Knowl Inf Syst (2019) 61:661–693
https://doi.org/10.1007/s10115-018-1213-y

REGULAR PAPER

Leveraging external information in topic modelling

He Zhao[1] · **Lan Du**[1] · **Wray Buntine**[1] · **Gang Liu**[2]

Received: 20 December 2017 / Revised: 4 April 2018 / Accepted: 6 May 2018 /
Published online: 12 May 2018

Abstract Besides the text content, documents usually come with rich sets of meta-information, such as categories of documents and semantic/syntactic features of words, like those encoded in word embeddings. Incorporating such meta-information directly into the generative process of topic models can improve modelling accuracy and topic quality, especially in the case where the word-occurrence information in the training data is insufficient. In this article, we present a topic model called MetaLDA, which is able to leverage either document or word meta-information, or both of them jointly, in the generative process. With two data augmentation techniques, we can derive an efficient Gibbs sampling algorithm, which benefits from the fully local conjugacy of the model. Moreover, the algorithm is favoured by the sparsity of the meta-information. Extensive experiments on several real-world datasets demonstrate that our model achieves superior performance in terms of both perplexity and topic quality, particularly in handling sparse texts. In addition, our model runs significantly faster than other models using meta-information.

Keywords Latent Dirichlet allocation · Side information · Data augmentation · Gibbs sampling

✉ Lan Du
lan.du@monash.edu

He Zhao
he.zhao@monash.edu

Wray Buntine
wray.buntine@monash.edu

Gang Liu
liugang@hrbeu.edu.cn

[1] Faculty of Information Technology, Monash University, Melbourne, VIC, Australia

[2] College of Computer Science and Technology, Harbin Engineering University, Harbin, China

1 Introduction

With the rapid growth of the internet, huge amounts of text data are generated in social networks, online shopping and news websites, etc. These data are generally short but may contain rich and complex kinds of information that can be difficult to find in traditional information sources [44], therefore create demand for both effective and efficient machine learning techniques. Probabilistic topic models such as Latent Dirichlet Allocation (LDA) [4] are among the popular approaches for this task. In topic modelling, a document is assumed to be generated from a mixture of topics, where each topic is a probability distribution over a vocabulary. However, most existing topic models discover topics purely based on the word-occurrences, ignoring the *meta-information* (a.k.a., *side information*) associated with the content, which often results in degraded performance. We argue that meta-information associated with diverse texts can play the role of background knowledge in human text comprehension. When we humans read text, it is natural for us to leverage metadata, such as categories, authors, timestamps, words' semantic/syntactic information, to improve our understanding of the text. Therefore, it is reasonable to expect topic models can also benefit from the meta-information and yield improved modelling accuracy and topic quality.

In practice, various kinds of meta-information are associated to tweets, product reviews, blogs, etc. They are often available at both the document level and the word level. At the document level, labels of documents can be used to guide topic learning so that more meaningful topics can be discovered. It is likely that documents with common labels should discuss similar topics, which can be modelled by similar distributions over topics. In the case of tweets, as shown in Fig. 1, they can have an author, hashtag, timestamp, etc. Previous work on tweet pooling [12,19] has shown that aggregating tweets according to their authors or hashtags can significantly improve topic modelling. Furthermore, if we use authors as labels for scientific papers, the research topics of the papers published by the same researcher can be closely related, and authors having similar research topics are more likely to collaborate [34].

At the word level, different semantic/syntactic features are also accessible. For example, there are features regarding word relationships, such as synonyms obtained from WordNet [22], word co-occurrence patterns obtained from a large corpus, and linked concepts from knowledge graphs. It is preferable that words having similar meaning but different morphological forms, like "dog" and "puppy", are likely to be assigned to the same topic,

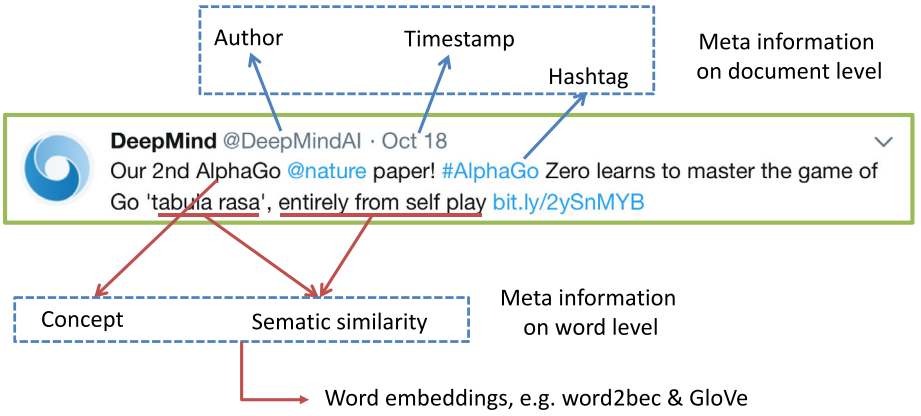


Fig. 1 Meta-information associated with a tweet

even if they barely co-occur in the modelled corpus. Recently, word embeddings generated by GloVe [27] and word2vec [20,21] have attracted a lot of attention in natural language processing and related fields. It has been shown that the word embeddings can capture both the semantic and syntactic features of words so that similar words are close to each other in the embedding space. It is reasonable to expect that these word embeddings will improve topic modelling [8,26]. Figure 1 also shows some word-level meta-information associated with the tweet.

It is known that most conventional topic models can suffer from a large performance degradation on short texts (e.g., tweets and news headlines) due to insufficient word co-occurrence information. In such cases, meta-information of documents and words can play the role of auxiliary information in analysing short texts, which can compensate for the lost information in word co-occurrences. At the document level, we can leverage the hashtags, users, locations, and timestamps of tweets so that the data sparsity problem can be alleviated. At the word level, word semantic similarity and embeddings obtained or trained on large external corpus (e.g., Google News or Wikipedia) can also be built into the generative process of topic models [17,26,36].

Recently, significant research effort has been devoted to handle short texts in topic modelling. Models along this line often take classical topic models, like LDA, as a building block, and manipulate the graphical structure to incorporate meta-information into the generative process [23,26,30]. However, what we found is that those models make use of either the document level or the word level meta-information, rather than both. The limitation is often caused by their complicated model structures, which lose conjugacy favoured by sampling methods, and further result in inefficient inference algorithms.

In this article, we propose MetaLDA,[1] a new topic model that can effectively and efficiently make use of arbitrary document and word meta-information encoded in binary form. Specifically, the labels of a document in MetaLDA are incorporated in the prior of the per-document topic distributions. If two documents have similar labels, their topic distributions should be generated with similar Dirichlet priors. Analogously, at the word level, the features of a word are incorporated in the prior of the per-topic word distributions, which encourages words with similar features to have similar proportions across topics. Therefore, both document and word meta-information, if and when they are available, can be flexibly and simultaneously incorporated in the generative process. MetaLDA has the following key properties:

1. MetaLDA jointly incorporates various kinds of document and word meta-information for both regular and short texts, yielding better modelling accuracy and topic quality.
2. With data augmentation techniques, the inference of MetaLDA can be done by an efficient and closed-form Gibbs sampling algorithm that benefits from the full local conjugacy of the model.
3. The simple structure of incorporating meta-information and the efficient inference algorithm give MetaLDA advantage in terms of running speed over other models with meta-information.
4. MetaLDA has an improved interpretability. For example, the inclusion of the document labels directly in the generative process gives the ability of both explaining each label with topics and assigning labels to each topic.

We conduct extensive experiments with several real datasets including regular and short texts in various domains. The experimental results demonstrate that MetaLDA outperforms

[1] Code at https://github.com/ethanhezhao/MetaLDA/.

all the competitors we considered in terms of perplexity, topic coherence and running time. The rest of the article, which extends our earlier contribution [42], is organised as follows. We first briefly discuss the related work in Sect. 2. Then, we elaborate on MetaLDA and derive its sampling algorithm in Sects. 3 and 4, respectively. The experimental results derived on several real-world datasets are reported in Sect. 5. We conclude the article in Sect. 6.

2 Related work

In this section, we review three lines of related work: models with document meta-information, models with word meta-information, and models for short texts.

At the document level, Supervised LDA (sLDA) [18] models document labels by learning a generalised linear model with an appropriate link function and exponential family dispersion function. But the restriction for sLDA is that one document can only have one label. Labelled LDA (LLDA) [29] assumes that each label has a corresponding topic and a document is generated by a mixture of the topics. Although multiple labels are allowed in LLDA, it requires that the number of topics must equal to the number of labels, i.e., exactly one topic per label. As an extension to LLDA, Partially Labelled LDA (PLLDA) [30] relaxes this requirement by assigning multiple topics to a label. The Dirichlet Multinomial Regression (DMR) model [23] incorporates document labels on the prior of the topic distributions like our MetaLDA but with the logistic-normal transformation. As full conjugacy does not exist in DMR, a part of the inference has to be done by numerical optimisation, which is slow for large sets of labels and topics. Similarly, in the Hierarchical Dirichlet Scaling Process (HDSP) [14], conjugacy is broken as well since the topic distributions have to be renormalised. A Poisson factorisation model with hierarchical document labels is introduced in [13], but the technique cannot be applied to regular topic models as the topic proportion vectors are also unnormalised.

There has been growing interest in incorporating word features in topic models. For example, DF-LDA [2] incorporates word must-links and cannot-links using a Dirichlet forest prior in LDA; MRF-LDA [35] encodes word semantic similarity in LDA with a Markov random field; WF-LDA [28] extends LDA to model word features with the logistic-normal transform; LF-LDA [26] integrates word embeddings into LDA by replacing the topic-word Dirichlet multinomial component with a mixture of a Dirichlet multinomial component and a word embedding component; Instead of generating word types (tokens), Gaussian LDA (GLDA) [8] directly generates word embeddings with the Gaussian distribution. Despite the exciting applications of the above models, their inference is usually less efficient due to the non-conjugacy and/or complicated model structures.

Analysis of short text with topic models has been an active area with the development of social networks. Generally, there are two ways to deal with the sparsity problem in short texts, either using the intrinsic properties of short texts or leveraging meta-information. For the first way, one popular approach is to aggregate short texts into pseudo-documents, for example, [12] introduces a model that aggregates tweets containing the same word; Recently, PTM [46] aggregates short texts into latent pseudo-documents. Another approach is to assume one topic per short document, known as mixture of unigrams or Dirichlet Multinomial Mixture (DMM) such as [36,39]. For the second way, document meta-information can be used to aggregate short texts, for example, [12] aggregates tweets by the corresponding authors and [19] shows that aggregating tweets by their hashtags yields superior performance over other aggregation methods. Closely related work to ours are models that use word features for short texts. For

example, [36] introduces an extension of GLDA on short texts which samples an indicator variable that chooses to generate either the type of a word or the embedding of a word and GPU-DMM [17] extends DMM with word semantic similarity obtained from embeddings for short texts. Although with improved performance, there still exist challenges for existing models:

- for aggregation-based models, it is usually hard to choose which meta-information to use for aggregation;
- the "single topic" assumption makes DMM models lose the flexibility to capture different topic ingredients of a document;
- the incorporation of meta-information in the existing models is usually less efficient.

To our knowledge, the attempts that jointly leverage document and word meta-information are relatively rare. For example, meta-information can be incorporated by first-order logic in Logit-LDA [3] and score functions in SC-LDA [37]. However, the first-order logic and score functions need to be defined for different kinds of meta-information and the definition can be infeasible for incorporating both document and word meta-information simultaneously.

3 The MetaLDA model

Given a corpus, LDA uses the same Dirichlet prior for all the per-document topic distributions and the same prior for all the per-topic word distributions [33]. While in MetaLDA, each document has a specific Dirichlet prior on its topic distribution, which is computed from the meta-information of the document, and the parameters of the prior are estimated during training. Similarly, each topic has a specific Dirichlet prior computed from the word meta-information. In this section we elaborate on our MetaLDA, in particular on how the meta-information is incorporated. Hereafter, we will use labels as document meta-information, unless otherwise stated. Table 1 summarises the notations used in this section.

The basic formulation mirrors that of standard LDA. Given a collection of D documents $\mathcal{D}$, MetaLDA generates document $d \in \{1, \ldots, D\}$ with a mixture of K topics and each topic $k \in \{1, \ldots, K\}$ is a distribution over the vocabulary with V tokens, denoted by $\boldsymbol{\phi}_k \in \mathbb{R}_+^V$. For document d with N_d words, to generate the ith ($i \in \{1, \ldots, N_d\}$) word $w_{d,i}$, we first sample a topic $z_{d,i} \in \{1, \ldots, K\}$ from the document's topic distribution $\boldsymbol{\theta_d} \in \mathbb{R}_+^K$, and then sample $w_{d,i}$ from $\boldsymbol{\phi}_{z_{d,i}}$. Now this is extended with meta-information. Assume the labels of document d are encoded in a binary vector $\boldsymbol{f_d} \in \{0, 1\}^{L_{\text{doc}}}$ where L_{doc} is the total number of unique labels. $f_{d,l} = 1$ indicates label l is active in document d and vice versa. MetaLDA allows each document to have multiple labels. Similarly, the L_{word} features of token v are stored in a binary vector $\boldsymbol{g}_v \in \{0, 1\}^{L_{\text{word}}}$. Therefore, the document and word meta-information associated with $\mathcal{D}$ are stored in the matrix $\mathbf{F} \in \{0, 1\}^{D \times L_{\text{doc}}}$ and $\mathbf{G} \in \{0, 1\}^{V \times L_{\text{word}}}$, respectively. Although MetaLDA incorporates binary features, categorical features and real-valued features can be converted into binary values with proper transformations such as discretisation and binarisation [10].

Figure 2 shows the graphical model of MetaLDA and the generative process is as follows:

1. For each topic k:
   (a) For each doc-label l: Draw $\lambda_{l,k} \sim \text{Ga}(\mu_0, \mu_0)$
   (b) For each word-feature l': Draw $\delta_{l',k} \sim \text{Ga}(\nu_0, \nu_0)$
   (c) For each token v: Compute $\beta_{k,v} = \prod_{l'=1}^{L_{\text{word}}} \delta_{l',k}^{g_{v,l'}}$
   (d) Draw $\boldsymbol{\phi}_k \sim \text{Dir}_V(\boldsymbol{\beta}_k)$

Table 1 List of notations

Notation	Description
D	Number of documents
V	Size of vocabulary
K	Number of topics
N_d	Number of words in document d
L_{doc}	Dimension of document labels
L_{word}	Dimension of word features
$\boldsymbol{f_d}$	Binary label vector of document d
$\boldsymbol{g_v}$	Binary feature vector of word v
$w_{d,i}$	ith word in document d
$z_{d,i}$	Topic of the ith word in document d
$\boldsymbol{\theta_d}$	Normalised topic weights (topic distribution) of document d
$\boldsymbol{\phi_k}$	Normalised word weights (word distribution) of topic k
$\boldsymbol{\alpha_d}$	Dirichlet parameter of the topic distribution of document d
$\boldsymbol{\beta_k}$	Dirichlet parameter of the word distribution of document k
$\lambda_{l,k}$	Weight between document label l and topic k
$\delta_{l',k}$	Weight between word feature l' and topic k
μ_0	Hyper-parameter of $\lambda_{l,k}$
ν_0	Hyper-parameter of $\delta_{l',k}$

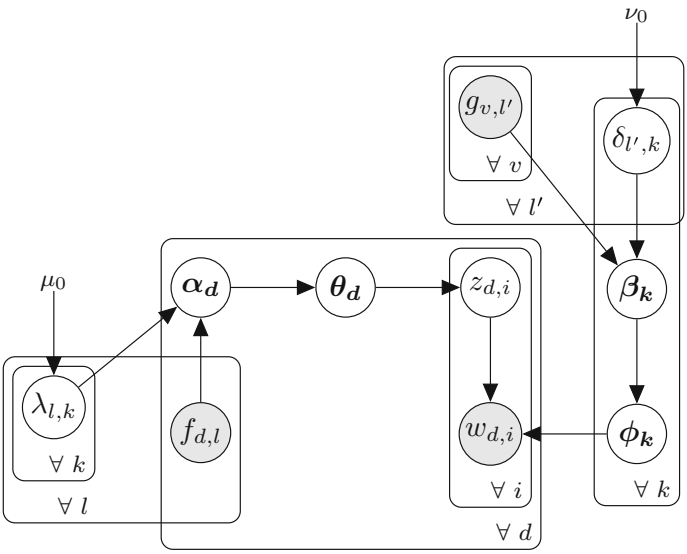


Fig. 2 The graphical model of MetaLDA

2. For each document d:
   (a) For each topic k: Compute $\alpha_{d,k} = \prod_{l=1}^{L_{\text{doc}}} \lambda_{l,k}^{f_{d,l}}$
   (b) Draw $\boldsymbol{\theta}_d \sim \text{Dir}_K(\boldsymbol{\alpha}_d)$
   (c) For each word in document d:
      (i) Draw topic $z_{d,i} \sim \text{Cat}_K(\boldsymbol{\theta}_d)$
      (ii) Draw word $w_{d,i} \sim \text{Cat}_V(\boldsymbol{\phi}_{z_{d,i}})$

where $\text{Ga}(\cdot,\cdot)$, $\text{Dir}(\cdot)$, $\text{Cat}(\cdot)$ are the gamma distribution with shape and rate parameters, the Dirichlet distribution, and the categorical distribution, respectively. K, μ_0, and ν_0 are the hyper-parameters.

To incorporate document labels, MetaLDA learns a specific Dirichlet prior over the topics for each document by using the label information. Specifically, the information of document d's labels is incorporated in $\boldsymbol{\alpha}_d$, the parameter of Dirichlet prior on $\boldsymbol{\theta}_d$. As shown in Step 2a, $\alpha_{d,k}$ is computed as a log linear combination of the labels $f_{d,l}$. Since $f_{d,l}$ is binary, $\alpha_{d,k}$ is indeed the multiplication of $\lambda_{l,k}$ over all the active labels of document d, i.e., $\{l \mid f_{d,l} = 1\}$. Drawn from the gamma distribution with mean 1, $\lambda_{l,k}$ controls the impact of label l on topic k. If label l has no or less impact on topic k, $\lambda_{l,k}$ is expected to be 1 or close to 1, and then $\lambda_{l,k}$ will have no or little influence on $\alpha_{d,k}$ and vice versa. The hyper-parameter μ_0 controls the variation of $\lambda_{l,k}$. The incorporation of word features is analogous but in the parameter of the Dirichlet prior on the per-topic word distributions as shown in Step 1c.

The intuition of our way of incorporating meta-information is as follows. At the document level, if two documents have more labels in common, their Dirichlet parameter $\boldsymbol{\alpha}_d$ will be more similar, resulting in more similar topic distributions $\boldsymbol{\theta}_d$; At the word level, if two words have similar features, their $\beta_{k,v}$ in topic k will be similar and then we can expect that their $\phi_{k,v}$ could be more or less the same. Finally, the two words will have similar probabilities of showing up in topic k. In other words, if a topic "prefers" a certain word, we expect that it will also prefer other words with similar features to that word. Moreover, at both the document and the word level, different labels/features may have different impact on the topics (λ/δ), which can be automatically learnt in MetaLDA from the data.

4 Inference

Unlike most existing methods, our way of incorporating the meta-information facilitates the derivation of an efficient Gibbs sampling algorithm. With two data augmentation techniques (i.e., the introduction of auxiliary variables), MetaLDA admits the local conjugacy that further gives us a close-form Gibbs sampling algorithm. Note that MetaLDA incorporates the meta-information on the Dirichlet priors, so we can still use LDA's collapsed Gibbs sampling algorithm for the topic assignment $z_{d,i}$. Thus, there is no need to use a hybrid learning algorithm (i.e., optimisation + sampling), such as those in [23,26]. Moreover, as shown in Step 2a and 1c, we only need to consider nonzero entries of $\mathbf{F}$ and $\mathbf{G}$ in computing the full conditionals, which further reduces the inference complexity, particularly when the feature space is sparse. This is often the case in real-world scenarios. In the rest of this section, we will focus on the derivation of the full conditionals for sampling the two Gamma random variables, $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$, used to modelling the influence of document labels and word features on topics. Table 2 shows the statistics that we need while running the inference.

Table 2 Summary of statistics

Notation	Description
$m_{d,k}$	Number of words in document d assigned to topic k
$n_{k,v}$	Number of word v assigned to topic k
q_d	Beta distributed axillary variable for document d
$t_{d,k}$	Axillary table counts drawn from CRP for document d and topic k
$\hat{q}_k$	Beta distributed axillary variable for topic k
$t'_{d,k}$	Axillary table counts drawn from CRP for document k and word v

Given $\boldsymbol{\phi}_{1:K}$ and $\boldsymbol{\theta}_{1:D}$, the complete model likelihood (i.e., joint distribution) of MetaLDA is exactly the same as LDA's likelihood, which is as follows:

$$\Pr(\boldsymbol{w}_{1:D}, \boldsymbol{z}_{1:D} | \boldsymbol{\theta}_{1:D}, \boldsymbol{\phi}_{1:K}) = \prod_{d=1}^{D} \prod_{i=1}^{N_d} \theta_{d,z_{d,i}} \phi_{z_{d,i},v} = \prod_{d=1}^{D} \prod_{k=1}^{K} \theta_{d,k}^{m_{d,k}} \prod_{k=1}^{K} \prod_{v=1}^{V} \phi_{k,v}^{n_{k,v}} \tag{1}$$

where $n_{k,v} = \sum_{d}^{D} \sum_{i=1}^{N_d} \mathbf{1}_{(w_{d,i}=v, z_{d,i}=k)}$ counts the number of words v assigned to topic k, $m_{d,k} = \sum_{i=1}^{N_d} \mathbf{1}_{(z_{d,i}=k)}$ counts the number of words in document d assigned to topic k, and $\mathbf{1}_{(\cdot)}$ is the indicator function. In the standard LDA model, we can marginalise out $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ using the Dirichlet multinomial conjugacy, and then yield

$$\begin{aligned}
&\Pr(\boldsymbol{z}_{1:D}, \boldsymbol{w}_{1:D}; \boldsymbol{\alpha}_{1:D}, \boldsymbol{\beta}_{1:K}) \\
&= \int_{\boldsymbol{\theta}} \prod_{d=1}^{D} \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_{d,k}\right)}{\prod_{k=1}^{K} \Gamma(\alpha_{d,k})} \prod_{k=1}^{K} \theta_{d,k}^{m_{d,k}+\alpha_{d,k}-1} \int_{\boldsymbol{\phi}} \prod_{k=1}^{K} \frac{\Gamma\left(\sum_{v=1}^{V} \beta_{k,v}\right)}{\prod_{v=1}^{V} \Gamma(\beta_{k,v})} \prod_{v=1}^{V} \phi_{k,v}^{n_{k,v}+\beta_{k,v}-1} \cdot \\
&= \prod_{d=1}^{D} \frac{\mathrm{Beta}_K(\boldsymbol{\alpha}_d + \boldsymbol{m}_d)}{\mathrm{Beta}_K(\boldsymbol{\alpha}_d)} \prod_{k=1}^{K} \frac{\mathrm{Beta}_V(\boldsymbol{\beta}_k + \boldsymbol{n}_k)}{\mathrm{Beta}_V(\boldsymbol{\beta}_k)}
\end{aligned} \tag{2}$$

where $\Gamma(\cdot)$ is the Gamma function, $\mathrm{Beta}_N(\cdot)$ is a N-dimensional beta function as

$$\mathrm{Beta}_N(\boldsymbol{x}) = \frac{\prod_n \Gamma(x_n)}{\Gamma\left(\sum_n x_n\right)}$$

and here we assume that the Dirichlet priors are document and topic specific. Given $\boldsymbol{\beta}_k$ and $\boldsymbol{\alpha}_d$, it is straightforward to compute the full conditional for sampling topic assignment $z_{d,i}$, i.e.,

$$\begin{aligned}
\Pr(z_{d,i} = k \mid \boldsymbol{z}_{1:D}^{-z_{d,i}}, \boldsymbol{w}_{1:D}, \boldsymbol{\alpha_{1:D}}, \boldsymbol{\beta_{1:K}}) &= \frac{\Pr(z_{d,i} = k, \boldsymbol{z}_{1:D}^{-z_{d,i}}, \boldsymbol{w}_{1:D}, \boldsymbol{\alpha}_{1:D}, \boldsymbol{\beta}_{1:K})}{\Pr(\boldsymbol{z}_{1:D}^{-z_{d,i}}, \boldsymbol{w}_{1:D}, \boldsymbol{\alpha}_{1:D}, \boldsymbol{\beta}_{1:K})} \\
&\propto (\alpha_{d,k} + m_{d,k}) \frac{\beta_{k,v} + n_{k,v}}{\beta_{k,\cdot} + n_{k,\cdot}} .
\end{aligned} \tag{3}$$

In MetaLDA, we have replaced $\boldsymbol{\alpha}_d$ and $\boldsymbol{\beta}_k$ with a log linear model in order to build informative priors from various side information associated with both documents and words. They are deterministically computed from a set of Gamma random variables, as shown in Step 2a and 1c in the generative process. Equation (3) can still be used in MetaLDA to sample the topic assignments. However, the major challenge is to sample the Gamma random variables, $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$ without significantly complicating the inference procedure.

4.1 Sampling Gamma random variable $\lambda_{l,k}$

$\lambda_{l,k}$ is involved in computing the Dirichlet prior over $\boldsymbol{\theta}_{1:D}$ via the parameter $\boldsymbol{\alpha}_{1:D}$. To sample $\lambda_{l,k}$, we expand the first Beta ratio in Eq. (2) with Gamma functions as follows:

$$\prod_{d=1}^{D} \frac{\text{Beta}_K(\boldsymbol{\alpha}_d + \boldsymbol{m}_d)}{\text{Beta}_K(\boldsymbol{\alpha}_d)} = \prod_{d=1}^{D} \underbrace{\frac{\Gamma(\alpha_{d,\cdot})}{\Gamma(\alpha_{d,\cdot} + m_{d,\cdot})}}_{\text{Gamma ratio 1}} \prod_{k=1}^{K} \underbrace{\frac{\Gamma(\alpha_{d,k} + m_{d,k})}{\Gamma(\alpha_{d,k})}}_{\text{Gamma ratio 2}} \tag{4}$$

where $\alpha_{d,\cdot} = \sum_{k=1}^{K} \alpha_{d,k}$, and $m_{d,\cdot} = \sum_{k=1}^{K} m_{d,k}$. It is not easy to directly work with these Gamma functions, while we replace α_k with $\prod_{l=1}^{L_{\text{doc}}} \lambda_{l,k}^{f_{d,l}}$. In order to retain the sampling efficiency of the standard LDA model, we appeal to data augmentation.

Gamma ratio 1 in Eq. (4) can be seen to be the marginalisation of a set of Beta random variables, therefore can be augmented as (similar to the sampling of the Pitman–Yor concentration parameter in [9]):

$$\underbrace{\frac{\Gamma(\alpha_{d,\cdot})}{\Gamma(\alpha_{d,\cdot} + m_{d,\cdot})}}_{\text{Gamma ratio 1}} \propto \int_{q_d} q_d^{\alpha_{d,\cdot}-1} (1 - q_d)^{m_{d,\cdot}-1} \tag{5}$$

where for each document d, $q_d \sim \text{Beta}(\alpha_{d,\cdot}, m_{d,\cdot})$. Given a set of $q_{1:D}$ for all the documents, Gamma ratio 1 can be approximated by the product of $q_{1:D}$, i.e., $\prod_{d=1}^{D} q_d^{\alpha_{d,\cdot}}$.

Gamma ratio 2 in Eq. (4) is the Pochhammer symbol for a rising factorial, which can be augmented with an auxiliary variable $t_{d,k}$ [7,31,40,45] as follows:

$$\underbrace{\frac{\Gamma(\alpha_{d,k} + m_{d,k})}{\Gamma(\alpha_{d,k})}}_{\text{Gamma ratio 2}} = \sum_{t_{d,k}=0}^{m_{d,k}} S_{t_{d,k}}^{m_{d,k}} \alpha_{d,k}^{t_{d,k}} \tag{6}$$

where S_t^m indicates an unsigned Stirling number of the first kind. Gamma ratio 2 is indeed a normalising constant for the probability of the number of tables in the Chinese Restaurant Process (CRP) [5], $t_{d,k}$ can be sampled by a CRP with $\alpha_{d,k}$ as the concentration and $m_{d,k}$ as the number of customers:

$$t_{d,k} = \sum_{i=1}^{m_{d,k}} \text{Bern}\left(\frac{\alpha_{d,k}}{\alpha_{d,k} + i}\right) \tag{7}$$

where $\text{Bern}(\cdot)$ samples a sequence of binary variables from the Bernoulli distribution. The complexity of sampling $t_{d,k}$ by Eq. (7) is $\mathcal{O}(m_{d,k})$. For large $m_{d,k}$, as the standard deviation of $t_{d,k}$ is $\mathcal{O}(\sqrt{\log m_{d,k}})$ [5], one can sample $t_{d,k}$ in a small window around the current value in complexity $\mathcal{O}(\sqrt{\log m_{d,k}})$.

By ignoring the terms unrelated to α, the augmentation of Eq. (6) can be simplified to a single term $\alpha_{d,k}^{t_{d,k}}$. With those auxiliary variables, we can simplify Eq. (4) as:

$$\prod_{d=1}^{D} q_d^{\alpha_{d,\cdot}} \prod_{k=1}^{K} \alpha_{d,k}^{t_{d,k}} = \prod_{d=1}^{D} \prod_{k=1}^{K} q_d^{\alpha_{d,k}} \alpha_{d,k}^{t_{d,k}} \tag{8}$$

Now, replacing $\alpha_{d,k}$ with $\lambda_{l,k}$ (i.e., $\alpha_{d,k} = \prod_{l=1}^{L_{\text{doc}}} \lambda_{l,k}^{f_{d,l}}$), we get:

$$\begin{aligned}
&\left(\prod_{d=1}^{D}\prod_{k=1}^{K} e^{\alpha_{d,k}\log q_d}\right)\left(\prod_{d=1}^{D}\prod_{k=1}^{K}\left(\prod_{l=1}^{L_{\text{doc}}}\lambda_{l,k}^{f_{d,l}}\right)^{t_{d,k}}\right)\\
&= \left(\prod_{d=1}^{D}\prod_{k=1}^{K} e^{-\alpha_{d,k}\log\frac{1}{q_d}}\right)\left(\prod_{l=1}^{L_{\text{doc}}}\prod_{k=1}^{K}\lambda_{l,k}^{\sum_{d=1}^{D} f_{d,l}t_{d,k}}\right)\\
&= \left(\prod_{k=1}^{K} e^{-\sum_{d=1}^{D}\alpha_{d,k}\log\frac{1}{q_d}}\right)\left(\prod_{l=1}^{L_{\text{doc}}}\prod_{k=1}^{K}\lambda_{l,k}^{\sum_{d=1}^{D} f_{d,l}t_{d,k}}\right) \qquad (9)
\end{aligned}$$

Recall that all the document labels are binary and $\lambda_{l,k}$ is involved in computing $\alpha_{d,k}$ if and only if $f_{d,l} = 1$. Extracting all the terms related to $\lambda_{l,k}$ in Eq. (9), we get the posterior likelihood of $\lambda_{l,k}$:

$$e^{-\lambda_{l,k}\left(\sum_{d=1:f_{d,l}=1}^{D}\frac{\alpha_{d,k}}{\lambda_{l,k}}\log\frac{1}{q_d}\right)}\lambda_{l,k}^{\sum_{d=1}^{D} f_{d,l}t_{d,k}}$$

where $\frac{\alpha_{d,k}}{\lambda_{l,k}}$ is the value of $\alpha_{d,k}$ with $\lambda_{l,k}$ removed when $f_{d,l} = 1$. With these data augmentation techniques, the likelihood is transformed into a form that is conjugate to the gamma prior of $\lambda_{l,k}$.

$$\begin{aligned}
\Pr(\lambda_{l,k}) &\propto e^{-\lambda_{l,k}\left(\sum_{d=1:f_{d,l}=1}^{D}\frac{\alpha_{d,k}}{\lambda_{l,k}}\log\frac{1}{q_d}\right)}\lambda_{l,k}^{\sum_{d=1}^{D} f_{d,l}t_{d,k}}\lambda_{l,k}^{\mu_0-1}e^{-\lambda_{l,k}\mu_0}\\
&= e^{-\lambda_{l,k}\left(\mu_0-\sum_{d=1:f_{d,l}=1}^{D}\frac{\alpha_{d,k}}{\lambda_{l,k}}\log q_d\right)}\lambda_{l,k}^{\mu_0+\sum_{d=1}^{D} f_{d,l}t_{d,k}-1}
\end{aligned}$$

Therefore, it is straightforward to yield the following sampling strategy for $\lambda_{l,k}$:

$$\lambda_{l,k} \sim \text{Ga}(\mu', \mu'') \qquad (10)$$

$$\mu' = \mu_0 + \sum_{d=1:f_{d,l}=1}^{D} t_{d,k} \qquad (11)$$

$$\mu'' = \mu_0 - \sum_{d=1:f_{d,l}=1}^{D}\frac{\alpha_{d,k}}{\lambda_{l,k}}\log q_d \qquad (12)$$

Before $\lambda_{l,k}$ is sampled, the value of $\alpha_{d,k}$ can be computed and cached. After a new value of $\lambda_{l,k}$ is sampled, $\alpha_{d,k}$ is updated by:

$$\alpha_{d,k} \leftarrow \frac{\alpha_{d,k}\lambda'_{l,k}}{\lambda_{l,k}},\ \forall\ 1 \le d \le D : f_{d,l} = 1 \qquad (13)$$

where $\lambda'_{i,k}$ is the newly sampled value of $\lambda_{i,k}$.

To sample/compute Eqs. (10)–(13), one only iterates over the documents where label l is active (i.e., $f_{d,l} = 1$). Thus, the sampling for all λ takes $\mathcal{O}(D'KL_{\text{doc}})$ where D' is the average number of documents where a label is active (i.e., the column-wise sparsity of $\mathbf{F}$). It is usually that $D' \ll D$ because if a label exists in nearly all the documents, it provides little discriminative information and can then be neglected. This demonstrates how the sparsity of document meta-information is leveraged. Moreover, sampling all the tables t takes $\mathcal{O}(\tilde{N})$ ($\tilde{N}$ is the total number of words in $\mathcal{D}$) which can be accelerated with the window sampling technique explained above.

4.2 Sampling Gamma random variable $\delta_{l',k}$

The derivation of sampling $\delta_{l',k}$ is analogous to $\lambda_{l,k}$. Here, we use the same data augmentation methods for re-parameterising the second Beta ratio in Eq. (2), i.e.,

$$\prod_{k=1}^{K} \frac{\text{Beta}_V(\boldsymbol{\beta}_k + \boldsymbol{n}_k)}{\text{Beta}_V(\boldsymbol{\beta}_k)} = \prod_k \frac{\Gamma(\beta_{k,\cdot})}{\Gamma(\beta_{k,\cdot} + n_{k,\cdot})} \prod_v \frac{\Gamma(\beta_{k,v} + n_{k,v})}{\Gamma(\beta_{k,v})} \tag{14}$$

as

$$\prod_{k=1}^{K} \prod_{v=1}^{V} \hat{q}_k^{\beta_{k,v}} \beta_{k,v}^{t'_{k,v}} \tag{15}$$

where $\hat{q}_k \sim \text{Be}(\beta_{k,\cdot}, n_{k,\cdot})$ and $t'_{k,v} = \sum_{i=1}^{n_{k,v}} \text{Bern}\left(\frac{\beta_{k,v}}{\beta_{k,v}+i}\right)$. Now, we replace $\beta_{k,v}$ with $\prod_{l'=1}^{L_{\text{word}}} \delta_{l',k}^{g_{v,l'}}$,

$$\begin{aligned}
&\left(\prod_{k=1}^{K}\prod_{v=1}^{V} e^{-\delta_{l',k}\frac{\beta_{k,v}}{\delta_{l',k}}\log\frac{1}{\hat{q}_k}}\right)\left(\prod_{k=1}^{K}\prod_{v=1}^{V}\left(\prod_{l'=1}^{L_{\text{word}}}\delta_{l',k}^{g_{v,l'}}\right)^{t'_{k,v}}\right)\\
&= \prod_{k=1}^{K} e^{-\delta_{l',k}\left(\sum_{v=1}^{V}\frac{\beta_{k,v}}{\delta_{l',k}}\right)\log\frac{1}{\hat{q}_k}}\left(\prod_{k=1}^{K}\prod_{l'=1}^{L_{\text{word}}}\delta_{l',k}^{\sum_{v=1}^{V} g_{v,l'}t'_{k,v}}\right)
\end{aligned}$$

and then extract all the terms related to $\delta_{l',k}$ in Eq. (15), and add the Gamma prior, we derive the posterior of $\delta_{l',k}$:

$$\Pr(\delta_{l',k}) \propto e^{-\delta_{l',k}\left(\nu_0 - \log\hat{q}_k \sum_{v=1:g_{v,l'}=1}^{V}\frac{\beta_{k,v}}{\delta_{l',k}}\right)} \delta_{l',k}^{\nu_0 + \sum_v g_{v,l'}t'_{k,v} - 1}$$

We can then sample $\delta_{l',k}$ from a Gamma distribution parameterised with

$$\delta_{l',k} \sim \text{Ga}(\nu', \nu'') \tag{16}$$

$$\nu' = \nu_0 + \sum_{v=1:g_{v,l'}=1}^{V} t'_{k,v} \tag{17}$$

$$\nu'' = \nu_0 - \log\hat{q}_k \sum_{v=1:g_{v,l'}=1}^{V} \frac{\beta_{k,v}}{\delta_{l',k}} \tag{18}$$

$\beta_{k,v}$ can be updated in a similar way to $\alpha_{d,k}$, i,e,

$$\beta_{k,v} \leftarrow \frac{\beta'_{k,v}\delta'_{l',k}}{\delta_{l',k}},\ \forall\ 1 \le k \le K : g_{v,l'} = 1 \tag{19}$$

where $\delta'_{l',k}$ is newly sampled value of $\delta_{l',k}$. Sampling all δ takes $\mathcal{O}(V'KL_{\text{word}})$ where V' is the average number of tokens where a feature is active (i.e., the column-wise sparsity of $\mathbf{G}$ and usually $V' \ll V$) and sampling all the tables t' takes $\mathcal{O}(\tilde{N})$. Figure 3 illustrates the full sampling algorithm.

Require: $\mathcal{D}$, $\mathbf{F}$ (if available), $\mathbf{G}$ (if available), K, μ_0, ν_0, $MaxIteration$
Ensure: topic assignments for all words: $z_{d,i}$

Randomly initialise $z_{d,i}$, $\lambda_{l,k}$ (Step 1a), $\delta_{l',k}$ (Step 1b)
Compute $\alpha_{d,k}$ (Step 2a), $\beta_{k,v}$ (Step 1c), $m_{d,k}$, $n_{k,v}$
for $iter \leftarrow 1$ **to** $MaxIteration$ **do**
  for all document d **do**
    for all word $w_{d,i} = v$ ($z_{d,i} = k$) in d **do**
      $m_{d,k} = m_{d,k} - 1$, $n_{k,v} = n_{k,v} - 1$
      Sample new topic k' according to Eq. (3)
      $z_{d,i} = k'$, $m_{d,k'} = m_{d,k'} + 1$, $n_{k',v} = n_{k',v} + 1$
    end for
  end for
  for all document d **do**
    Sample q_d by $q_d \sim \text{Beta}(\alpha_{d,\cdot}, m_{d,\cdot})$
    for all topic k **do**
      Sample $t_{d,k}$ according to Eq. (7)
    end for
  end for
  for all document label l and topic k **do**
    Sample $\lambda_{l,k}$ according to Eq. (10) to Eq. (12)
    Update $\alpha_{d,k}$ according to Eq. (13)
  end for
  for all topic k **do**
    Sample $\hat{q}_k$ by $\hat{q}_k \sim \text{Beta}(\beta_{k,\cdot}, n_{k,\cdot})$
    for all word v **do**
      Sample $t'_{k,v}$ by $t'_{k,v} = \sum_{i=1}^{n_{k,v}} \text{Bern}\left(\frac{\beta_{k,v}}{\beta_{k,v}+i}\right)$
    end for
  end for
  for all word feature l' and topic k **do**
    Sample $\delta_{l',k}$ according to Eq. (16) to Eq. (18)
    Update $\beta_{k,v}$ according to Eq. (19)
  end for
end for

Fig. 3 Collapsed Gibbs sampling algorithm for MetaLDA

4.3 MetaLDA as a hyper-parameter sampling approach

Besides the observed labels/features associated with the datasets, a default label/feature for each document/word is introduced in MetaLDA, which is always equal to 1. The default can be interpreted as the bias term in α/β, which is supposed to capture the information unrelated to the labels/features. When working without document labels with the default, MetaLDA samples the Dirichlet parameters (i.e., Hyper-parameters of LDA) of the document-topic distributions, α, according to the statistics in the target corpus. Similarly, without word features, the Dirichlet parameters of the topic-word distributions, β, are sampled. We demonstrate this by taking the document-topic distributions as an example.

Now assume each document only has a default label that is always equal to 1, i.e., $f_{d,0} = 1$ and $f_{d,l} = 0$ for all $l > 0$. According to our construction (Step 1 and 2a), $a_{d,k} = \lambda_{0,k}$ for all the document. In other words, all the documents share the same asymmetric Dirichlet prior

on the document-topic distributions ($\boldsymbol{\theta}_d$) which is constructed as follows:

$$\alpha_k \sim \text{Ga}(\mu_0, \nu_0) \tag{20}$$

$$\boldsymbol{\theta}_d \sim \text{Dir}_K(\boldsymbol{\alpha}) \tag{21}$$

In this case, we can sample α_k as follows:

$$\alpha_k \sim \text{Ga}\left(\mu_0 + t_{\cdot,k}, \mu_0 - \sum_{d=1}^{D} \log q_d\right) \tag{22}$$

Alternatively, we can vary MetaLDA to have a **symmetric Dirichlet prior**:

$$\alpha \sim \text{Ga}(\mu_0, \mu_0) \tag{23}$$

$$\boldsymbol{\theta}_d \sim \text{Dir}_K(\alpha, \ldots, \alpha) \tag{24}$$

In this case, we can sample α as follows:

$$\alpha \sim \text{Ga}\left(\mu_0 + t_{\cdot,\cdot}, \mu_0 - \sum_{d=1}^{D} \log q_d\right) \tag{25}$$

Discussed in [6,33], sampling the Dirichlet priors can gain significant performance improvement in topic models. In the case where document labels/word features are not used, MetaLDA offers an alternative hyper-parameter sampling approach to the methods such as fixed-point iterations [24] and Newton–Raphson [32]. These methods use MAP to optimise the hyper-parameters while ours uses MCMC sampling. We would like to point out that MetaLDA's sampling of symmetric Dirichlet prior is similar to the approach introduced in [31]. However the sampling of asymmetric prior was not considered in [31]. Compared with the built-in hyper-parameter sampling methods in Mallet[2] which are based on histograms of the statistics, our approach is more robust in the case where the statistics are not sufficient (e.g., short texts). This is further discussed with experiments in Sect. 5.4.3.

5 Experiments

In this section, we evaluate the proposed MetaLDA against several recent alternatives that also incorporate meta-information, using 6 real datasets including both regular and short texts. We will focus on the evaluation of

- the modelling accuracy of MetaLDA in terms of perplexity, a standard measure used in topic modelling. The goal is to study how the meta-information contributes to the predictive likelihood of unseen documents.
- the quality of topics learned by MetaLDA. It is interesting to see whether or not the meta-information will positively affect the topic coherence. We will report both quantitative and qualitative analyses.
- the running time of MetaLDA. The introduction of meta-information increases the modelling complexity to some extend. However, as we discussed in previous sections, MetaLDA can benefit from the local conjugacy given by the data augmentation methods, and also be parallelised using the same distributed framework [25] in Mallet. Therefore, we will empirically study the efficiency of MetaLDA.

[2] http://mallet.cs.umass.edu.

Besides, we will also study how word embeddings learnt by different techniques affect both perplexity and topic coherence.

5.1 Datasets

In the experiments, we used three regular and three short text datasets, which are as follows:

- *Reuters* is a widely used corpus extracted from the Reuters-21578 dataset where documents without any labels are removed.[3] There are 11,367 documents and 120 labels. Each document is associated with multiple labels. The vocabulary size is 8817, and the average document length is 73.
- *20NG* 20 Newsgroups is a widely used dataset consists of 18,846 news articles with 20 categories. The vocabulary size is 22,636 and the average document length is 108.
- *NYT* New York Times is extracted from the documents in the category "Top/News/Health" in the New York Times Annotated Corpus.[4] There are 52,521 documents and 545 unique labels. Each document is with multiple labels. The vocabulary contains 21,421 tokens, and there are 442 words in a document on average.
- *WS* Web Snippets, used in [17], contains 12,237 web search snippets and each snippet belongs to one of 8 categories. The vocabulary contains 10,052 tokens, and there are 15 words in one snippet on average.
- *TMN* Tag My News, used in [26], consists of 32,597 English RSS news snippets from Tag My News. With a title and a short description, each snippet belongs to one of 7 categories. There are 13,370 tokens in the vocabulary, and the average length of a snippet is 18.
- *AN* ABC News, is a collection of 12,495 short news descriptions and each one is in multiple of 194 categories. There are 4255 tokens in the vocabulary, and the average length of a description is 13.

All the datasets were tokenised by Mallet (see footnote 2) and we removed the words that exist in less than 5 documents and more than 95% of the documents.

5.2 Meta-information settings

At the document level, the labels associated with documents in each dataset were used as the meta-information. At the word level, we used a set of binarised word embeddings as word features (see footnote 3), which are obtained from real-valued word embeddings such as GloVe or word2vec. To binarise word embeddings, we first adopted the following method similar to [11]:

$$g'_{v,j} = \begin{cases} 1, & \text{if } g''_{v,j} > \text{Mean}_+(\boldsymbol{g}''_v) \\ -1, & \text{if } g''_{v,j} < \text{Mean}_-(\boldsymbol{g}''_v) \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

where $\boldsymbol{g}''_v$ is the original embedding vector for word v, $g'_{v,j}$ is the binarised value for jth element of $\boldsymbol{g}''_v$, and $\text{Mean}_+(\cdot)$ and $\text{Mean}_-(\cdot)$ are the average value of all the positive elements and negative elements, respectively.

[3] MetaLDA is able to handle documents/words without labels/features. But for fair comparison with other models, we removed the documents without labels and words without features.

[4] https://catalog.ldc.upenn.edu/ldc2008t19.

The insight is that we only consider features with strong opinions (i.e., large positive or negative value) on each dimension. To transform $g' \in \{-1, 1\}$ to the final $g \in \{0, 1\}$, we use two binary bits to encode one dimension of $g'_{v,j}$: the first bit is on if $g'_{v,j} = 1$ and the second is on if $g'_{v,j} = -1$. This means that if the original embeddings are 100-dimensional, the binarised embeddings will be with 200 dimensions. In our experiments, we also tried some other word embedding binarisation methods including the one in [10]. However, the performance with those binarisation methods is not comparable with the one we proposed above. Therefore, the experimental results with different binarisation methods will not be reported.

In the perplexity and topic coherence evaluation, i.e., Sects. 5.4 and 5.5, we will use the 50-dimensional GloVe word embeddings pre-trained on Wikipedia[5] as the source of word features. We then study how different word embedding sources influence the performance of our model in Sect. 5.6. It is noteworthy that MetaLDA can also work with other word features such as semantic similarity.

5.3 Compared models and parameter settings

We evaluate the performance of the following models:

- *MetaLDA* and its variants: the proposed model and its variants. Here we use MetaLDA to indicate the model considering both document labels and word features. Several variants of MetaLDA with document labels and word features separately were also studied, which are shown in Table 3. These variants differ in the method of estimating $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. All the models listed in Table 3 were implemented on top of Mallet. The hyper-parameters μ_0 and ν_0 were set to 1.0.
- *LDA* [4]: the baseline model. The Mallet implementation of SparseLDA [38] is used.
- *LLDA*, Labelled LDA [29] and **PLLDA**, Partially Labelled LDA [30]: two models that make use of multiple document labels. The original implementation[6] is used.
- *DMR*, LDA with Dirichlet Multinomial Regression [23]: a model that can use multiple document labels. The Mallet implementation of DMR based on SparseLDA was used. Following Mallet, we set the mean of λ to 0.0 and set the variances of λ for the default label and the document labels to 100.0 and 1.0, respectively.
- *WF-LDA*, Word Feature LDA [28]: a model with word features. We implemented it on top of Mallet and used the default settings in Mallet for the optimisation.
- *LF-LDA*, Latent Feature LDA [26]: a model that incorporates word embeddings. The original implementation[7] was used. Following the original paper, we used 1500 and 500 MCMC iterations for initialisation and sampling, respectively, and set λ to 0.6, and used the original 50-dimensional GloVe word embeddings as word features.
- *GPU-DMM*, Generalized Pólya Urn DMM [17]: a model that incorporates word semantic similarity. The original implementation[8] was used. The word similarity was generated from the distances of the word embeddings. Following the original paper, we set the hyper-parameters μ and ϵ to 0.1 and 0.7, respectively, and the symmetric document Dirichlet prior to $50/K$.

[5] https://nlp.stanford.edu/projects/glove/.

[6] https://nlp.stanford.edu/software/tmt/tmt-0.4/.

[7] https://github.com/datquocnguyen/LFTM.

[8] https://github.com/NobodyWHU/GPUDMM.

Table 3 MetaLDA and its variants

	Compute α with	Compute β with
MetaLDA	Document labels	Word features
MetaLDA-dl-def	Document labels	Default feature
MetaLDA-dl-0.01	Document labels	Symmetric 0.01 (fixed)
MetaLDA-def-wf	Default label	Word features
MetaLDA-0.1-wf	Symmetric 0.1 (fixed)	Word features
MetaLDA-def-def	Default label	Default feature

Table 4 Summary of the compared models

Meta Info used	Model
None	LDA [4]
	PTM [46]
	MetaLDA-def-def
Document labels	LLDA [29]
	PLLDA [30]
	DMR [23]
	MetaLDA-dl-def
	MetaLDA-dl-0.01
Word features	WF-LDA [28]
	LF-LDA [26]
	MetaLDA-def-wf
	MetaLDA-0.1-wf
	GPU-DMM [17]
Both	MetaLDA

– *PTM*, Pseudo document based Topic Model [46]: a model for short text analysis. The original implementation[9] was used. Following the paper, we set the number of pseudo-documents to 1000 and λ to 0.1.

All the models, except where noted, the symmetric parameters of the document and the topic Dirichlet priors were set to 0.1 and 0.01, respectively, and 2000 MCMC iterations are used to train the models. We summarise the compared models in terms of their usage of meta-information in Table 4.

5.4 Perplexity evaluation

Perplexity is a measure that is widely used [33] to evaluate the modelling accuracy of topic models. The lower the score, the higher the modelling accuracy. To compute perplexity, we randomly selected some documents in a dataset as the training set and the remaining as the test set. We first trained a topic model on the training set to get the word distributions of each topic k ($\boldsymbol{\phi}_k^{\text{train}}$). Each test document d was split into two halves containing every first and every second word, respectively. We then fixed the topics and trained the models on the first half to get the topic proportions ($\boldsymbol{\theta}_d^{\text{test}}$) of test document d and compute perplexity for

[9] http://ipv6.nlsde.buaa.edu.cn/zuoyuan/.

predicting the second half. With regard to MetaLDA, we fixed the matrices Φ^{train} and Λ^{train} output from the training procedure. On the first half of test document d, we computed the Dirichlet prior $\boldsymbol{\alpha}_d^{\text{test}}$ with Λ^{train} and the labels $\boldsymbol{f}_d^{\text{test}}$ of test document d (See Step 2a), and then point-estimated $\boldsymbol{\theta}_d^{\text{test}}$. We ran all the models 5 times with different random number seeds and report the average scores and the standard deviations.

In testing, we may encounter words that never occur in the training documents (a.k.a., unseen words or out-of-vocabulary words). There are two strategies for handling unseen words for calculating perplexity on test documents: ignoring them or keeping them in computing the perplexity. Here we investigate both strategies:

5.4.1 Perplexity computed without unseen words

In this experiment, the perplexity is computed only on the words that appear in the training vocabulary. Here we used 80% documents in each dataset as the training set and the remaining 20% as the test set.

Tables 5 and 6 show[10] the average perplexity scores with standard deviations for all the models. Note that: (1) The scores on AN with 150 and 200 topics are not reported due to overfitting observed in all the compared models. (2) Given the size of NYT, the scores of 200 and 500 topics are reported. (3) The number of latent topics in LLDA must equal to the number of document labels. (4) For PLLDA, we varied the number of topics per label from 5 to 50 (2 and 5 topics on NYT). The total number of topics used by PPLDA is the product of the number of labels and the number of topics per label.

The results show that the proposed MetaLDA outperformed all the competitors in terms of perplexity on nearly all the datasets, showing the benefit of using both document and word meta-information. Specifically, we have the following remarks:

- By looking at the models using only the document-level meta-information, we can see the significant improvement of these models over LDA, which indicates that document labels can play an important role in guiding topic modelling. Although the performance of the two variants of MetaLDA with document labels and DMR is comparable, our models run much faster than DMR, which will be studied later in Sect. 5.8.
- It is interesting that PLLDA with 50 topics for each label has better perplexity than MetaLDA with 200 topics in the 20NG dataset. With the 20 unique labels, the actual number of topics in PLLDA is 1000. However, if 10 topics for each label in PLLDA are used, which is equivalent to 200 topics in MetaLDA, PLLDA is outperformed by MetaLDA significantly.
- At the word level, MetaLDA-def-wf performed the best among the models with word features only. Moreover, our model has a clear advantage in running speed (see Table 13). Furthermore, comparing MetaLDA-def-wf with MetaLDA-def-def and MetaLDA-0.1-wf with LDA, we can see using the word features indeed improved perplexity.
- The scores show that the improvement gained by MetaLDA over LDA on the short text datasets is larger than that on the regular text datasets. This is expected because meta-information serves as complementary information in MetaLDA and can have significant impact when the data is sparse.
- It can be observed that models usually gained improved perplexity, if the Dirichlet parameter α is sampled/optimised, in line with [33]. We further study this in Sect. 5.4.3.

[10] For GPU-DMM and PTM, perplexity is not evaluated because the inference code for unseen documents is not public available. The random number seeds used in the code of LLDA and PLLDA are pre-fixed in the package. So the standard deviations of the two models are not reported.

Table 5 Perplexity comparison on the regular text datasets

Dataset	Reuters				20NG				NYT	
#Topics	50	100	150	200	50	100	150	200	200	500
LDA	677 ± 1	634 ± 2	629 ± 1	631 ± 1	2147 ± 7	1930 ± 7	1820 ± 5	1762 ± 3	2293 ± 8	2154 ± 4
MetaLDA-def-def	648 ± 3	592 ± 2	559 ± 1	540 ± 1	2093 ± 6	1843 ± 7	1708 ± 5	1626 ± 4	2258 ± 9	2079 ± 8
DMR	640 ± 1	577 ± 1	544 ± 2	526 ± 2	2080 ± 8	1811 ± 8	1670 ± 4	1578 ± 1	**2231** ± 13	**2013** ± 6
MetaLDA-dl-0.01	649 ± 2	582 ± 2	551 ± 3	530 ± 2	2067 ± 9	1821 ± 7	1680 ± 5	1590 ± 1	**2219** ± 4	**2018** ± 4
MetaLDA-dl-def	642 ± 3	576 ± 3	543 ± 1	526 ± 1	2050 ± 4	1804 ± 6	1675 ± 8	1589 ± 2	**2230** ± 3	**2022** ± 5
LF-LDA	841 ± 4	787 ± 4	772 ± 3	771 ± 4	2855 ± 21	2576 ± 3	2433 ± 7	2326 ± 8	2831 ± 2	2700 ± 5
WF-LDA	659 ± 2	616 ± 2	615 ± 1	613 ± 1	2089 ± 7	1875 ± 2	1784 ± 2	1727 ± 3	2287 ± 6	2134 ± 6
MetaLDA-0.1-wf	659 ± 3	621 ± 1	619 ± 1	623 ± 1	2098 ± 7	1887 ± 8	1796 ± 8	1744 ± 4	2283 ± 4	2143 ± 2
MetaLDA-def-wf	643 ± 2	582 ± 4	552 ± 3	535 ± 1	2068 ± 6	1819 ± 1	1685 ± 7	1600 ± 3	2260 ± 7	2095 ± 6
MetaLDA	**633** ± 2	**568** ± 2	**536** ± 2	**517** ± 1	**2025** ± 12	**1781** ± 8	**1640** ± 5	**1551** ± 6	**2217** ± 6	**2020** ± 6

Dataset	Reuters				20NG				NYT	
#Topics per label	5	10	20	50	5	10	20	50	2	5
PLLDA	714	708	733	829	1997	1786	1605	**1482**	2839	2846
LLDA	834				2607				2948	

The best results are highlighted in boldface

Table 6 Perplexity comparison without unseen words on the short text datasets. The best results are highlighted in boldface

Dataset	WS				TMN				AN	
#Topics	50	100	150	200	50	100	150	200	50	100
LDA	961 ± 6	878 ± 8	869 ± 6	888 ± 5	1969 ± 14	1873 ± 6	1881 ± 9	1916 ± 4	406 ± 14	422 ± 12
MetaLDA-def-def	884 ± 10	733 ± 6	671 ± 6	625 ± 6	1800 ± 11	1578 ± 19	1469 ± 4	1422 ± 6	352 ± 16	336 ± 11
DMR	845 ± 7	683 ± 4	607 ± 1	562 ± 2	1750 ± 8	1506 ± 3	1391 ± 7	1323 ± 5	326 ± 6	**290** ± 5
MetaLDA-dl-0.01	840 ± 7	693 ± 6	618 ± 3	588 ± 4	1767 ± 11	1528 ± 10	1416 ± 7	1345 ± 13	321 ± 13	303 ± 8
MetaLDA-dl-def	832 ± 4	679 ± 5	622 ± 7	582 ± 5	1720 ± 7	1505 ± 16	1395 ± 11	1325 ± 12	319 ± 9	293 ± 7
LF-LDA	1164 ± 6	1039 ± 17	1019 ± 11	992 ± 6	2415 ± 35	2393 ± 11	2371 ± 10	2374 ± 14	482 ± 17	514 ± 19
WF-LDA	894 ± 6	839 ± 6	827 ± 10	842 ± 4	1853 ± 6	1766 ± 12	1830 ± 60	1854 ± 45	397 ± 5	410 ± 6
MetaLDA-0.1-wf	889 ± 6	832 ± 3	839 ± 2	853 ± 4	1865 ± 4	1784 ± 2	1799 ± 9	1831 ± 6	388 ± 3	410 ± 8
MetaLDA-def-wf	830 ± 6	688 ± 8	624 ± 5	584 ± 4	1730 ± 14	1504 ± 3	1402 ± 13	1342 ± 4	346 ± 15	332 ± 8
MetaLDA	**774** ± 9	**627** ± 6	**572** ± 3	**534** ± 4	**1657** ± 4	**1415** ± 16	**1304** ± 6	**1235** ± 6	**314** ± 9	**293** ± 9

Dataset	WS				TMN				AN	
#Topics per label	5	10	20	50	5	10	20	50	5	10
PLLDA	1060	886	735	642	2181	1863	1647	1456	440	525
LLDA	1543				2958				392	

– On the AN dataset, there is no statistically significant difference between MetaLDA and DMR. On NYT, a similar trend is observed: the improvement in the models with the document labels over LDA is obvious but not in the models with the word features. Given the number of the document labels (194 of AN and 545 of NYT), it is possible that the document labels already offer enough information and the word embeddings have little contribution in the two datasets.

5.4.2 Perplexity computed with unseen words

To test the hypothesis that the incorporation of meta-information in MetaLDA can significantly improve the modelling accuracy in the cases where the corpus is sparse, we varied the proportion of documents used in training from 20 to 80% and used the remaining for testing. It is natural that when the proportion is small, the number of unseen words in testing documents will be large. Instead of simply excluding the unseen words in the previous experiments, here we compute the perplexity with unseen words for LDA, DMR, WF-LDA and the proposed MetaLDA. For perplexity calculation, $\phi_{k,v}^{\text{test}}$ for each topic k and each token v in the test documents is needed. If v occurs in the training documents, $\phi_{k,v}^{\text{test}}$ can be directly obtained. While if v is unseen, $\phi_{k,v}^{\text{unseen}}$ can be estimated by the prior:

$$\frac{\beta_{k,v}^{\text{unseen}}}{n_{k,\cdot}^{\text{train}} + \beta_{k,\cdot}^{\text{train}} + \beta_{k,\cdot}^{\text{unseen}}}.$$

For LDA and DMR which do not use word features, $\beta_{k,v}^{\text{unseen}} = \beta_{k,v}^{\text{train}}$; For WF-LDA and MetaLDA which are with word features, $\beta_{k,v}^{\text{unseen}}$ is computed with the features of the unseen token. Following Step 1c, for MetaLDA, $\beta_{k,v}^{\text{unseen}} = \prod_{l'}^{L_{\text{word}}} \delta_{l',k}^{g_{v,l}^{\text{unseen}}}$.

Figure 4 shows the perplexity scores on Reuters, 20NG, TMN and WS with 200, 200, 100 and 50 topics, respectively. MetaLDA outperformed the other models significantly with a lower proportion of training documents and relatively higher proportion of unseen words. The gap between MetaLDA and the other three models increases while the training proportion decreases. It indicates that the meta-information helps MetaLDA to achieve better modelling accuracy on predicting unseen words.

5.4.3 Perplexity evaluation for using MetaLDA as a hyper-parameter sampling approach

We further study how MetaLDA performs in terms of perplexity when used as a hyper-parameter sampling approach without meta-information. The experimental settings are the same as the ones used in Sect. 5.4.1. Table 7 shows the results of different variants of MetaLDA on hyper-parameter sampling. We would like to point out that MetaLDA-0.1-asym is equivalent to MetaLDA-0.1-def, MetaLDA-asym-0.01 is equivalent to MetaLDA-def-0.01, and MetaLDA-asym-asym is equivalent to MetaLDA-def-def in Table 3. Here we use the former to make the comparison clear. We have the following observations:

– In general, the best perplexity score is derived with the use of both asymmetric α and asymmetric β.

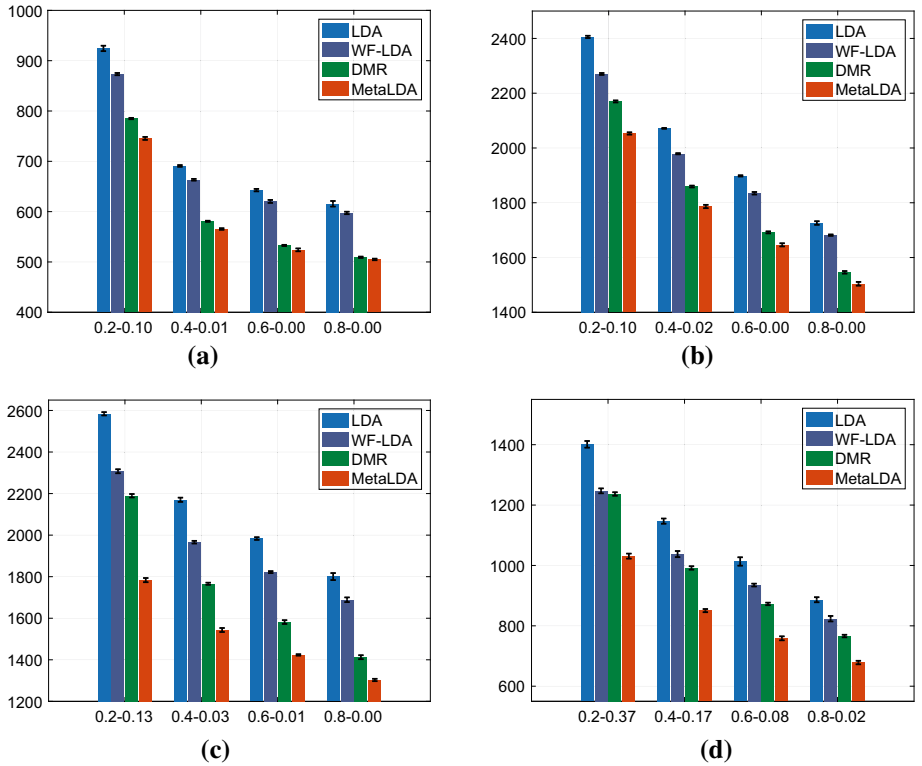


Fig. 4 Perplexity comparison with unseen words in different proportions of the training documents. Each pair of the numbers on the horizontal axis are the proportion of the training documents and the proportion of unseen tokens in the vocabulary of the test documents, respectively. For each setting, the four coloured bars from left to right correspond to LDA, WF-LDA, DMR and MetaLDA. The error bars are the standard deviations over 5 runs. **a** Reuters with 200 topics, **b** 20NG with 200 topics, **c** TMN with 100 topics, **d** WS with 50 topics

– If we fix the setting for the topic side and vary the setting for the document side (for example, compare MetaLDA-0.1-0.01, MetaLDA-sym-0.01 and MetaLDA-asym-0.01), we can derive that 1) the use of sampled priors (either symmetric or asymmetric) can significantly lower the perplexity scores, This is in line with the findings in [33]; 2) using asymmetric prior can further decrease perplexity.
– Similarly, fixing the setting for the document side and varying the setting for the topic side (for example, comparing MetaLDA-sym-0.01, MetaLDA-sym-sym and MetaLDA-sym-asym), we found that sampling either symmetric or asymmetric prior on per-topic word distributions does not significantly affect the perplexity scores, which also complies with [33]. However, there is a subtle difference: for our method an asymmetric prior on per-topic word distributions is marginally better, whereas it is often worse in [33].
– Now comparing the last row in Table 7 with the corresponding results in Tables 5 and 6 shows that constructing the priors with meta-information can further decrease the perplexity scores, which further proves our assumption that it is beneficial to use meta-information in topic modelling.

Table 7 Perplexity on 20NG with 200 topics, Reuters with 200 topics, WS with 100 topics

MetaLDA variants	20NG-200	Reuters-200	WS-100
MetaLDA-0.1-0.01 (LDA)	1762 ± 3	631 ± 1	878 ± 8
MetaLDA-0.1-sym	1774 ± 4	633 ± 1	888 ± 4
MetaLDA-0.1-asym	1764 ± 6	629 ± 2	884 ± 6
MetaLDA-sym-0.01	1652 ± 7	557 ± 5	744 ± 7
MetaLDA-sym-sym	1652 ± 6	557 ± 2	748 ± 6
MetaLDA-sym-asym	1641 ± 8	545 ± 2	743 ± 8
MetaLDA-asym-0.01	1618 ± 10	543 ± 1	726 ± 10
MetaLDA-asym-sym	1618 ± 11	542 ± 1	741 ± 11
MetaLDA-asym-asym	1626 ± 4	540 ± 1	733 ± 6

5.5 Topic coherence evaluation

We further evaluate the semantic coherence of the words in a topic learnt by LDA, PTM, DMR, LF-LDA, WF-LDA, GPU-DMM and MetaLDA. Here we use the normalised pointwise mutual information (NPMI) [1,16] to calculate topic coherence score for topic k with top T words:

$$\text{NPMI}(k) = \sum_{j=2}^{T} \sum_{i=1}^{j-1} \log \frac{p(w_j, w_i)}{p(w_j)p(w_i)} / -\log p(w_j, w_i),$$

where $p(w_i)$ is the probability of word i, and $p(w_i, w_j)$ is the joint probability of words i and j that co-occur together within a sliding window. Those probabilities were computed on an external large corpus, i.e., a 5.48 GB Wikipedia dump in our experiments. The NPMI score of each topic in the experiments is calculated with top 10 words ($T = 10$) by the Palmetto package.[11] Again, we report the average scores and the standard deviations over 5 random runs.

It is known that conventional topic models directly applied to short texts suffer from low quality topics, caused by the insufficient word co-occurrence information. Here we study whether or not the meta-information helps MetaLDA improve topic quality, compared with other topic models that can also handle short texts. Table 8 shows the NPMI scores on the three short text datasets. Higher scores indicate better topic coherence. All the models were trained with 100 topics. Besides the NPMI scores averaged over all the 100 topics, we also show the scores averaged over top 20 topics with highest NPMI, where "rubbish" topics are eliminated, following [37]. It is clear that MetaLDA performed significantly better than all the other models in WS and AN dataset in terms of NPMI, which indicates that MetaLDA can discover more meaningful topics with the document and word meta-information. We would like to point out that on the TMN dataset, even though the average score of MetaLDA is still the best, the score of MetaLDA overlaps with the others' when allowing for standard deviation, which indicates the difference is not statistically significant.

[11] http://palmetto.aksw.org.

Table 8 Topic coherence (NPMI) on the short text datasets

	All 100 topics			Top 20 topics		
	WS	TMN	AN	WS	TMN	AN
LDA	-0.0030 ± 0.0047	0.0319 ± 0.0032	-0.0636 ± 0.0033	0.1025 ± 0.0067	0.137 ± 0.0043	-0.0010 ± 0.0052
PTM	-0.0029 ± 0.0048	0.0355 ± 0.0016	-0.0640 ± 0.0037	0.1033 ± 0.0081	0.1527 ± 0.0052	0.0004 ± 0.0037
DMR	0.0091 ± 0.0046	0.0396 ± 0.0044	-0.0457 ± 0.0024	0.1296 ± 0.0085	0.1472 ± 0.1507	0.0276 ± 0.0101
LF-LDA	0.0130 ± 0.0052	0.0397 ± 0.0026	-0.0523 ± 0.0023	0.1230 ± 0.0153	0.1456 ± 0.0087	0.0272 ± 0.0042
WF-LDA	0.0091 ± 0.0046	0.0390 ± 0.0051	-0.0457 ± 0.0024	0.1296 ± 0.0085	0.1507 ± 0.0055	0.0276 ± 0.0101
GPU-DMM	-0.0934 ± 0.0106	-0.0970 ± 0.0034	-0.0769 ± 0.0012	0.0836 ± 0.0105	0.0968 ± 0.0076	-0.0613 ± 0.0020
MetaLDA	$\mathbf{0.0311} \pm 0.0038$	$\mathbf{0.0451} \pm 0.0034$	$-\mathbf{0.0326} \pm 0.0019$	$\mathbf{0.1511} \pm 0.0093$	$\mathbf{0.1584} \pm 0.0072$	$\mathbf{0.0590} \pm 0.0065$

Table 9 Perplexity comparison for MetaLDA with different word embeddings on WS and TMN

Dataset	WS		TMN	
#Topics	50	100	50	100
GloVe-50	774 ± 9	627 ± 6	1657 ± 4	1415 ± 16
SkipGram-50	782 ± 11	643 ± 5	1678 ± 3	1449 ± 10
CBOW-50	781 ± 6	636 ± 9	1683 ± 11	1430 ± 6
GloVe-100	776 ± 3	648 ± 3	1653 ± 8	1418 ± 12
SkipGram-100	786 ± 14	651 ± 5	1685 ± 17	1444 ± 4
CBOW-100	778 ± 3	645 ± 7	1675 ± 11	1442 ± 16

Table 10 Topic coherence (NPMI) comparison for MetaLDA with different word embeddings on WS and TMN

	All 100 topics		Top 20 topics	
	WS	TMN	WS	TMN
GloVe-50	0.0311 ± 0.0038	0.0451 ± 0.0034	0.1511 ± 0.0093	0.1584 ± 0.0072
SkipGram-50	0.0251 ± 0.0052	0.0385 ± 0.0046	0.1405 ± 0.0081	0.1521 ± 0.0086
CBOW-50	0.0324 ± 0.0035	0.0430 ± 0.0048	0.1580 ± 0.0055	0.1532 ± 0.0027
GloVe-100	0.0286 ± 0.0043	0.0455 ± 0.0026	0.1473 ± 0.0082	0.1522 ± 0.0043
SkipGram-100	0.0277 ± 0.0041	0.0424 ± 0.0046	0.1508 ± 0.0058	0.1545 ± 0.0051
CBOW-100	0.0308 ± 0.0046	0.0408 ± 0.0035	0.1439 ± 0.0092	0.1505 ± 0.0102

5.6 Changing word embeddings

In the above experiments, we used the binarised 50-dimensional GloVe embeddings as word features to demonstrate the superiority of MetaLDA over all the other competitors. It is also interesting to study how the performance of MetaLDA changes while we use different word embeddings. In this set of experiments, we varied the sources (i.e., the methods used to train the word embeddings) as well as the dimensions of those word embeddings. Here we used the embeddings pre-trained by three methods: GloVe, SkipGram[12] and CBOW [20].[12] For each word embedding method, 50 and 100 dimensional embeddings were used.

Tables 9 and 10 show the perplexity and topic coherence performance of MetaLDA, respectively, on the WS and TMN datasets. We followed the experiment settings used in the previous sections, except for the word features. MetaLDA work marginally better with GloVe embeddings than with word2vec embeddings. However, the difference is not significant, given the standard errors. The reasons might be:

1. The binarisation could water down the differences between word embeddings. Therefore, minor differences in word embedding might not significantly influence the performance. But it is interesting to develop a model that can directly utilise the real-valued word embeddings.
2. Using the embeddings as the prior information could make MetaLDA insensitive to the quality of binarised word embeddings.

[12] http://vsmlib.readthedocs.io/en/latest/tutorial/getting_vectors.html.

Table 11 Top 5 related topics of the document labels in the WS dataset with 100 topics

Label	Topic number	Top 5 words	$\lambda_{l,k}$
Business	72	Exchange stock estate currency trading	12.11
	93	Trade capital export venture import	8.63
	94	Jobs marketing job stress advertising	7.99
	49	Bank financial banking finance insurance	7.06
	28	Business management services resources solutions	6.51
Computers	20	intel device digital apple chip	9.49
	66	Internet bandwidth speed connection test	6.57
	35	Computer software engineering architecture graphics	6.19
	48	Linux operating system unix library	5.10
	86	Memory computer virtual cache security	4.77
Culture&Arts&Entertainment	47	Art arts museum painting surrealism	11.16
	45	Guitar piano jazz orchestra instruments	6.87
	7	Religion ancient culture roman christian	6.41
	41	Album tom beatles band julia	6.32
	22	Culture American Chinese history Japanese	5.54
Education and science	68	Journal journals international conference research	7.36
	19	Theoretical models model reasoning framework	7.21
	81	Thesis dissertation technical empirical edu	7.04
	15	Physics quantum theory mechanics mathematics	6.40
	37	Research discovery scientific science scientists	5.77
Engineering	70	wheels car rims custom truck	5.95
	24	Electrical products equipment electric motor	5.80
	74	Car cars automobile models howstuffworks	5.68
	80	Automatic gear transmission China manual	4.84
	88	Engine diesel fuel cylinder turbine	4.72

Table 11 continued

Label	Topic number	Top 5 words	$\lambda_{l,k}$
Health	51	Diet calorie nutrition health energy	6.65
	96	HIV disease aids prevention heart	6.55
	98	Drug system respiratory effects drugs	5.89
	82	Physical therapy american therapists checkup	5.85
	52	Cancer lung tobacco smoking risk	5.69
Politics & Society	97	Cabinet prime minister appointment pbs	7.59
	18	System republic government parliamentary election	7.58
	83	Military revolution force navy army	7.27
	89	House gov congress legislation senate	5.21
	16	Democracy party democratic communist social	5.04
Sports	10	Football league rugby team stadium	11.21
	38	Tennis golf tournament woods volleyball	10.17
	27	Match cricket quarterfinal game playoff	8.45
	21	Tickets chicago bulls basketball boxing	6.68
	14	Soccer goalkeeper diego maradona kick	5.58

5.7 Qualitative analysis

Now we show that besides better quantitative performance, MetaLDA with meta-information also allows more informative and interesting interpretation of the discovered topics.

As discussed in Sect. 3, the latent variable $\lambda_{l,k}$ is the weight measuring the association between document label l and topic k. Each label can be interpreted as an unnormalised mixture of topics, represented by a K-dimensional vector $\boldsymbol{\lambda}_l$. Therefore, similar to finding the top words for each topic, ranking $\lambda_{l,k}$ can give us the most related topics for each label. Table 11 shows the top 5 related topics among 100 discovered by MetaLDA for the 9 document labels in the WS dataset. For each topic, the top 5 words (ranked with $\phi_{k,v}$) are listed. The results show that the topics are closely related to the labels. For example, the top 5 topics for the "Computers" category describe hardware, software, internet, and system, which are different aspects of computers. The "Sports" category broadly covers football, rugby, tennis, golf, cricket, etc. The major topics discussed in the "Health" related documents include diet, infectious diseases, lung cancer and its causes, and so on.

Table 12 Top 3 related labels of the topics in the WS dataset with 100 topics

Topic number	Top 5 words	Labels
46	Programming web java server code	Computers
		Education and science
		Engineering
54	Diet calorie nutrition health energy	Health
		Engineering
		Business
20	Intel device digital apple chip	Computers
		Culture&Arts&Entertainment
		Business
17	Movie fiction documentary film soundtrack	Culture&Arts&Entertainment
		Education and Science
		Sports

Furthermore, MetaLDA can also automatically assign the labels to the latent topics, which is known as automatic topic labelling [15]. The method proposed in [15] generates label from the top-ranked topic terms and the titles of Wikipedia articles containing these terms. It is an ad hoc process. In contrast, MetaLDA automatically learns the association between the document labels and the latent topics via the association matrix $\boldsymbol{\lambda}$. Specifically, for each topic k, we rank the labels according the weight $\lambda_{l,k}$, and then retrieve the most likely labels for each topic. Table 12 shows some examples derived one the WS dataset. For instance, topic 46 is about web programming. The most probable label for this topic assigned by MetaLDA is "Computers". The second and third probable labels are also very related to this topic. Topic 17 is about movies, and the most probable label found by MetaLDA is "Culture&Arts&Entertainment". It is clear that topics and their most probable labels are well correlated. All these findings demonstrate that MetaLDA is able to discover meaningful topics and label the topics automatically.

5.8 Running time

In this section, we empirically study the efficiency of the models in term of per-iteration running time. The implementation details of our MetaLDA are as follows:

– The SparseLDA framework [38] reduces the complexity of LDA to be sub-linear by breaking the conditional of LDA into three "buckets", where the "smoothing only" bucket is cached for all the documents and the "document only" bucket is cached for all the tokens in a document. We adopted a similar strategy when implementing MetaLDA. When only the document meta-information is used, the Dirichlet parameters α for different documents in MetaLDA are different and asymmetric. Therefore, the "smoothing only" bucket has to be computed for each document, but we can cache it for all the tokens, which still gives us a considerable reduction in computing complexity. However, when the word meta-information is used, the SparseLDA framework no longer works in MetaLDA as the β parameters for each topic and each token are different.

Table 13 Running time (seconds per iteration) on 80% documents of each dataset

Dataset	Reuters				WS				NYT	
#Topics	50	100	150	200	50	100	150	200	200	500
LDA	0.0899	0.1023	0.1172	0.1156	0.0219	0.0283	0.0301	0.0351	0.7509	1.1400
PTM	4.9232	5.8885	7.2226	7.7670	1.1840	1.6375	1.8288	2.0030	–	–
DMR	0.6112	0.9237	1.2638	1.6066	0.4603	0.8549	1.2521	1.7173	13.7546	31.9571
MetaLDA-dl-0.01	0.1187	0.1387	0.1646	0.1868	0.0396	0.0587	0.0769	0.1121	2.4679	4.9928
LF-LDA	2.6895	5.3043	8.3429	11.4419	2.4920	6.0266	9.1245	11.5983	95.5295	328.0862
WF-LDA	1.0495	1.6025	3.0304	4.8783	1.8162	3.7802	6.1863	8.6599	14.0538	31.4438
GPU-DMM	0.4193	0.7190	1.0421	1.3229	0.1206	0.1855	0.2487	0.3118	–	–
MetaLDA-0.1-wf	0.2427	0.4274	0.6566	0.9683	0.1083	0.1811	0.2644	0.3579	4.6205	12.4177
MetaLDA	0.2833	0.5447	0.7222	1.0615	0.1232	0.2040	0.3282	0.4167	6.4644	16.9735

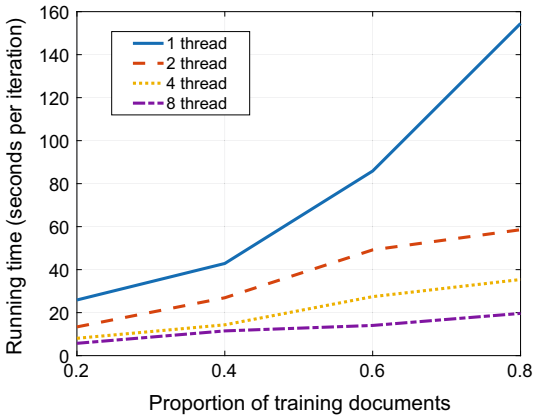


Fig. 5 MetaLDA's running time (seconds per iteration) on the NYT dataset with 500 topics with different proportions of training documents and different number of threads

– By adapting the Distributed framework in [25], our MetaLDA implementation runs in parallel with multiple threads, which makes MetaLDA able to handle larger document collections. The parallel implementation was tested on the NYT dataset.

The per-iteration running time of all the models is shown in Table 13. Note that:

– On the Reuters and WS datasets, all the models ran with a single thread on a desktop PC with a 3.40 GHz CPU and 16 GB RAM.
– Due to the size of NYT, we report the running time for the models that are able to run in parallel. All the parallelised models ran with 10 threads on a cluster with a 14-core 2.6 GHz CPU and 128 GB RAM.
– All the models were implemented in JAVA.
– As the models with meta-information add extra complexity to LDA, the per-iteration running time of LDA can be treated as the lower bound.

At the document level, both MetaLDA-df-0.01 and DMR use priors to incorporate the document meta-information and both of them were implemented in the SparseLDA framework. However, our variant is about 6 to 8 times faster than DMR on the Reuters dataset and more than 10 times faster on the WS dataset. Moreover, it can be seen that the larger the number of topics, the faster our variant is over DMR. At the word level, similar patterns can be observed: our MetaLDA-0.1-wf ran significantly faster than WF-LDA and LF-LDA especially when more topics are used (20–30 times faster on WS). It is not surprising that GPU-DMM has comparable running speed with our variant, because only one topic is allowed for each document in GPU-DMM. With both document and word meta-information, MetaLDA still ran several times faster than DMR, LF-LDA, and WF-LDA. On NYT with the parallel settings, MetaLDA maintains its efficiency advantage as well.

To further examine our model's scalability, we report the per-iteration running time of MetaLDA on NYT with 500 topics in Fig. 5. For this, we varied the proportion of training documents from 20 to 80% as well as the number of threads from 1 to 8. For the single thread version, when the training proportions change from 40 to 80% the per-iteration running time becomes 4 times slower. However, with multi-threading, our model scales much better. The per-iteration running time is only doubled while the training proportions quadruple. In terms of speed-up, the per-iteration running time increases nearly linearly with the number of threads. For example, given 60% training data, the per-iteration running time is reduced to half while the number of thread doubles.

6 Conclusion

In this article, we have presented a topic modelling framework named MetaLDA that can efficiently incorporate document and word meta-information. This results in a significant improvement over other models in terms of perplexity and topic quality. With two data augmentation techniques, MetaLDA enjoys full local conjugacy, allowing efficient Gibbs sampling, demonstrated by superiority in the per-iteration running time. MetaLDA[1] has been implemented within Mallet using the `DistributedLDA` framework, and works efficiently in a multicore context. Furthermore, without losing generality, MetaLDA can work with both regular texts and short texts. The improvement of MetaLDA over other models that also use meta-information is remarkable, particularly when the word-occurrence information is insufficient. Moreover, MetaLDA efficiently demonstrates that asymmetric-asymmetric LDA does beat regular symmetric LDA.

MetaLDA takes a particular approach for incorporating meta-information on topic models. However, the approach is general enough to be applied to other Bayesian probabilistic models that go beyond topics modelling, such as multi-label learning with sparse features [43]. Moreover, it would be interesting to extend our method to use real-valued meta-information directly without binarisation [41], which is the subject of future work.

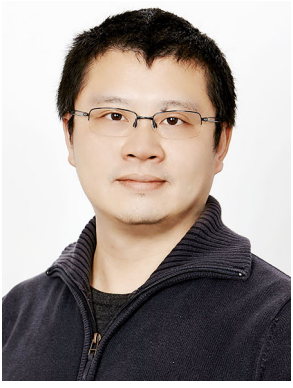
References

1. Aletras N, Stevenson M (2013) Evaluating topic coherence using distributional semantics. In: Proceedings of the 10th international conference on computational semantics, p 13–22
2. Andrzejewski D, Zhu X, Craven M (2009) Incorporating domain knowledge into topic modeling via Dirichlet forest priors. In: Proceedings of the 26th annual international conference on machine learning, p 25–32
3. Andrzejewski D, Zhu X, Craven M, Recht B (2011) A framework for incorporating general domain knowledge into Latent Dirichlet Allocation using first-order logic. In: Proceedings of the twenty-second international joint conference on artificial intelligence, p 1171–1177
4. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3:993–1022
5. Buntine W, Hutter M (2010) A Bayesian view of the Poisson–Dirichlet process. arXiv preprint arXiv:1007.0296
6. Buntine WL, Mishra S (2014) Experiments with non-parametric topic models. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, p 881–890
7. Chen C, Du L, Buntine W (2011) Sampling table configurations for the hierarchical Poisson–Dirichlet process. In: Proceedings of the 2011 European conference on Machine learning and knowledge discovery in databases, p 296–311
8. Das R, Zaheer M, Dyer C (2015) Gaussian LDA for topic models with word embeddings. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing, p 795–804
9. Du L, Buntine W, Jin H, Chen C (2012) Sequential latent Dirichlet allocation. Knowl Inf Syst 31(3):475–503
10. Faruqui M, Tsvetkov Y, Yogatama D, Dyer C, Smith N (2015) Sparse overcomplete word vector representations. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing, p 1491–1500
11. Guo J, Che W, Wang H, Liu T (2014) Revisiting embedding features for simple semi-supervised learning. In: Proceedings of the 2014 conference on empirical methods in natural language processing, p 110–120
12. Hong L, Davison BD (2010) Empirical study of topic modeling in Twitter. In: Proceedings of the first workshop on social media analytics, p 80–88
13. Hu C, Rai P, Carin L (2016) Non-negative matrix factorization for discrete data with hierarchical side-information. In: Proceedings of the 19th international conference on artificial intelligence and statistics, p 1124–1132
14. Kim D, Oh A (2017) Hierarchical Dirichlet scaling process. Mach Learn 106(3):387–418

15. Lau JH, Grieser K, Newman D, Baldwin T (2011) Automatic labelling of topic models. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, p 1536–1545
16. Lau JH, Newman D, Baldwin T (2014) Machine reading tea leaves: automatically evaluating topic coherence and topic model quality. In: Proceedings of the 14th conference of the european chapter of the association for computational linguistics, p 530–539
17. Li C, Wang H, Zhang Z, Sun A, Ma Z (2016) Topic modeling for short texts with auxiliary word embeddings. In: Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval, p 165–174
18. Mcauliffe JD, Blei DM (2008) Supervised topic models. Adv Neural Inf Process Syst 20:121–128
19. Mehrotra R, Sanner S, Buntine W, Xie L (2013) Improving LDA topic models for microblogs via tweet pooling and automatic labeling. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, p 889–892
20. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. In: International conference on learning representations (workshop)
21. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionally. Adv Neural Inf Process Syst 26:3111–3119
22. Miller GA (1995) WordNet: a lexical database for English. Commun ACM 38(11):39–41
23. Mimno D, McCallum A (2008) Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In: Proceedings of the 24th conference in uncertainty in artificial intelligence, p 411–418
24. Minka T (2000) Estimating a dirichlet distribution
25. Newman D, Asuncion A, Smyth P, Welling M (2009) Distributed algorithms for topic models. J Mach Learn Res 10:1801–1828
26. Nguyen DQ, Billingsley R, Du L, Johnson M (2015) Improving topic models with latent feature word representations. Trans Assoc Comput Linguist 3:299–313
27. Pennington J, Socher R, Manning C (2014) GloVe: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing, p 1532–1543
28. Petterson J, Buntine W, Narayanamurthy SM, Caetano TS, Smola AJ (2010) Word features for latent Dirichlet allocation. Adv Neural Inf Process Syst 23:1921–1929
29. Ramage D, Hall D, Nallapati R, Manning CD (2009) Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In: Proceedings of the 2009 conference on empirical methods in natural language processing, p 248–256
30. Ramage D, Manning CD, Dumais S (2011) Partially labeled topic models for interpretable text mining. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, p 457–465
31. Teh YW, Jordan MI, Beal MJ, Blei DM (2006) Hierarchical Dirichlet processes. J Am Stat Assoc 101(476):1566–1581
32. Wallach HM (2008) Structured topic models for language. Ph.D. thesis, University of Cambridge
33. Wallach HM, Mimno DM, McCallum A (2009) Rethinking LDA: why priors matter. Adv Neural Inf Process Syst 22:1973–1981
34. Wang C, Blei DM (2011) Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining, p 448–456
35. Xie P, Yang D, Xing E (2015) Incorporating word correlation knowledge into topic modeling. In: Proceedings of the 2015 conference of the north American chapter of the association for computational linguistics: human language technologies, p 725–734
36. Xun G, Gopalakrishnan V, Ma F, Li Y, Gao J, Zhang A (2016) Topic discovery for short texts using word embeddings. In: Proceedings of IEEE 16th international conference on data mining, p 1299–1304
37. Yang Y, Downey D, Boyd-Graber J (2015) Efficient methods for incorporating knowledge into topic models. In: Proceedings of the 2015 conference on empirical methods in natural language processing, p 308–317
38. Yao L, Mimno D, McCallum A (2009) Efficient methods for topic model inference on streaming document collections. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, p 937–946
39. 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, p 233–242
40. Zhao H, Du L, Buntine W (2017) Leveraging node attributes for incomplete relational data. In: Proceedings of the 34th international conference on machine learning, p 4072–4081

41. Zhao H, Du L, Buntine W (2017) A word embeddings informed focused topic model. In: Proceedings of the ninth Asian conference on machine learning, p 423–438
42. Zhao H, Du L, Buntine W, Liu G (2017) MetaLDA: a topic model that efficiently incorporates meta information. In: Proceedings of 2017 IEEE international conference on data mining, p 635–644
43. Zhao H, Rai P, Du L, Buntine W (2018) Bayesian multi-label learning with sparse features and labels, and label co-occurrences. In: Proceedings of the 21st international conference on artificial intelligence and statistics (**in press**)
44. Zhao WX, Jiang J, Weng J, He J, Lim EP, Yan H, Li X (2011) Comparing twitter and traditional media using topic models. In: Proceedings of the 33rd European conference on advances in information retrieval, p 338–349
45. Zhou M, Carin L (2015) Negative binomial process count and mixture modeling. IEEE Trans Pattern Anal Mach Intell 37(2):307–320
46. Zuo Y, Wu J, Zhang H, Lin H, Wang F, Xu K, Xiong H (2016) Topic modeling of short texts: a pseudo-document view. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, p 2105–2114

He Zhao is a Ph.D. student in the Faculty of Information Technology, Monash University, Australia. He received his B.Eng from Nankai University, China in 2011 and M.Eng from Nanjing University, China in 2014, respectively. His Ph.D. research project focuses on Bayesian (nonparametric) probabilistic models for discrete data (e.g., documents and networks).



Lan Du is currently a lecturer in the Faculty of Information Technology, Monash University. He previously worked in the language technology group in the Department of Computing, Macquarie university from 2012 to 2015. He received B.Sc. with honours and Ph.D. in computer science from the Australian National University (ANU) in 2017 and 2012, respectively. His research interests focus on statistical machine learning and its application in text analysis, relational learning, social network analysis, etc. With more than 30 papers published in these areas, he served/is serving on program committees for many top conferences in machine learning, data mining and natural language processing.

Wray Buntine is a full professor at Monash University in February 2014 after 7 years at NICTA in Canberra Australia. At Monash he is director of the Master of Data Science, the Faculty of IT's newest and in-demand degree, and was founding director of the innovative (online) Graduate Diploma of Data Science. He was previously at NICTA (Australia), Helsinki Institute for Information Technology, NASA Ames Research Center, University of California, Berkeley, and Google. He is known for his theoretical and applied work and in probabilistic methods for document and text analysis, social networks, data mining and machine learning. His recent focus has been with nonparametric methods in these areas.

Gang Liu Male, Ph.D., Associate Professor, Born in 1976. He got Ph.D. degree in Harbin Engineering University in China (2008), and major in computer applied technology. He conducted research in University of Illinois at Urbana–Champaign as visiting scholar in the group of Professor Jiawei Han in 2005. As a member of China Computer Federation, he has conducted and is conducting about 10 research projects such as National Science and Technology Support Plan and Chinese NSFC project as main researcher. He has published 30 papers in well-known journals such as JCIS, etc, which has been cited 20 times by SCI, EI. Dr. Liu has authored and co-authored 4 books in Chinese. He has filed 10 computer software copy authorities, and all of them have been authorized. He has developed and applied the advanced intelligent analysis and policy consistency verification technology, in auditing over 20 million attendees of Chinese social security.