Publication 6

Jaakko Peltonen, Janne Sinkkonen and Samuel Kaski. Sequential Information Bottleneck for Finite Data. In Russ Greiner and Dale Schuurmans, editors, *Proceedings of the Twenty-First International Conference on Machine Learning (ICML 2004)*, pages 647-654, Omnipress, Madison, WI, 2004.

Sequential Information Bottleneck for Finite Data

Jaakko Peltonen[1] JAAKKO.PELTONEN@HUT.FI
Janne Sinkkonen[1] JANNE.SINKKONEN@HUT.FI
Samuel Kaski[1,2] SAMUEL.KASKI@HUT.FI
[1] Neural Networks Research Centre, Helsinki University of Technology, P.O. Box 5400, FIN-02015 HUT, Finland
[2] Department of Computer Science, P.O. Box 26, FIN-00014 University of Helsinki, Finland

Abstract

The sequential information bottleneck (sIB) algorithm clusters co-occurrence data such as text documents vs. words. We introduce a variant that models sparse co-occurrence data by a generative process. This turns the objective function of sIB, mutual information, into a Bayes factor, while keeping it intact asymptotically, for non-sparse data. Experimental performance of the new algorithm is comparable to the original sIB for large data sets, and better for smaller, sparse sets.

1. Introduction

In text document analysis, word order is commonly ignored and documents are treated as "bags of words". Under this model, the sufficient statistics for a document are the word counts n_y telling how many times each word y appears in the document.

For clustering purposes, we formally consider a set of co-occurrences of two categorical variables X and Y. Given exchangeability, a sufficient statistic for the co-occurrences are the counts n_{xy} of all value combinations (x, y). The counts form a two-dimensional matrix or *contingency table*. For a document collection, the documents (X, rows) and words (Y, columns) form the margins of this table. Clustering the document collection, or any similarly expressible data, is equivalent to merging rows of the corresponding contingency table.

Methods for clustering contingency table margins have been proposed earlier both in statistics (Gilula, 1986) and in machine learning (Tishby et al., 1999). Some component models are closely related (e.g., Blei et al., 2003; Buntine, 2002; Hofmann, 1999). We focus on the sequential information bottleneck algorithm (sIB; Slonim et al., 2002), motivated by the information bottleneck principle (IB; Tishby et al., 1999).

Clustering the margins of a contingency table loses information about the original margin variables. A criterion for clustering would be to minimize loss of such information that is common to the margins. In other words, statistical dependences between the margins should be preserved. In the case of text documents, document clusters obtained by this criterion would be informative about words of the documents.

IB and sIB perform this kind of clustering; they measure dependency of the margins by mutual information. Mutual information is not defined for counts obtained from finite data, but for probability distributions. Its accuracy for finite, especially sparse data sets, is then uncertain. We propose an alternative objective function defined for counts, and an algorithm very similar to sIB. For large data sets the new objective function approaches mutual information, and the algorithm approaches sIB. Empirical performance is comparable to sIB and better for very sparse data.

The objective function is interpretable as a Bayes factor, or as a likelihood. The likelihood is a marginalized version of the crisp-cluster likelihood interpretation of IB (Slonim & Weiss, 2002, but see also the ACM model in Hofmann & Puzicha, 1998).

1.1. Information Bottleneck

The information bottleneck (IB) principle for clustering rows of contingency tables is based on Shannon's information theory: It deals with the (asymptotic) joint distribution $p(x, y)$ of X and Y.

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.

In information-theoretic terms, IB finds a representation T for the margin variable X, identified with the distribution $p(t|x)$. For clustering, t can be interpreted as clusters, and $p(t|x)$ as degrees of cluster memberships.

Objective function of IB. IB minimizes $I(X,T) - \beta I(T,Y)$, a compromise between two mutual informations. The dependency term $I(T,Y)$, tries to preserve the structure of $p(X,Y)$, while the complexity term $I(X,T)$ tries to keep T non-informative of X, which can be seen as simplicity of the representation T. In a sense, the objective function implements a "bottleneck" for the dependency between X and Y through T. (Note that as explained below, the simplicity constraint is relaxed in sIB.)

Variational optimization of the cost with respect to $p(t|x)$ leads to the set of equations

$$\begin{cases} p(t|x) = \frac{p(t)}{Z(\beta,x)} \exp(-\beta D_{KL}(p(y|x), p(y|t))) \\ p(y|t) = \frac{1}{p(t)} \sum_x p(y|x)p(t|x)p(x) \\ p(t) = \sum_x p(t|x)p(x) \end{cases} \quad (1)$$

where $Z(\beta,x)$ is a normalization term. The first equation characterizes the optimal clusters in the document space X. Before normalization, they decay exponentially with the Kullback-Leibler divergence from "document prototypes" $p(y|t)$.

Crisp clusters. From (1) it is clear that as $\beta \to \infty$, optimal clusters tend to become crisp. That is, $p(t|x)$ approaches zero or one almost everywhere, and the cost function $I(X,T) - \beta I(T,Y)$ essentially becomes $I(T,Y)$, a measure of dependency between T and Y. Conversely, when $I(T,Y)$ is variationally maximized with respect to the clusters, the optimal clusters are of the form $p(t|x) \in \{0,1\}$.

If the motivation is to group the data, it is tempting to at least finally set $\beta \to \infty$. We focus on this case.

For crisp clusters, the representation T, defined by $p(t|x)$, becomes a deterministic function $t(x) : X \to T$. The co-occurrences of T and Y then form a (smaller) *contingency table*, where the rows of the (X,Y) table are merged according to $t(x)$.

Below we will denote the (T,Y) table by $\mathbf{N}$ and its entries by n_{ty}. Margins of the table are denoted by $n_t = \sum_y n_{ty}$, $n_y = \sum_t n_{ty}$, and the total number of co-occurrences in the table by $N = \sum_{t,y} n_{ty}$. Although the setting is more general, for convenience x, y, and t are below called documents, words, and clusters, respectively.

1.2. Sequential Information Bottleneck (sIB)

The sequential information bottleneck method (here sIB for brevity; Slonim et al., 2002) produces crisp clusters, $p(t|x) \in \{0,1\}$, where each document belongs to a single cluster. This corresponds to maximizing the dependency between the clusters T and the words Y, measured by the mutual information $I(T,Y)$. It may therefore be seen as a margin-clustering contingency table algorithm in the sense explained above.

Given an initial assignment of documents x to clusters t, the sIB algorithm sequentially draws a random document x from the clusters, and finds a new cluster for it by minimizing a merging criterion. In the sIB algorithm the merging criterion is (weighted) Jensen-Shannon divergence D_{JS} between the word distributions of the relocated document and the candidate cluster, i.e.,

$$\begin{aligned} &(p(x) + p(t)) \cdot D_{JS}(p(Y|x), p(Y|t)|\pi_1, \pi_2) \\ &\quad = (p(x) + p(t)) \cdot [\pi_1 D_{KL}(p(Y|x), \bar{p}(Y)) \\ &\qquad + \pi_2 D_{KL}(p(Y|t), \bar{p}(Y))] \end{aligned} \quad (2)$$

where D_{KL} is the Kullback-Leibler divergence, $\pi_1 = p(x)/(p(x)+p(t))$, $\pi_2 = p(t)/(p(x)+p(t))$ and $\bar{p}(y) = \pi_1 p(y|x) + \pi_2 p(y|t)$.

In experiments sIB outperformed agglomerative IB, iterative double clustering, K-means and a number of heuristic alternatives for sequential clustering (Slonim et al., 2002). The sequential approach may also perform better than alternating optimization by equations (1) (Slonim et al., 2002). It is therefore a good application for the finite-data objective presented next.

2. Sequential Information Bottleneck for Finite Data

Mutual information is an asymptotic quantity, defined for document and word probabilities $p(x)$ and $p(y|x)$. The implicit solution (1) of the IB optimization problem then also refers to probabilities. In straightforward data analysis tasks these are unknown; for IB they need to be estimated from data. Slonim et al. (2002) used the estimate $p(y|x) = n_{xy}/|X|$, where n_{xy} is the count of occurrences of word y in document x, and $|X|$ is the total number of words in the document. But relative word counts are noisy descriptors of the content of the documents, and the noise is not visible in such a straightforward estimate.

A solution is to replace the point estimates by the posterior distributions of document content, given the counts and the bag-of-words model. Uncertainty is in-

tegrated out only at the final stage of evaluating the evidence for the model. Low word counts carrying only little information are then correctly given less weight. Assuming the bag-of-words model is sound, such analysis should better match the underlying topical categories, and improve generalization to new data.

We introduce a finite-data variant of the sIB method, called *finite sequential information bottleneck* (fsIB), that takes the finite counts into account probabilistically. The novelty is contained in the new objective function; otherwise the resulting algorithm is identical to sIB with a similar time complexity and similar convergence proofs.

We derive the objective from generative models and their Bayes factors; at the limit of large co-occurrence counts it is equivalent to mutual information.

2.1. Objective Function

We aim to measure the dependency between the rows and the columns of the contingency table, given a certain clustering T of rows. The table is then indexed by t (rows; for example document groups) and y (words).

The objective function should account for the uncertainty of probabilities estimated from sparse data. This is achieved by assuming a multinomial generative process: the observed frequencies n_{ty} are produced by an underlying multinomial distribution with parameters θ_{ty}.

The objective function for fsIB, C_{fsIB}, will compare evidence for two hypotheses, or model families, that could produce the observed data. The first family H_1 assumes a free multinomial distribution, parameterized by θ_{ty}, over all cells. The second family H_2 assumes independent multinomial margins θ_t and θ_y; the cell-wise probabilities are then restricted to products $\theta_{ty} = \theta_t\theta_y$ of the margin probabilities. The evidence for these two hypotheses, dependent vs. independent margins, can be compared by a Bayes factor (Good, 1976), where the uncertainty of the parameters θ is integrated out to compare the two hypotheses without regard to the parameter values.

The Bayes factor can be shown to be (for details see Peltonen et al., 2003)

$$\frac{p(\mathbf{N}|H_1)}{p(\mathbf{N}|H_2)} \propto \frac{\prod_{t,y}\Gamma(n_{ty}+\alpha_{ty})}{\prod_t \Gamma(n_t+\alpha_t)} \equiv C_{fsIB} \qquad (3)$$

where the $\alpha_{ty} \geq 0$ are parameters of Dirichlet priors of the co-occurrence counts and the $\alpha_t \geq 0$ of the sizes of the clusters. These parameters can be seen as prechosen counts of "virtual data", co-occurrence counts and counts of data within clusters, respectively. We call the priors "consistent" if the cluster counts match the co-occurrence counts of virtual data, that is, if $\alpha_t = \sum_y \alpha_{ty}$.

For document clustering we will use two kinds of priors. Both are empirical priors computed from the whole corpus to distribute the "virtual data" according to the overall proportions of Y in the corpus, with the same total as in the hypergeometric formula (see below). In both priors $\alpha_{ty} = n_y \frac{|Y|}{N}$, where $|Y|$ is the number of possible values of Y (size of the dictionary) and N is the total number of co-occurrences. The priors differ for α_t: we set either $\alpha_t = 1$ or $\alpha_t = |Y|$. The first choice (**fsIB 1**) is used in the hypergeometric formula but it leaves the priors defined by α_{ty} and α_t inconsistent; the second choice (**fsIB 2**) makes them consistent.

2.2. Some interpretations of the Bayes factor

Connection to the hypergeometric formula. Note that if $\alpha_{ty} = \alpha_t = 1$, the gamma functions in (3) turn to factorials, and (3) becomes (the inverse of) the likelihood of data given the observed margin counts and the assumption of independent margins (H_2) (Rao, 1973).

Interpretation as evidence for independent margins. A special interpretation of the Bayes factor is possible if the priors are "consistent". It can then be shown to be the inverse of the *evidence term* in the posterior probability of independent margins (Peltonen et al., 2003): $p(H_2|H_1,\mathbf{N}) \propto p(H_2|H_1)/C_{fsIB}$, where H_2 is a subfamily of H_1 for consistent priors, and $p(H_2|H_1)$ and $p(H_2|H_1,\mathbf{N})$ are the prior and posterior probabilities for the contingency table being generated from H_2 given that it is generated from H_1. On the right-hand side C_{fsIB} contains all terms that depend on the data. Maximizing the Bayes factor with respect to the clustering T minimizes the evidence that the margins are independent.

Interpretation as marginalized likelihood. When consistent priors are used, it turns out that the objective function (3) can also be interpreted as the marginal likelihood $p(\mathbf{N}|H_3)$ under a family H_3 that models each cluster with a separate multinomial distribution, with a Dirichlet prior α_{ty} for each cluster t (cf. Sinkkonen et al., 2002). The likelihood is marginalized over the individual models (parameter values) in H_3.

2.3. Finite Data and the Original Sequential IB

On document priors. The equations (1) are defined for any distributions $p(x)$ and $p(y|x)$. Slonim et al. (2002) tested sIB using a uniform prior over documents, $p(x) = 1/|X|$, where $|X|$ is the number of documents in the dataset. However, emphasizing long documents as more informative may sometimes be more desirable. Then the non-uniform document prior proportional to the number of words in the documents, $p(x) = n_x/N$, is an obvious candidate (n_x is the number of word occurrences in the document and N is the total number).

The objectives are asymptotically equivalent. In the limit of large co-occurrence counts, i.e., $N \to \infty$, the fsIB objective function C_{fsIB} can be shown to be equivalent to the sIB objective $I(T, Y)$ with non-uniform document priors (Peltonen et al., 2003).

For a uniform document prior such a connection is not known, however. In Section 3 we test the sIB method with both priors.

Other connections. The proposed model is additionally closely related to the asymmetric clustering model ACM (Hofmann & Puzicha, 1998). ACM uses a maximum likelihood point estimate, whereas we integrate over the word distribution parameters.

2.4. Optimization

Slonim et al. (2002) introduced a "template algorithm" for clustering with so-called *decomposable* objective functions, i.e., sums over functions that depend only on an individual cluster. Applying this template for mutual information yielded the sIB algorithm.

The same template algorithm can be applied to fsIB, since the log of (3) is decomposable. The resulting algorithm resembles sIB; only the objective function and similarity measure are replaced.

Cluster merging criterion. In sIB, an extracted document was assigned to the new cluster for which a weighted Jensen-Shannon divergence was minimized. For fsIB, we instead maximize the Bayes factor (3).

If a document $\mathbf{d}$ is extracted from some cluster t_0 and merged to a cluster t, the Bayes factor is changed by a factor of

$$d_{fsIB}(\mathbf{d}, t) = \left(\prod_y \frac{\Gamma(n_{ty} + d_y + \alpha_{ty})}{\Gamma(n_{ty} + \alpha_{ty})} \right) \cdot \frac{\Gamma(n_t + \alpha_t)}{\Gamma(n_t + |\mathbf{d}| + \alpha_t)} \tag{4}$$

times a constant (Peltonen et al., 2003). Here n_{ty} and n_t are the co-occurrence counts after the document has been extracted from t_0, d_y is the co-occurrence count of word y in the document, and $|\mathbf{d}| = \sum_y d_y$. Note that this quantity depends only on the co-occurrences in the document and the cluster t, not on the rest of the contingency table. The document is assigned to the cluster t that yields the largest value of (4).

Interpretation of the criterion. It can be shown that the cluster merging criterion (4) is asymptotically equal to the weighted Jensen-Shannon divergence (2) used in sequential IB (Peltonen et al., 2003).

Notice that when $n_{t_0 y}$ and n_{t_0} have been updated after extracting the document, the criterion (4) does not depend further on the original cluster t_0. In fact, adding a completely new document to the contingency table yields the same criterion. This allows probabilistic interpretations of the criterion.

In general, if new data $\mathbf{N}_2$ is added to the contingency table, the change in the Bayes factor is given by

$$\frac{p(\mathbf{N}, \mathbf{N}_2|H_1)}{p(\mathbf{N}, \mathbf{N}_2|H_2)} \cdot \frac{p(\mathbf{N}|H_2)}{p(\mathbf{N}|H_1)} = \frac{p(\mathbf{N}_2|H_1, \mathbf{N})}{p(\mathbf{N}_2|H_2, \mathbf{N})}\,, \tag{5}$$

i.e., it is a Bayes factor comparing H_1 and H_2 with data $\mathbf{N}_2$ after witnessing $\mathbf{N}$. For the fsIB algorithm the "new data" is the previously extracted document, i.e., $\mathbf{N}_2 = \mathbf{d}$.

Furthermore, if "consistent priors" are used, the criterion (4) is the inverse of the *new evidence* term for updating the Bayesian posterior probability of independent margins. That is, we have

$$p(H_2|H_1, \mathbf{N}, \mathbf{N}_2) = \frac{p(\mathbf{N}_2|H_1, H_2, \mathbf{N})}{p(\mathbf{N}_2|H_1, \mathbf{N})} \cdot p(H_2|H_1, \mathbf{N}) \tag{6}$$

where the first term on the right is (inverse to) the cluster merging criterion when $H_2 \subset H_1$, and the second term is independent of $\mathbf{N}_2$. The first term contains all factors that depend on the new data; minimizing it keeps the evidence of independent margins minimal.

In the case of "consistent priors", there is a third probabilistic interpretation: equation (4) is equal (up to a constant factor) to $p(\mathbf{d}|\mathbf{n}_t)$, the probability that the (unknown) multinomial distribution that generated the words $\mathbf{n}_t$ in cluster t also generated the words

in **d**, when a Dirichlet prior with parameters α_{ty} is used for the multinomial distribution.

The algorithm. Pseudo-code for the fsIB optimization algorithm is presented in Figure 1 (compare to the sIB algorithm by Slonim et al., 2002). To find a good local minimum, fsIB takes n restarts from random initial conditions (just as the sIB algorithm; multiple initializations help avoid bad local minima). For each restart, the algorithm continues until it has performed $maxL$ iterations over the data set, or until the number of cluster changes per iteration is at most a fraction ϵ of the number of objects.

Input:
$|X|$ objects to be clustered
Parameters: K, n, $maxL$, ϵ

Output:
A partition T of X into K clusters

Main Loop:
For $i = 1, \ldots, n$
  $T_i \leftarrow$ random partition of X.
  $c \leftarrow 0$, $C \leftarrow 0$, $done = FALSE$
  While not $done$
    For $j = 1, \ldots, |X|$
      Draw x_j out of $t(x_j)$
      $t^{new}(x_j) = \arg\min_{t'} d_{fsIB}(\{x_j\}, t')$
      If $t^{new}(x_j) \neq t(x_j)$ then $c \leftarrow c + 1$
      Merge x_j into $t^{new}(x_j)$
    $C \leftarrow C + 1$
    If $C \geq maxL$ or $c \leq \epsilon \cdot |X|$ then
      $done \leftarrow TRUE$
$T \leftarrow \arg\max_{T_i} C_{fsIB}(T_i)$

Figure 1. Pseudo-code for the fsIB optimization algorithm. In the context of clustering text documents, the x_j are documents with word co-occurrences $\{x_j\}$, and $t(x_j)$ and $t^{new}(x_j)$ are the old and new clusters of extracted document x_j. The overall criterion C_{fsIB} is computed by (3) and the cluster merging criterion d_{fsIB} by (4).

Convergence and complexity. The convergence result for the sIB algorithm (Slonim et al., 2002) also holds for the fsIB algorithm; since cluster changes are directly chosen to maximize the objective function, it never decreases during the iteration. Moreover, the fsIB objective is upper bounded for any dataset as long as α_{ty} and α_t are positive, since the gamma functions are then bounded from above and below. The empirical priors used in Section 3 satisfy this requirement.

Since sIB and fsIB use the same optimization method with different objective functions (which are equally complex to evaluate), both methods have the same time complexity $\mathcal{O}(K|Y|)$ per iteration. The total time complexity is bounded by $\mathcal{O}(nLK|X||Y|)$ for both methods, where n is the number of restarts, L is the number of cycles over X and K is the number of clusters. For relative speeds in practice, see Section 3.4.

3. Empirical Tests

In this section we test experimentally whether fsIB improves unsupervised clustering results. Since the sequential IB (sIB) algorithm has already been compared to several alternatives in (Slonim et al., 2002), and because sIB outperformed the alternatives, we will compare fsIB only to sIB.

3.1. The Data Sets

We used the same datasets as Slonim et al. (2002). The Twenty Newsgroups (**20NG**; Lang, 1995) and Reuters-21578 (**Reuters**; available from http://www.daviddlewis.com/resources/testcollections/reuters21578/) datasets are standard test corpora of text documents. The 20NG set consists of articles from 20 Usenet newsgroups, and the Reuters set consists of short news articles from the Reuters stream.

In both datasets, a categorization of the documents is known; for the 20NG set the newsgroups are the 20 categories. For the Reuters set news articles belong in one or more topic categories; the ten largest categories were chosen. For both datasets, the number of clusters sought equaled the number of categories. The categories themselves were not used in training, but were only used to evaluate the clustering results.

Slonim et al. (2002) additionally compared the methods on a selection of subproblems from the two corpora; here we only tested the full problems.

Preprocessing. For the 20NG set the preprocessing included removing stopwords, setting characters to lowercase, replacing digits with zeroes, ignoring non-alphanumeric characters and pruning words that only occurred once. For the Reuters set a similar but slightly more complicated preprocessing was done, including more advanced word splitting, and replacing numeric expressions with indicator tags.

After the initial preprocessing the 2000 most informative words were selected, having the largest terms

in the equation of mutual information between documents and words (Slonim et al., 2002). That is, the selected words y had the largest values of

$$\sum_x p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \ . \quad (7)$$

Note that the "informativeness" is based on a joint distribution estimate $p(x,y)$ which could be computed with either a uniform or non-uniform document prior, as described in Section 2.3. Here we used the non-uniform prior. Lastly, documents that had less than 10 words were discarded. This yielded 18656 documents for the 20NG set and 8452 for the Reuters set.

3.2. Test Setup

Comparison methods. The fsIB algorithm is compared against the sIB algorithm, with two variants for both methods. For sequential IB, both uniform (**sIB 1**) and nonuniform (**sIB 2**) document priors are tried. For fsIB, the two kinds of empirical priors described in Section 2.1 will be used, the inconsistent (**fsIB 1**) and consistent (**fsIB 2**). For all variants we set the parameters of the optimization algorithm to $maxL = 30$, $\epsilon = 0$ and $n = 15$.

Data subdivision. The fsIB algorithm is expected to be especially effective for sparse data; in the large data limit it approaches sIB 2. To test the effect of sparseness, both datasets were divided into an increasing number of subsets (from 1 to 160) by stratified sampling; with a fixed number of clusters, this leaves progressively less samples per cluster.

For each subdivision of a dataset (20NG or Reuters), each subset was clustered by all four methods. Their performances were then compared across the subsets by paired t-test.

Goodness measure. The goodness of the clustering solutions is evaluated by their micro-averaged precision (see below) with respect to human-given categories: Usenet newsgroup for the 20NG set, and news categories for the Reuters set. (In the Reuters set a document may be listed under several news categories.)

Assume first that each cluster of a clustering solution T is given the class label of the dominant class in the cluster. The clustering classifies each document to the label of its cluster. Given this classification, the micro-averaged precision of the entire clustering T is defined by

$$P(T) = \frac{\sum_c f_1(c,T)}{\sum_c f_1(c,T) + f_2(c,T)} = \frac{1}{|X|} \sum_c f_1(c,T) \quad (8)$$

where $f_1(c,T)$ is the number of documents correctly classified to class c (their true classes include c), $f_2(c,T)$ is the number of documents incorrectly classified to c and $|X|$ is the total number of documents. That is, $P(T)$ is a kind of classification success rate.

Slonim et al. (2002) introduced a thresholding strategy to increase precision at the expense of recall. The same strategy could be used for fsIB but is not used here.

3.3. Results

Figure 2 presents the micro-averaged precisions, averaged over subsets, for the 20NG and Reuters sets. As expected, performance decreases for all methods as the data becomes sparse (documents/cluster becomes small). However, the fsIB algorithm with consistent priors (fsIB 2) yields the best results on the sparsest data. As the amount of data increases the fsIB results converge toward sIB results obtained with the nonuniform prior (sIB 2). Again, this is expected due to the asymptotical equivalence of the objective functions.

For fsIB, the consistent priors are clearly better for small data sets. For sequential IB, the differences between document priors are less clear: a uniform prior is better on the small Reuters subsets, but worse on the whole set. The non-uniform prior is better on the small 20NG subsets, but worse on the whole set.

It is surprising that the prior seems to affect performance in a way that depends on the size of the data set. As explained above, the significance of the whole-set results (rightmost points in Figure 2) is not known, though, and for the single available run the algorithms may, for instance, have been caught in local minima.

Why are sIB 2 and the fsIB 1 & 2 bad for the full 20NG set? The length of Usenet articles varies a lot, so our first hypothesis was that the three algorithms, by weighting documents by their length, give some documents bad weights. Evidently this is not the case, for removing the longest and shortest five percent of the documents did not made a significant difference. Other potential reasons for the poor performance include relatively constant headers, repetition due to quotations or binary postings, or other kind of deviations from the bag of the words model. Pure chance cannot be completely ruled out either; there is only one full set,

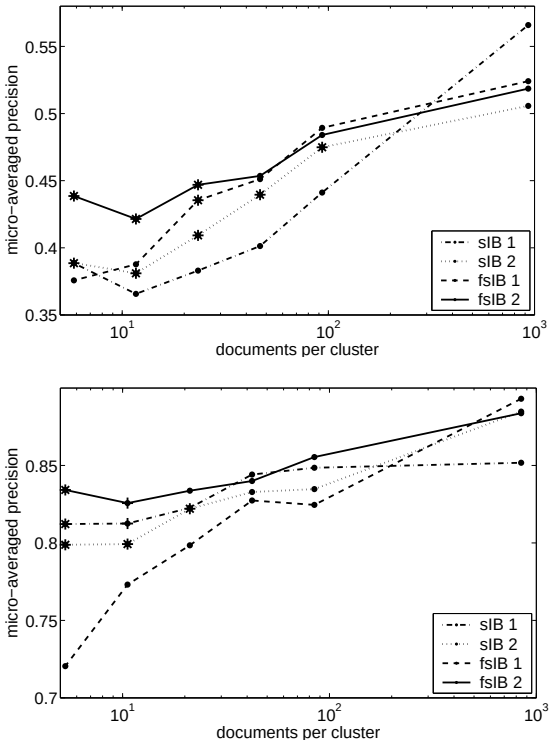

Figure 2. Performance of the methods as a function of data sparsity. Top: Twenty Newsgroups data, bottom: Reuters-21578 data, both divided into 1-160 subsets to simulate sparsity. Performance is measured as the average micro-averaged precision (eqn 8; higher is better), and data sparsity as the number of documents per cluster. Both values are averages over data subsets. The methods are: sIB with uniform document prior (sIB 1), sIB with normalized document length as document prior (sIB 2), fsIB with inconsistent cluster prior (fsIB 1), fsIB with consistent cluster prior (fsIB 2). The best method was compared to the second best, the second to the third and so on, by a two-tailed paired t-test. A point is marked with a plus sign if the t-test returned $p < 0.05$, and with an asterisk if $p < 0.01$.

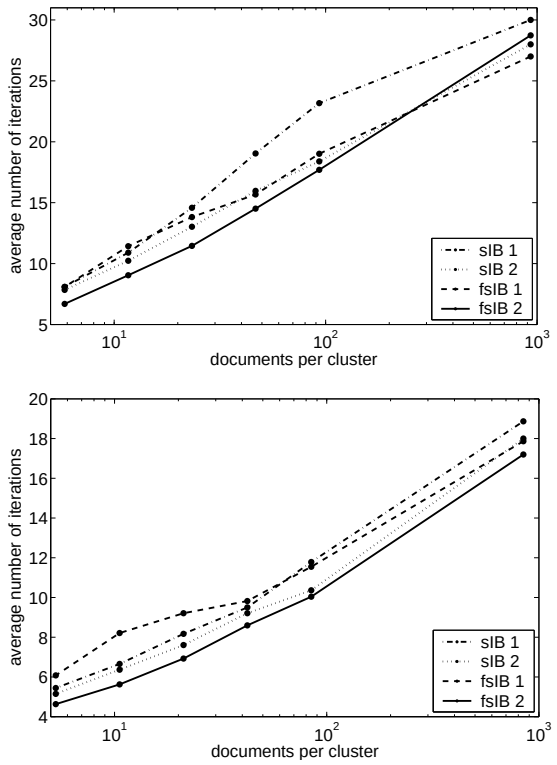

Figure 3. Convergence speed measured by the number of iterations over the data, averaged over restarts and subsets. Top: Twenty Newsgroups data, bottom: Reuters-21578 data.

and hence no significance testing can be done.

Why is fsIB 1 bad for the small subsets of Reuters data? With small data sets, the relatively strong prior in the denominator of the fsIB 1 cost favors uniform cluster sizes, which causes difficulties if the real cluster sizes diverge. And indeed, the sizes of the clusters extracted from the full Reuters set are unequal but similar irrespective of the prior (fsIB 1 vs. fsIB 2), while from the small Reuters subsets fsIB 1 finds much more uniform-sized clusters than fsIB 2.

3.4. Convergence Speed

Figure 3 presents convergence speeds for the algorithms, in average numbers of iterations over the respective datasets. For all algorithms, the convergence seems roughly logarithmic with respect to the number of documents per cluster.

The sIB with uniform document prior (sIB 1) converges on average slightly slower than sIB 2. The fsIB with consistent priors (fsIB 2) on average converges with the fewest iterations, except on the whole 20NG set (rightmost point). However, since the log-Gamma function in fsIB is much slower to compute than the logarithm in sIB, fsIB still takes more time to run than sIB (roughly 2.5 times as much).

4. Conclusions and Discussion

The finite sequential information bottleneck (fsIB) algorithm gives a probabilistic Bayes factor interpretation to distributional crisp clustering by the sequential Information Bottleneck method. Asymptotically, the

two objective functions become equivalent. fsIB inherits its desirable properties such as convergence, computational complexity and a thresholding strategy from sequential IB.

While Bayes factors are commonly used in "Bayesian hypothesis testing" to compare two alternative model families, it is not common to use them as criteria for optimizing models. We are not aware of their use for clustering before the related continuous-space model (Sinkkonen et al., 2002). Their main advantage is that uncertainty in the parameter values can be properly and easily taken into account. There are two potential difficulties in their use. First, although they are easily computed in closed form for our models, they may in general be hard to compute. Second, it may be hard to choose suitable priors. This may not be a severe problem, however, since even simple choices outperformed alternative methods, and more sophisticated hierarchical priors would probably be even better.

In empirical tests, fsIB yields significantly improved clustering performance on small (sparse) data sets. On large sets it performs comparably to sequential IB; the insignificant relative goodness varies with the data.

More experiments are required to find out which kinds of data fsIB is best for.

Of the two tested fsIB alternatives, the one with "consistent" priors (in the sense discussed in Section 2.1) was clearly better. This suggests a more general conjecture to favor consistent priors. In this paper the strengths of the priors were pre-chosen (they were not optimized in the experiments)—better results could be obtained by making this more flexible.

Although this paper only considered one-sided clustering, the Bayes factor objective function is straightforwardly applicable to two-sided clustering.

Acknowledgments

This work was supported by the Academy of Finland, decision number 79017.

References

Blei, D., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.

Buntine, W. (2002). Variational extensions to EM and multinomial PCA. In T. Elomaa, H. Mannila and H. Toivonen (Eds.), *Proceedings of the ECML'02, 13th European Conference on Machine Learning*, Lecture Notes in Artificial Intelligence 2430, 23–34. Berlin: Springer.

Gilula, Z. (1986). Grouping and association in contingency tables: An exploratory canonical correlation approach. *Journal of the American Statistical Association, 81*, 773–779.

Good, I. J. (1976). On the application of symmetric Dirichlet distributions and their mixtures to contingency tables. *Annals of Statistics, 4*, 1159–1189.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, 289–296. San Francisco, CA: Morgan Kaufmann Publishers.

Hofmann, T., & Puzicha, J. (1998). *Statistical models for co-occurrence data* (A.I. Memo 1625). MIT.

Lang, K. (1995). NewsWeeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, 331–339. San Mateo, CA: Morgan Kaufmann Publishers.

Peltonen, J., Sinkkonen, J., & Kaski, S. (2003). *Finite sequential information bottleneck (fsIB)* (Technical Report A74). Helsinki University of Technology, Publications in Computer and Information Science, Espoo, Finland.

Rao, C. R. (1973). *Linear statistical inference and its applications.* Wiley. 2nd edition, Originally published 1965.

Sinkkonen, J., Kaski, S., & Nikkilä, J. (2002). Discriminative clustering: Optimal contingency tables by learning metrics. In T. Elomaa, H. Mannila and H. Toivonen (Eds.), *Proceedings of the ECML'02, 13th European Conference on Machine Learning*, 418–430. Berlin: Springer.

Slonim, N., Friedman, N., & Tishby, N. (2002). Unsupervised document classification using sequential information maximization. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 129–136. ACM Press.

Slonim, N., & Weiss, Y. (2002). Maximum likelihood and the information bottleneck. In *Advances in neural information processing systems 14*. Cambridge, MA: MIT Press.

Tishby, N., Pereira, F. C., & Bialek, W. (1999). The information bottleneck method. In *37th Annual Allerton Conference on Communication, Control, and Computing*, 368–377. Urbana, Illinois.