**Behaviormetrika manuscript No.**
(will be inserted by the editor)

# Experiments with Learning Graphical Models on Text

**Joan Capdevila · He Zhao · François Petitjean · Wray Buntine**

Received: date / Accepted: date

**Abstract** A rich variety of models are now in use for unsupervised modelling of text documents, and, in particular, a rich variety of graphical models exist, with and without latent variables. To date, there is inadequate understanding about the comparative performance of these, partly because they are subtly different, and they have been proposed and evaluated in different contexts. This paper reports on our experiments with a representative set of state of the art models: chordal graphs, matrix factorisation, and hierarchical latent tree models. For the chordal graphs we use different scoring functions. For matrix factorisation models we use different hierarchical priors, asymmetric priors on components. We use Boolean matrix factorisation rather than topic models so we can do comparable evaluations. The experiments perform a number of evaluations: probability for each document, omni-directional prediction which predicts different variables, and anomaly detection. We find that matrix factorisation performed well at anomaly detection but poorly on the prediction task. Chordal graph learning performed the best generally, and probably due to its lower bias, often out-performed hierarchical latent trees.

**Keywords** graphical models · document analysis · unsupervised learning · matrix factorisation · latent variables · evaluation

## 1 Introduction

Research in previous decades has led to an embarrassment of riches when it comes to alternative unsupervised graphical models for text documents. There

Joan Capdevila E-mail: jc@ac.upc.edu
C6E201, 1-3 Jordi Girona, 08034 Barcelona, Spain

He Zhao E-mail: he.zhao@monash.edu
François Petitjean E-mail: francois.petitjean@monash.edu
Wray Buntine E-mail: wray.buntine@monash.edu
25 Exhibition walk, 3800 Monash University, Australia

are early clustering models (Aggarwal and Zhai, 2012), non-probabilistic semantic spaces built using sliding windows to build co-occurrence statistics (Lund and Burgess, 1996), many varieties of word and sentence embeddings built using deep neural networks (Mikolov et al, 2013), different kinds of topic models (Blei, 2012), augmented with document structure, bibliographic and semantic relationships (Lim and Buntine, 2016), and related areas of matrix factorisation (Cai et al, 2011) and tree-structured graphical models (Liu et al, 2014). This list only considers models based on bag-of-words and similar assumptions, and does not begin to consider those also modelling the sentence structure built using deep neural networks, sometimes supported by parse structure (Collobert and Weston, 2008; Socher et al, 2012).

Earliest graphical models were Bayesian networks over discrete variables, Gaussians or a mixture and a variety of algorithms have been developed given the convenient exponential family nature of the models (Heckerman and Chickering, 1995). Standard implementations, however, were usually restricted to less than 100 variables. More recently, improved data structures and algorithms have been developed that allow models to be built with more variables. Branch and bound techniques allow best model search (Suzuki and Kawahara, 2017), but for thousands of discrete variables one uses cached local search and restriction to chordal graphs (Petitjean and Webb, 2016). The algorithm for doing this is known as *Chordalysis* and it allows learning graphical models on text corpora with vocabulary sizes on the order of thousands. However, *Chordalysis* was initially proposed for association discovery and the metrics (i.e. SMT, G-tests) sought to minimize the probability of false discoveries. Therefore, new metrics are required when using Chordalysis to learn statistical models on text aiming to maximise predictability.

The research question we seek answers for in this work is as follows: *could Chordalysis models be a competitive alternative to existing latent variable models on text prediction?*

In this paper we restrict ourselves to methods that correspond to graphical models using a bag-of-words representation. However, they still have many variations: data can be Boolean or count, bagged or sequential, different forms of latent variables can be included, and document length can be modelled or excluded. To keep things comparable, we restrict our experiments to the Boolean representation of bag of words data, and select/modify algorithms accordingly. Topic models are comparable to matrix factorisation (Gaussier and Goutte, 2005) (ignoring document length modelling) so we use matrix factorisation (Zhou, 2015) instead of topic modelling. Another type of graphical model built on binarised text is the Hierarchical latent tree analysis (HLTA) (Liu et al, 2014; Chen et al, 2017). These graphical models yield intriguing "local" topics, that only interact with a limited set of variables.

To the best of our knowledge, no prior work has looked more broadly at comparing these models. The contributions of this paper therefore are as follows:

– we do the first performance evaluation of three different bag-of-word document models across three different tasks;
– we introduce the anomaly detection task, well known in machine learning but not used for these kinds of models;
– we show that matrix factorisation performs well on anomaly detection but not so well on prediction
– we show that Chordalysis generally beats HLTA but believe that is explained by the lower bias available with Chordalysis.

In Section 2 we review a number of different graphical models suitable for text and explain our particular choices. In Section 3 we discuss the experimental evaluation. There are a number of subtleties here so we discuss alternatives. The results of experiments are reported in Sections 4–6.

## 2 Graphical Models for Text

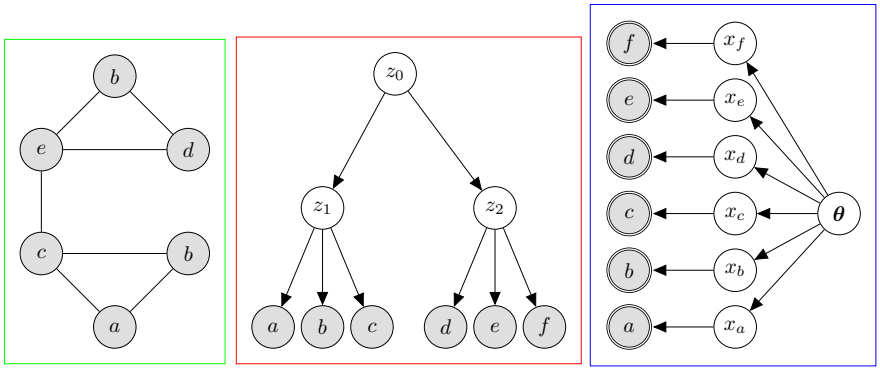



**Fig. 1** Examples of the three models for a single document. See text for explanation.

The three graphical models selected for comparison, Chordalysis, HLTA and BPFA are depicted in Figure 1. This figure ignores other model parameters and hyper-parameters, and has six simple words "a", "b", "c", "d", "e" and "f". Green is Chordalysis where the structure is learnt but restricted to be a chordal graph (which has many valid variable orders). Red is HLTA where the structure is learnt but restricted to be a tree with three latent variable ($z_1$ etc.). Blue is BPFA where the structure is fixed, the Booleans are computed from the latent counts ($x_a$ etc.), and correlations are got indirectly via the document topic proportions $\boldsymbol{\theta}$.

### 2.1 *Chordalysis* Graphical Models

*Chordalysis* (Petitjean and Webb, 2015) is a forward selection algorithm to learn the structure of probabilistic models from discrete data. It leverages the decomposability of chordal graphs to scale the learning of statistical models up to thousands of variables. This method also requires a scoring function that is incremental with changes in the graph, and several scores can be used. The Subfamilywise Multiple Testing (SMT) score (Petitjean and Webb, 2016) was initially proposed for Chordalysis to keep the probability of making at least one false discovery low. The well known BDeu score is not used because of well-understood problems with its application (Suzuki, 2017). However, the SMT score is too conservative for prediction purposes and other existing metrics such as the Bayesian Information Criteria (BIC) seems more suitable for the task. Similarly, the Quotient Normalised Maximum Likelihood (QNML) (Silander, 2016) has been lately proposed as a competitive score in predictive terms. As we shown in Figure 2, the QNML metric learns models with higher number of free parameters (right) which have higher held-out likelihood (left) for different sizes of the training set.

After the structure is learnt, the variable order is set as per the perfect elimination order, which exists in any chordal graph, and by decreasing TF-IDF score. Then probabilities are estimated from the data using m-estimation, which smooths the maximum likelihood probabilities by introducing some pseudo-counts $m$. On top of that, we also use the classical Back-off scheme introduced for Bayesian Classifiers (Friedman et al, 1997). This technique first builds a probability tree for each Conditional Probability Table (CPT) in the network based on a specific variable order. Nodes at the leafs of the tree correspond to the entries in the CPT, while the internal nodes represent the corresponding marginal values. Then, probabilities for each cell are computed as weighted averages between the m-estimates at the leaves and its parent probabilities (i.e. marginal values) in the tree. The weighting controls how much we back off to the parent's probability through the parameter $N_o$ and the counts for that cell. This enables smoothing the probability distribution to deal with data scarcity when the cliques of the network are large.

### 2.2 Hierarchical Latent Tree Analysis (HLTA)

Hierarchical latent tree analysis (*HLTA*) (Liu et al, 2014; Chen et al, 2017) has recently been developed that yield intriguing "local" topics, that only interact with a limited set of variables. This is achieved by introducing a hierarchy of Boolean latent variables, so that the final model is a tree with the observed words, represented as present/absent, at the leaves. HLTA is comparable to Boolean matrix factorisation, and has been scaled to work with thousands of Boolean variables. The hierarchical nature of the latents leads to insightful structures that seem inherently more interpretable than standard topic models (e.g. Chen et al, 2017, Figure 8). The HLTA models, being restricted to a

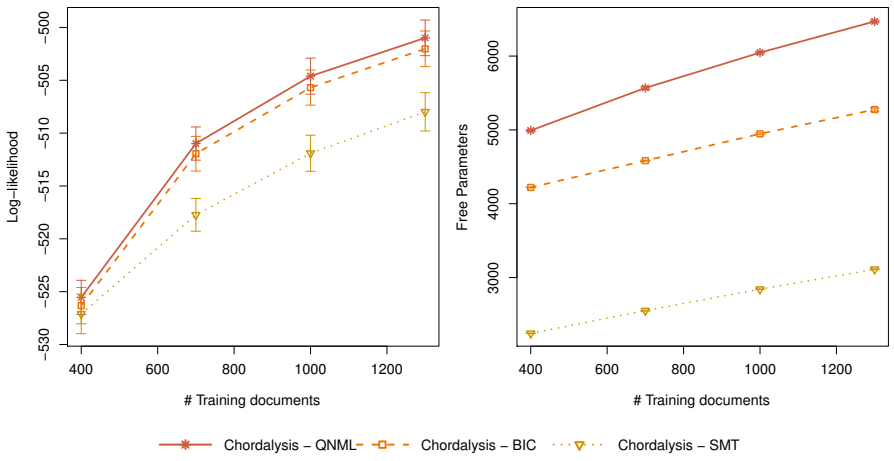



**Fig. 2** Metrics comparison for Chordalysis models in NIPS dataset.

Boolean vector of word "presence/absence", due to their algorithm, acts as a lowest common denominator for us. That is, we can only use other algorithms working with Boolean vectors.

HLTA was compared with a number of hierarchical topic models by Chen et al (2017). This has the disadvantage that the topic models are not run natively: they are being trained on Boolean data for which they were not designed. Moreover, the nHDP algorithm used has only demonstrated a marginal improvement in perplexity (Paisley et al, 2015) over HDP-LDA (Teh et al, 2006). Significantly more improvement is readily gained by using superior algorithms for the HDP-LDA training (Buntine and Mishra, 2014). Of course, nHDP has superior comprehensibility, with its hierarchical topics, however, we are evaluating various performance metrics, so we will not use the hierarchical topic modelling methods compared with by Chen et al (2017).

The algorithm to learn these trees from data operates as follows. First, leaves are grouped using a "common latent Boolean factor" statistical test, latent Boolean factors are added and then the grouping process repeated, with progressive expectation-maximisation runs to re-estimate probability tables as the trees are grown. This can be done by using all the data (BATCH) or with less accurate mini-batch updating of parameters (STEP) on large datasets.

### 2.3 Bernoulli-Poisson Factor Analysis (BPFA)

Zhou (2015) introduced the Bernoulli-Poisson link technique to extend Poisson factorisation methods to Boolean data. With this we can take the Poisson factorisation model of Zhou et al (2012), which is a flexible model, and add the Bernoulli-Poisson link on top to obtain a representative factor analysis

method for Boolean vector data. We refer to this as Bernoulli-Poisson factor analysis (*BPFA*).

The BPFA model does matrix factorisation to create matrices $\mathbf{\Phi}$ (the loading matrix) and $\mathbf{\Theta}$ (the factor matrix) with the following probability forms:

$$\begin{aligned} \boldsymbol{\phi}_k &\sim \text{Dirichlet}_V(\beta \mathbf{1}) & \theta_{d,k} &\sim \text{Gamma}\left(g_k, \tfrac{q_k}{1-q_k}\right) \\ x_{d,v} &\sim \text{Poisson}\left(\textstyle\sum_{k=1}^{K} \theta_{d,k}\phi_{k,v}\right) & y_{d,v} &= 1_{x_{d,v} \geq 1} \ , \end{aligned} \tag{1}$$

where $K$ is the number of topics, $d$ indexes documents, $V$ is the size of the vocabulary, and $\beta$, $\mathbf{g}$ and $\mathbf{q}$ (where $0 < q_k < 1$) are hyper-parameters with their own priors. Note $\mathbf{\Phi}$ is made up of rows $\boldsymbol{\phi}_k$ which normalise. This parameterisation means that BPFA is comparable to HDP-LDA in terms of hyper-parameters, but with a fixed dimension for topics. More details can be found in Zhou et al (2012); Hu et al (2016). The observed data is the Boolean matrix $\mathbf{Y}$ which has a corresponding latent count matrix $\mathbf{X}$.

The algorithm to learn the model parameters from data goes as follows. After random initialisation, a Gibbs sampler iterates over parameters and latent counts and topic values. The sampler is built on the conditional exponential family structure of the model, and in some cases using data augmentation to create fast simple sampling. Hyper-parameters are similarly sampled.

### 2.4 Other Models

In a very different manner to HLTA, but perhaps with a similar high-level goal, focused topic models allow focusing on limited vocabularies per topic, as implemented in latent IBP compound Dirichlet allocation (Archambeau et al, 2015). This means the words distribution for a topic is now limited to a small subset of words, perhaps 5-10% of the full vocabulary. Current implementations, however, neither deal with Boolean data nor matrix factorisation and thus we could not perform the comparisons.

As mentioned in the introduction, there are also rich and high-performing classes of deep neural network models. Most, however, use richer document structures and build more complex document architectures, not simple graphical models, so again we have not included them here.

## 3 Experimental Methodology

We first discuss general evaluation methods. Because of the variety of different algorithms, this turns out to be a challenging, so we discuss the literature and report our conclusions on how evaluations should be done. We then present the software implementations, datasets and parameter setting for the experimentation.

### 3.1 Evaluation Methods

Perplexity for topic models is a measure of the *predictive log-likelihood* of a held-out document scaled to a "per-word" measure. Various methods exist for estimating it where the computation is very different (Wallach et al, 2009). The only reliable technique giving an unbiased estimate of log-likelihood is the left-to-right algorithm of Wallach et al (2009). Given a sequence of words $w_1, w_2, ...w_L$, it estimates $p(w_l|w_1, w_2, ..., w_{l-1})$ in turn using MCMC for $l = 1, ..., L$. It is computationally intense, however, and the preferred method instead is document completion, which estimates the topic proportions $\boldsymbol{\theta}$ on $w_1, w_2, ...w_{L/2}$, and then computes $p(w_{L/2+1}, ..., w_{l-1}|\boldsymbol{\theta})$ exactly. This does not give an unbiased estimate of $p(w_{L/2+1}, ..., w_{l-1})$ because it is informed by an estimate for $\boldsymbol{\theta}$, but it is *comparatively* unbiased for different topic modelling algorithms/methods/models that all similarly require an estimate of $\boldsymbol{\theta}$.

In our case, a document is represented as a Boolean vector, and words are not supplied in a sequence. For matrix factorisation, things are more subtle, but a similar technique to document completion has been proposed by Zhou et al (2012). This splits the 1-valued entries of the document vector $\mathbf{y}$ into two parts. The first part is zeroed, and from this the document factor vector $\boldsymbol{\theta}$ is estimated. Then the second part is zeroed and the document vector can have its probability computing exactly using $\boldsymbol{\theta}$. As before this is biased, though comparatively unbiased. So it is suitable for their comparison of Poisson factorisation models, but is unsuitable for our comparison with Chordalysis or HLTA.

So, could a left-to-right algorithm be developed for estimating log probabilities of Poisson factorisation models? While in principle the same sequential logic applies, gradually adding words (which may be 0 or 1) to a partially filled document vector, there is a catch that makes it computationally much harder. Computation of the document likelihoods is non-trivial in the case of a partial document vector. The normalising constant for a Poisson is $e^{-\lambda}$ for rate $\lambda$. For a full data vector, the product of these over all features is, from Equation (1)

$$\left(\prod_{v=1}^{V} e^{-\sum_{k=1}^{K}\theta_{d,k}\phi_{k,v}}\right) = e^{-\sum_{v=1}^{V}\sum_{k=1}^{K}\theta_{d,k}\phi_{k,v}} = e^{-\sum_{k=1}^{K}\theta_{d,k}} \ .$$

Thus, with a full data vector, exponential terms in $\phi_{k,v}$ disappear so the likelihood is Dirichlet on $\boldsymbol{\phi}_k$. Not so for a partial data vector!

Thus, for now, we claim that *efficient unbiased estimation of the probability of a document vector for Poisson matrix factorisation is challenging.* We avoid it in the experiments.

Another common evaluation method is to do link prediction (Zhou, 2015). The idea is to hold out some of the variables (which may be positive or negative), and then evaluate how well their occurrence is predicted from the remainder of the record. For this to be done correctly, the missing link/variable needs to be made temporarily "missing", and the various models ran to predict

its probability of occurrence. This computation is not always done correctly: some researchers simple zero the variable rather than making it missing. We refer to this task as *omni-directional learning*, as the task is to do predictive modelling, but on a random selection of variables, rather than on a single target variable as is done for classification.

A popular task is to use the topics or factors derived from the matrix factorisation or topic model as features for a classification algorithm, for instance, a support vector machine (Buntine and Jakulin, 2004). In our context, this is not realistic for Chordalysis, there is no vector of factors, so we have not done it. The omni-directional learning task is a more direct replacement for the use of the models in classification.

Another earlier task was to use the models for information retrieval, for instance for topic models (Wei and Croft, 2006). The results here have not stood the test of time against the onslaught of the BM25 paradigm (Robertson and Zaragoza, 2009). The conceptual task of retrieving information with suitable "aboutness" or relevance to query words is sufficiently ill-defined that we believe better understanding is needed before suitable use of general document models can be used for information retrieval.

A final task we consider is anomaly detection (Chandola et al, 2009), an important task in security and engineering domains for instance. This is a broad area but we consider the problem of point anomaly detection (whether a single data item or document as an anomaly). There are a broad number of techniques in use, and we use ranking by log probability (lower is more likely to be an anomaly) as a straw-man algorithm to compare with. Note text anomaly detection is more challenging because of the huge number of variables. Clustering is sometimes used for collective anomaly detection with text, a different task to point anomaly detection.

Thus we use three different evaluation protocols, briefly described here, but more detail of implementation is given later.

Log-likelihood: Simple measure of predictive probability for documents held out from the training set. Not done for BPFA.

Omni-directional prediction: for each document, a variable (word) is drawn at random from a candidate set and then a prediction is made for it (is it in or out of the document). This can be reported in terms of AUC or root mean square error (measuring quality of the probability predictions).

Anomaly detection: an infrequent subclass of documents are held out from training and then added to a test set and fed to the model. The subclass are presumed the anomalies in the test set, and are presumed to be low probability documents. Log probability gives a base ranking to predict if it is an anomaly and other derived measures can be used. Which ever measure is used, it can be evaluated with AUC.

### 3.2 Implementation

We modified the Java code of Chordalysis[1] to include QNML and BIC. We used the existing Java code of HLTA[2]. The Matlab code for the Bernoulli matrix factorisation was originally reported in Hu et al (2016) and some code was added to estimate hyper-parameters and some computational speed-ups were done. The three evaluations were done in Java and Matlab respectively and care was taken to ensure standardisation across implementations.

Note that most of the evaluations for HLTA and Chordalysis are simple to implement because of their convenient structure as simple Bayesian networks. Suppose that a document is represented by a Boolean vector $\mathbf{y}$, and if entry $v$, $y_v$, is converted to a missing value, this is denoted as $\mathbf{y}^{-v}$. Then computation of the measures works as follows, given a specific model represented by $M$.

Log-likelihood: Compute $\log Pr(\mathbf{y}|M)$. Not done for BPFA.

Omni-directional prediction: We have a candidate set of words $S$. For each $v \in S$ compute $Pr(y_v|\mathbf{y}^{-v}, M)$, and note the correct value $y_v$ is given in the data. That gives a set of $|S|$ scores calibrated as probabilities. So use these and by changing the threshold $q$ ($0 < q < 1$), compute AUC. Alternatively, compute root mean square error by averaging $(y_v - Pr(y_v|\mathbf{y}^{-v}, M))^2$ across all words $S$ and all test documents, and then reporting the square root.

Anomaly detection: We tested various scores based on $\log Pr(\mathbf{y}|M)$ or $Pr(y_v|\mathbf{y}^{-v}, M)$ for $v \in S$. These are reported in Section 6. For BFFA, scores do not need to be unbiased, so $\log Pr(\mathbf{y}|M)$ can be recorded for an aposterior sample of $\boldsymbol{\theta}$ (the document factors).

The harder computations here are for BPFA. We argued previously that estimating $\log Pr(\mathbf{y}|M)$ in an unbiased manner efficiently is an open question. To estimate predictive probabilities such as $Pr(y_v|\mathbf{y}^{-v}, M)$, however, is feasible. Note it is well known how to construct a Gibbs sampler for a given document, for $\boldsymbol{\theta}_d$ and $\mathbf{x}_d$, and also the case when one of the words $y_{d,v}$ are missing. The following formulas are then recorded during the respective MCMC runs and an estimate made:

$$\lambda_{d,v} = \sum_{k=1}^{K} \phi_{v,k}\theta_{d,k}$$

$$Pr(y_v = 1|\mathbf{y}^{-v}, M) = \left(1 - e^{-\lambda_{d,v}}\right)$$

Omni-directional prediction and anomaly detection for BPFA can be evaluated with these.

### 3.3 Datasets

We selected three regular and three short text corpora for the experimentation. Collections were preprocessed with the text mining tool assembled in

[1] `https://github.com/fpetitjean/Chordalysis`

[2] `https://github.com/kmpoon/hlta`

Scala included in the HLTA software[3], which is suited to build Boolean vector data. For each collection, we tokenised text strings by space, lower-cased tokens, normalised them according to the Normalization Form KC (NFKC), removed stopwords based on the Lewis list and filtered out words with less than 3 characters. From the resulting vocabularies, we selected the top-500 and top-2000 words with highest TF-IDF score (the raw counts of a term normalised by the negative logarithm of the fraction of documents that contain that term) to build two vocabularies for each collection. All datasets were tokenised and binarised based on these vocabularies and documents without any word were removed. The final Boolean datasets are available from `https://doi.org/BLINDED-URL` and have the following features:

NIPS: consists of 1,740 conference papers published at NIPS between 1988 and 1999[4].

20NG: 20 Newsgroups, consists of 18,828 news articles and each article is in one of 20 categories[5]. An article has on average 65 different words.

NYT: New York Times Annotated Corpus supplied by the Linguistic Data Consortium[6]. It contains 1,855,658 news articles. An article has on average 196 different words.

WS: Web Snippet, used by Li et al (2016), contains 12,327 web search snippets and each snippet belongs to one of 8 categories. Documents are typically 15 words long before reducing the vocabulary.

TMN: Tag My News, consists of 32,573 English RSS news snippets from Tag My News, used by Nguyen et al (2015). Belonging to one of 7 categories, each snippet contains a title and a short description, average length 18 words.

Twitter: is extracted in 2011 and 2012 microblog tracks at Text REtrieval Conference (TREC) 3, preprocessed by Yin and Wang (2014). It has 11,109 tweets in total and a tweet contains 21 words on average.

In the likelihood and omni-directional prediction experiments, we looked at the performance of the different graphical models as function of the amount of training data. For each dataset, we randomly generated four training splits of different size and evaluated the trained models in a held-out set, which we kept the same for all training splits.

In the omni-directional prediction task, we held-out some words from the test set. In particular, we have randomly selected $S = 10$ words per test document for all datasets except for NIPS, in which we chose $S = 50$ given that the test set was smaller and there was too much variance in the results.

In the anomaly detection task, we used the 20Newsgroups and the WS dataset given that both are labeled and they are a good representatives of long and short text. For this task, datasets were split in the classical 80% training, 20% testing framework and the anomalous class was held-out from

[3] `https://github.com/kmpoon/hlta`

[4] `http://www.cs.nyu.edu/~roweis/data.html`

[5] `http://qwone.com/~jason/20Newsgroups`

[6] `http://catalog.ldc.upenn.edu/LDC2008T19`

**Table 1** Model hyper-parameters and algorithm parameters.

| Chord - BIC | Chord - QNML | Chord - SMT |
|---|---|---|
| $K_{max} = 20$ | $K_{max} = 20$ | $p = 0.05$ |
| $N_o = 5$ | $N_o = 5$ | $N_o = 5$ |
| $m_{est} = 0.5$ | $m_{est} = 0.5$ | $m_{est} = 0.5$ |
| | | $p_{err} = 0.05$ |
| **BPFA** | **HLTA - Batch** | **HLTA - Step** |
| $\beta \sim \text{Gamma}(1, 1)$ | max-EM-steps: 50 | max-EM-steps: 50 |
| $q_k \sim Beta(1/K, (1 - 1/K))$ | num-EM-starts: 5 | num-EM-starts: 5 |
| $g_k \sim \text{Gamma}(1, 1)$ | EM-threshold: 0.01 | EM-threshold: 0.01 |
| $K = 200$ | UD-test-threshold: 3 | UD-test-threshold: 3 |
| train-burnin: 500 | max-island: 10 | max-island: 10 |
| train-collect: 500 | max-top: 15 | max-top: 15 |
| test-burnin: 100 | Global-batch-size: 1,000 | |
| test-collect: 100 | Global-max-epochs: 10 | |
| | Global-max-EM-steps: 128 | |
| | Structural-batch-size: 1,000/10,000 | |

the training set. To do a fair comparison, we report results by holding out each category in the collection.

In all tasks, each experiment was performed 5 times and different training-test splits were randomly generated at each repetition.

### 3.4 Model parameters

Next, we report all model parameters set in this experimentation. A full summary can be found in Table 1.

For the Chordalysis models based on BIC and QNML, we set a safety parameter limiting the tree-width of the network $K_{max}$ equal to 20 to avoid models with big cliques. Nonetheless, this value was never reached. For Chordalysis with the SMT score, we specified the maximum family-wise error rate $p_{err} = 0.05$. For all three Chordalysis models, we use the simple Back-Off estimates introduced for Bayesian Classifiers (Friedman et al, 1997) that computes each cell in the conditional probability table as a weighted average between the cell itself and its parents, in the probability tree. $N_0$ controls how much we back-off to the parent estimate, being 0, no back-off and $\infty$ complete back off to the parent value. As in Friedman et al (1997), we use $N_0 = 5$. Moreover, we also smooth each cell in the CPT with m-estimates with parameter $m_{est} = 0.5$.

For HLTA, we used the default values reported by Chen et al (2017), except for the structural-batch-size parameter, which we set to 1,000 in the small datasets (20Newsgroups, WS, Twitter) and to 10,000 in the large ones (TMN, NYT).

For BPFA, all hyper-parameters were sampled using benign priors using standard augmented Gibbs samplers. Details of the priors are in the table.

## 4 Likelihood Experiments

In the document likelihood experiments, we consider Chordalysis and HLTA models. After training each model in the corresponding data split, we report the per-document log-likelihood on the test set. Computing the log-likelihood on the set of held-out documents consists in a inference task on the discrete graphical model.

Results in Figure 3 show that graphical models learned out of Chordalysis give higher log-likelihood to held-out documents than HLTA, specially when the training data is larger. Full plots in all cases are reported in the Appendix. This can be explained by Figure 4 where we see that the number of parameters in Chordalysis grows with the dataset size. Also, we see that the stepwise version of HLTA, HLTA-STEP, harms performance over the batch version considerably in some cases.

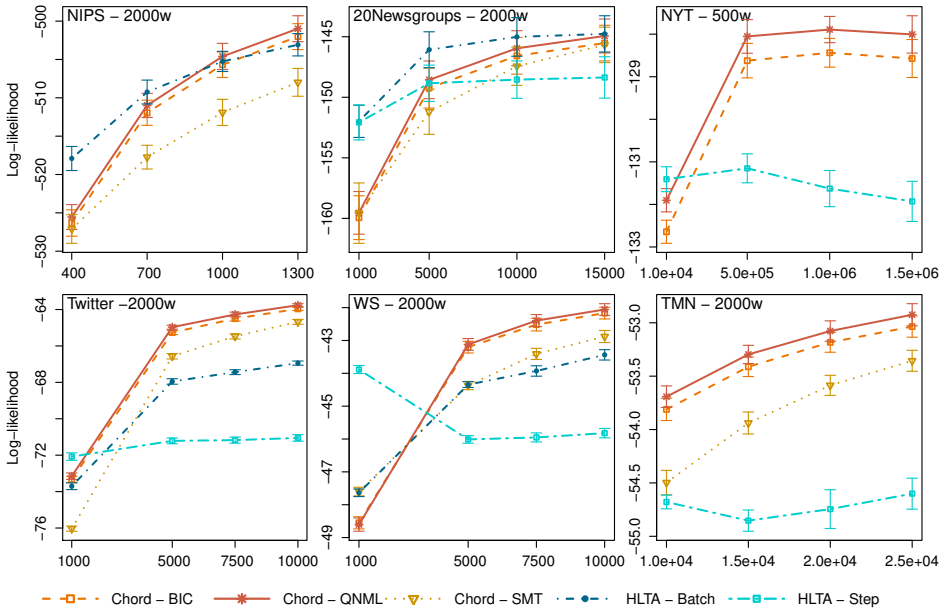



**Fig. 3** Per-document log-likelihood as function of training data.

## 5 Omni-directional Prediction Experiments

For each document in the test set we randomly hold out $S$ words ($S = 50$ in NIPS, $S = 10$ in the rest) and predict their presence or absence given all other words in the document and the graphical model.

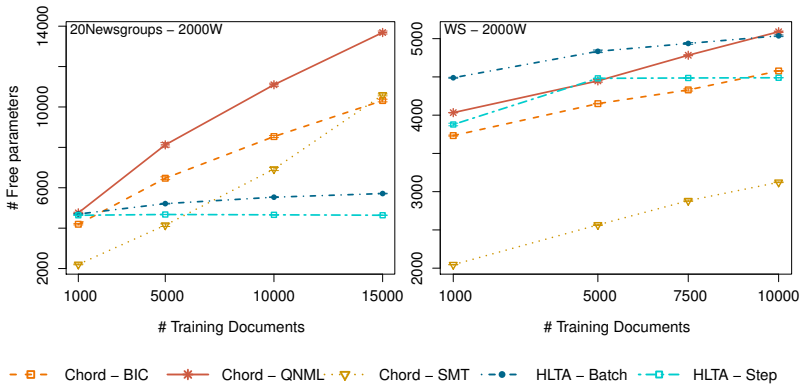



**Fig. 4** Number of Parameters as function of training data.

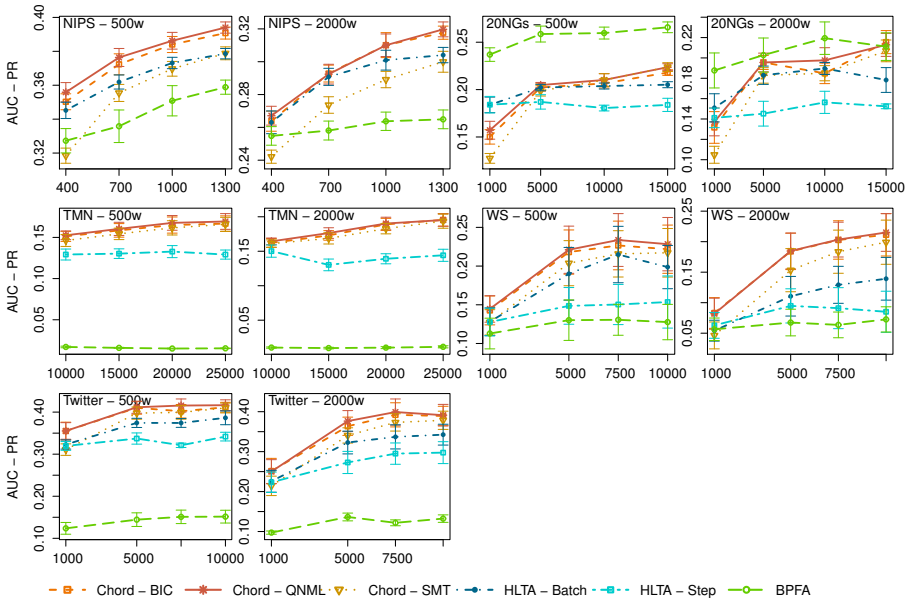



**Fig. 5** AUC-PR scores in the task of Omni-directional prediction.

Predictions are compared to the true word labels (present or absent) and assessed in terms of Area Under the Precision Recall Curve (AUC-PR). The choice of AUC-PR instead of AUC-ROC is motivated by the skewness in the dataset, i.e the "anomaly" class is unlikely (Davis and Goadrich, 2006). We also compared them in terms of Root Mean Square Error (RMSE) and see that both scores brings us similar conclusions.

Figure 5 plots the AUC-PR score for 5 different datasets with vocabulary sizes of 500 and 2,000 words. In this task, we compare all 6 models in scope

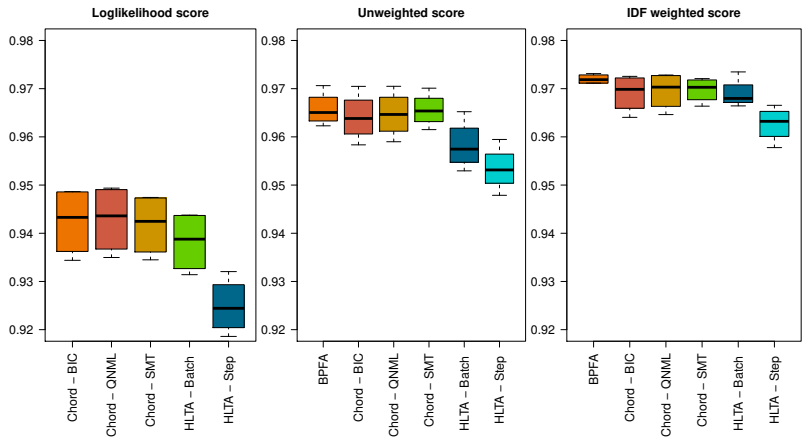



**Fig. 6** AUC-PR boxplot for the anomaly class Health in WS dataset for different scores.

and evaluate their prediction capabilities. Full plots in all cases are reported in the Appendix.

We observe that Chordalysis models outperform in four out of five datasets. Of the Chordalysis variants, QNML is marginally superior. The performance of BPFA is quite poor, for short text particularly.

## 6 Anomaly Detection Experiments

A classical statistical approach to assess how anomalous a test document is w.r.t. a set of "nominal" documents is to compute its likelihood under a null model which has been learned out of the "nominal" set (Chandola et al, 2009). Next, we aim to compare the different graphical models on the task of anomaly detection.

However, we first show that the straightforward use of log-likelihood as a score for anomaly detection is not enough in the context of text data, where documents have different lengths and some features (i.e. words) might be more discriminative than others. This motivates the development of a tailored score for this task, which despite being based on likelihood it also normalises by the document length and its weights each word by its IDF.

In Figure 6, we show the detection performance for three different scores in uncovering the anomalous class in the WS dataset. This is for a representative class, and note corresponding comparative results were obtained in all cases. That is, the relative performance of the three scores was consistent across anomalous classes. The first score corresponds to the naive log-likelihood, which can be computed for all models except BPFA. The second score is built from the probabilities average over all the words that are present in the document. The last score weights these probabilities with the Inverse Document Frequency (IDF). From the first to the second score, we correct the fact that longer documents will always be more anomalous, whereas from

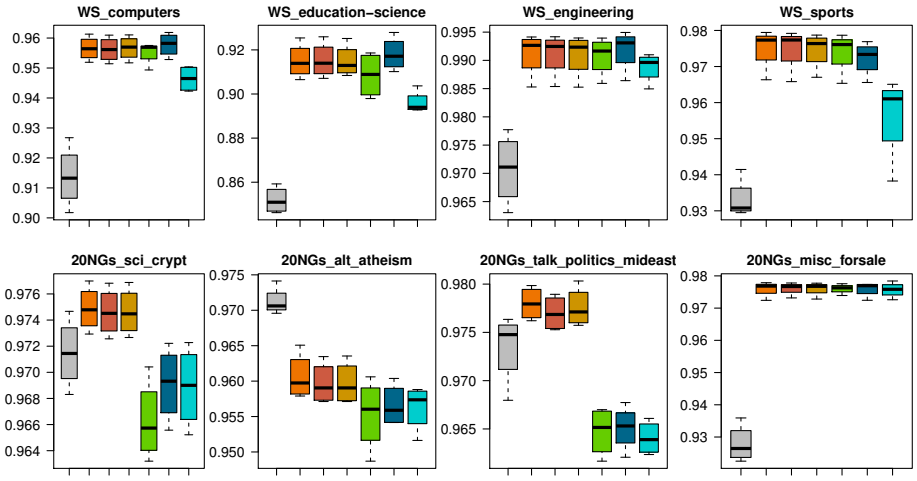



**Fig. 7** AUC-PR scores in 20Newsgroup and WS for different anomalous classes. See text for explanation of colours.

the second to third, we adjust the score to make it less sensitive to common words, which less discriminative in discovering the anomaly. Intuitions from information retrieval (Robertson and Zaragoza, 2009) clearly work for anomaly detection.

In Figure 7, we show the detection performance for 4 of the categories that we kept as anomalous in WS and 20Newsgroup. The colouring of box-plots follows the legend of Figure 6, and the grey result corresponds to a simple baseline just using log-probability. Full plots for more cases are reported in the Appendix, but the results in the figure are representative.

We note that we also tried OC-SVM (Chang and Lin, 2011), using the distance to the hyperplane as anomaly score, but its performance was not competitive given that it was not proposed for high-dimensional data.

In summary, the three Chordalysis variants are similar and more consistent than the others. HLTA and BPFA perform better sometimes, but more generally are marginally worse. The baseline works extremely well in a 5/20 cases on 20Newsgroups in politics and religion, for reasons we have not yet understood.

## 7 Running Times

Finally, we measured the running times for the 6 models under study in the two datasets 20Newsgroup and WS. Although these times have been measured in similar conditions, the implementations do not only differ from the algorithmic point of view, but also from the programming language used. Therefore, the aim of these results is more to give some insights on the running times of each algorithm rather than doing a proper comparison.

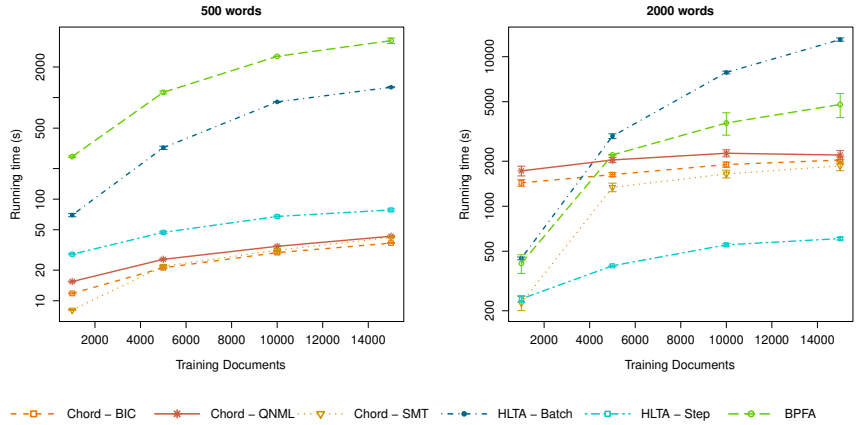



**Fig. 8** Running time in the 20newgroups dataset.

As we can see in Figure 8, BPFA and HLTA-Batch are the most time consuming algorithms, specially for large data sets. This hampered the execution of these algorithms in large datasets, such as NYT.

We also note that the running time for Chordalysis models ramps up when increasing the vocabulary size from 500 to 2000 words. This hinders the use of Chordalysis models for vocabularies larger than thousands of words. Although the differences between the 3 Chordalysis scores are small, we highlight the fact that QNML takes more time than BIC and SMT. This is due to the search procedure for QNML takes more steps than that of BIC and SMT given that the QNML score enables the finding of richer structures.

The HLTA Stepwise algorithm keeps the running time constant by subsampling the vocabulary to learn the structure (structural-batch-size in Table 1) and sub-sampling the observations to learn the parameters (Global-batch-size in Table 1). However, the performance penalty was often severe.

## 8 Conclusion

We sought to compare three very different styles of unsupervised text models that are based on the bag-of-words representation: Chordalysis, which learns chordal graphs, Hierarchical latent tree analysis (HLTA), which learns trees with observed variables at the leaves, and Bernoulli-Poisson factor analysis (BPFA), which is a matrix factorisation method. We restricted ourselves to Boolean models, and for evaluation used document log-likelihood, omni-directional prediction, and anomaly detection. These tasks were chosen as being best suited from a range of others, although for BPFA unbiased document log-likelihoods could not feasibly be computed. We did not evaluate the interpretability or explainability of the different models, but we note that all

three have excellent though different properties in some respects, with BPFA being perhaps the least explainable generally.

For Chordalysis, the new QNML scoring function was superior, though BIC was close. For document log-likelihood Chordalysis was generally superior but we expect that is because it supports much lower bias models. For omni-directional prediction, Chordalysis was generally superior but HLTA was good in some cases. BPFA was quite poor at this task. For anomaly detection, BPFA was generally best, but Chordalysis was close, and HLTA was poor.

Matrix factorisation which is BPFA, despite its reputation, did not perform well in the prediction tasks, probably because its models have no local characteristics like Chordalysis and HLTA. HLTA was very effective, but clearly with its limited number of parameters could not compete with Chordalysis as the dataset size grew. Chordalysis, however, did not scale well with a larger vocabulary.

Future work is needed in a number of areas: extending HLTA and Chordalysis models to richer structures and count data, doing unbiased document probability estimates for matrix factorisation, and better exploring focused topic models and deep neural networks, which were not evaluated here.

**Acknowledgements** This is a post-peer-review, pre-copyedit version of an article published in Behaviormetrika. The final authenticated version is available online at:
`https://doi.org/10.1007/s41237-018-0050-3`

## References

Aggarwal C, Zhai C (2012) A survey of text clustering algorithms. In: Aggarwal C, Zhai C (eds) Mining Text Data, pp 77–128

Archambeau C, Lakshminarayanan B, Bouchard G (2015) Latent IBP compound Dirichlet allocation. IEEE Trans on Pattern Analysis and Machine Intelligence 37(2):321–333

Blei D (2012) Probabilistic topic models. Communications of the ACM 55(4):77–84

Buntine W, Jakulin A (2004) Applying discrete PCA in data analysis. In: 20th Conference on Uncertainty in Artificial Intelligence, Banff, Canada

Buntine W, Mishra S (2014) Experiments with non-parametric topic models. In: Proc. of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 881–890

Cai D, He X, Han J, Huang T (2011) Graph regularized nonnegative matrix factorization for data representation. IEEE Trans, on Pattern Analysis and Machine Intelligence 33(8):1548–1560

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: A survey. ACM Comput Surv 41(3):15:1–15:58

Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. ACM Trans Intellent Systems Technology 2(3):27:1–27:27

Chen P, Zhang N, Liu T, Poon L, Chen Z, Khawar F (2017) Latent tree models for hierarchical topic detection. Artificial Intelligence 250:105–124

Collobert R, Weston J (2008) A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proc. of the 25th International Conference on Machine Learning, ACM, pp 160–167

Davis J, Goadrich M (2006) The relationship between precision-recall and ROC curves. In: Proc. of the 23rd international conference on Machine learning, ACM, pp 233–240

Friedman N, Geiger D, Goldszmidt M (1997) Bayesian network classifiers. Machine Learning 29(2):131–163

Gaussier E, Goutte C (2005) Relation between PLSA and NMF and implications. In: Proc. of 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05, pp 601–602

Heckerman D, Chickering D (1995) Learning Bayesian networks: The combination of knowledge and statistical data. In: Machine Learning, pp 20–197

Hu C, Rai P, Carin L (2016) Non-negative matrix factorization for discrete data with hierarchical side-information. In: Gretton A, Robert C (eds) Proc. of the 19th International Conference on Artificial Intelligence and Statistics, pp 1124–1132

Li C, Wang H, Zhang Z, Sun A, Ma Z (2016) Topic modeling for short texts with auxiliary word embeddings. In: Proc. of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pp 165–174

Lim K, Buntine W (2016) Bibliographic analysis on research publications using authors, categorical labels and the citation network. Machine Learning 103:185–213

Liu T, Zhang N, Chen P (2014) Hierarchical latent tree analysis for topic detection. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, pp 256–272

Lund K, Burgess C (1996) Producing high-dimensional semantic spaces from lexical co-occurrence. Behavior Research Methods, Instruments, & Computers 28(2):203–208

Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionally. In: Advances in Neural Information Processing Systems, pp 3111–3119

Nguyen D, Billingsley R, Du L, Johnson M (2015) Improving topic models with latent feature word representations. Trans of the Association for Computational Linguistics 3:299–313

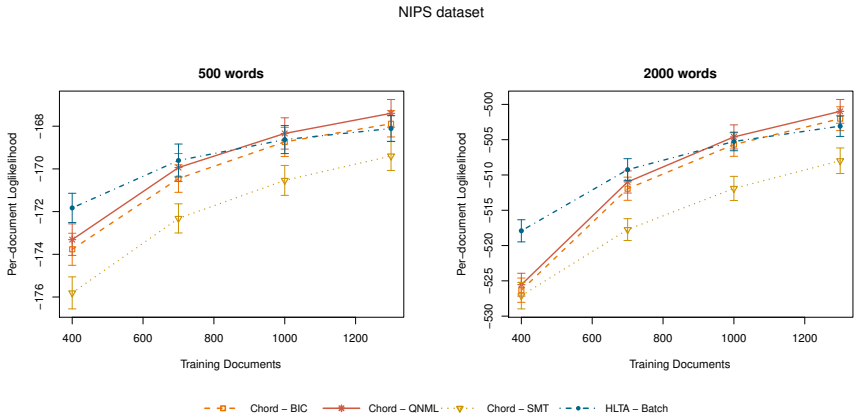
Paisley J, Wang C, Blei D, Jordan M (2015) Nested hierarchical Dirichlet processes. IEEE Trans on Pattern Analysis and Machine Intelligence 37(2):256–270

Petitjean F, Webb G (2015) Scaling log-linear analysis to datasets with thousands of variables. In: Proc. of the 2015 SIAM International Conference on Data Mining, SIAM, pp 469–477

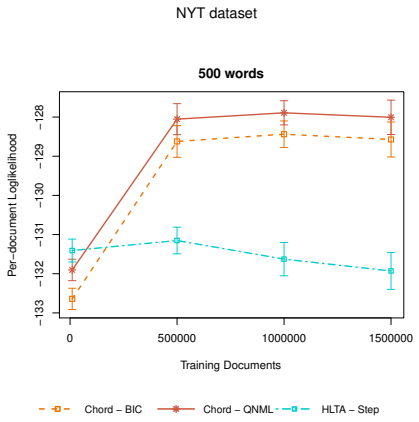
Petitjean F, Webb G (2016) Scalable learning of graphical models. In: Proc. of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery

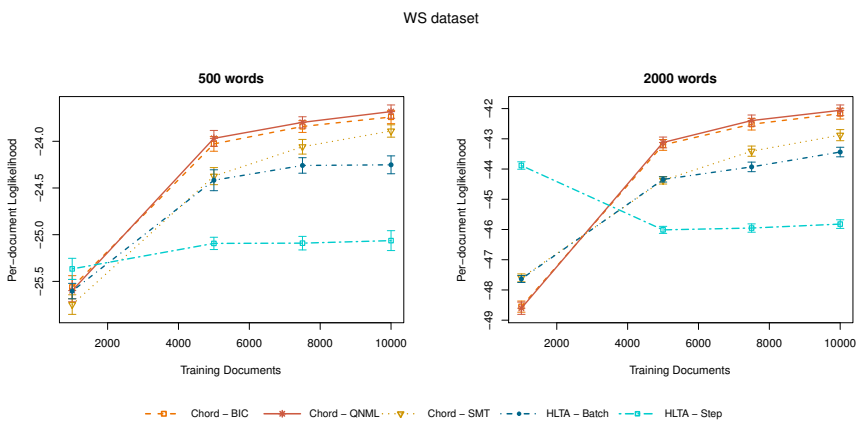
and Data Mining, KDD '16, pp 2131–2132
Robertson S, Zaragoza H (2009) The Probabilistic Relevance Framework. Now Publishers Inc., Hanover, MA, USA
Silander T (2016) Bayesian network structure learning with a quotient normalized maximum likelihood criterion. In: Proc. of the Ninth Workshop on Information Theoretic Methods in Science and Engineering, Helsinki, Finland, pp 32–35
Socher R, Huval B, Manning C, Ng A (2012) Semantic compositionality through recursive matrix-vector spaces. In: Proc. of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL '12, pp 1201–1211
Suzuki J (2017) A theoretical analysis of the BDeu scores in Bayesian network structure learning. Behaviormetrika 44(1):97–116
Suzuki J, Kawahara J (2017) Branch and bound for regular Bayesian network structure learning. In: Conference on Uncertainty in Artificial Intelligence, Sydney, Australia
Teh Y, Jordan M, Beal M, Blei D (2006) Hierarchical Dirichlet processes. Journal of the ASA 101(476):1566–1581
Wallach H, Murray I, Salakhutdinov R, Mimno D (2009) Evaluation methods for topic models. In: Bottou L, Littman M (eds) Proc. of the 26th International Conference on Machine Learning
Wei X, Croft W (2006) LDA-based document models for ad-hoc retrieval. In: Proc. of the 29th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, pp 178–185
Yin J, Wang J (2014) A Dirichlet multinomial mixture model-based approach for short text clustering. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 233–242
Zhou M (2015) Infinite edge partition models for overlapping community detection and link prediction. In: Proc. of 18th International Conference on Artificial Intelligence and Statistics, pp 1135–1143
Zhou M, Hannah L, Dunson D, Carin L (2012) Beta-negative binomial process and Poisson factor analysis. In: Proc. of the 15th International Conference on Artificial Intelligence and Statistics, La Palma, Canary Islands, pp 1462–1471

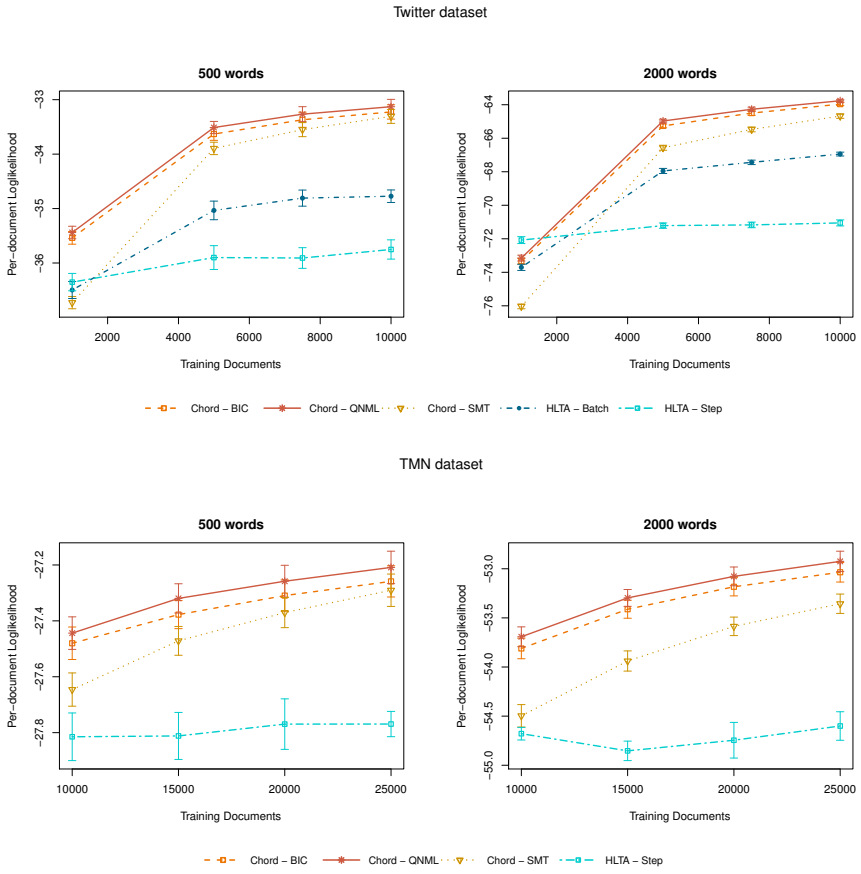
## A Log-likelihood



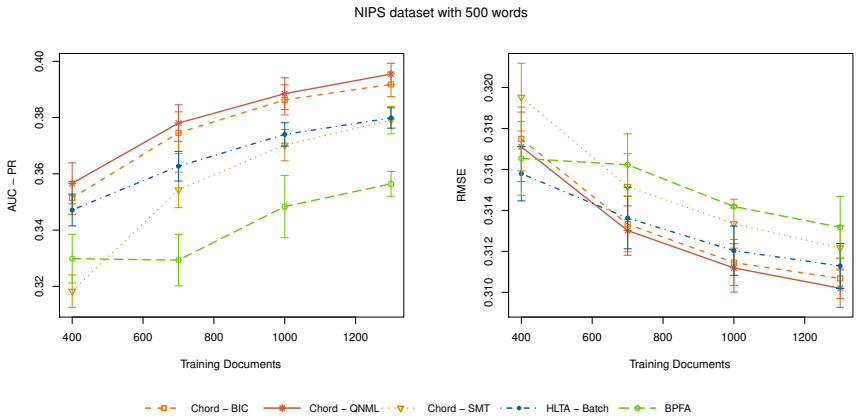


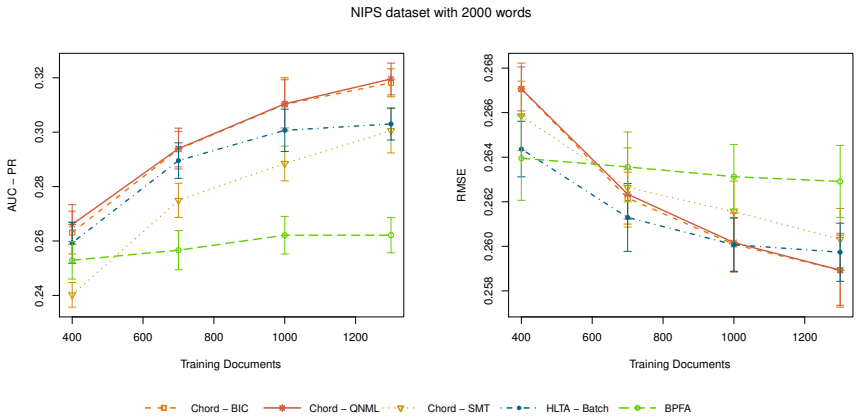


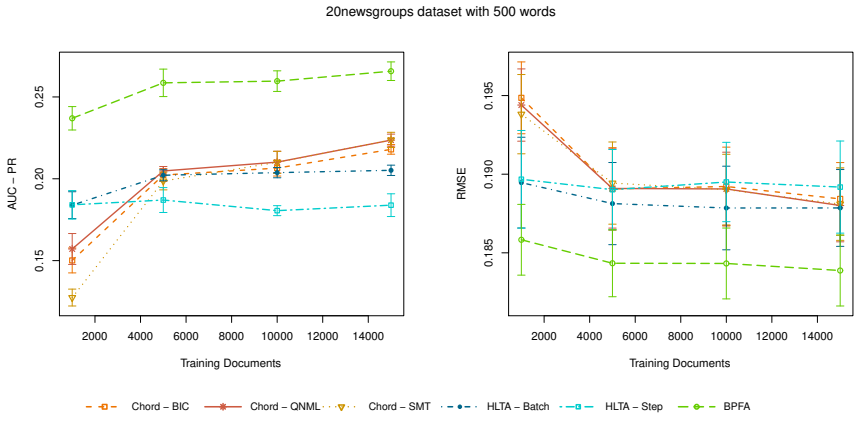



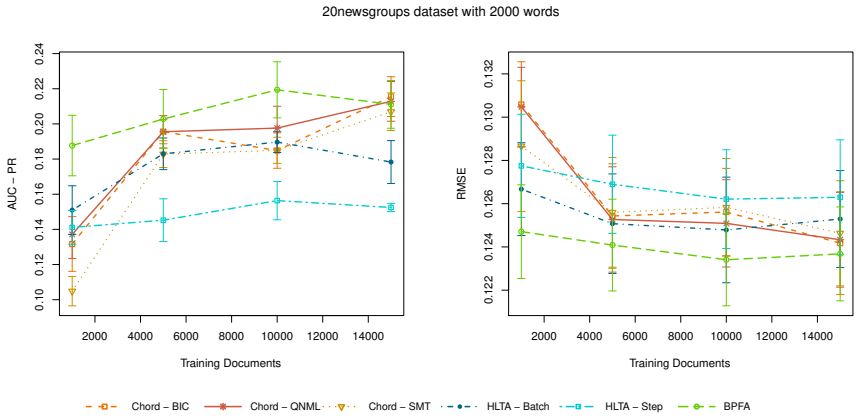

Twitter dataset
500 words
2000 words
Per−document Loglikelihood
Training Documents
Chord – BIC
Chord – QNML
Chord – SMT
HLTA – Batch
HLTA – Step
TMN dataset
500 words
2000 words
Per−document Loglikelihood
Training Documents
Chord – BIC
Chord – QNML
Chord – SMT
HLTA – Step

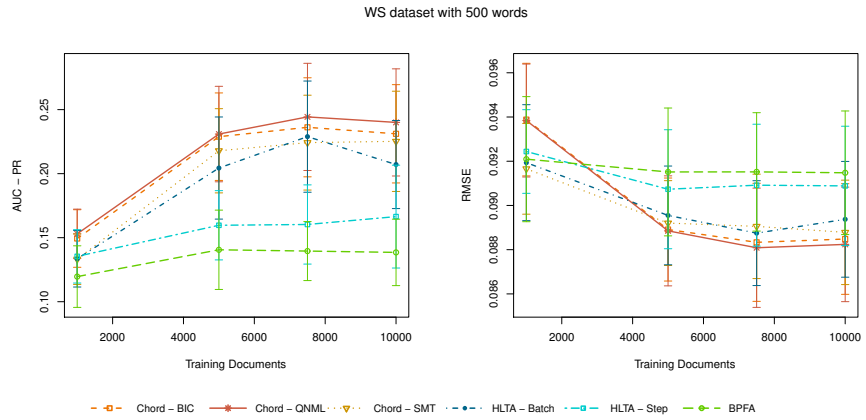
## B Omni-directional Prediction

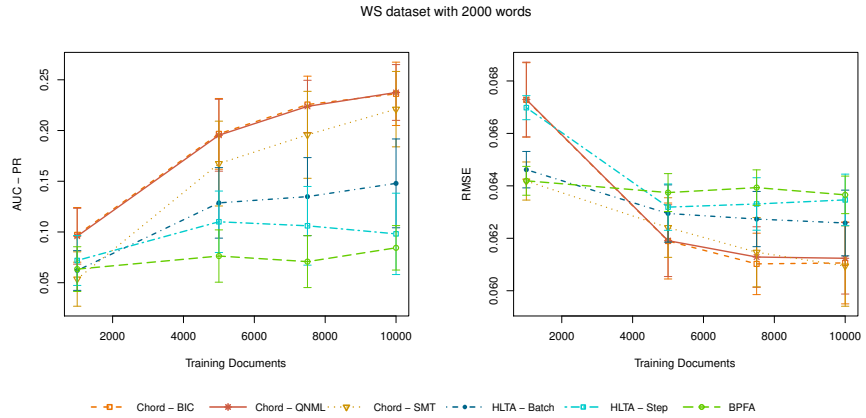


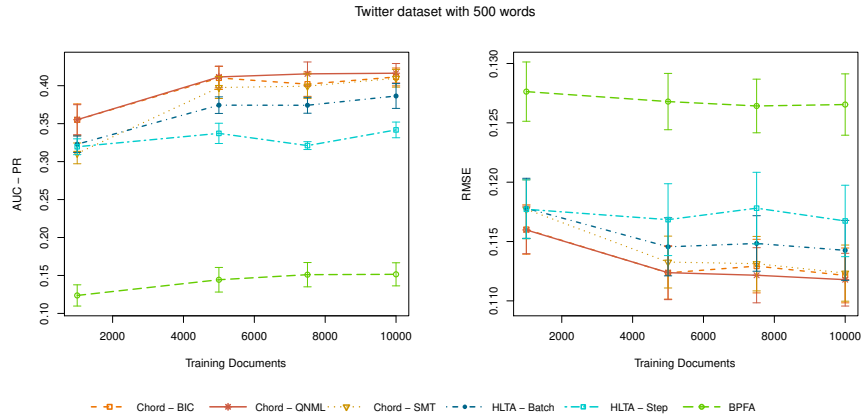



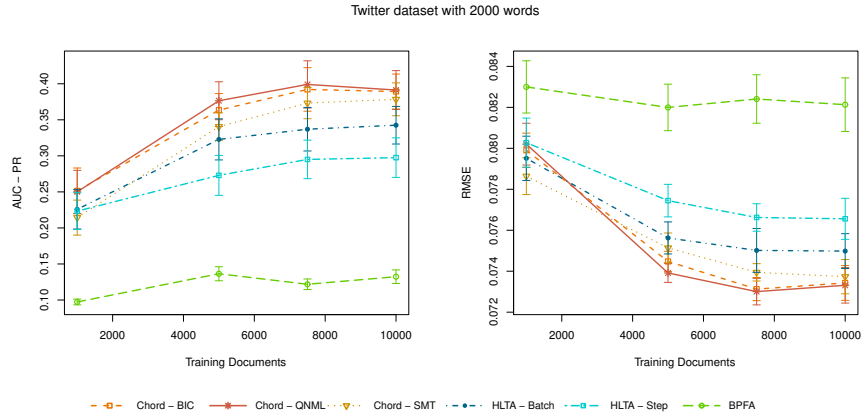



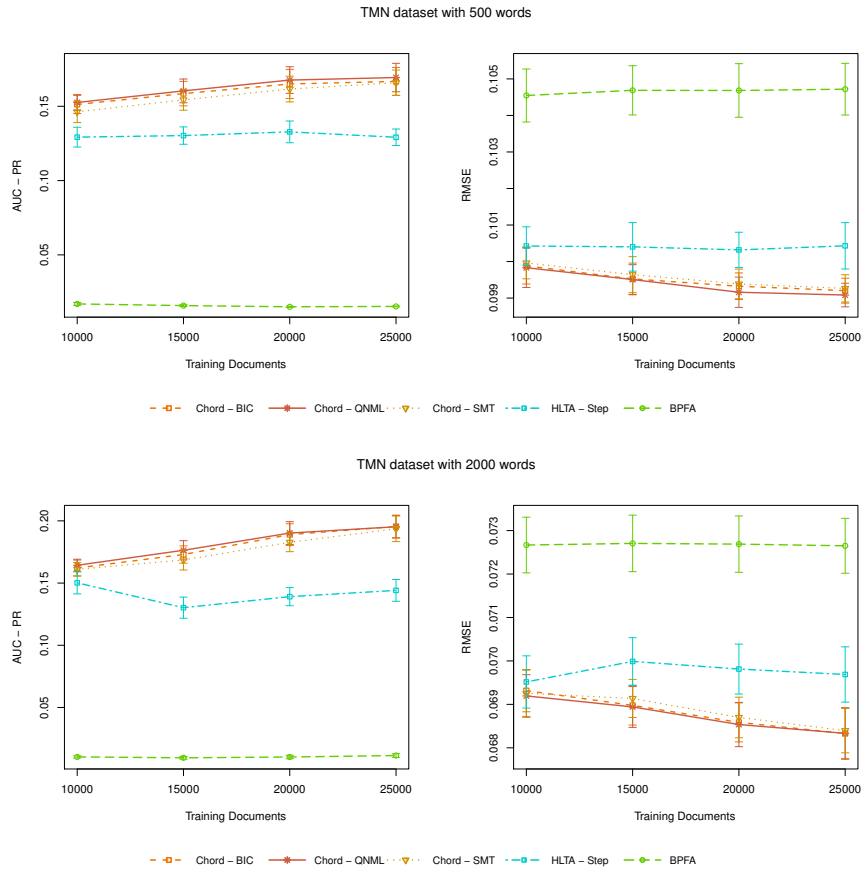




WS dataset with 500 words
AUC – PR
RMSE
Training Documents
Training Documents
Chord – BIC
Chord – QNML
Chord – SMT
HLTA – Batch
HLTA – Step
BPFA


WS dataset with 2000 words
AUC – PR
RMSE
Training Documents
Training Documents
Chord – BIC
Chord – QNML
Chord – SMT
HLTA – Batch
HLTA – Step
BPFA


Twitter dataset with 500 words
AUC – PR
RMSE
Training Documents
Training Documents
Chord – BIC
Chord – QNML
Chord – SMT
HLTA – Batch
HLTA – Step
BPFA


Twitter dataset with 2000 words
AUC – PR
RMSE
Training Documents
Training Documents
Chord – BIC
Chord – QNML
Chord – SMT
HLTA – Batch
HLTA – Step
BPFA


TMN dataset with 500 words
AUC – PR
0.05
0.10
0.15
10000
15000
20000
25000
Training Documents
RMSE
0.099
0.101
0.103
0.105
10000
15000
20000
25000
Training Documents
Chord – BIC
Chord – QNML
Chord – SMT
HLTA – Step
BPFA
TMN dataset with 2000 words
AUC – PR
0.05
0.10
0.15
0.20
10000
15000
20000
25000
Training Documents
RMSE
0.068
0.069
0.070
0.071
0.072
0.073
10000
15000
20000
25000
Training Documents
Chord – BIC
Chord – QNML
Chord – SMT
HLTA – Step
BPFA

## C Anomaly Detection

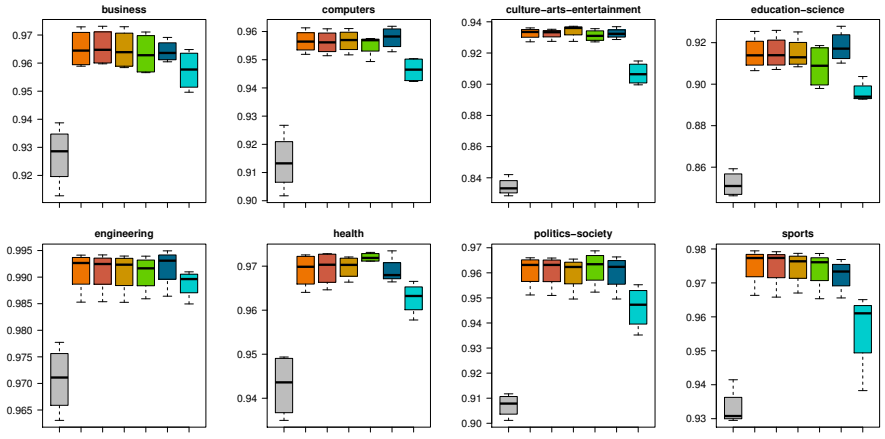



**Fig. 9** Anomaly Detection in W'S dataset

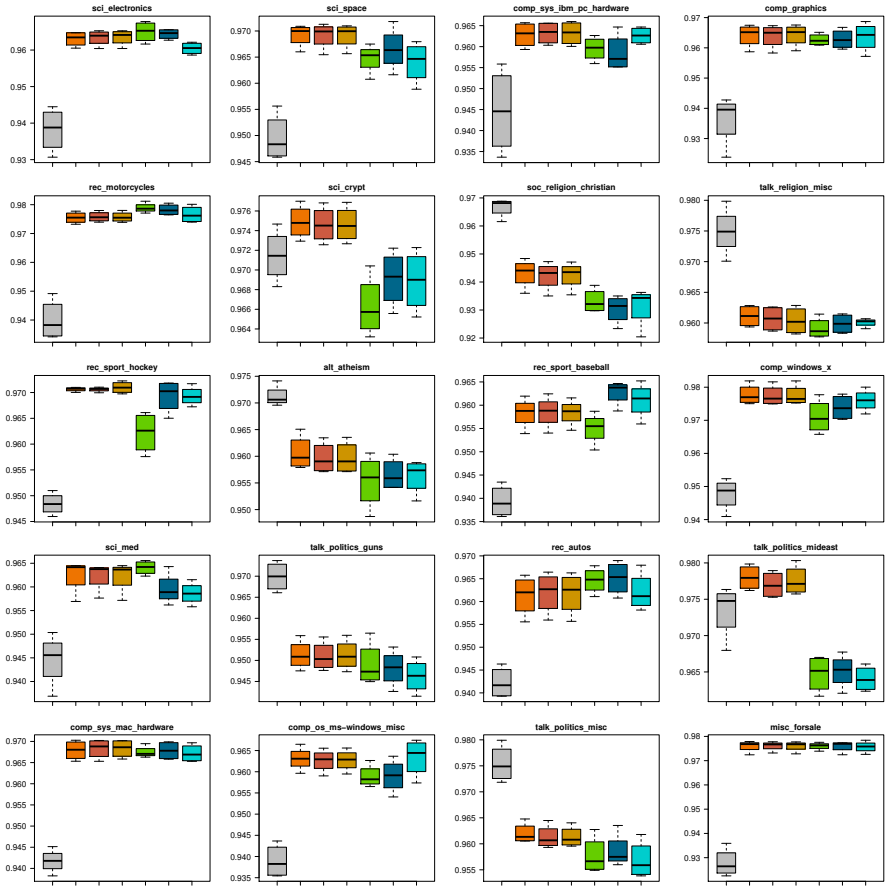



**Fig. 10** Anomaly Detection in 20Newsgroups dataset

## D Running Times

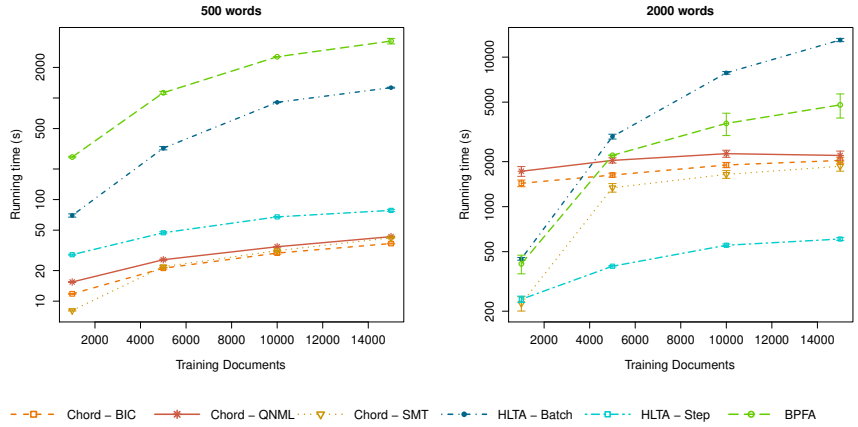



**Fig. 11** Running time in WS dataset

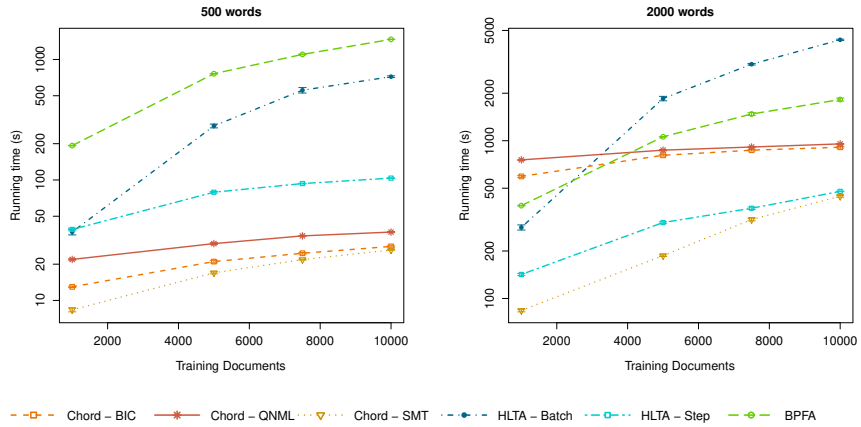



**Fig. 12** Running time in 20Newgroups dataset