM can converge to, consequently, they have less probability of EM converging to the optimal solution when both are initialized from the same point. In other words, more complex models are more frequently subject to *overfitting*. This problem may be overcome by employing an *annealed* EM [13] for parameter estimation. The second conclusion may be justified by a consideration that a hierarchical model may compensate for the lack of flexibility of individual plain models by the overall flexibility of the complete hierarchy. All the above conclusions were tested and are consistent for documents defined by a wide range of vocabulary sizes.

One should notice that although we have a criterion for assessing hierarchies we do not suggest any algorithm for searching for them and this still remains an open question as full search is exponential in time. The problem has been identified in the Bayesian networks literature as the induction of networks from data [6]. Although a number of elegant methods have been suggested [6][17] we found them intractable in our case or at least requiring some modification and further investigation.

We have also derived a Fisher kernel for this method which when incorporated in an SVM classifier 1) highlighted the superiority of the hierarchical method over a previously developed flat model and 2) demonstrated an example of an application of information mined from a text corpus by this method in classification and consequently in improved information retrieval (IR). Both points above grant us the possibility to build an efficient IR system operating mostly in an automatic manner that may process large text collections and derive a `YAHOO!`-like hierarchic catalogue from it that would help users in navigation through the collection and can be used for efficient and intelligent execution of user queries.

Table 2. Classification errors percentage for SVMs with different kernels

subsample	earn	acq	money-fx	grain	crude
linear					
0.05	7.65	8.84	5.83	4.84	5.50
0.1	5.53	7.32	5.47	4.84	5.43
0.2	4.34	6.82	5.27	3.81	4.84
PLSA (K=128)					
0.05	4.44	7.78	5.83	4.34	4.54
0.1	4.21	7.45	6.13	4.27	4.80
0.2	3.74	7.12	5.20	3.38	4.17
MASA (K=64)					
0.05	5.03	6.46	4.54	4.67	4.14
0.1	3.31	5.07	4.31	4.01	3.78
0.2	2.88	5.07	4.11	3.35	3.28
MASHA (6,8)					
0.05	5.00	7.42	3.94	3.41	3.31
0.1	3.21	5.86	3.64	3.01	2.98
0.2	2.68	5.20	3.61	2.29	2.91

Table 3. Average classification errors percentage for SVMs with different kernels

subsampl. factor	MASHA (6,8)	MASHA/ MASA %	MASHA/ PLSA %	MASHA/ linear %
0.05	4.62	7.63	16.68	41.51
0.1	3.74	9.52	43.64	52.89
0.2	3.34	10.98	41.46	50.27

Appendix

A.1. Model selection

Using denotations introduced in Section 3 let us write the expected likelihood for HPLSA model as the following:

$$p(S'|\theta,\mathcal{T}) = \prod_d \prod_w \prod_{\mathbf{c}\in\wp(\mathcal{T}_L)} \left\{ \left(\prod_{m=1}^{L} p(c_m|c_{m-1}) \right) p(d|\mathbf{c}) p(w|\mathbf{c}) \right\}^{p(\mathbf{c}_L|d,w)n_{dw}} \tag{A.1}$$

Summing up the power at $p(c_m|c_{m-1})$ we can rewrite the product

$$\prod_d \prod_w \prod_{\mathbf{c}\in\wp(\mathcal{T}_L)} \prod_{m=1}^{L} p(c_m|c_{m-1})^{p(\mathbf{c}|d,w)n_{dw}} = \tag{A.2}$$

$$\prod_{\substack{c_m\in\mathcal{T}:\\ \exists c_{m-1}=Pa(c_m)}} p(c_m|c_{m-1})^{\sum_d \sum_w n_{dw} \sum_{\mathbf{c}\in Ds(c_m)} p(\mathbf{c}|d,w)} = \tag{A.3}$$

$$\prod_{\substack{c_{m-1}\in\mathcal{T}:\\ Ch(c_{m-1})\neq\emptyset}} \prod_{c_m\in Ch(c_{m-1})} p(c_m|c_{m-1})^{\sum_d \sum_w n_{dw} \sum_{\mathbf{c}\in Ds(c_m)} p(\mathbf{c}|d,w)} \tag{A.4}$$

Taking into account the constraint $\sum_{\alpha_m} p(\alpha_m|\alpha_{m-1}) = 1$ we write the *a priori* distribution of parameters $p(\alpha_m|\alpha_{m-1})$

$$p(\theta_{CP}|\mathcal{T}) = \prod_{\substack{c_{m-1}\in\mathcal{T}:\\ Ch(c_{m-1})\neq\emptyset}} \frac{\Gamma(\beta_1+\beta_2+\cdots)}{\Gamma(\beta_1)\Gamma(\beta_2)\cdots\Gamma(\beta_{|Ch(c_{m-1})|})} \times \tag{A.5}$$

$$\prod_{c_m\in Ch(c_{m-1})} p(c_m|c_{m-1})^{\beta_{\alpha_m}-1} \tag{A.6}$$

where hyperparameters of the Dirichlet distribution are $\forall m\ \beta_{\alpha_m} > 0$. Now integrating (A.2) over distribution (A.6) we obtain

$$\prod_{\substack{c_{m-1}\in\mathcal{T}:\\ Ch(c_{m-1})\neq\emptyset}} \frac{\Gamma(\beta_1+\beta_2+\cdots+\beta_{|Ch(c_{m-1}|)})}{\Gamma(\beta_1)\Gamma(\beta_2)\cdots\Gamma(\beta_{|Ch(c_{m-1})|})} \times \tag{A.7}$$

$$\frac{\prod_{c_m\in Ch(c_{m-1})} \Gamma\left(\sum_d \sum_w n_{dw} \sum_{\mathbf{c}\in Ds(c_m)} p(\mathbf{c}|d,w) + \beta_m\right)}{\Gamma\left(\sum_{c_m\in Ch(c_{m-1})} \left\{\sum_d \sum_w n_{dw} \sum_{\mathbf{c}\in Ds(\alpha_m)} p(\mathbf{c}|d,w) + \beta_m\right\}\right)} \tag{A.8}$$

If we take all hyperparameters equal to 1 and remembering that $\Gamma(n+1) = n!$ we obtain a simplified expression for the θ_{CP}-term in $p(S'|\mathcal{T})$:

$$\prod_{\substack{c_{m-1}\in\mathcal{T}:\\ Ch(c_{m-1})\neq\emptyset}} \frac{\prod_{c_m\in Ch(c_{m-1})} \left(\sum_d \sum_w n_{dw} \sum_{\mathbf{c}\in Ds(c_m)} p(\mathbf{c}|d,w)\right)!}{[|Ch(c_{m-1})|]^{\sum_d \sum_w n_{dw} \sum_{\mathbf{c}\in Ds(c_{m-1})} p(\mathbf{c}|d,w)}} \tag{A.9}$$

For the $p(w|\mathbf{c})$ group of parameters the constraint is $\sum_w p(w|\mathbf{c}) = 1$ and consequently we should suppose the density function is as following

$$p(\theta_{WC}|\mathcal{T}) = \prod_{\mathbf{c}\in\wp(\mathcal{T}_L)} \frac{\Gamma(\alpha_{1|\mathbf{c}} + \alpha_{2|\mathbf{c}} + \cdots + \alpha_{M|\mathbf{c}})}{\Gamma(\alpha_{1|\mathbf{c}})\Gamma(\alpha_{2|\mathbf{c}})\cdots\Gamma(\alpha_{M|\mathbf{c}})} \prod_w p(w|\mathbf{c})^{\alpha_{w|\mathbf{c}}-1} \tag{A.10}$$

Integrating the part of (A.1) containing $p(w|\mathbf{c})$ we obtain the θ_{WC}-term of $p(S'|\mathcal{T})$:

$$\prod_{\mathbf{c}\in\wp(\mathcal{T}_L)} \frac{\Gamma(\alpha_{1|\mathbf{c}} + \alpha_{2|\mathbf{c}} + \cdots + \alpha_{M|\mathbf{c}})}{\Gamma(\alpha_{1|\mathbf{c}})\Gamma(\alpha_{2|\mathbf{c}})\cdots\Gamma(\alpha_{M|\mathbf{c}})} \frac{\prod_w \Gamma\left(\alpha_{w|\mathbf{c}} + \sum_w n_{dw} p(\mathbf{c}|d,w)\right)}{\Gamma\left(\sum_d \left\{\alpha_{w|\mathbf{c}} + \sum_w n_{dw} p(\mathbf{c}|d,w)\right\}\right)} \tag{A.11}$$

and after expanding Γ function as a factorial and some rearrangements we obtain (15). But if we take unit hyperparameters we get a simplified version of (A.11):

$$\prod_{\mathbf{c}\in\wp(\mathcal{T}_L)} \frac{\prod_w \left(\sum_d n_{dw} p(\mathbf{c}|d,w)\right)!}{[M]^{\sum_d \sum_w n_{dw} p(\mathbf{c}|d,w)}} \tag{A.12}$$

Finally, the result for $p(d|\mathbf{c})$-group of parameters is just the same up to substitutions d by w and M by N.

References

1. Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.
2. Christopher M. Bishop and Michael E. Tipping. A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):281–293, 1998.
3. P. Cheeseman and J. Stutz. Bayesian classification (autoclass): Theory and results. *Advances in knowledge discovery and data mining*, pages 153–180, 1995.
4. H. Chen and S. Dumais. Bringing order to the web: Automatically categorising search results. In *Proceedings of CHI-00, ACM International Conference on Human Factors in Computing Systems*, pages 145–152, Den Haag, NL, 2000.
5. David Maxwell Chickering and David Heckerman. Efficient approximations for the marginal likelihood of bayesian networks with hidden variables. *Machine Learning*, 29:181–212, 1996.
6. Gregory F. Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.
7. S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41, 1990.
8. Platt J. Heckerman D/ Dumais, S. T. and M. Sahami. Inductive learning algorithms and representations for text categorisation. In *Proceedings of the Seventh International Conference on Information and Knowledge Management*, pages 148–155, 1998.
9. Susan T. Dumais and Hao Chen. Hierarchical classification of Web content. In Nicholas J. Belkin, Peter Ingwersen, and Mun-Kew Leong, editors, *Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*, pages 256–263, Athens, GR, 2000. ACM Press, New York, US.
10. Robin Hanson, John Stutz, and Peter Cheeseman. Bayesian classification with correlation and inheritance. In Ray Myopoulos, John; Reiter, editor, *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 692–698, Sydney, Australia, August 1991. Morgan Kaufmann.
11. Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–99)*, pages 289–296, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

12. Thomas Hofmann. Learning the similarity of documents: an information-geometric approach to document retrieval and categorization. In *Proceedings of Advances in Neural Information Processing Systems (NIPS'99)*, volume 12. MIT Press, 2000.
13. Thomas Hofmann and Joachim M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):1–14, 1997.
14. T. Jaakola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Proceedings of Advances in Neural Information Processing Systems (NIPS'98)*, volume 11, pages 487–493. MIT Press, 1999.
15. Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398 in Lecture Notes in Computer Science, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.
16. Michael I. Jordan. Hierarchical mixture of experts and the EM algorithm. *Neural Computation*, (6):181–214, 1994.
17. Marina Marin̆a) and Michael I. Jordan. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1:1–48, 2000.
18. A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.
19. G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. New York; John and Wiley and Sons, Inc., 1997.
20. Bo Thiesson. Score and information for recursive exponential models with incomplete data. In Dan Geiger and Prakash Pundalik Shenoy, editors, *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI-97)*, pages 453–463, San Francisco, August 1–3 1997. Morgan Kaufmann Publishers.
21. Shivakumar Vaithyanatham and Byron Dom. Generalized model selection for unsupervised learning in high dimensions. In *Proceedings of Advances in Neural Information Processing Systems (NIPS'99)*, volume 12. MIT Press, 2000.
22. C.J. van Rijsbergen. *Information Retrieval*. London: Butterworths, 2 edition, 1979.
23. Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
24. Nuno Vasconcelos and Andrew Lippman. Learning mixture hierarchies. In *Proceedings of Advances in Neural Information Processing Systems (NIPS'98)*, volume 11, pages 606–612. MIT Press, 1999.
25. Alexei Vinokourov and Mark Girolami. A probabilistic hierarchical clustering method for organizing collections of text documents. In *15th International Conference on Pattern Recognition (ICPR'2000)*, volume 2, pages 182–185. IEEE Computer Society, 2000.
26. A.S. Weigend, E.D. Wiener, and O. Pedersen, J. Exploiting hierarchy in text categorisation. *Information Retrieval*, 1(3):193–216, 1999.