Machine Learning, 42, 177–196, 2001

Unsupervised Learning by Probabilistic Latent Semantic Analysis

THOMAS HOFMANN th@cs.brown.edu
Department of Computer Science, Brown University, Providence, RI 02912, USA

Editor: Douglas Fisher

Abstract. This paper presents a novel statistical method for factor analysis of binary and count data which is closely related to a technique known as Latent Semantic Analysis. In contrast to the latter method which stems from linear algebra and performs a Singular Value Decomposition of co-occurrence tables, the proposed technique uses a generative latent class model to perform a probabilistic mixture decomposition. This results in a more principled approach with a solid foundation in statistical inference. More precisely, we propose to make use of a temperature controlled version of the Expectation Maximization algorithm for model fitting, which has shown excellent performance in practice. Probabilistic Latent Semantic Analysis has many applications, most prominently in information retrieval, natural language processing, machine learning from text, and in related areas. The paper presents perplexity results for different types of text and linguistic data collections and discusses an application in automated document indexing. The experiments indicate substantial and consistent improvements of the probabilistic method over standard Latent Semantic Analysis.

Keywords: unsupervised learning, latent class models, mixture models, dimension reduction, EM algorithm, information retrieval, natural language processing, language modeling

1. Introduction

The development of algorithms that enable computers to automatically process text and natural language has always been one of the great challenges in Artificial Intelligence. In recent years, this research direction has increasingly gained importance, last not least due to the advent of the World Wide Web, which has amplified the need for intelligent text and language processing. The demand for computer systems that manage, filter and search through huge repositories of text documents has created a whole new industry, as has the demand for smart and personalized interfaces. Consequently, any substantial progress in this domain will have a strong impact on numerous applications ranging from information retrieval, information filtering, and intelligent agents, to speech recognition, machine translation, and human-machine interaction.

There are two schools of thought: On one side, there is the traditional linguistics school, which assumes that linguistic theory and logic can instruct computers to "learn" a language. On the other side, there is a statistically-oriented community, which believes that machines can learn (about) natural language from training data such as document collections and text corpora. This paper follows the latter approach and presents a novel method for learning

the *meaning* of words in a purely data-driven fashion. The proposed unsupervised learning technique called *Probabilistic Latent Semantic Analysis* (PLSA) aims at identifying and distinguishing between different *contexts of word usage* without recourse to a dictionary or thesaurus. This has at least two important implications: Firstly, it allows us to disambiguate *polysems*, i.e., words with multiple meanings, and essentially every word is polysemous. Secondly, it reveals topical similarities by grouping together words that are part of a common context. As a special case this includes *synonyms*, i.e., words with identical or almost identical meaning.

As the name PLSA indicates, our approach has been largely inspired and influenced by *Latent Semantic Analysis* (LSA) (Deerwester et al., 1990), although there are also notable differences. The key idea in LSA is to map high-dimensional count vectors, such as term-frequency (tf) vectors arising in the vector space representation of text documents (Salton & McGill, 1983), to a lower dimensional representation in a so-called *latent semantic space*. In doing so, LSA aims at finding a data mapping which provides information beyond the lexical level of word occurrences. The ultimate goal is to represent semantic relations between words and/or documents in terms of their proximity in the semantic space. Due to its generality, LSA has proven to be a valuable analysis tool for many different problems in practice and thus has a wide range of possible applications (e.g., Deerwester et al., 1990; Foltz & Dumais, 1992; Landauer & Dumais, 1997; Wolfe et al., 1998; Bellegarda, 1998).

Despite its success, there are a number of shortcomings of LSA. First of all, the methodological foundation remains to a large extent unsatisfactory and incomplete. The original motivation for LSA stems from linear algebra and is based on a L_2-optimal approximation of matrices of word counts based on a *Singular Value Decomposition* (SVD) (Berry, Dumais, & Obrien, 1995). While SVD by itself is a well-understood and principled method (Golub & Van Loan, 1996), its application to count data in LSA remains somewhat *ad hoc*. From a statistical point of view, the utilization of a L_2-norm approximation principle is reminiscent of a Gaussian noise assumption which is hard to justify in the context of count variables. On a deeper, conceptual level the representation obtained by LSA is unable to handle polysemy. For example, it is easy to show that in LSA the coordinates of a word in the latent space can be written as a linear superposition of the coordinates of the documents that contain the word. The superposition principle, however, is unable to explicitly capture multiple senses of a word, nor does it take into account that every word occurrence is typically intended to refer to only one meaning at a time.

Probabilistic Latent Semantics Analysis (PLSA) stems from a statistical view of LSA. In contrast to standard LSA, PLSA defines a proper generative data model. This has several advantages: On the most general level it implies that standard techniques from statistics can be applied for model fitting, model selection and complexity control. For example, one can assess the quality of a PLSA model by measuring its predictive performance, e.g., with the help of cross-validation. More specifically, PLSA associates a latent context variable with each word occurrence, which explicitly accounts for polysemy. A more technical discussion of the differences between LSA and PLSA can be found in Section 3.3.

2. Latent semantic analysis

2.1. Count data and co-occurrence tables

LSA can be applied to any type of count data over a discrete dyadic domain, so-called *two-mode data* (Hofmann, Puzicha & Jordan, 1999). Yet, since the most prominent application of LSA is in the analysis and retrieval of text documents, we focus on this setting. Suppose therefore that we have given a collection of text documents $\mathcal{D} = \{d_1, \ldots, d_N\}$ with terms from a vocabulary $\mathcal{W} = \{w_1, \ldots, w_M\}$. By ignoring the sequential order in which words occur in a document, one may summarize the data in a rectangular $N \times M$ *co–occurrence table* of counts $\mathbf{N} = \big(n(d_i, w_j)\big)_{ij}$, where $n(d_i, w_j)$ denotes the number of times the term w_j occurred in document d_i. In this particular case, $\mathbf{N}$ is also called the term-document matrix and the rows/columns of $\mathbf{N}$ are referred to as document/term vectors, respectively. The key assumption is that the simplified 'bag-of-words' or vector-space representation (Salton & McGill, 1983) of documents will in many cases preserve most of the relevant information, e.g., for tasks like text retrieval based on keywords.

The co-occurrence table representation immediately reveals the problem of *data sparseness* (Katz, 1987), also known as the *zero-frequency problem* (Witten & Bell, 1991). A typical term-document matrix derived from short articles, text summaries or abstracts may only have a small fraction of non-zero entries (typically well below 1%), which reflects the fact that only very few of the words in the vocabulary are actually used in any single document. This has consequences, for example, in applications that are based on matching queries with documents or evaluating similarities between documents by comparing common terms. The likelihood to find many common terms even in closely related articles may be small, just because they might not use *exactly* the same terms.

For example, most of the matching functions utilized in this context are based on similarity functions that rely on inner products between pairs of document vectors. The encountered problems are two-fold: On one hand, one has to account for synonyms in order not to underestimate the true similarity of documents. On the other hand, one has to deal with polysems to avoid overestimating the true similarity between documents by counting common terms that are used in different meanings. Both problems may lead to inappropriate lexical matching scores which may not reflect the 'true' similarity hidden in the semantics of words.

2.2. Latent semantic analysis by singular value decomposition

As mentioned in the introduction, the key idea of LSA is to map documents—and by symmetry terms—to a vector space of reduced dimensionality, the *latent semantic space*, which in a typical application in document indexing is chosen to have of the order $\approx 100-300$ dimensions (Deerwester et al., 1990; Dumais, 1995). The mapping of the given document/term vectors to its latent space representatives is restricted to be linear and is based on a decomposition of the co-occurrence matrix by SVD. One thus starts with the standard SVD given by

$$\mathbf{N} = \mathbf{U}\Sigma\mathbf{V}^t, \tag{1}$$

where $\mathbf{U}$ and $\mathbf{V}$ are matrices with orthonormal columns $\mathbf{U}^t\mathbf{U} = \mathbf{V}^t\mathbf{V} = \mathbf{I}$ and the diagonal matrix Σ contains the singular values of $\mathbf{N}$. The LSA approximation of $\mathbf{N}$ is computed by thresholding all but the largest K singular values in Σ to zero ($=\tilde{\Sigma}$), which is rank K optimal in the sense of the L_2-matrix or Frobenius norm as is well-known from linear algebra, i.e., one obtains the approximation

$$\tilde{\mathbf{N}} = \mathbf{U}\tilde{\Sigma}\mathbf{V}^t \approx \mathbf{U}\Sigma\mathbf{V}^t = \mathbf{N}. \tag{2}$$

Notice that if we want to compute the document-to-document inner products based on (2), we would get $\tilde{\mathbf{N}}\tilde{\mathbf{N}}^t = \mathbf{U}\tilde{\Sigma}^2\mathbf{U}^t$ and hence one might think of the rows of $\mathbf{U}\tilde{\Sigma}$ as defining coordinates for documents in the latent space. While the original high-dimensional vectors are sparse, the corresponding low-dimensional latent vectors will typically not be sparse. This implies that it is possible to compute meaningful association values between pairs of documents, even if the documents do not have any terms in common. The hope is that terms having a common meaning are roughly mapped to the same direction in the latent space.

3. Probabilistic latent semantic analysis

3.1. The aspect model

The starting point for our novel *Probabilistic Latent Semantic Analysis* is a statistical model which has been called the *aspect model* (Hofmann, Puzicha, & Jordan, 1999). The aspect model has independently been proposed by Saul and Peveira (1997) in the context of language modeling, where it is referred to as *aggregate Markov model*. In the statistical literature similar models have been discussed for the analysis of contingency tables (cf. Gilula & Haberman, 1986). Another closely related technique called non-negative matrix decomposition has been investigated in Lee and Seung (1999).

The aspect model is a latent variable model for co-occurrence data which associates an unobserved class variable $z_k \in \{z_1, \ldots, z_K\}$ with each observation, an observation being the occurrence of a word in a particular document. Let us introduce the following probabilities: $P(d_i)$ is used to denote the probability that a word occurrence will be observed in a particular document d_i, $P(w_j \mid z_k)$ denotes the class-conditional probability of a specific word conditioned on the unobserved class variable z_k, and finally $P(z_k \mid d_i)$ denotes a document-specific probability distribution over the latent variable space. Using these definitions, one may define a generative model for word/document co-occurrences by the following scheme:

1. select a document d_i with probability $P(d_i)$,
2. pick a latent class z_k with probability $P(z_k \mid d_i)$,
3. generate a word w_j with probability $P(w_j \mid z_k)$.

As a result one obtains an observation pair (d_i, w_j), while the latent class variable z_k is discarded. Translating the data generation process into a joint probability model results in the expression

$$P(d_i, w_j) = P(d_i)P(w_j \mid d_i), \quad P(w_j \mid d_i) = \sum_{k=1}^{K} P(w_j \mid z_k)P(z_k \mid d_i). \tag{3}$$

Essentially, to obtain (3) one has to sum over the possible choices of z_k by which an observation could have been generated. Like virtually all statistical latent variable models the aspect model introduces a conditional independence assumption, namely that d_i and w_j are independent conditioned on the state of the associated latent variable. A very intuitive interpretation for the aspect model can be obtained by a closer examination of the conditional distributions $P(w_j \mid d_i)$ which are seen to be convex combinations of the K class-conditionals or *aspects* $P(w_j \mid z_k)$. Loosely speaking, the modeling goal is to identify conditional probability mass functions $P(w_j \mid z_k)$ such that the document-specific word distributions are as faithfully as possible approximated by convex combinations of these aspects. More formally, one can use a maximum likelihood formulation of the learning problem, i.e., one has to maximize

$$\begin{aligned} \mathcal{L} &= \sum_{i=1}^{N} \sum_{j=1}^{M} n(d_i, w_j) \log P(d_i, w_j) \\ &= \sum_{i=1}^{N} n(d_i) \left[\log P(d_i) + \sum_{j=1}^{M} \frac{n(d_i, w_j)}{n(d_i)} \log \sum_{k=1}^{K} P(w_j \mid z_k) P(z_k \mid d_i) \right], \end{aligned} \tag{4}$$

with respect to all probability mass functions. Here, $n(d_i) = \sum_j n(d_i, w_j)$ refers to the document length. A representation of the aspect model in terms of a graphical model is depicted in Figure 1(a). Since the cardinality of the latent variables is typically smaller than the number of documents (and terms) in the collection, $K \ll \min\{N, M\}$, it acts as a bottleneck variable in predicting words.

It is worth noticing that an equivalent parameterization of the model can be obtained by reversing the arc between D and Z in the graphical model representation (cf. Figure 1(a) and (b)) resulting in the equivalent parameterization of the joint probability in (3) by

$$P(d_i, w_j) = \sum_{k=1}^{K} P(z_k) P(d_i \mid z_k) P(w_j \mid z_k), \tag{5}$$

which is perfectly symmetric in both entities, documents and words.

3.2. Model fitting with the EM algorithm

The standard procedure for maximum likelihood estimation in latent variable models is the Expectation Maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977). EM

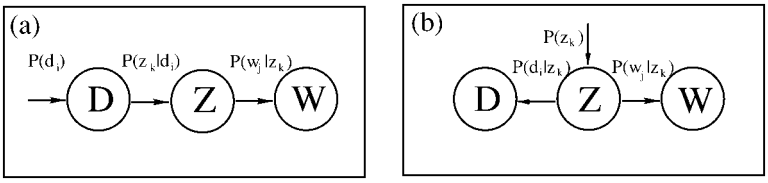

Figure 1. Graphical model representation of the aspect model in the asymmetric (a) and symmetric (b) parameterization.

alternates two steps: (i) an expectation (E) step where posterior probabilities are computed for the latent variables, based on the current estimates of the parameters, (ii) a maximization (M) step, where parameters are updated based on the so-called expected complete data log–likelihood which depends on the posterior probabilities computed in the E-step.

For the E-step one simply applies Bayes' formula, e.g., in the parameterization of (3), to obtain

$$P(z_k \mid d_i, w_j) = \frac{P(w_j \mid z_k) P(z_k \mid d_i)}{\sum_{l=1}^{K} P(w_j \mid z_l) P(z_l \mid d_i).} \tag{6}$$

In the M-step one has to maximize the expected complete data log-likelihood $\mathbf{E}[\mathcal{L}^c]$. Since the trivial estimate $P(d_i) \propto n(d_i)$ can be carried out independently, the relevant part is given by

$$\mathbf{E}\left[\mathcal{L}^c\right] = \sum_{i=1}^{N} \sum_{j=1}^{M} n(d_i, w_j) \sum_{k=1}^{K} P(z_k \mid d_i, w_j) \log\left[P(w_j \mid z_k) P(z_k \mid d_i)\right]. \tag{7}$$

In order to take care of the normalization constraints, (7) has to be augmented by appropriate Lagrange multipliers τ_k and ρ_i,

$$\mathcal{H} = \mathbf{E}\left[\mathcal{L}^c\right] + \sum_{k=1}^{K} \tau_k \left(1 - \sum_{j=1}^{M} P(w_j \mid z_k)\right) + \sum_{i=1}^{N} \rho_i \left(1 - \sum_{k=1}^{K} P(z_k \mid d_i)\right). \tag{8}$$

Maximization of $\mathcal{H}$ with respect to the probability mass functions leads to the following set of stationary equations

$$\sum_{i=1}^{N} n(d_i, w_j) P(z_k \mid d_i, w_j) - \tau_k P(w_j \mid z_k) = 0, \quad 1 \le j \le M, \quad 1 \le k \le K, \tag{9}$$

$$\sum_{j=1}^{M} n(d_i, w_j) P(z_k \mid d_i, w_j) - \rho_i P(z_k \mid d_i) = 0, \quad 1 \le i \le N, \quad 1 \le k \le K. \tag{10}$$

After eliminating the Lagrange multipliers one obtains the M-step re-estimation equations

$$P(w_j \mid z_k) = \frac{\sum_{i=1}^{N} n(d_i, w_j) P(z_k \mid d_i, w_j)}{\sum_{m=1}^{M} \sum_{i=1}^{N} n(d_i, w_m) P(z_k \mid d_i, w_m)}, \tag{11}$$

$$P(z_k \mid d_i) = \frac{\sum_{j=1}^{M} n(d_i, w_j) P(z_k \mid d_i, w_j)}{n(d_i)}. \tag{12}$$

The E-step and the M-step equations are alternated until a termination condition is met. This can be a convergence condition, but one may also use a technique known as *early stopping*.

In early stopping one does not necessarily optimize until convergence, but instead stops updating the parameters once the performance on hold-out data is not improving. This is a standard procedure that can be used to avoid overfitting in the context of iterative fitting methods, EM being a special case.

Before discussing further algorithmic questions, we will study the relationship between the proposed model and LSA in more detail.

3.3. Latent probability spaces and probabilistic latent semantic analysis

Consider the class-conditional probability mass functions $P(\cdot \mid z_k)$ over the vocabulary $\mathcal{W}$ which can be represented as points on the $M-1$ dimensional simplex of all probability mass functions over $\mathcal{W}$. Via its convex hull, this set of K points defines a $K-1$ dimensional convex region $\mathcal{R} \equiv \text{conv}(P(\cdot \mid z_1), \ldots, P(\cdot \mid z_K))$ on the simplex (provided they are in general position). The modeling assumption expressed by (3) is that all conditional probabilities $P(\cdot \mid d_i)$ for $1 \leq i \leq N$ are approximated by a convex combination of the K probability mass functions $P(\cdot \mid z_k)$. The mixing weights $P(z_k \mid d_i)$ are coordinates that uniquely define for each document a point within the convex region $\mathcal{R}$. A simple sketch of the geometry is shown in Figure 2. This demonstrates that despite of the discreteness of the introduced latent variables, a *continuous latent space* is obtained within the space of all probability mass functions over $\mathcal{W}$. Since the dimensionality of the convex region $\mathcal{R}$ is $K-1$ as opposed to $M-1$ for the probability simplex, this can also be thought of in terms of dimensionality reduction and $\mathcal{R}$ can be identified with a *probabilistic latent semantic space*. Each "direction" in this space corresponds to a particular context as quantified by $P(\cdot \mid z_k)$ and each document d_i participates in each context with some specific fraction $P(z_k \mid d_i)$. Note that since the aspect model is symmetric with respect to terms and documents, by

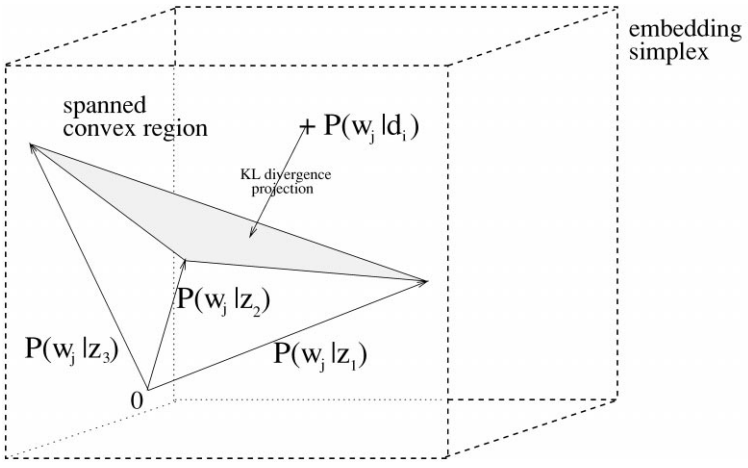

Figure 2. Sketch of the probability simplex and a convex region spanned by class-conditional probabilities in the aspect model.

reversing their role one obtains a corresponding region $\mathcal{R}'$ in the simplex of all probability mass functions over $\mathcal{D}$. Here each term w_j will participate in each context with some fraction $P(z_k \mid w_j)$, i.e., the probability of an occurrence of w_j as part of the context z_k.

To stress this point and to clarify the relation to LSA, let us rewrite the aspect model as parameterized by (5) in matrix notation. Hence define matrices by $\hat{\mathbf{U}} = (P(d_i \mid z_k))_{i,k}$, $\hat{\mathbf{V}} = (P(w_j \mid z_k))_{j,k}$, and $\hat{\Sigma} = \mathrm{diag}(P(z_k))_k$. The joint probability model $\mathbf{P}$ can then be written as a matrix product $\mathbf{P} = \hat{\mathbf{U}}\hat{\Sigma}\hat{\mathbf{V}}^t$. Comparing this decomposition with the SVD decomposition in LSA, one can point out the following re-interpretation of concepts of linear algebra: (i) the weighted sum over outer products between rows of $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ reflects conditional independence in PLSA, (ii) the K factors are seen to correspond to the mixture components of the aspect model, (iii) the mixing proportions in PLSA substitute for the singular values of the SVD in LSA. The crucial difference between PLSA and LSA, however, is the objective function utilized to determine the optimal decomposition/approximation. In LSA, this is the L_2- or Frobenius norm, which corresponds to an implicit additive Gaussian noise assumption on (possibly transformed) counts. In contrast, PLSA relies on the likelihood function of multinomial sampling and aims at an explicit maximization of the predictive power of the model. As is well known, this corresponds to a minimization of the cross entropy or Kullback–Leibler divergence between the empirical distribution and the model, which is different from any type of squared deviation. On the modeling side this offers important advantages, for example, the mixture approximation $\mathbf{P}$ of the co-occurrence table is a well-defined probability distribution and factors have a clear probabilistic meaning in terms of mixture component distributions. In contrast, LSA does not define a properly normalized probability distribution and, even worse $\tilde{\mathbf{N}}$, may contain negative entries. In addition, there is no obvious interpretation of the directions in the LSA latent space, while the directions in the PLSA latent space are interpretable as class-conditional word distributions that define a certain topical context. The probabilistic approach can also take advantage of the well-established statistical theory for model selection and complexity control, e.g., to determine the optimal number of latent space dimensions. Choosing the number of dimensions in LSA on the other hand is typically based on ad hoc heuristics.

A comparison in terms of computational complexity might suggest some advantages for LSA: ignoring potential problems of numerical stability, the SVD can be computed exactly, while the EM algorithm is an iterative method which is only guaranteed to find a local maximum of the likelihood function. There are two independent issues: (i) How much does the model accuracy suffer from the fact that EM is only able to find a local maximum? (ii) Provided that the local maximum is (almost) as good as the global maximum, what is the time complexity to compute a solution, i.e., how does the EM algorithms scale with the size of the data set? In many experiments conducted on various data sets, we have observed that the global maximum (as well as any local maximum) is often plagued by overfitting. In order to avoid overfitting, one may use regularization techniques like early stopping or tempered EM (cf. Section 3.6). It turns out that the variability of the solution quality obtained from various randomized initial conditions is usually small, the upshot being that even a "poor" local maximum in PLSA might be better than the exact solution in LSA. The number of arithmetic operations of course depends on the number of EM iterations that have to be performed. Typically 20–50 iterations are sufficient, each iteration requiring

$O(R \cdot K)$ operations, where R is the number of distinct observation pairs (d_i, w_j), i.e., $N \cdot M$ times the degree of sparseness of the term-document matrix. This can easily be verified by noticing that $R \cdot K$ posterior probabilities have to be computed in the E-step (6) each of which contributes to exactly one re-estimation equation in (11) and one in (12). As a consequence, the computation time of EM has not been significantly worse than computing an SVD on the co-occurrence matrix in any of the performed experiments. There is also a large potential for improving run-time performance of EM by on-line update schemes, which has not been explored so far.

3.4. Intermezzo: Word usage analysis with the aspect model

Let us briefly discuss an elucidating example application of the aspect model at this point. We have generated a dataset (CLUSTER) with abstracts of 1568 documents on *clustering* and trained an aspect model with 128 latent classes. As a particularly interesting term in this domain we have chosen the word 'segment'. Figure 3 shows the most probable words from 4 out of the 128 aspects which have the highest probability to generate the term 'segment'. This sketchy characterization of the aspects reveals very meaningful sub-domains: the first three aspects deal with *image segmentation* while the fourth concerns *phonetic segmentation* in the context of speech recognition. The further division of *image segmentation* into three slightly different types of word usage for 'segment' is also highly plausible: the first aspect seems to capture some of the word statistics in *medical imaging*, the second deals with *image sequence analysis*, while the third aspect is related to image segmentation mainly in the context of *contour and boundary detection*. Notice that the term 'region' and 'segment' have a very similar meaning in this context, which is indeed reflected by this aspect.

Figure 4 shows the abstracts of four exemplary documents, which have been pre-processed by a standard stop-word list and a stemmer. The posterior probabilities for the classes given

Aspect 1	Aspect 2	Aspect 3	Aspect 4
imag	video	region	speaker
SEGMENT	sequenc	contour	speech
textur	motion	boundari	recogni
color	frame	descrip	signal
tissu	scene	imag	train
brain	SEGMENT	SEGMENT	hmm
slice	shot	precis	sourc
cluster	imag	estim	speakerindepend
mri	cluster	pixel	SEGMENT
algorithm	visual	paramet	sound

Figure 3. The 4 aspects to most likely generate the word 'segment', derived from a $K = 128$ aspect model of the CLUSTER document collection. The displayed word stems are the most probable words in the class-conditional distribution $P(w_j \mid z_k)$, from top to bottom in descending order.

Document 1, $P\{z_k|d_1, W = \text{'segment'}\} = (0.951, 0.002, 0.001, 0.0001, \ldots)$
$P\{W = \text{'segment'}|d_1\} = 0.06$

SEGMENT medic imag challeng problem field imag analysi diagnost base proper **SEGMENT** digit imag **SEGMENT** medic imag need applic involv estim boundari object classif tissu abnorm shape analysi contour detec textur **SEGMENT** despit exist techniqu **SEGMENT** specif medic imag remain crucial problem [...]

Document 2, $P\{z_k|d_2, W = \text{'segment'}\} = (0.025, 0.956, 0.0002, 0.0002, \ldots)$
$P\{W = \text{'segment'}|d_2\} = 0.014$

describ new techniqu extract hierarch decomposi complex video selec brows purpos techniqu combin visual tempor inform captur import relat scene scene video allow analysi underli stori structur priori knowledg content defin gener model hierarch scene transition graph appli model implement brows video shot identifi collec kei frame repres video **SEGMENT** collec classifi accord gross visual inform [...]

Document 3, $P\{z_k|d_3, W = \text{'segment'}\} = (0.025, 0.003, 0.897, 0.016, \ldots)$
$P\{W = \text{'segment'}|d_3\} = 0.010$

paper describ contour extrac scheme refin roughli estim initi contour outlin precis object boundari author approach mixtur densiti descrip parametr descrip decompos subregion obtain region cluster descrip likelihood pixel belong object background evalu unlik activ contour extrac scheme region edgebas estim scheme integr energi minim process [...]

Document 4, $P\{z_k|d_3, W = \text{'segment'}\} = (0.025, 0.076, 0.001, 0.867, \ldots)$
$P\{W = \text{'segment'}|d_4\} = 0.010$

consid signal origin sequenc sourc specif problem **SEGMENT** signal relat **SEGMENT** sourc address issu wide applic field report describ resolu method ergod hidden markov model hmm hmm state correspond signal sourc signal sourc sequenc determin decod procedur viterbi algorithm forward algorithm observ sequenc baumwelch train estim hmm paramet train materi applic multipl signal sourc identif problem experi perform unknown speaker identif [...]

Figure 4. Abstracts of 4 exemplary documents from the CLUSTER collection along with latent class posterior probabilities $P\{z_k \mid d_i, W = \text{'segment'}\}$ and word probabilities $P\{W = \text{'segment'} \mid d_i\}$.

the different occurrences of 'segment' indicate how likely it is for each of the 4 aspects to have generated this observation. By inspection one can verify that all occurrences are assigned to the 'correct' aspect. We have also displayed estimates of the conditional word probabilities $P\{W = \text{'segment'} \mid d_i\}$ for the 4 documents. Notice that although 'segment' does not occur explicitly in document 3, the estimated probability for 'segment' is still significant. This implies, for example, that we might consider returning the document in response to a query with the keyword 'segment', although this term was actually never used in the (available part of the) document.

To provide a second example from a different domain, we have used the Topic Detection and Tracking (TDT1) corpus (http://www.ldc.upenn.edu/TDT, 1997) which consists of approximately 7 million word occurrences in 15863 documents. For illustrative purposes, an aspect model with $K = 128$ dimensions was trained. Figure 5 displays the two most probable aspects that generate the term 'flight' (left) and 'love' (right), revealing interesting types of word usage for 'flight' (aviation vs. space missions) as well as 'love' (family love vs. Hollywood love).

3.5. Aspects versus clusters

It is worth comparing the aspect model with statistical clustering models (cf. also Hofmann, Puzicha, & Jordan, 1999). In clustering models for documents, one typically associates a latent class variable with each document in the collection. Most closely related to our approach is the *distributional clustering model* (Pereira, Tishby, & Lee, 1993; Baker & McCallum, 1998) and the multinomial (maximum likelihood) version of Autoclass clustering (Cheeseman & Stutz, 1996), an unsupervised version of a naive Bayes' classifier.

Aspect 1	Aspect 2	Aspect 3	Aspect 4
plane	space	home	film
airport	shuttle	family	movie
crash	mission	like	music
flight	astronauts	love	new
safety	launch	kids	best
aircraft	station	mother	hollywood
air	crew	life	love
passenger	nasa	happy	actor
board	satellite	friends	entertainment
airline	earth	cnn	star

Figure 5. The 2 aspects to most likely generate the word 'flight' (left) and 'love' (right), derived from a $K = 128$ aspect model of the TDT1 document collection. The displayed terms are the most probable words in the class-conditional distribution $P(w_j \mid z_k)$, from top to bottom in descending order.

It can be shown (Hofmann, Puzicha, & Jordan, 1999) that the conditional word probability of a probabilistic clustering model is given by

$$P(w_j \mid d_i) = \sum_{k=1}^{K} P\{c(d_i) = c_k\} P(w_j \mid c_k), \tag{13}$$

where $P\{c(d_i) = c_k\}$ is the posterior probability of document d_i belonging to cluster c_k. In maximum likelihood Autoclass, it is a simple implication of Bayes' rule that the class posterior probabilities will concentrate their probability mass on one cluster c_k with an increasing number of observations (i.e., with the length of the document). The likelihood contributions for each word occurrence enter a product, reflecting the conditional independence assumption

$$P\{c(d_i) = c_k\} = \frac{P(c_k) \prod_{j=1}^{M} P(w_j \mid c_k)^{n(d_i, w_j)}}{\sum_{l=1}^{K} P(c_l) \prod_{j=1}^{M} P(w_j \mid c_l)^{n(d_i, w_j)}}. \tag{14}$$

This means that although (3) and (13) are algebraically equivalent, they are conceptually very different. In a clustering model for documents, it is assumed that each document belongs to exactly one cluster and it is only the finiteness of the number of observations per document that induces uncertainty about a document's cluster membership, as expressed in the class posterior probabilities $P\{c(d_i) = c_k\}$. In contrast, the aspect model assumes that every *occurrence* of a word in a document is associated with a unique state z_k of the latent class variable. This does by no means exclude that different word occurrences within the same document or occurrences of the same word within different documents can be "explained" by different aspects. However, since latent class variables associated with occurrences in the same document share their prior probabilities $P(z_k \mid d_i)$, observations within a document get effectively coupled. By symmetry this also holds for different occurrences of the same word. As a result of this coupling, the probabilities $P(z_k \mid d_i)$ and $P(z_k \mid w_j)$ tend to be "sparse", i.e., for given d_i or w_j typically only few entries are significantly different from zero.

The aspect model clusters document-word pairs, which is different from clustering either documents or words or both. As a consequence the document-specific word probabilities in the aspect model are a convex combination of aspects, while the clustering model assumes there is just *one* cluster-specific distribution which is inherited by all documents in the cluster (cf. also Hofmann, Puzicha, & Jordan, 1999).

3.6. Model fitting revisited: Improving generalization by tempered EM

So far we have focused on maximum likelihood estimation to fit a model to a given document collection. Although the likelihood is the quantity we believe to be crucial in assessing the quality of a model, one clearly has to distinguish between the performance on the training data and on unseen test data. We will use the *perplexity*, a measure commonly used in language modeling, to assess the generalization performance of a model. The perplexity is defined to be the log-averaged inverse probability on unseen data, i.e.,

$$\mathcal{P} = \exp\left[-\frac{\sum_{i,j} n'(d_i, w_j)\, \log P(w_j \mid d_i)}{\sum_{i,j} n'(d_i, w_j)}\right], \tag{15}$$

where $n'(d_i, w_j)$ denotes counts on hold-out or test data.

To derive conditions under which generalization on unseen data can be guaranteed is actually *the* fundamental problem of statistical learning theory. Here, we propose a generalization of maximum likelihood for mixture models which is known as *annealing* and is based on an entropic regularization term. The resulting method is called *Tempered Expectation Maximization* (TEM) and is closely related to *deterministic annealing* (Rose, Gurewitz, & Fox, 1990). The combination of deterministic annealing with the EM algorithm has been investigated before in Ueda and Nakano (1998), Hofmann, Puzicha, and Jordan (1999).

The starting point of TEM is a derivation of the E-step based on an optimization principle. As has been pointed out in Neal and Hinton (1998), the EM procedure in latent variable models can be obtained by minimizing a common objective function—the (Helmholtz) *free energy*—which for the aspect model is given by

$$\begin{aligned}\mathcal{F}_\beta \;=\; & -\beta \sum_{i=1}^{N}\sum_{j=1}^{M} n(d_i, w_j) \sum_{k=1}^{K} \tilde{P}(z_k; d_i, w_j)\, \log\left[P(d_i \mid z_k) P(w_j \mid z_k) P(z_k)\right] \\ & + \sum_{i=1}^{N}\sum_{j=1}^{M} n(d_i, w_j) \sum_{k=1}^{K} \tilde{P}(z_k; d_i, w_j)\, \log \tilde{P}(z_k; d_i, w_j)\,. \end{aligned} \tag{16}$$

Here $\tilde{P}(z_k; d_i, w_j)$ are variational parameters which define a conditional distribution over $\{z_1, \ldots, z_K\}$ and β is a parameter which—in analogy to physical systems—is called the *inverse computational temperature*. Notice that the first contribution in (16) is the negative expected log-likelihood scaled by β. Thus in the case of $\tilde{P}(z_k; d_i, w_j) = P(z_k \mid d_i, w_j)$ minimizing $\mathcal{F}$ w.r.t. the parameters defining $P(d_i, w_j \mid z_k) P(z_k)$ amounts to the standard

M-step in EM. In fact, it is straightforward to verify that the posteriors are obtained by minimizing $\mathcal{F}$ w.r.t. $\tilde{P}$ at $\beta = 1$. In general $\tilde{P}$ is determined by

$$\tilde{P}(z_k; d_i, w_j) = \frac{[P(z_k)P(d_i \mid z_k)P(w_j \mid z_k)]^\beta}{\sum_l [P(z_l)P(d_i \mid z_l)P(w_j \mid z_l)]^\beta} = \frac{[P(z_k \mid d_i)P(w_j \mid z_k)]^\beta}{\sum_l [P(z_l \mid d_i)P(w_j \mid z_l)]^\beta}. \quad (17)$$

This shows that the effect of the entropy at $\beta < 1$ is to dampen the posterior probabilities such that they will get closer to the uniform distribution with decreasing β.

Somewhat contrary to the spirit of annealing as a continuation method, we propose an 'inverse' annealing strategy which first performs EM iterations and then *decreases* β until performance on hold-out data deteriorates. Compared to annealing this may accelerate the model fitting procedure significantly (e.g., by a factor of $\approx 10 - 50$) and we have not found the test set performance of "heated" models to be significantly worse than the one achieved by carefully "annealed" models. The TEM algorithm can be implemented in the following way:

1. Set $\beta \leftarrow 1$ and perform EM with early stopping.
2. Decrease $\beta \leftarrow \eta\beta$ (with $\eta < 1$) and perform one TEM iteration.
3. As long as the performance on hold-out data improves (non-negligible) continue TEM iterations at this value of β, otherwise goto step 2
4. Perform stopping on β, i.e., stop when decreasing β does not yield further improvements.

4. Experimental results

In this paper, we have presented a novel approach to latent semantic analysis based on a statistical latent variable model, the general problem of text analysis being the main thread of our presentation. In the experimental evaluation, however, we focus on two more specific tasks to assess the performance of PLSA: (i) perplexity minimization for a document-specific unigram model and noun-adjective pairs, and (ii) automated indexing of documents. The evaluation of LSA and PLSA on the first task will demonstrate the advantages of explicitly minimizing perplexity by (tempered) maximum likelihood estimation, the second task will then show that the advantages of PLSA in terms of a solid statistical foundation do pay off in applications which superficially do not seem to be directly related to perplexity reduction.

4.1. Perplexity evaluation for PLSA and LSA

In order to compare the predictive performance of PLSA and LSA one has to specify how to extract probabilities from a LSA decomposition. This problem is not trivial, since negative entries prohibit a straightforward re-normalization of the approximating matrix $\tilde{\mathbf{N}}$. We have thus followed the approach of Coccaro and Jurafsky (1998) to derive LSA probabilities. The latter is *ad hoc* and involves the optimization of a free parameter γ, but the extracted probabilities are more accurate than the ones obtained by other straightforward normalization steps.

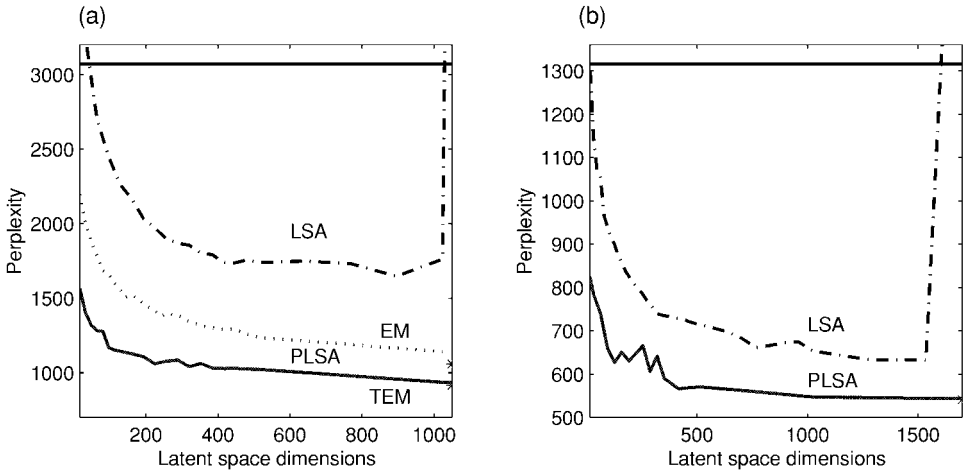

Figure 6. Perplexity results as a function of the latent space dimensionality for (a) the MED data (rank 1033) and (b) the LOB data (rank 1674). Plotted results are for LSA (dashed-dotted curve) and PLSA (trained by TEM = solid curve, trained by early stopping EM = dotted curve). The upper baseline is the unigram model corresponding to marginal independence. The star at the right end of the PLSA denotes the perplexity of the largest trained aspect models ($K = 2048$).

Two data sets that have been used to evaluate the perplexity performance: (i) a standard information retrieval test collection MED with 1033 document, (ii) a dataset with noun-adjective pairs generated from a tagged version of the LOB corpus. In Figure 6 we report perplexity results for LSA and PLSA on the MED (a) and LOB (b) datasets dependent on the number of dimensions of the (probabilistic) latent semantic space. For the noun-adjective pairs the reported perplexity corresponds to predicting nouns conditioned on the corresponding adjective. PLSA outperforms the statistical model derived from standard LSA by far. On the MED collection PLSA reduces perplexity relative to the unigram baseline by more than a factor of three ($3073/936 \approx 3.3$), while LSA achieves less than a factor of two in reduction ($3073/1647 \approx 1.9$). On the less sparse LOB data, the difference between LSA and PLSA is somewhat less drastic, but still very significant. With PLSA the reduction in perplexity is $1316/547 \approx 2.41$ while the reduction achieved by LSA is only $1316/632 \approx 2.08$. In order to demonstrate the advantages of TEM, we have also trained aspect models on the MED collection by standard EM with early stopping. As can be seen from the curves in Figure 6(a), the difference between EM and TEM model fitting is significant. Although both strategies—annealing and early stopping—are successful in controlling the model complexity, EM training performs worse, since it makes a very inefficient use of the available degrees of freedom. Notice, that with both methods it is possible to train high-dimensional models with a continuous improvement in performance. The number of latent space dimensions may even exceed the rank of the co-occurrence matrix $\mathbf{N}$ and the choice of the number of dimensions becomes merely an issue of possible limitations of computational resources.

In order to investigate the effect of the final choice of β and to further stress the advantages of TEM, we have performed another series of experiments on the TDT1 corpus using

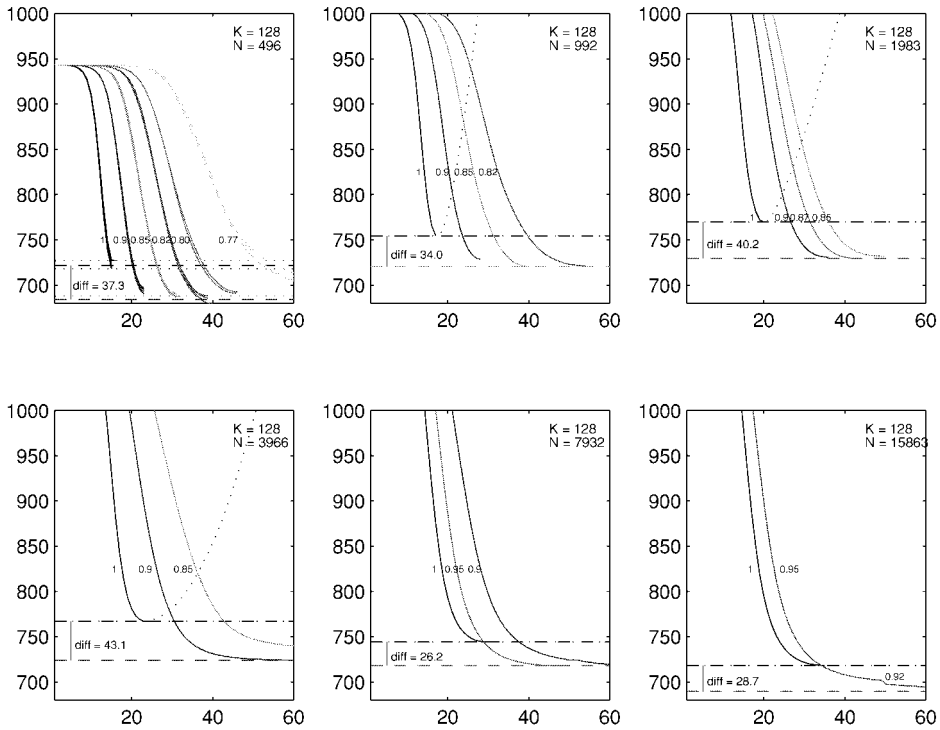

Figure 7. Annealing experiments on the TDT1 dataset for a model with $K = 128$ aspects at different subsampling factors (from upper left to lower right: 32x,16x,8x,4x,2x,1x). N denotes the number of documents in the training data set, small numbers indicate the inverse temperature β utilized for the respective training curves. The dotted line shows the performance of EM past the optimal stopping point.

randomized subsampling to study the effect of the size of the document collection. Since we want to focus on the control of overfitting and not on the problem of local maxima, all models have been trained at a fixed temperature. Figure 7 shows perplexity curves for different inverse temperatures β for a model with $K = 128$ as a function of the number of tempered EM iterations at various subsampling levels. At all temperatures we have performed early stopping once the perplexity on hold-out data increased. For the $32\times$ subsampling experiments ($N = 496$ documents), we have repeated runs 5 times in order to evaluate the solution variability for different (randomized) initial conditions. The following observations can be made: (i) Although early stopping prevents overfitting, the use of temperature control yields a significant and consistent improvement for all sample sizes. (ii) The advantages of tempered EM are considerable, even for the full TDT1 dataset with 7 million tokens. Thus overfitting is also an important problem in large-scale data sets. (iii) The computational complexity of tempered EM is slightly higher compared to standard EM (i.e., $\beta = 1$), typically twice as many iterations are necessary to converge. (iv) The variablity in the quality of solutions achieved from different initial conditions is small compared to the perplexity difference between EM and TEM. (v) Although β is a critical parameter, there

is a broad range of values for which a comparable performance is achieved. Hence, TEM is fairly robust with respect to the choice of the optimal β.

4.2. Information retrieval with PLSA and LSA

One of the key problems in information retrieval is *automatic indexing* which has its main application in query-based retrieval. The most popular family of information retrieval techniques is based on the *Vector–Space Model* (VSM) for documents (Salton & McGill, 1983). A VSM variant is characterized by three ingredients: a transformation function (also called local term weight), a term weighting scheme (also called global term weight), and a similarity measure. Since this paper deals with PLSA as a general unsupervised learning technique, we have not taken advantage of the huge variety of sophisticated transformation functions and term weighting schemes which are known to yield largely varying results on different data sets. Instead, we have utilized a rather straightforward representation based on the (untransformed) term frequencies $n(d_i, w_j)$ together with the standard cosine matching function. Given that our interest is mainly in assessing the relative performance differences between direct term matching, LSA, and PLSA, this choice seems to be justified. The same representation used for documents applies to queries q as well, so that the similarity function for the baseline method can be written as

$$s(d_i, q) = \frac{\sum_{j=1}^{M} n(d_i, w_j) n(q, w_j)}{\sqrt{\sum_{j=1}^{M} n(d_i, w_j)^2} \sqrt{\sum_{j=1}^{M} n(q, w_j)^2}}, \tag{18}$$

In Latent Semantic Indexing (LSI), the original vector space representation of documents is replaced by a representation in the low–dimensional latent space and the similarity is computed based on that representation. Queries or documents which were not part of the original collection can be *folded in* by a simple matrix multiplication (cf. (Deerwester et al., 1990) for details). In our experiments, we have actually considered linear combinations of the original similarity score (18) (weight λ) and the one derived from the latent space representation (weight $1 - \lambda$), as suggested in Pereira et al. (1993).

The same ideas have been applied in Probabilistic Latent Semantic Indexing (PLSI) in conjunction with the PLSA model. More precisely, the low–dimensional representation in the *factor space* $P(z_k \mid d_i)$ and $P(z_k \mid q)$ have been utilized to evaluate similarities. To achieve this queries have to be folded in, which is done in the PLSA by fixing the $P(w_j \mid z_k)$ parameters and calculating weights $P(z_k \mid q)$ by TEM.

One advantage of using statistical models vs. SVD techniques is that it allows us to systematically combine different models. While this should optimally be done according to a Bayesian model combination scheme, we have utilized a much simpler approach in our experiments which has nevertheless shown excellent performance and robustness. Namely, we have simply combined the cosine scores of all models with a uniform weight. The resulting method is referred to as PLSI*. Empirically we have found the performance to be very robust w.r.t. different (non-uniform) weights and also w.r.t. the λ-weight used in combination with the original cosine score. This is due to the noise reducing benefits of model averaging. Notice that LSA representations for different K form a nested sequence,

which is not true for the statistical models which are expected to capture a larger variety of reasonable decompositions.

We have utilized the following four medium–sized standard document collection with relevance assessment: (i) MED (1033 document abstracts from the National Library of Medicine), (ii) CRAN (1400 document abstracts on aeronautics from the Cranfield Institute of Technology), (iii) CACM (3204 abstracts from the CACM journal), and (iv) CISI (1460 abstracts in library science from the Institute for Scientific Information). For all document collections queries are annotated with ground truth, i.e., a set of relevant documents has been determined for each query by human experts. The condensed results in terms of average precision (at the 9 recall levels 10%−90%) are summarized in Table 1, while the corresponding precision-recall curves can be found in Figure 8. Here are some additional details of the experimental setup: PLSA models at $K = 32, 48, 64, 80, 128$ have been trained by TEM for each data set with 10% hold-out data. For PLSI we report the best result obtained by any of these models, for LSI we report the best result obtained for the optimal dimension (exploring 32–512 dimensions at a step size of 8). The combination weight λ for the cosine baseline score has been manually optimized, MED, CRAN: $\lambda = 1/2$, CACM, CISI: $\lambda = 2/3$; in general slightly smaller weights have been utilized for the combined models, although the results are highly robust with respect to the exact choice of λ.

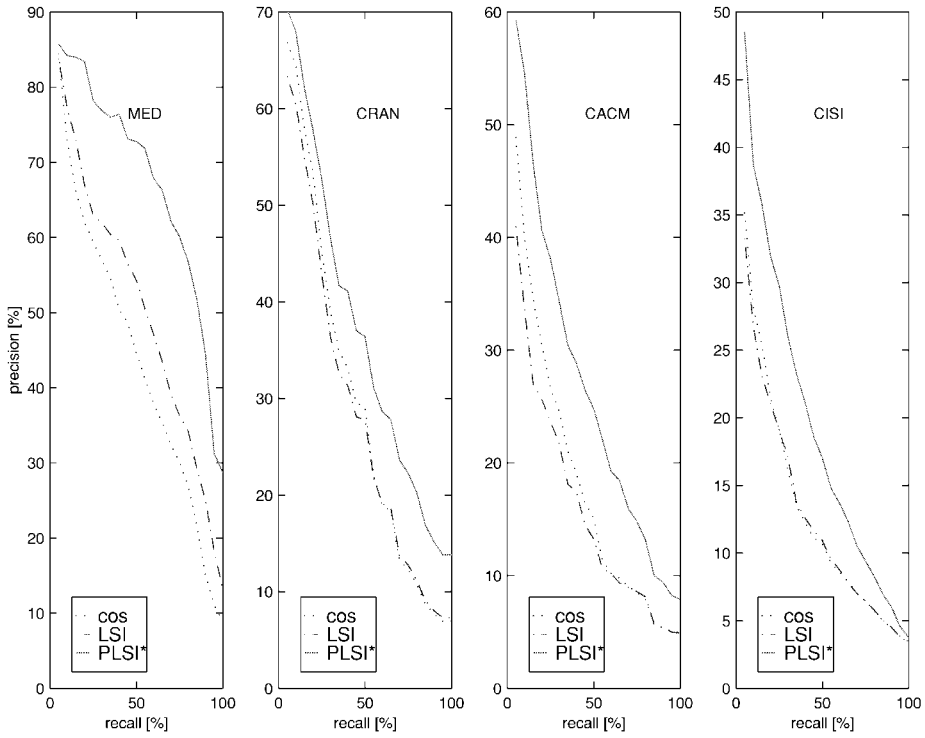

Figure 8. Precision-recall curves for the 4 test collections. Depicted are curves for direct term matching, LSI, and PLSI*.

Table 1. Average precision results and relative improvement w.r.t. the baseline method cos + tf for the 4 standard test collections.

	MED		CRAN		CACM		CISI	
	prec.	impr.	prec.	impr.	prec.	impr.	prec.	impr.
cos + tf	44.3	–	29.9	–	17.9	–	12.7	–
LSI	51.7	+16.7	*28.7	−4.0	*16.0	−11.6	12.8	+0.8
PLSI	63.9	+44.2	35.1	+17.4	22.9	+27.9	18.8	+48.0
PLSI*	66.3	+49.7	37.5	+25.4	26.8	+49.7	20.1	+58.3

Compared are LSI, PLSI, as well as results obtained by combining PLSI models (PLSI*). An asterix for LSI indicates that no performance gain could be achieved over the baseline, the result at 256 dimensions with $\lambda = 2/3$ is reported in this case.

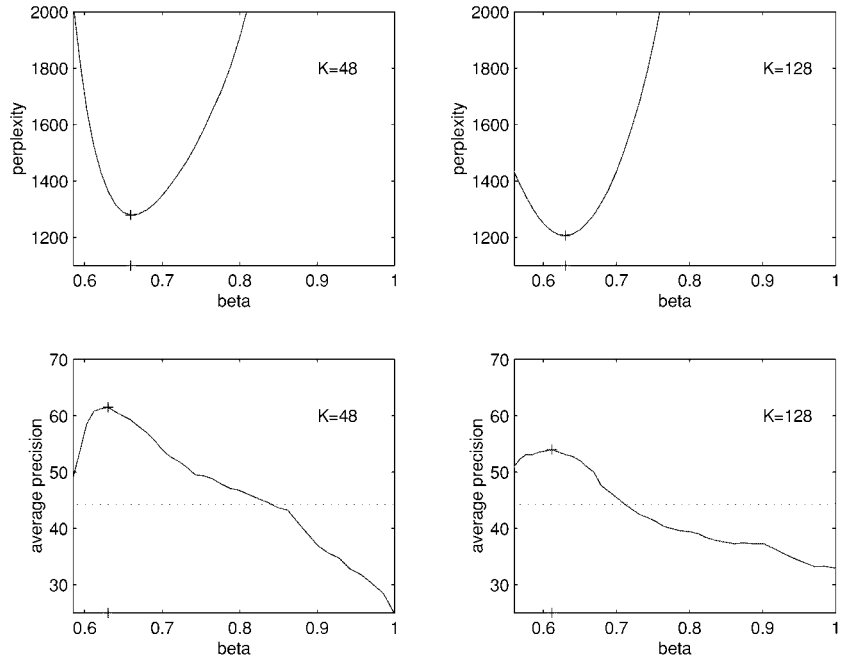

Figure 9. Perplexity and average precision as a function of the inverse temperature β for an aspect model with $K = 48$ (left) and $K = 128$ (right).

The experiments consistently validate the advantages of PLSI over LSI. Substantial performance gains have been achieved for all 4 data sets. Notice that the relative precision gain compared to the baseline method is typically around 100% in the most interesting intermediate regime of recall! In particular, PLSI works well even in cases where LSI fails completely (these problems of LSI are in accordance with the original results reported in Deerwester et al. (1990)). The benefits of model combination are also very substantial. In all cases the (uniformly) combined model performed better than the best single model.

These experiments clearly demonstrate that the advantages of PLSA over standard LSA are not restricted to applications with performance criteria directly depending on

the perplexity. Statistical objective functions like the perplexity (log-likelihood) may thus provide a general yardstick for analysis methods in text learning and information retrieval. To stress this point we ran a series of experiments, where both, perplexity and average precision, have been monitored simultaneously as a function of β. The resulting curves for the MED collection are plotted in Figure 9. The results on the other collections are very similar. Although the two curves do not attain their respective extrema for exactly the same value of β, the correlation is quite striking. In fact, the best retrieval performance is achieved for slightly lower values of β than the one determined on the hold-out data.

5. Conclusion

We have proposed a novel method for unsupervised learning, called *Probabilistic Latent Semantic Analysis*, which is based on a statistical latent-class model. We have argued that this approach is more principled than standard Latent Semantic Analysis, since it possesses a sound statistical foundation and utilizes the (annealed) likelihood function as an optimization criterion. *Tempered Expectation Maximization* has been presented as a powerful fitting procedure. We have experimentally verified the claimed advantages in terms of perplexity evaluation on text data as well as on linguistic data and for an application in automated document indexing, achieving substantial performance gains in all cases. Probabilistic Latent Semantic Analysis has thus to be considered as a promising novel unsupervised learning method with a wide range of application in text learning, computational linguistics, information retrieval, and information filtering. Future work and publications will deal in larger detail with specific applications as well as with extensions and generalizations of the presented method.

Acknowledgment

The author would like to thank Jan Puzicha, Joachim Buhmann, Andrew McCallum, Dan Gildea, Andrew Ng, Mike Jordan, Nelson Morgan, Jerry Feldman, Sebastian Thrun, and Tom Mitchell for stimulating discussions and helpful hints.

References

Baker, L. D. & McCallum, A. K. (1998). Distributional clustering of words for text classification. In *Proceedings of the 21st ACM-SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*.

Bellegarda, J. R. (1998). Exploiting both local and global constraints for multi-span statistical language modeling. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98*, pp. 677–680.

Berry, M. W. Dumais, S. T., & O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review*, *37*(4), 573–595.

Cheeseman, P. & Stutz, J. (1996). Bayesian classification (AutoClass): Theory and results. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, & Ramasamy Uthurusamy, (Eds.), *Advances in Knowledge Discovery and Data Mining*. AAAI Press/MIT Press.

Coccaro, N. & Jurafsky, D. (1998). Towards better integration of semantic predictors in statistical language modeling. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP)*.

Deerwester, S., Dumais, G. W., Furnas, S. T., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*, 391–407.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. B*, *39*, 1–38.

Dumais, S. T. (1995). Latent semantic indexing (LSI): TREC-3 report. In D.K Harman, (Ed.), *Proceedings of the Text REtrieval Conference (TREC-3)*, pp. 219–230.

Foltz, P. W. & Dumais, S. T. (1992). An analysis of information filtering methods. *Communications of the ACM*, *35*(12), 51–60.

Gilula, Z., & Haberman, S. J. (1986). Canonical analysis of contingency tables by maximum likelihood. *Journal of the American Statistical Association*, *81*(395), 780–788.

Golub, G. H. & Van Loan, C. F. (1996). *Matrix Computations*. Johns Hopkins University Press, 3rd (ed.).

Hofmann, T., Puzicha, J., & Jordan, M. I. (1999). Unsupervised learning from dyadic data. In *Advances in Neural Information Processing Systems*, Vol. 11, MIT Press.

Katz, S. M. (1987). Estimation of probabilities for sparse data for the language model component of a speech recogniser. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *35*(3), 400–401.

Landauer, T. K. & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*.

LDC. Linguistic Data Consortium: TDT pilot study corpus documentation. http://www.ldc.upenn.edu/TDT, 1997.

Lee, D. D. & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(675), 788–791.

Neal, R. M. & Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental and other variants. In M.I. Jordan, (Ed.), *Learning in Graphical Models*, Dordrecht, MA: Kluwer Academic Publishers, pp. 355–368.

Pereira, F. C. N., Tishby, N. Z., & Lee, L. (1983). Distributional clustering of english words. In *Proceedings of the ACL*, pp. 183–190.

Rose, K., Gurewitz, E., & Fox, G. (1990). A deterministic annealing approach to clustering. *Pattern Recognition Letters*, *11*(11), 589–594.

Salton, G. & McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. New York: McGraw–Hill.

Saul, L. & Pereira, F. (1997). Aggregate and mixed–order Markov models for statistical language processing. In *Proceedings of the 2nd International Conference on Empirical Methods in Natural Language Processing*, pp. 81–89.

Ueda, N. & Nakano, R. (1988). Deterministic annealing EM algorithm. *Neural Networks*, *11*(2), 271–282.

Witten, I. H. & Bell, T. C. (1991). The zero-frequency problem—estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4);1085–1094.

Wolfe, M. B. W., Schreiner, M. E., Rehder, B., Laham, D., Foltz, P. W., Kintsch, W. & Landauer, T. K. (1998). Learning from text: Matching readers and texts by latent semantic analysis. *Discourse Processes*, *25*(2/3), 309–336.

Received February 15, 1999
Revised July 26, 2000
Accepted July 31, 2000
Final manuscript July 31, 2000