

A hierarchical model for clustering and categorising documents

E. Gaussier¹, C. Goutte¹, K. Popat², and F. Chen²

¹ Xerox Research Center Europe, 6 Ch. de Maupertuis, F-38240 Meylan, France
Eric.Gaussier,Cyril.Goutte@xrce.xerox.com

² Xerox PARC, 3333 Coyote Hill Rd, Palo Alto, CA94304, USA
popat,fchen@parc.xerox.com

Abstract. We propose a new hierarchical generative model for textual data, where words may be generated by topic specific distributions at any level in the hierarchy. This model is naturally well-suited to clustering documents in preset or automatically generated hierarchies, as well as categorising new documents in an existing hierarchy. Training algorithms are derived for both cases, and illustrated on real data by clustering news stories and categorising newsgroup messages. Finally, the generative model may be used to derive a Fisher kernel expressing similarity between documents.

1 Overview

Many IR tasks, such as clustering, filtering or categorisation, rely on *models* of documents. The basic approach to document modelling considers that each document is an (independent) observation, with words used as features and word frequencies, combined with various normalisation schemes, are the feature values. This so-called *bag of word* approach is exemplified by the Naïve Bayes method [24] and hierarchical extensions [14, 19], where the features are modelled using multinomial distributions. Other related techniques rely eg on Gaussian mixture models[18], fuzzy k-means, etc.

In the following, we adopt a different standpoint, using the concept of co-occurrence [10]. In this approach, one basic observation is the co-occurrence of a word in a document. There is no numerical feature, only the absence or presence of co-occurrences and associated counts. A document (and thus a document collection) arises as an assortment of such co-occurrences.

In this article we address the problem of clustering and categorising documents, using probabilistic models. Clustering and categorisation can be seen as two sides of the same coin, and differ by the fact that categorisation is a supervised task, ie labels identifying categories are provided for a set of documents (the training set), whereas, in the case of clustering the aim is to automatically organise unlabelled documents into clusters, in an unsupervised way. Popular document categorisation methods include nearest neighbours, Naïve Bayes [24] or support vector machines [12], while document clustering has been tackled with eg k-means, latent semantic indexing [3] or hierarchical agglomerative

methods [23, 21]. Even though clustering and categorisation can be studied independently of each other, we propose a general model that can be used for both tasks. One strength of our model lies in its capacity to deal with hierarchies of categories/clusters, based on soft assignments while maintaining a distinction between document and word structures. This has to be contrasted with traditional approaches which result in hard partitions of documents.

In the next section, we will give a detailed presentation of the generative hierarchical probabilistic model for co-occurrences that we will use for performing clustering and categorisation of documents. One interesting feature of this model is that it generalises several well-known probabilistic models used for document processing. We then describe the motivations behind our model (section 3), then give further details on the implementation of clustering (section 4) and categorisation (section 5) using it. After presenting some encouraging experimental results (section 6), we discuss in section 7 some technical and methodological aspects, and in particular we show how our model can be used to obtain a measure of similarity between documents using Fisher kernels [11].

2 Modelling documents

In this contribution we address the problem of modelling documents as an assortment of co-occurrence data [10]. Rather than considering each document as a vector of word frequency (bag-of-words), we model the term-document matrix as the result of a large number of co-occurrences of words in documents. In that setting, the data can be viewed as a set of triples $(i(r), j(r), r)$, where $r = 1 \dots L$ is an index over the triples, each triple indicating the occurrence of word $j(r)$ in document $i(r)$. Note that a word can occur several times in a document. For example, the fact that word “line” (with number $i = 23$) occurs twice (for $r = 6$ and $r = 9$) in document #1 will correspond to two triples: $(1, 23, 6)$ and $(1, 23, 9)$, ie $i(6) = i(9) = 1$ and $j(6) = j(9) = 23$. An alternative representation is (i, j, n_{ij}) indicating that word j occurs n_{ij} times in document i (leading to $(1, 23, 2)$ in the previous example). Note that the index is then over i and j , with several instances of $n_{ij} = 0$.

In order to structure the data, we adopt a generative probabilistic approach, where we assume it was generated from a hierarchical model. Some early attempts at defining such models are Hofmann’s Hierarchical Asymmetric Clustering Model (HACM) [10] or Hierarchical Shrinkage [14]. In this paper we propose a new model which has some additional flexibility. In our model the data are generated by the following process:

1. Pick a document class α with probability $P(\alpha)$,
2. Choose document i using the class-conditional probability $P(i|\alpha)$,
3. Sample a word topic ν with the class-conditional probability $P(\nu|\alpha)$,
4. Generate word j using the topic-conditional distribution $P(j|\nu)$.

Here we use the term of (document) class α to denote a group of documents sharing some common thematic feature (for example they all deal with “computer graphics”). We use the term of (word) topic to denote a homogeneous

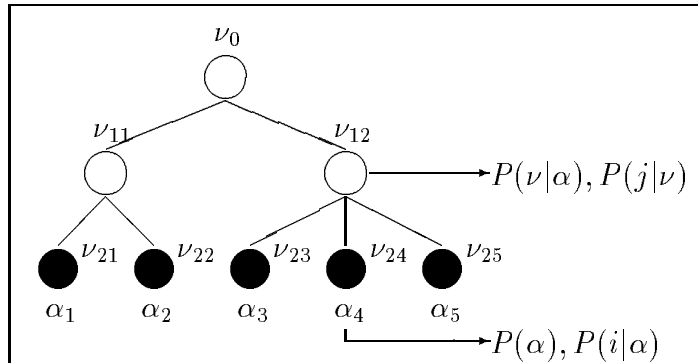


Fig. 1. An example hierarchical model with classes at the leaves: document classes $\alpha_1 \dots \alpha_5$ are indicated by filled circles, topics $\nu_0 \dots \nu_{25}$ are indicated by circles (accordingly, classes are also topics). Sampling of a co-occurrence is done by first sampling a class (ie selecting the fourth leaf here) and a document from the class-conditional $P(i|\alpha)$, then sampling a topic from the class conditional $P(\nu|\alpha)$ (second topic at the second level here) and a word from the topic-conditional $P(j|\nu)$.

semantic field described by a specific word distribution. One document may be partly relevant to different topics, and several documents from different classes may share common, higher level topics. For example, a document on “computer graphics” and a document on “algorithmics” may have parts relevant to the more general “computer science” topic.

In a typical hierarchy, we may for example assign documents to classes at the leaves of the hierarchy, while words are sampled from topics which are any node in the hierarchy (figure 1). This therefore allows classes to be linguistically related by sharing some common ancestors/topics, as exemplified in our previous example. In a more general hierarchy, documents could be assigned to classes at any node of the hierarchy, like in Yahoo! or Google directories. In all cases, topics ν will here be restricted to the set of ancestors of a class α , a fact that we note $\nu \uparrow \alpha$. This can also be achieved by setting $P(\nu|\alpha) = 0$ whenever $\nu \not\uparrow \alpha$.

According to this model, the probability of a given co-occurrence (i, j) is:

$$P(i, j) = \sum_{\alpha} P(\alpha) P(i|\alpha) \sum_{\nu \uparrow \alpha} P(\nu|\alpha) P(j|\nu) \quad (1)$$

where we have used the conditional independence of documents and words given the class α . The parameters of the model are the discrete probabilities $P(\alpha)$, $P(i|\alpha)$, $P(\nu|\alpha)$ and $P(j|\nu)$. One interesting feature of this model is that several known models can be written as special cases of ours. For example:

- $P(i|\alpha) = P(i), \forall \alpha$ (documents independent of class) yields the HACM [10];

- Dropping $P(\alpha)$ and $P(i|\alpha)$ (or imposing uniform probabilities on these quantities) yields a hierarchical version of the Naïve Bayes model;¹
- Flat (ie non-hierarchical) models are naturally represented using $P(\nu|\alpha) = 1$ iff $\nu = \alpha$ (ie one topic per class), yielding $P(i, j) = \sum_{\alpha} P(\alpha)P(i|\alpha)P(j|\alpha)$, aka Probabilistic Latent Semantic Analysis (PLSA, [8]).
- A flat model with a uniform distribution over classes α and documents i reduces to the (flat) Naïve Bayes model [24].

Our general model does not account eg for the “hierarchical” model described in [22], where probabilities for classes α are directly conditioned on the parent node $Pa(\alpha)$. The co-occurrence model in this framework is, at each level l in the hierarchy: $P(i, j) \propto \sum_{\alpha \in l} P(\alpha|Pa(\alpha))P(i|\alpha)P(j|\alpha)$. In this framework, however, the parent node mainly serves as a pooling set for its children, whereas in our model (eq. 1), the whole hierarchical structure is used and co-occurrences are generated by a single class-topic combination. In contrast, the model in [22] leads in fact to a series of flat clustering distributing all data across the nodes at each level, a strategy that we do not consider fully hierarchical in the context of this paper.

In addition, we will see in the Discussion (section 7) that the document similarity induced by the Fisher kernel [11] gives similar contribution for the model in [22] and for (flat) PLSA [9], whereas fully hierarchical models introduce a new contribution from the hierarchy.

Because our model can be seen as a hierarchical extension to PLSA, where both documents and words may belong to different clusters, we will refer to our hierarchical model as HPLSA, ie Hierarchical Probabilistic Latent Semantic Analysis, in the case of clustering (unsupervised classification). In the context of categorisation (supervised classification), we will call it HPLC, ie Hierarchical Probabilistic Latent Categoriser, or PLC when no hierarchy is used.

3 Motivations behind our model

Depending on the meaning we may give to i and j , different problems can be addressed with our model. Generally, we can view i and j as representing any pairs of co-occurring objects. In the preceding section, we have focused on modelling documents, with i objects representing documents, and j objects representing words. Within the general framework of clustering a large collection of unlabelled documents,² an important application we envisage is topic detection, where the goal is to *automatically* identify topics covered by a set of documents. In such a case, a cluster can be interpreted as a topic defined by the word probability distributions, $P(j|\nu)$. Our soft hierarchical model takes into account several important aspects of to this task: 1) a document can cover (or be explained by)

¹ Several models implement hierarchical extensions to Naïve Bayes, eg Hierarchical Shrinkage [14], Hierarchical Mixture Model [19], and the unpublished [1]. One key difference between these models lie in the estimation procedures.

² This contrasts with the task of categorising documents in pre-defined classes induced by a corpus of labelled documents.

several themes (soft assignment of i objects provided by $P(i|\alpha)$), 2) a theme is described by a set of words, which may belong to different topics due to polysemy and specialisation (soft assignment of j objects provided by $P(j|\nu)$), and 3) topics are in many instances hierarchically organized, which corresponds to the hierarchy we induce over clusters. Moreover, our use of a general probabilistic model for hierarchies allows us to deal with document collections in which topics cannot be hierarchically organized. In that case, probabilities $P(\nu|\alpha)$ are concentrated on $\nu = \alpha$, inducing a flat set of topics rather than a hierarchy. We obtained such a result on an internal, highly heterogeneous, collection.

Another important application we envisage for our model is knowledge structuring (see for example [5]), where it is important first to recognize the different realisations (ie terms and their variants) of the main concepts used in a domain, and secondly to organize them into ontologies.³ A common feature of all ways of organizing terms in taxonomies is the central role of the “generalisation/specialisation” relation. Traditional approaches [6, 17] induce hierarchies by repeatedly clustering terms into nested classes, each identified with a concept. Such clustering relies on some measure of similarity of the contexts in which terms occur as (inverse) distance between terms themselves.⁴ Different kinds of contexts can be considered, from local ones, such as direct syntactic relations, or small windows centered on the term under consideration, to global ones, such as sentences, or paragraphs. However, problems in such clustering approaches arise from the following:

- Terms may be polysemous, and thus may belong to several classes;
- Contexts (eg a verb of which the term of interest is the direct object) may be ambiguous, suggesting the inclusion of similar context in different classes.

The collocation “hot line” illustrates the above two points: the meaning of the (polysemous) head “line” is determined by the collocate “hot”, the meaning of which in the complete expression differs from its usual, daily one. We thus should be able to assign both “line” and “hot” to different clusters.

Polysemy of words, whether regarded as terms under focus or contexts of terms, and polytopicality of textual units, at various levels of granularity (from sentences to complete documents), thus call for models able to induce hierarchies of clusters while assigning objects to different clusters. The additional flexibility provided by our model over previously proposed ones exactly amounts to the ability of soft clustering both objects in the hierarchy instead of one.

4 Soft hierarchical document clustering

We first consider the task of automatically organising unlabelled documents, given as an i.i.d. collection of co-occurrences $\mathcal{D} = \{(i(r), j(r), r)\}_{r=1\dots L}$. The

³ Here ontologies are the taxonomies structuring concepts of a domain.

⁴ This approach is reminiscent of Harris’ distributionalism, where classes of linguistic units are identified by the different contexts they share [7].

class membership of the documents (and co-occurrences) is unknown and must be inferred from the unlabelled data alone. The likelihood of the parameters can be expressed (using the independence assumption) as the product of the individual likelihoods (1) as $P(\mathcal{D}) = \prod_r P(i(r), j(r))$. Due to the presence of the multiple sums under the product of examples, the log-likelihood

$$\log P(\mathcal{D}) = \sum_{r=1}^L \log \left(\sum_{\alpha} P(\alpha) P(i(r)|\alpha) \sum_{\nu} P(\nu|\alpha) P(j(r)|\nu) \right) \quad (2)$$

must in general be maximised numerically. This is elegantly done using the Expectation-Maximisation (EM) algorithm [4] by introducing the unobserved (binary) indicator variables specifying the class and topic choices for each observed co-occurrence.

Unfortunately, the iterative EM method is often sensitive to initial conditions. As a remedy, we used deterministic annealing [16, 20] in conjunction with the EM iterations. Deterministic annealing EM is also called tempered EM in a statistical physics analogy, as it corresponds to introducing a temperature which is used to “heat” and “cool” the likelihood, in order to obtain a better and less sensitive maximum. Deterministic annealing has two interesting features: first it has been shown empirically [20, 8] to yield solutions that are more stable with respect to parameter initialisation; second, it provides a natural way to grow the hierarchy (cf. [15] and below).

Tempered EM iteratively estimates the maximum likelihood parameters for the model using the completed likelihood, ie the likelihood of the data plus unobserved indicator variables. Let us note $C_{\alpha}(r)$ the (binary) class indicator for observation r , such that $C_{\alpha}(r) = 1$ iff document $i(r)$ is sampled from class α ($C_{\alpha}(r) = 0$ otherwise), and $T_{\alpha\nu}(r)$ the (binary) topic indicator such that $T_{\alpha\nu}(r) = 1$ iff $(i(r), j(r))$ is sampled from class α and topic ν ($T_{\alpha\nu}(r) = 0$ otherwise). Noting C (resp. T) the indicator vector (resp. matrix) of all C_{α} (resp. $T_{\alpha\nu}$), the likelihood of a generic completed observation consisting of (i, j, C, T) is:

$$P(i, j, C, T) = \sum_{\alpha} C_{\alpha} P(\alpha) P(i|\alpha) \sum_{\nu} T_{\alpha\nu} P(\nu|\alpha) P(j|\nu) \quad (3)$$

Again, the (completed) log-likelihood over the entire dataset is expressed by combining (3) for all observations as $\sum_r \log P(i(r), j(r), C(r), T(r))$. At each EM iteration t , we first take the expectation of the completed log-likelihood over a “tempered” version (using temperature β) of the posterior probability of the indicator variables:

$$Q_{\beta}^{(t)} = \sum_{C, T} \left[\sum_r \log P^{(t)}(i(r), j(r), C(r), T(r)) \frac{P^{(t)}(i(r), j(r), C(r), T(r))^{\beta}}{\sum_{C, T} P^{(t)}(i(r), j(r), C, T)^{\beta}} \right] \quad (4)$$

For $\beta = 0$, the tempered distribution is uniform, and for $\beta = 1$, we retrieve the posterior at iteration t , $P^{(t)}(C(r), T(r)|i(r), j(r))$. $Q_{\beta}^{(t)}$ can be conveniently

expressed using two quantities:

$$\langle C_\alpha(r) \rangle_\beta^{(t)} = \frac{P(\alpha)^\beta P(i(r)|\alpha)^\beta \sum_\nu P(\nu|\alpha)^\beta P(j(r)|\nu)^\beta}{\sum_\alpha P(\alpha)^\beta P(i(r)|\alpha)^\beta \sum_\nu P(\nu|\alpha)^\beta P(j(r)|\nu)^\beta} \quad (5)$$

$$\langle T_{\alpha\nu}(r) \rangle_\beta^{(t)} = \frac{P(\alpha)^\beta P(i(r)|\alpha)^\beta P(\nu|\alpha)^\beta P(j(r)|\nu)^\beta}{\sum_\alpha P(\alpha)^\beta P(i(r)|\alpha)^\beta \sum_\nu P(\nu|\alpha)^\beta P(j(r)|\nu)^\beta} \quad (6)$$

also known as the E-step formula. Note that all probabilities on the right-hand sides of (5) and (6) are estimates at iteration t (not indicated for notational convenience). The iterated values for the model parameters are obtained by maximising $Q_\beta^{(t)}$ with respect to these parameters, leading to:

$$P^{(t+1)}(\alpha) = \frac{1}{L} \sum_r \langle C_\alpha(r) \rangle_\beta^{(t)} \quad P^{(t+1)}(i|\alpha) = \frac{\sum_r \langle C_\alpha(r) \rangle_\beta^{(t)} \mathbb{1}_{r,i(r)=i}}{\sum_r \langle C_\alpha(r) \rangle_\beta^{(t)}} \quad (7)$$

$$P^{(t+1)}(\nu|\alpha) = \frac{\sum_r \langle T_{\alpha\nu}(r) \rangle_\beta^{(t)}}{\sum_r \sum_\nu \langle T_{\alpha\nu}(r) \rangle_\beta^{(t)}} \quad P^{(t+1)}(j|\nu) = \frac{\sum_r \sum_\alpha \langle T_{\alpha\nu}(r) \rangle_\beta^{(t)} \mathbb{1}_{r,j(r)=j}}{\sum_r \sum_\alpha \langle T_{\alpha\nu}(r) \rangle_\beta^{(t)}} \quad (8)$$

also known as the M-step formulas. At each value of β , iterating the E-step and M-step formulas is guaranteed to converge to a local maximum.

Tempered EM has an additional advantage in connection to hierarchical clustering, as increasing the “temperature” parameter β provides a natural way to grow a hierarchy. Starting with $\beta = 0$, only one class exists. As β increases, classes start to differentiate by splitting in two or more (typically one class will split into two). In the statistical physics analogy, this is the result of “phase transitions” in the system. Ideally, we could track the generalisation abilities (using eg held-out data or cross-validation) while β increases, and retain the hierarchy which gives the lowest estimated generalisation error. However, we have not implemented this complete process currently, and we use the following annealing schedule instead. Starting from $\beta = 0.3$, we iterate the E-step and M-step with one class only. β is then increased by increments of 0.1 (until $\beta = 0.8$) and 0.05 (above 0.8). At each increment of β , we track phase transitions by duplicating the classes, perturbing them slightly and running the EM iterations. We then check whether class duplicates have diverged, and if so, replace the former class by the new ones, while if duplicates of a class haven’t diverged, we return to the original class. This process continues until a pre-specified number of classes is reached. Note that only the number of classes, not the hierarchy, is specified, such that the resulting hierarchy need not be a balanced tree.

In order to schematically represent the relationships between the parameters and the observed and latent variables, we present in figure 2 the graphical models corresponding to our clustering model (on the left). Note the difference with the HACM presented on the right: the generation of the document depends on the

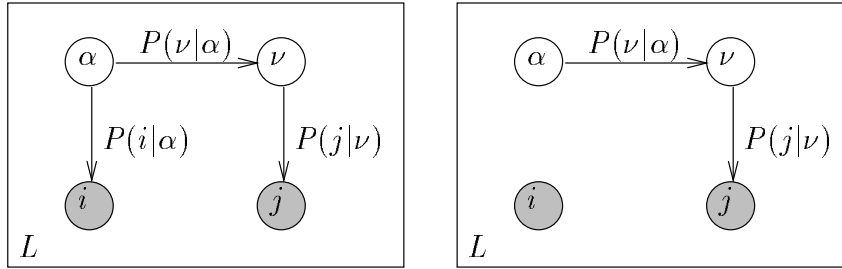


Fig. 2. Graphical models for the Hierarchical Probabilistic Latent Semantic Analysis, or HPLSA, model (left) discussed in this paper (equation 1) and the Hierarchical Asymmetric Clustering Model (HACM) from [10]. Gray-filled circles indicate observed variables.

class which introduces extra flexibility. If we force i to be independent of α , the rightmost downward arrow in our model disappears, which reduces to the HACM, as mentioned earlier.

5 Hierarchical document categorisation

We will now consider the supervised counterpart of the previous problem, where the goal is to categorise, ie assign labels to, incoming document, based on the information provided by a dataset of labelled documents. Each document $i(r)$ now has a category label $c(r)$,⁵ so the training data is $\mathcal{D} = \{(i(r), j(r), c(r), r)\}_{r=1 \dots L}$, recording the category label associated with each co-occurrence. The purpose of the categorisation task is to assign a category to a new document d expressed as a set of co-occurrences $(d, j(s), s)$, where s runs across the number L_d of co-occurrences in document d (or alternatively (d, j, n_{dj})).

We will further assume that each document category c corresponds to a single class α , such that $\forall r, \exists! \alpha, c(r) = \alpha$.

In the training phase, the model parameters are obtained again by maximising the likelihood. Note that the category/class labels actually give additional information, and the only remaining latent variable is the choice of the topic. For the full hierarchical model:

$$P(i, j, c = \alpha) = P(\alpha)P(i|\alpha) \sum_{\nu} P(\nu|\alpha)P(j|\nu) \quad (9)$$

The (log-)likelihood may be maximised analytically over both $P(\alpha)$ and $P(i|\alpha)$, leading to estimates which are actually the empirical training set frequencies. Noting $\#\alpha$ the number of examples with category $c(r) = \alpha$ in the training set (ie $\#\alpha = \#\{r|c(r) = \alpha\}$), and $\#(i, \alpha)$ the number of examples from

⁵ We assume that each example is assigned to a single category. Examples assigned to multiple categories are replicated such that each replication has only one.

document i and category $c = \alpha$,⁶ the maximum likelihood estimates are:

$$P(\alpha) = \frac{\#\alpha}{L} \quad \text{and} \quad P(i|\alpha) = \frac{\#(i, \alpha)}{\#\alpha} \quad (10)$$

Due to the presence of the sum on ν in (9), the remaining parameters cannot be obtained analytically⁷ and we again use an iterative (tempered) EM algorithm. The re-estimation formula are identical to (6) for the E-step and (8) for the M-step. The only difference with the unsupervised case is that the expressions of $P(\alpha)$ and $P(i|\alpha)$ are known and fixed from eq. 10. Note also that as the ML estimate for $P(i|\alpha)$ (eq. 10) is 0 for all documents i not in class α , the expression for the E-step (6) will simplify, and most notably, $\langle T_{\alpha\nu}(r) \rangle$ is 0 for all examples r not in class α . As a consequence, the re-estimation formulas are identical to those of the Hierarchical Mixture Model (equations (5-7) of [19]). The difference between both models lies in the additional use of $P(\alpha)$ and $P(i|\alpha)$ (10) in the training phase and our use of EM in the categorisation step (in addition to training, see below), as opposed to the approach presented in [19].

Categorisation of a new document is carried out based on the posterior class probability $P(\alpha|d) \propto P(d|\alpha)P(\alpha)$. The class probability $P(\alpha)$ are known from the training phase, and $P(d|\alpha)$ must again be estimated iteratively using EM to maximise the log-likelihood of the document to categorise:

$$\mathcal{L}(d) = \sum_s \log \left(\sum_{\alpha} P(\alpha)P(d|\alpha) \sum_{\nu} P(\nu|\alpha)P(j(s)|\nu) \right) \quad (11)$$

$\mathcal{L}(d)$ will be maximised wrt $P(d|\alpha)$ using the following EM steps, for $s = 1 \dots L_d$:

$$\text{E-step:} \quad \langle C_{\alpha}(s) \rangle^{(t)} = \frac{P(\alpha)P^{(t)}(d|\alpha) \sum_{\nu} P(\nu|\alpha)P(j(s)|\nu)}{\sum_{\alpha} P(\alpha)P^{(t)}(d|\alpha) \sum_{\nu} P(\nu|\alpha)P(j(s)|\nu)} \quad (12)$$

$$\text{M-step:} \quad P^{(t+1)}(d|\alpha) = \frac{\sum_s \langle C_{\alpha}(s) \rangle^{(t)}}{\#\alpha + \sum_s \langle C_{\alpha}(s) \rangle^{(t)}} \quad (13)$$

Notice that this is usually much faster than the training phase, as only $P(d|\alpha)$ (for all α) are calculated and all other probabilities are kept fixed to their training values. Once $P(d|\alpha)$ are obtained for all values of α , document d may be assigned to the class α_d for which the class posterior $P(\alpha|d)$ is maximised.

Note that as mentioned above, only $P(d|\alpha)$ is estimated, although clearly one may decide to benefit from the additional (unsupervised) information given by document d to update the class, topic and word distributions using a full EM. However we have decided not to implement this feature in order to classify all incoming documents on the same grounds.

For categorisation, the following tables present a summary of different models. In the tables, we indicate the model parameters, and whether parameter

⁶ As document i is assigned to one category only, $\#(i, \alpha)$ will be zero for all but one class α , for which it will be equal to the number of co-occurrences in document i .

⁷ For the flat model the ML estimate of $P(j|\alpha)$ is obtained directly as in (10).

estimation or categorisation of a new example is performed directly or using an EM-type iterative algorithm.

Flat models		
Naïve Bayes	Parameters	$P(j \alpha), P(\alpha)$
	Estimation	Direct
	Categorisation	Direct
PLSA[8]	Parameters	$P(j \alpha), P(i \alpha), P(\alpha)$
	Estimation	EM
	Categorisation	EM
PLC	Parameters	$P(j \alpha), P(i \alpha), P(\alpha)$
	Estimation	Direct
	Categorisation	EM for $P(d \alpha)$

Hierarchical models		
Hierarchical Naïve Bayes ⁸	Parameters	$P(j \nu), P(\nu \alpha), P(\alpha)$
	Estimation	EM
	Categorisation	Direct
HACM[10]	Parameters	$P(j \nu), P(\nu \alpha), P(i), P(\alpha)$
	Estimation	EM
	Categorisation	Direct
HPLC	Parameters	$P(j \nu), P(\nu \alpha), P(i \alpha), P(\alpha)$
	Estimation	EM
	Categorisation	EM for $P(d \alpha)$

where PLC stands for Probabilistic Latent Categorisation and HPLC for Hierarchical PLC. Notice that by using class labels efficiently, PLC estimates its parameters directly. In contrast standard PLSA implementations such as [19] apply a standard EM for both estimation and categorisation.

6 Experiments

6.1 Clustering

For evaluating our model in the context of document clustering, we used as a test collection the labeled documents in TDT-1. TDT-1 is a collection of documents provided by the Linguistic Data Consortium for the Topic Detection and Tracking project, consisting of news articles from CNN and Reuters, and covering various topics such as Oklahoma City Bombing, Kobe earthquake, etc.

The labeled portion of the TDT-1 collection was manually labeled with the main topic of each document as one of twenty-two topics. We randomly selected a subset of sixteen topics with the corresponding documents, obtaining 700 documents with 12700 different words. We then removed the labels from

⁸ This is valid for several variants related to the hierarchical structure and/or estimation procedure [1, 14, 19].

the documents and clustered the obtained collection with our model. For easy comparison, we used a binary tree with 16 leaves as hierarchy (corresponding to 15 cluster splits in tempered EM), where documents are assigned to the leaves of the induced hierarchy. We followed the same methodology using a version of Hierarchical Shrinkage described in [14, 1], which we will refer to as HS, and PLSA, a flat model we retained to test the influence of the induced hierarchy (note that the original document collection is not hierarchically organized).

To measure the adequacy between obtained clusters and manual labels, we used the average, over the labels and over the clusters, of the Gini function, defined as:

$$G_l = \frac{1}{L} \sum_l \sum_{\alpha} \sum_{\alpha' \neq \alpha} P(\alpha|l)P(\alpha'|l)$$

$$G_{\alpha} = \frac{1}{A} \sum_{\alpha} \sum_l \sum_{l' \neq l} P(l|\alpha)P(l'|\alpha)$$

where L is the number of different labels and A the number of different clusters ($L = A = 16$ here). G_l measures the impurity of the obtained clusters α with respect to the labels l , and reciprocally for G_{α} . Smaller values of these functions indicate better results since clusters and labels are in closer correspondence, ie if the “data clusters” and “label clusters” contain the same documents with the same weights, the Gini index is 0. Furthermore, these functions have an upper bound of 1. Our choice for the Gini index was motivated by the fact that it is more fine-grained, and therefore more informative, than the direct misclassification cost in many instances.

The results we obtained are summarized in the following table:

	G_l	G_{α}
PLSA	0.34	0.30
HS	0.40	0.45
HPLSA	0.20	0.16

We thus see that PLSA, which is a flat model, ranks between the two hierarchical models we tested. HS, which results in most cases in a hard assignment of documents to clusters, is significantly worse than the other two models, which result in a soft assignment. Furthermore, our hierarchical model significantly outperforms the flat model. This last result is interesting since we assigned documents to the leaves of the hierarchy only, so, except for the words, the documents are assigned in the same manner as in a flat model. However, by making use of the hierarchy for words, we have better results than a flat model for both words and documents (even though we do not display the results here, experiments, eg [10], show that a standard approach to flat clustering, like K-means, yields results similar to, if not worse than, PLSA).

As an example, we present the hierarchy we obtained on the 273 documents related to Oklahoma City Bombing. These documents contain 7684 different non-empty words (empty words, such as determiners, prepositions, etc., were removed

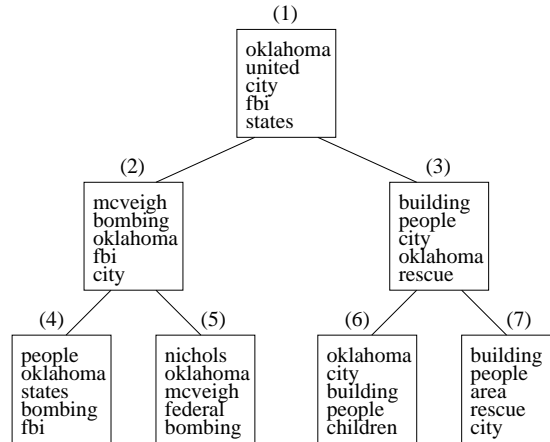


Fig. 3. Cluster hierarchy for Oklahoma City Bombing

using a stop list). An experiment, conducted with a flat model on the same data, revealed that the documents were best explained with three topics/clusters. We have chosen to cluster the data with a binary tree containing 4 leaves and 7 nodes. Figure 3 shows the hierarchy obtained with our model. For each node, we provide the first five words associated with the node, ie the five words for which $p(j|\nu)$ is the highest.

We can see that the data is first separated into two topics/clusters, respectively related to the investigation and the description of the event itself (nodes (2) and (3)). Node (2) is then divided into two parts, the investigation itself (node (4)) and the trial (node (5)), whereas node (3) is split between the description of the bombing and casualties (node (6)) and the work of the rescue teams (node (7)). Note also that despite our use of upper nodes to describe a given topic (through $P(\nu|\alpha)$ and $P(j|\nu)$), certain words, eg *Oklahoma*, appear in different nodes of the hierarchy. This is due to the fact that these words appear frequently in all documents. Our data is thus best explained by assigning these words to different topics/clusters.

6.2 Categorisation

In order to illustrate the behaviour of the hierarchical categorisation method, we address the task of categorising messages from 15 different newsgroups organised in a simple two-level hierarchy ([19] and figure 4). The data is taken from the benchmark “20 newsgroups” dataset.⁹ The only pre-processing we perform on the data is that all tokens that do not contain at least one alphabetic character are removed. In particular, we do not filter out words or tokens with very low frequency.

⁹ http://www.ai.mit.edu/~jrennie/20_newsgroups/

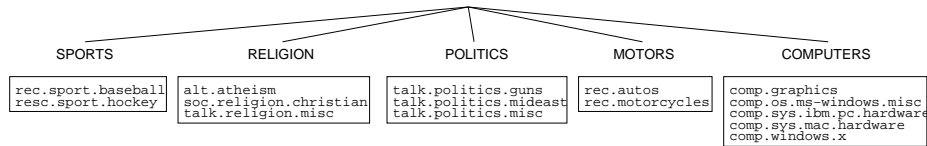


Fig. 4. The newsgroup hierarchy. There are 15 newsgroups at the leaves, organised with 5 mid-level topics and a root topic.

We consider several sample sizes, using from 7 to 134 messages per class for training, and 3 to 66 messages for testing, such as the total amount of data varies from 10 to 200. For each sample size, we average results over ten replications of the same size.

We consider 4 categorisation methods:

- Naïve Bayes, with smoothed¹⁰ class-conditional word distributions;
- Probabilistic Latent Categorisation (PLC) with 15 classes, no hierarchy;
- Hierarchical Mixture [19] with uniform conditional topic distributions $P(\nu|\alpha)$;
- Hierarchical PLC where the conditional topic distributions are estimated using maximum likelihood implemented with EM (HPLC).

The choice of uniform class-conditional topic distributions is due to a tendency to overfit, especially with small samples. This will be discussed in more details below.

Results are displayed in figure 5. Somewhat surprisingly, Naïve Bayes performs quite badly on our data, with some 10 to 20 percent lower classification rate compared to results published in [19] on similar data (as each document belongs to and is assigned a single category, the micro-averaged $F1$ reported in table 5 is equal to the percentage of correct class assignment). We blame the preprocessing for this somewhat dismal performance: as we retain all tokens which contain at least one alphabetic character, we introduce a high level of noise in the estimation of the topic-conditional word distributions by including many words which occur only once or twice in the entire collection.

The performance of both the flat and hierarchical PLC models with our noisy data is close to that reported by [19] for their hierarchical mixture model with cleaner pre-processed data. Although it uses a hierarchy to constrain the classes, HiMi doesn't manage to outperform the non-hierarchical PLC in our experiments. This is due to 2 factors. Firstly, HiMi is a hierarchical version of Naïve Bayes, such that the increased noise level mentioned above will also hurt HiMi significantly. Secondly, the estimates of the class-conditional topic distributions are different here and in [19]. As the ML estimates of $P(\nu|\alpha)$ will tend to overfit, we need to somehow address this problem. Typical solutions

¹⁰ We used a particular version of Lidstone smoothing [13] with $\lambda = 0.5$, which corresponds to the MAP estimate for a multinomial with a non-informative Dirichlet(1/2) prior on the parameters.

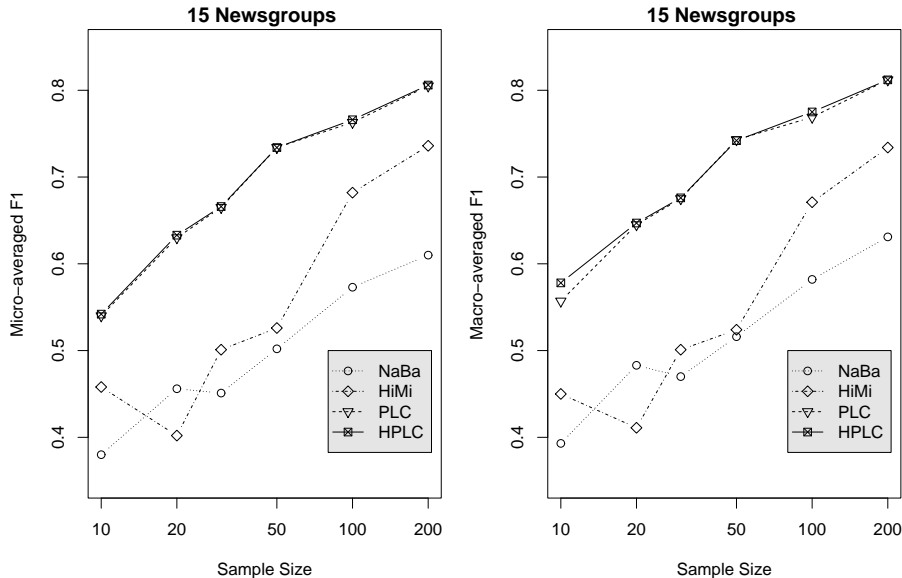


Fig. 5. Categorisation results for Naïve Bayes and PLC (both flat models) and for hierarchical mixture (HiMi) using uniform class-conditional topic distributions and hierarchical PLC (HPLC) using maximum likelihood class-conditional topic distributions trained by EM. The plots give the micro-averaged (left) and macro-averaged (right) $F1$ for sample sizes ranging from 10 to 200.

include using held-out data to estimate $P(\nu|\alpha)$ in EM, or using a resampling strategy [14, 19] to leverage available data more efficiently. We have chosen here to use a uniform distribution on topics instead, meaning that words can be sampled in equal parts from either the class or one of its ancestors. Although sub-optimal, this solution performs much better than the degenerate ML solution, which is equivalent to (flat) Naïve Bayes.

HPLC also suffers from this overfitting problem for small sample sizes, which explains why the performance of HPLC in table 5 is not significantly better than PLC’s, suggesting that the hierarchy is not used to its full extent. In the case of HPLC, using a uniform distribution on the class-conditional topic distributions does not improve (or harm) performance. Although there is clearly room for improvement here, we take this as an indication that HPLC is not overly sensitive to these parameters, in particular when overfitting is concerned.

7 Discussion

Other EM variants. The co-occurrence model discussed here is a mixture model and relies on the EM algorithm for estimating the maximum likelihood parameters in cases where no analytical solution exist. Problems with the EM algorithm are the stability of the obtained solution and the convergence speed. The

use of deterministic annealing [20, 8] addresses the first problem in a satisfactory way, at the expense of the second, as typically more iterations will be needed during the annealing procedure than for a typical standard EM run. Although the gain in stability of the solution partially offsets this problem, by requiring only a single EM run, training times can still be problematic for large scale problems. With Gaussian mixture models, several strategies have been proposed, such as classification EM (CEM) or stochastic EM (SEM) [2]. Classification EM consists in assigning at every iteration each example r to the most likely class and topic, ie the α and ν with the highest values of $\langle C_\alpha(r) \rangle_\beta$ and $\langle T_{\alpha\nu}(r) \rangle_\beta$. The well-known K-means algorithm is an instance of CEM in the context of a uniform mixture of isotropic Gaussian distributions. CEM has been observed to operate quite fast, usually reaching a stable partition of the data in very few iterations. Similarly, stochastic EM assigns examples to classes and topics at random with probabilities given by $\langle C_\alpha(r) \rangle_\beta$ and $\langle T_{\alpha\nu}(r) \rangle_\beta$, rather than averaging over those values as standard EM does.

Fisher Kernels. Probabilistic models may be used to derive a model-based measure of similarity between examples, using the so-called Fisher kernel [11, 9]. Fisher kernels are defined using the gradient (wrt parameters) of the log-likelihood of a document d , $\nabla_\theta \ell(d)$, aka Fisher score, and the Fisher information matrix. The Fisher information matrix is $\mathbf{I} = \mathbf{E} \left(\nabla_\theta \ell(d) \nabla_\theta \ell(d)^\top \right)$, where the expectation is taken over $P(d|\theta)$. With this notation, the similarity between two documents d_i and d_n induced by the Fisher kernel is:

$$k(d_i, d_n) = \nabla_\theta \ell(d_i)^\top \mathbf{I}^{-1} \nabla_\theta \ell(d_n) \approx \nabla_\theta \ell(d_i)^\top \nabla_\theta \ell(d_n) \quad (14)$$

using the standard approximation that the Fisher information \mathbf{I} is approximately the unit matrix. Notice that the original expression (including \mathbf{I}) is independent of the parameterisation. However, this is clearly not true for the simplified expression. It is therefore very relevant to use the parameterisation for which the approximation is best. In particular, we will here follow [9] and use a square root parameterisation for the multinomial parameters in our model.

Previous works [11, 9, 22] have shown how to make efficient use of unlabelled data into discriminative classifiers using Fisher kernels. Even though we have not experimented with Fisher kernels, we want to give here their forms for the different models, since we believe it sheds some light on the models.

For HPLSA, the likelihood of a document d , normalised by document length, is:

$$\ell(d) = \sum_j \widehat{P}(j|d) \log \left(\sum_\alpha P(\alpha) P(d|\alpha) \sum_\nu P(\nu|\alpha) P(j|\nu) \right) \quad (15)$$

where $\widehat{P}(j|d)$ is the empirical word frequency for word j in document d . The relevant derivatives are:

$$\frac{\partial \ell(d)}{\partial P(\alpha)} = \frac{P(\alpha|d)}{P(\alpha)} \sum_j \frac{\widehat{P}(j|d)}{P(j|d)} P(j|\alpha) \approx \frac{P(\alpha|d)}{P(\alpha)} \quad (16)$$

$$\frac{\partial \ell(d)}{\partial P(\nu|\alpha)} = P(\alpha|d) \sum_j \frac{\hat{P}(j|d)}{P(j|d)} P(j|\nu) \approx P(\alpha|d) \quad (17)$$

$$\frac{\partial \ell(d)}{\partial P(j|\nu)} = \frac{\hat{P}(j|d) P(\nu|d, j)}{P(j|d)} \quad (18)$$

where we have used the approximation $\hat{P}(j|d) \approx P(j|d)$. Using this and the square root parameterisation, the similarity between two documents d_i and d_n is evaluated by Fisher kernel as the following expression:

$$k(d_i, d_n) = k_1(d_i, d_n) + k_2(d_i, d_n) + k_3(d_i, d_n)$$

$$k_1(d_i, d_n) = \sum_{\alpha} \frac{P(\alpha|d_i) P(\alpha|d_n)}{P(\alpha)} \quad (19)$$

$$k_2(d_i, d_n) = \sum_j \hat{P}(j|d_i) \hat{P}(j|d_n) \sum_{\nu} \frac{P(\nu|j, d_i) P(\nu|j, d_n)}{P(j|\nu)} \quad (20)$$

$$k_3(d_i, d_n) = \sum_{\alpha} P(\alpha|d_i) P(\alpha|d_n) \quad (21)$$

Contributions k_1 and k_3 can be summed up in a single contribution, involving the (weighted) inner product of the document mixing weights. This captures, up to a certain point, synonymy between words. Contribution k_2 performs a weighted inner product between the empirical distributions of words in the document, with a weight depending on whether the words have similar conditional topic distributions over the whole hierarchy. This distinguishes words used with different meanings in different contexts (polysems). Similar contributions can also be found in the kernel expressions of PLSA [9] and in [22].

Interestingly, under the above assumption, the contribution from parameters $P(\nu|\alpha)$ vanishes in the kernel function. This suggests that the values obtained for these parameters do not play a direct role in computing the similarity between documents, or, in other words, if the other parameters are fixed, varying the parameters $P(\nu|\alpha)$ does not impact the similarity function. The same is not true for hierarchical versions of Naïve Bayes, since the partial derivative of the log-likelihood function with respect to the $p(\nu|\alpha)$ parameters is given by (for brevity, we omit the details of the derivation):

$$\frac{\partial \ell(d)}{\partial P(\nu|\alpha)} = P(\alpha|d) \sum_j \hat{P}(j|d) \frac{P(j|\nu)}{P(j|\alpha)} \quad (22)$$

leading to the following contribution to the kernel function:

$$k_3^{NB}(d_i, d_n) = \sum_{\alpha} P(\alpha|d_i) P(\alpha|d_n) \times$$

$$\sum_{\nu} \left(\sum_j \hat{P}(j|d_i) \frac{P(j|\nu)}{P(j|\alpha)} \right) \left(\sum_j \hat{P}(j|d_n) \frac{P(j|\nu)}{P(j|\alpha)} \right) \quad (23)$$

Thus, in this case, varying the parameters $P(\nu|\alpha)$ yields different similarities.

Although a more complete study of the relations between the Fisher kernel and the Maximum Likelihood estimates is needed, we believe this explains our observation that HPLC does not seem overly sensitive to the parameters $P(\nu|\alpha)$, whereas HiMi (a hierarchical version of Naïve Bayes) is. Future work should focus on working out the exact conclusions one can draw from the above facts.

Model selection. One of the objectives of this paper was to propose a very flexible model (1) which encompasses a number of previously proposed methods [8, 14, 19, 24]. In addition to shedding some light on the assumptions built in the different models, this opens a possibility for using standard tools for performing model selection. Indeed, we have shown that we have a family of embedded models of increasing complexity, where we could select a model, typically using arguments based on generalisation relying on eg algebraic estimators or cross-validation. Note however that typically we would expect more flexible models to perform worse for small sample sizes. The results we have reported suggest that the most flexible model (HPLC) actually does quite well even for small sample sizes.

8 Conclusion

In this paper, we proposed a unifying model for hierarchical clustering and categorisation of co-occurrence data, with particular application to organising documents. This generative mixture model encompasses several other models, both hierarchical and flat, already proposed in the literature. We gave details on how to perform parameter estimation, either using a tempered version of the EM algorithm, or using direct formula when applicable.

The use of this model is illustrated on two task: clustering incoming news articles and categorising newsgroup messages. In both cases, the proposed model was compared to competing alternatives. Although there is certainly room for improvement on several fronts, we note that the combined use of the hierarchy and soft class assignment yields performance that is at least as good (and in most cases better) as previously proposed generative models. We also presented the Fisher kernels associated with the model and discussed how they relate to its behaviour.

Acknowledgements

We wish to thank Nicola Cancedda and Jean-Michel Renders for useful discussions on several aspects of this paper, and the anonymous reviewers for their helpful and constructive comments.

References

1. L. Douglas Baker, Thomas Hofmann, Andrew McCallum, and Yiming Yang. A hierarchical probabilistic model for novelty detection in text. <http://www-2.cs.cmu.edu/mccallum/papers/tdt-nips99s.ps.gz>.
2. Gilles Celeux and Gérard Govaert. A Classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, 14:315–332, 1992.
3. 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(6):391–407, 1990.
4. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
5. E. Gaussier and N. Cancedda. Probabilistic models for terminology extraction and knowledge structuring from documents. In *Proceedings of the 2001 IEEE International Conference on Systems, Man & Cybernetics*, 2001.
6. G. Grefenstette. *Explorations in Automatic Thesaurus Construction*. Kluwer Academic Publishers, 1994.
7. Z. S. Harris. Distributional structure. *Word*, 10:146–162, 1954.
8. Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 289–296. Morgan Kaufmann, 1999. <http://www2.sis.pitt.edu/dsl/UAI/uai99.html>.
9. Thomas Hofmann. Learning the similarity of documents: An information-geometric approach to document retrieval and categorization. In *Advances in Neural Information Processing Systems 12*, page 914. MIT Press, 2000.
10. Thomas Hofmann and Jan Puzicha. Statistical models for co-occurrence data. A.I. Memo 1625, A.I. Laboratory, February 1998.
11. Tommi S. Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems 11*, pages 487–493, 1999.
12. Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning (ECML98)*, number 1398 in Lecture Notes in Computer Science, pages 137–142. Springer Verlag, 1998.
13. Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.
14. Andrew McCallum, Ronald Rosenfeld, Tom Mitchell, and Andrew Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 359–367, 1998.
15. Fernando Pereira, Naftali Tishby, and Lillian Lee. Distributional clustering of english words. In *Proceedings of the International Conference of the Association for Computational Linguistics*, 1993.
16. K. Rose, E. Gurewitz, and G. Fox. Statistical mechanics and phase transitions in clustering. *Physical Review Letters*, 65(8):945–848, 1990.
17. G. Salton. *Automatic Thesaurus Construction for Information Retrieval*. North Holland Publishing, 1972.
18. D. M. Titterton, A. F. Smith, and U. E. Makov. *Statistical Analysis of Finite Mixture Distribution*. John Wiley & Sons, San Diego, 1985.

19. Kristina Toutanova, Francine Chen, Kris Popat, and Thomas Hofmann. Text classification in a hierarchical mixture model for small training sets. In *Proceedings of the ACM Conference on Information and Knowledge Management*, 2001.
20. Naonori Ueda and Ryohei Nakano. Deterministic annealing variant of the EM algorithm. In Gerry Tesauro, David Touretzky, and Todd Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 545–552. MIT Press, 1995.
21. C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 2nd edition edition, 1979.
22. Alexei Vinokourov and Mark Girolami. A probabilistic framework for the hierarchical organisation and classification of document collections. *Journal of Intelligent Information Systems*, 18(2–3):153–172, 2002.
23. Peter Willett. Recent trends in hierarchical document clustering: A critical review. *Information Processing & Management*, 24(5):577–597, 1988.
24. Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49, 1999.