Neurocomputing 424 (2021) 143–159

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Boosting prior knowledge in streaming variational Bayes ☆

Duc Anh Nguyen [a], Van Linh Ngo [b], Kim Anh Nguyen [b], Canh Hao Nguyen [a], Khoat Than [b],*

[a] Bioinformatics Center, Institute for Chemical Research, Kyoto University, Japan
[b] School of Information & Communication Technology, Hanoi University of Science and Technology, No. 1, Dai Co Viet road, Hanoi, Viet Nam

ARTICLE INFO

Article history:
Received 19 August 2019
Revised 12 August 2020
Accepted 10 October 2020
Available online 20 October 2020

Keywords:
Streaming variational Bayes
prior knowledge
online learning
short text
sparse data
noisy data streams

ABSTRACT

Exploiting prior/human knowledge is an effective way to enhance Bayesian models, especially in cases of sparse or noisy data, for which building an entirely new model is not always possible. There is a lack of studies on the effect of external prior knowledge in streaming environments, where the data come sequentially and infinitely. In this work, we show the problem of vanishing prior knowledge in *streaming variational Bayes*. This is a serious drawback in various applications. We then develop a simple framework to boost the external prior when learning a Bayesian model from data streams. By boosting, the prior knowledge can be maintained and efficiently exploited through each minibatch of streaming data. We evaluate the performance of our framework in four scenarios: streaming in synthetic data, streaming sentiment analysis, streaming learning for latent Dirichlet allocation, and streaming text classification, in comparison with the methods that do not keep priors. From extensive experiments, we find that when provided good external knowledge, our framework can improve the performance of a Bayesian model, often by a significant margin for noisy and short text streams.

1. Introduction

In the world of data explosion, designing a model that can capture continuously upcoming data is a basic need. By considering that new data depend on the past data in a probabilistic way, the Bayesian approach is highly suitable to model data streams where the data come sequentially and infinitely. Many researches have been successful in using this idea [1–5]. Through streaming learning methods, a model can be learned in a forward way without revisiting the old data, and hence more efficiently in terms of time and memory space. However, many challenges exist, including *extreme sparsity* and *noisy data*.

Sparsity in which the observed data are (extremely) sparse in nature is prevalent in practice, such as user ratings in recommender systems [6,7], and posts/comments from social networks [8–10]. Sparse data contain little information, and thus pose a severe challenge for modeling even in cases of very large number of samples [11].[1] Further, *noisy data* (e.g., comming from social networks or online forums) with contradicting or irrelevant information present inherent difficulties for modelling [12–16]. Fig. 1 shows two examples where the learnt models seem to encounter severe overfitting as learning from noisy and sparse data. For those challenges, designing an entirely new model is not always possible and possibly takes a high cost. Therefore providing human knowledge is often a good choice to improve a Bayesian model.

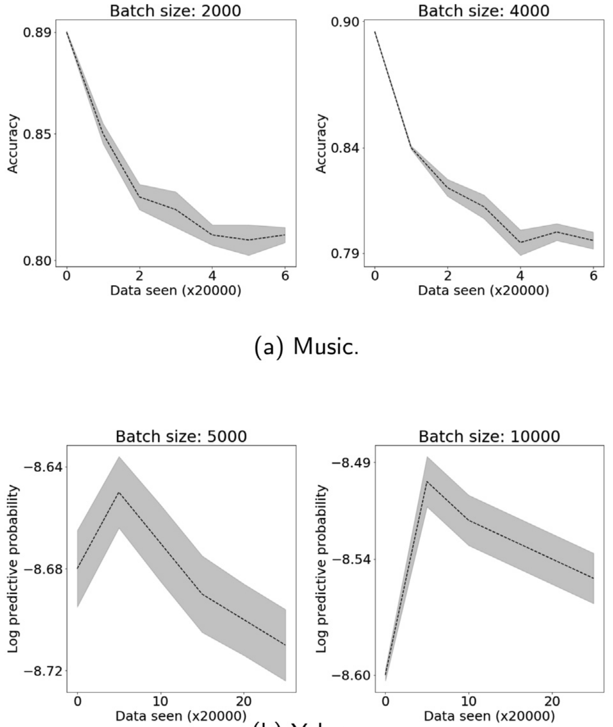
Consider the task of learning from a data stream, where the data come sequentially and infinitely. Existing learning methods have difficulties to effectively exploit prior knowledge. *Streaming variational Bayes* (*SVB*) [1] can use the external knowledge as initial prior at the first step of the learning process. However, in other steps, the prior is replaced by the posterior learned from the previous learning step. This strategy leads to losing information from the past, as shown in Section 3.2, and limits the effect of external knowledge. *Population variational Bayes* (*PVB*) [2] employs a population distribution to capture streaming data. Masegosa et al. [3] develop SVB further to balance the old and new knowledge, all learned from data, in a Bayesian way. Faraji et al. [5] study the same problem as [3], while Theis and Hoffman [19] develop a variant of stochastic variational inference (SVI) [20] for data streams. Due to the ignorance of external knowledge, those studies are limited when we want to exploit valuable knowledge in order to deal with the three challenges above. Note that good prior knowledge are now available in various forms including ontologies, open knowledge bases (e.g., Wikipedia), word embeddings, unsupervised pre-trained models [21,22], Zipf's law, etc.

☆ This research is funded by Vingroup Innovation Foundation (VINIF) annual research support program in project code VINIF.2019.DA18, and by the Office of Naval Research Global (ONRG), Air Force Office of Scientific Research (AFOSR), Asian Office of Aerospace Research & Development (AOARD) under Award Number N62909-18-1-2072.

* Corresponding author.
E-mail address: khoattq@soict.hust.edu.vn (K. Than).

[1] Although the proof of [11] focuses on a particular model, we believe that their result holds true for a large class of Bayesian models.

https://doi.org/10.1016/j.neucom.2020.10.026
0925-2312/© 2020 Elsevier B.V. All rights reserved.



(a) Music.

(b) Yahoo.

Fig. 1. Examples of severe overfitting when *Streaming variational Bayes* [1] does learning from noisy and short text streams, where 'Batch size' is the size of the data received at each time step. (a) shows the accuracy of the *Aspect and Sentiment Unification model* [17] for sentiment analysis. (b) shows the predictive probability of the *latent Dirichlet allocation* model [18] for text analysis. Higher is better. Music contains 136 K online reviews about musical instruments, collected from http://www.cs.jhu.edu/~mdredze/datasets/sentiment/. Yahoo is the dataset consisting of 500 K questions, crawled from http://answers.yahoo.com/, each question is noisy and extremely short [10]. The shadow shows the variance of the empirical result w. r.t. different runs.

In this paper, we make the following contributions:

- We show that *SVB* will quickly forget the prior knowledge through the streaming learning process.
- We then propose a simple framework for *boosting prior knowledge in streaming Bayesian learning* (*BPS*), a variant of SVB. Our framework emphasizes the role of the prior by encoding it in every step, therefore the valuable prior can be efficiently exploited in the entire learning process.
- We conduct experiments in four scenarios: streaming learning with synthetic data, streaming learning for unsupervised sentiment analysis [17], streaming learning for latent Dirichlet allocation (*LDA*) [18], and streaming classification [23,24]. Comparing with the framework that does not keep prior, *BPS* gives a significant improvement in performance. We further find that the improvement of *BPS* over *SVB* is often with a large margin for short text. This suggests that an appropriate exploitation of prior human knowledge in streaming environments will provide significant benefits, especially for short text and sparse data.

1.1. Further related work

Many researches have gained significant evidences in improving Bayesian models by incorporating prior knowledge. Diamond and Kaul [25] showed the effect of using prior from megatrials in Bayesian analysis for clinical data. Alfaro and Holder [26] used prior in phylogenetics to learn a Bayesian model of biological data. For text data, by capturing the Zipf's law [27,28], the performance of topic models can be improved [29] in comparison with *LDA* [18]. In text classification, Lin et al. [30] used an asymmetric prior which gave high weight for seed words of each class to gain better performance. In sentiment analysis, some studies [17,31] exploited a word list that consists of a predefined set of positive and negative words in learning Bayesian models to determine the sentiment of each document. However, most of those existing researches focus on static conditions, leaving an unexplored problem of how to exploit human knowledge in streaming environments where the data come sequentially and infinitely. Such an unexplored problem is the object of interest in this work.

AA Another term related to prior is *power prior* [32,33]. *Power prior* is used to balance the old knowledge learned from past data and the new knowledge learned from the current minibatch of data. While the initial prior is gained even when there is no data received, it can be provided from external knowledge that is outside of the observed data. Nonetheless, existing studies [5,3,19] did not emphasize in the role of the external prior but concern more on balancing the old and new knowledge which are all learned from data.

Other related topic is stochastic methods for training a Bayesian model from a dataset of big size. Examples include stochastic variational inference (SVI) [20] and memorized online variational inference (MOVI) [34]. The problem setting in our work is learning from a data stream, where the data come sequentially and infinitely. SVI and MOVI seem to be unsuitable for this problem setting. SVI not only requires to know the data size in advance, but also assumes the ability to have multiple learning passes over the whole data. MOVI divides dataset into the fixed number of minibatches and iterates through each minibatch multiple times. Both SVI and MOVI can work well on a big (but fixed) training data. However, in the streaming environment, since the dataset size is infinite and the underlying properties of the data may change over time, iteratively going through all minibatches is intractable. Some works [19,2] already pointed out those limitations of SVI (and MOVI).

1.2. Roadmap

Section 2 briefly presents some preliminaries with basic notations in Table 1. Section 3 reviews the *SVB* framework and the problem of vanishing priors. In Section 4, we propose the *BPS* framework. Section 5 presents case studies in applying *SVB* and *BPS* in streaming learning. Finally, Section 6 concludes our work.

2. Background

2.1. Variational inference

Variational inference is an approximate technique to solve Bayesian inference problem [35]. Formally, datum x is assumed to be generated from a Bayesian model with the prior η, the model parameter β and the hidden variable z. The task is to find the value of β and z to maximize the log likelihood function of the observed data x:

$\log p(x|\eta)$

In the variational inference, instead of searching for the whole space of β and z, we seek a variational distribution $q(\beta, z|\epsilon, \gamma)$, which approximates $p(\beta, z)$. We have:

$$\begin{aligned}\log p(x|\eta) &= \log \int\int_{\beta,z} \tfrac{p(x,\beta,z|\eta).q(\beta,z|\epsilon,\gamma)}{q(\beta,z|\epsilon,\gamma)} d\beta dz \\ &= E_q[\log(p(x,\beta,z|\eta)] - E_q[\log(q(\beta,z|\epsilon,\gamma)] \qquad (1) \\ &\quad + E_q[\log \frac{q(\beta,z|\epsilon,\gamma)}{p(\beta,z|\eta,x)}] \qquad (2)\end{aligned}$$

Table 1
Some basic notations.

x	a data point
C_i	a minibatch – a collection of data points
β, z, Φ	model's paremeters
η	prior
$\|\|X\|\|$	ℓ_1 norm of X
ϵ, γ	variational parameter symbols
ξ	natural parameter of an exponential family
$\tilde{\xi}_i$	learned information of a minibatch i
i.i.d	independent and identically distributed
s	scale ratio
r	boosting rate

Assuming that both β and z are continuous, otherwise integration is replaced by summation.

Note that the element in (2) is the Kullback Leibler (KL) divergence [36] between the approximate variational distribution $q(\beta, z|\epsilon, \gamma)$ and the true distribution $p(\beta, z|\eta, x)$ which has $KL(q||p) \geqslant 0$. Now denoting the element in (1) by $L(\epsilon, \gamma; \eta, x)$, we get:

$$\log p(x|\eta) = L(\epsilon, \gamma; \eta, x) + KL(q||p) \tag{3}$$

so that:

$$\log p(x|\eta) \geqslant L(\epsilon, \gamma; \eta, x)$$

$L(\epsilon, \gamma; \eta, x)$ is a lower bound of the likelihood function. Maximizing the lower bound L is equal to minimizing the KL distance between the variational distribution q and the posterior p. The form of q is selected in order to make L to be tractable or easier to optimize than the original likelihood function. The solution q is an approximation for the posterior p; hence:

$$p(\beta, z|\eta, x) \approx q(\beta, z|\epsilon, \gamma)$$

2.2. Additive property of the prior of exponential family

In probability and statistics, exponential family is a popular class of distributions that subsumes many common distributions including Bernoulli, Poisson, normal, gamma, multinomial and Dirichlet distributions (Table 2). Their density function is of the form:

$$f_X(x|\eta) = h(x)\exp((\eta \cdot T(x) - A(\eta))$$

where η is the natural parameter, $T(x)$ is the sufficient statistics, $h(x)$ is a known function, and $A(\eta)$ is a normalizing element.

Consider an observed datum x generated from an exponential distribution with prior which is encoded in the natural parameter η, and we know one more information that x is also contributed independently by another prior $\hat{\eta}$ in the same family of distribution, see Fig. 2. We have:

$$\begin{aligned} p(x|\eta, \tilde{\eta}) &\propto p(x|\eta)p(x|\hat{\eta}) & (4)\\ &\propto h(x)\hat{h}(x)\exp((\eta + \hat{\eta}) \cdot T(x)) & (5) \end{aligned}$$

which has $f_X(x|\eta + \hat{\eta})$ as its density function. The new prior for x has the natural parameter that is the sum of the two natural parameters: η and $\hat{\eta}$, this is the additive property of exponential family.

2.3. Latent Dirichlet allocation

We briefly describe *latent Dirichlet allocation* (LDA) [18], a well-known model, which will be referred in several parts later.

Table 2
Some distributions in the exponential family.

Distribution	Parameters	Natural parameter η	Sufficient statistic $T(x)$
Bernoulli	p	$\ln \frac{p}{1-p}$	x
Poisson	λ	$\ln \lambda$	x
Normal	μ, σ^2	$\begin{bmatrix} \frac{\mu}{\sigma^2} \\ \frac{1}{2\sigma^2} \end{bmatrix}$	$\begin{bmatrix} x \\ x^2 \end{bmatrix}$
Gamma	α, β	$\begin{bmatrix} \alpha - 1 \\ -\beta \end{bmatrix}$	$\begin{bmatrix} \ln x \\ x \end{bmatrix}$
Multinomial	$p_1, \ldots, p_k$ $\sum_1^k p_i = 1$	$\begin{bmatrix} \ln p_1 \\ \ldots \\ \ln p_k \end{bmatrix}$	$\begin{bmatrix} x_1 \\ \ldots \\ x_k \end{bmatrix}$
Dirichlet	$\alpha_1, \ldots, \alpha_k$	$\begin{bmatrix} \alpha_1 \\ \ldots \\ \alpha_k \end{bmatrix}$	$\begin{bmatrix} \ln x_1 \\ \ldots \\ \ln x_k \end{bmatrix}$

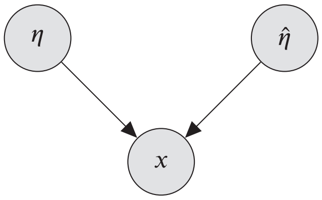


Fig. 2. Adding prior of exponential family.

Formally, *LDA* (Fig. 3) is a generative model for modeling text data. It assumes that a corpus C containing M documents is composed from K topics of V words, $\beta = (\beta_1, \ldots, \beta_K)$, each of which is drawn from a V-dimensional Dirichlet distribution: $\beta_k \sim Dirichlet(\eta)$. Each document d is a mixture of those topics and is presumed to be created by the following generative process:

1. Draw topic mixture $\theta|\alpha \sim Dirichlet(\alpha)$
2. For the i^{th} word in document d:
   - Draw topic index $z_i|\theta \sim Multinomial(\theta)$
   - Draw word $w_i|z_i, \beta \sim Multinomial(\beta_{z_i})$

The learning problem of *LDA* is to estimate the posterior $p(\theta, z, \beta|C, \alpha, \eta)$ of the latent variables. Nonetheless, this problem is intractable. Applying variational inference, we approximate

$$p(\theta, z, \beta|C, \alpha, \eta) \approx q(\theta, z, \beta|\gamma, \phi, \lambda) = q(\theta|\gamma)q(z|\phi)q(\beta|\lambda)$$

where γ, ϕ, λ are the parameters of the following variational distributions:

$$\begin{aligned} q(\theta) &= Dirichlet(\theta|\gamma)\\ q(\beta) &= Multinomial(\beta|\lambda)\\ q(z) &= Multinomial(z|\phi) \end{aligned}$$

and $\sum_k \phi_{dkv} = 1$. According to [18], we have the lower bound of the log likelihood of C:

$$\begin{aligned} L &= \sum_d (E_q \log(w_d, \theta_d, z, \beta) - E_q \log q(\gamma, \phi, \lambda))\\ &= \sum_d (\mathbb{E}_q[\log p(w_d|z_d, \boldsymbol{\beta})] + \mathbb{E}_q[\log p(z_d|\theta_d)]\\ &-\mathbb{E}_q[\log q(z_d)] + \mathbb{E}_q[\log p(\theta_d|\alpha)] - \mathbb{E}_q[\log q(\theta_d)])\\ &+\mathbb{E}_q[\log p(\boldsymbol{\beta}|\eta)] - \mathbb{E}_q[\log q(\boldsymbol{\beta})] \end{aligned}$$

Taking partial derivative of L with respect to ϕ and γ and set them to zero, we have the following inference equations for local variables:

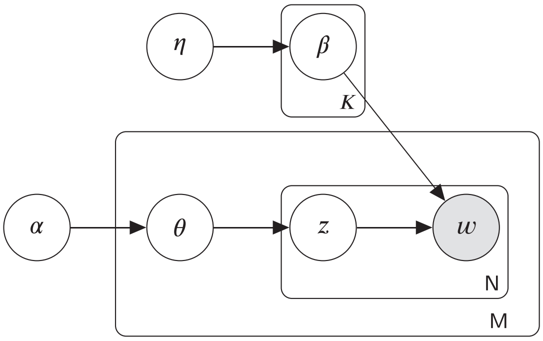


Fig. 3. Graphical representation for *LDA*.

```
Algorithm 1 Inference of local variables in LDA
Input: Document $d$, global variable $\lambda$
Output: Local variables $\gamma_d$, $\phi_d$
  Randomly initialize $\gamma_d, \phi_d$.
  repeat
    Compute $\phi_d$ as in Eq. 6
    Compute $\gamma_d$ as in Eq. 7
until convergence
```

```
Algorithm 2 Learning LDA using variational inference
Input: Prior $\eta$, hyper-parameter $\alpha$, data $C$
Output: $\lambda$
  Initialize: $\lambda_0 \leftarrow \eta$
  repeat
    Inference:
    for each document $d$ in $C$ do
      Infer $\phi_d, \gamma_d$ by Algorithm 1
    end for
    Update:
    $\lambda_{kv} = \eta + \sum_d \sum_v n_{dv} \phi_{dkv}$
    $\beta \quad \propto \lambda$
until convergence
```

$$\phi_{dkv} \propto \exp\{E_q[\log \theta_{dk}] + E_q[\log \beta_{kv}]\} \tag{6}$$

$$\gamma_{dk} = \alpha + \sum_v n_{dv} \phi_{dkv} \tag{7}$$

Similarity, by taking the partial derivative of L with to λ and set it to zero, the update equation for λ is:

$$\lambda_{kv} = \eta + \sum_d \sum_v n_{dv} \phi_{dkv} \tag{8}$$

where n_{dv} is the frequency of word $v \, (v \in \{1, \ldots, V\})$ appearing in document d.

We sum up the inference procedure using variational inference for *LDA* in Algorithm 1 and the learning algorithm for *LDA* in Algorithm 2.

3. Revisiting streaming variational Bayes

In this section, we briefly review the streaming variational Bayes framework (*SVB*) [1] for learning a Bayesian model from a data stream, then discuss the problem of vanishing prior in this framework.

3.1. Streaming variational Bayes

Streaming data is considered as a sequence of minibatches $\{C_i\}_{i=1 \to b}$, where b can be infinite. Each instance is assumed to be generated from a Bayesian model with parameter Φ. The learning problem is to find $\{\Phi\}$ that maximizes the posterior:

$$p(\Phi | C_1, C_2, \ldots, C_b) \tag{9}$$

There is a general framework for computing the posterior in Eq. 9. The idea is to use the prior from the past data and information from the current data. Fig. 4 provides a graphical representation of their idea.

Given the prior η, presuming that $b - 1$ minibatches have been processed, the posterior after b minibatches can be calculated using Bayes rule:

$$.p(\Phi | C_1, C_2, \ldots, C_{b-1}, C_b, \eta) \tag{10}$$

$$\propto p(C_b | \Phi, \eta) p(\Phi | C_1, C_2, \ldots, C_{b-1}, \eta) \tag{11}$$

$$\propto \frac{p(\Phi | C_b, \eta)}{p(\Phi | \eta)} p(\Phi | C_1, C_2, \ldots, C_{b-1}, \eta) \tag{12}$$

Which means that the posterior after b minibatches is proportional to the product of the likelihood of current data $p(C_b | \Phi, \eta)$ and its prior $p(\Phi | C_1, C_2, \ldots, C_{b-1}, \eta)$. Note that the prior element is also the posterior of $b - 1$ minibatches, so the posterior can be computed sequentially. Unfortunately, the posterior in Eq. 11 is often intractable to precisely compute. This problem leads to the need for an approximation method. Using the variational inference, we approximate:

$$p(\Phi | C) \approx q(\Phi | \xi) \propto \exp(\xi \cdot T(\Phi)) \tag{13}$$

and:

$$p(\Phi | C_1, C_2, \ldots, C_{b-1}, \eta) \approx q(\Phi | \xi_{b-1}) \tag{14}$$

$$p(\Phi | C_b, \eta) \approx q(\Phi | \hat{\xi}_b) \tag{15}$$

Assume that the initial prior $p(\Phi | \eta) = q(\Phi | \xi_0)$. Here the prior η is transformed to a natural parameter of an exponential family: $\xi_0 \leftarrow \eta$, so ξ_0 is also called as the initial prior knowledge. Now combining these approximates with Eq. 12, we have:

$$q(\Phi | \xi_b) \approx \frac{q(\Phi | \hat{\xi}_b)}{q(\Phi | \xi_0)} q(\Phi | \xi_{b-1}) \tag{16}$$

Taking the log function for both sides and use (13), we obtain

$$\xi_b = (\hat{\xi}_b - \xi_0) + \xi_{b-1} \tag{17}$$

Denote:

$$\hat{\xi}_b - \xi_0 = \tilde{\xi}_b \tag{18}$$

We have:

$$\xi_b = \tilde{\xi}_b + \xi_{b-1} \tag{19}$$

$$= \tilde{\xi}_b + \ldots + \tilde{\xi}_1 + \xi_0 \tag{20}$$

Eq. 19 is the streaming update function of the *SVB* framework [1]. It shows that the current model's parameter is equal to the sum of the past parameter ξ_{b-1} with the learned information $\tilde{\xi}_b$ from current data, and hence *SVB* provides us a principled way to model streaming data without revisiting past data.

3.2. The problem of vanishing prior

A drawback in the *SVB* framework is the vanishing prior problem. In other words, the information from the prior is probably vanished through learning process. This subsection will show this

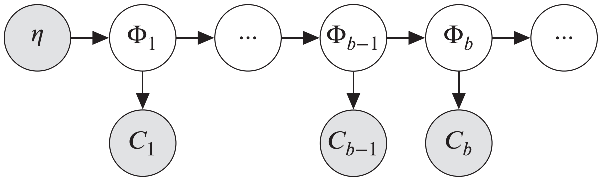


Fig. 4. Graphical representation of *SVB* for learning from streaming data.

phenomenon for *LDA* model in particular, and for any model learned by SVB.

Replacing Eq. 18 with $\xi_0 = \eta$ and $\hat{\xi} = \lambda$, we have the following for the bth minibatch:

$$\tilde{\xi}_{b_{kv}} = \hat{\xi}_{b_{kv}} - \xi_{0_{kv}} = \sum_{d \in C_b} \phi_{dkv} n_{dv}. \tag{21}$$

As a result, the prior ξ_0 in *SVB* is used only one time in the initial stage as in Eq. 20. It is clear that the learned parameters $\tilde{\xi}_b$ represent the learned information from minibatch data $b, \tilde{\xi}_1 + \ldots + \tilde{\xi}_b$ represents the total information learned from the data, and ξ_0 represents the information from the prior.

We will analyze the ratio between the norms of information learned from the data and the information from the prior.

Lemma 3.1 (*LDA*). *The SVB update of the LDA model has the following property:*

$$\begin{aligned} ||\xi_b - \xi_0|| = \quad & ||\widetilde{\xi}_1 + \ldots + \widetilde{\xi}_b|| \to +\infty \text{ as } b \to +\infty, \\ & \tfrac{||\widetilde{\xi}_1 + \ldots + \widetilde{\xi}_b||}{||\xi_0||} \to +\infty \text{ as } b \to +\infty, \end{aligned} \tag{22}$$

suggesting that as b increases, the information from the prior ξ_0 quickly becomes vanishing in comparison with the learned information $\tilde{\xi}_1 + \ldots + \tilde{\xi}_b$ from data.

Proof. It is clear that $||\xi_0||$ is a constant. We now prove $||\tilde{\xi}_1 + \ldots + \tilde{\xi}_b|| \to +\infty as b \to +\infty$. Because $\sum_k \phi_{dkv} = 1$ and $\tilde{\xi}_{bkv} \geqslant 0$, from Eq. 21 we have the following for any $b \geqslant 1$:

$$\begin{aligned} ||\tilde{\xi}_b|| = \sum_k \sum_v \tilde{\xi}_{bkv} &= \sum_{d \in C} \sum_k \sum_v \phi_{dkv} n_{dv} && (23) \\ &= \sum_{d \in C} \sum_v n_{dv} \sum_k \phi_{dkv} && (24) \\ &= \sum_{d \in C} n_d && (25) \\ &\geqslant 1. && (26) \end{aligned}$$

So

$$||\tilde{\xi}_1 + \ldots + \tilde{\xi}_b|| \geqslant b,$$

which means $||\widetilde{\xi}_1 + \ldots + \widetilde{\xi}_b|| \to +\infty$ as $b \to +\infty$. □

The result in Lemma 3.1 can be interpreted as following: In *SVB* with the initial prior only exists in the first stage, through the stream learning process, the new knowledge is added into the model's parameter. When data is big enough, almost all information in the model's parameter comes from the data and the role of the initial prior will vanish quickly. This problem can be easilly seen in LDA.Next, we discuss the vanishing prior phenomenon of *SVB* in broader contexts. For more complex models, such a phenomenon is not easily observed. We have to employ the well-known law of large numbers under some assumptions. Formally, we have the following.

Lemma 3.2 (*Other models*). *Assuming that the learned information $\{\tilde{\xi}_h\}_{h=1}^{b}$ are i.i.d samples from a probability distribution with mean $\bar{\xi} \neq 0$, we have the following property for SVB with probability 1:*

$$\begin{aligned} ||\xi_b - \xi_0|| = \quad & ||\widetilde{\xi}_1 + \ldots + \widetilde{\xi}_b|| \to +\infty \text{ as } b \to +\infty, \\ & \tfrac{||\widetilde{\xi}_1 + \ldots + \widetilde{\xi}_b||}{||\xi_0||} \to +\infty \text{ as } b \to +\infty, \end{aligned} \tag{27}$$

suggesting that as b increases, the prior ξ_0 quickly becomes vanishing in comparison with the learned information $\tilde{\xi}_1 + \ldots + \tilde{\xi}_b$ from data.

Proof. Using the law of large numbers, we have:

$$\Pr\left(\lim_{b \to +\infty} \frac{\tilde{\xi}_1 + \ldots + \tilde{\xi}_b}{b} = \bar{\xi} \right) = 1,$$

suggesting that

$$\Pr\left(\lim_{b \to +\infty} \frac{||\tilde{\xi}_1 + \ldots + \tilde{\xi}_b||}{b} = ||\bar{\xi}|| \right) = 1,$$

and because $||\xi_0||$ is a constant, we have:

$$\Pr\left(\lim_{b \to +\infty} \frac{||\tilde{\xi}_1 + \ldots + \tilde{\xi}_b||}{||\xi_0||} = +\infty \right) = 1.$$

□

This lemma shows that, in general using *SVB* in streaming learning will lead to the problem of losing the prior information. Although the assumption of Lemma 3.2 is not always met, the result provides a significant message for practice of streaming learning, especially when we have valuable prior knowledge about the domain/task of interest.

Considering the circumstances that we have valuable prior human knowledge that describes properties of data, losing the information of these priors in the learning process is highly wasteful. Bayesian methods often assume the data to be generated from a probabilistic process. If no knowledge is present, those methods can learn information efficiently only from data. However, note that most models are mis-specified, and as a consequence, might not model the data exactly. From this reason, combining prior human knowledge into a Bayesian model is an essential method to improve the quality of Bayesian models.

To overcome the problem of vanishing prior in *SVB*, we propose a framework for Bayesian learning to maintain the valuable information from the initial prior.

4. Boosting prior knowledge in streaming Bayesian learning

In this section, we first present the *BPS* framework for boosting the valuable prior knowledge in *SVB*. We then propose to choose the linear boosting function and analyze some advantages of such a choice. Finally, we discuss how to balance between boosted prior and the knowledge learned from each minibatch of data.

4.1. The BPS framework

The idea of *BPS* is to directly add one more prior information $\hat{\eta}$ to each minibatch. Fig. 5 depicts this idea clearly. The new prior $\hat{\eta}_b$ of the minibatch b is a boosting function of the initial prior η so that it reflects how the prior knowledge impacts to each minibatch of the streaming learning process.

$$\hat{\eta}_b = \hat{f}_b(\eta) \tag{28}$$

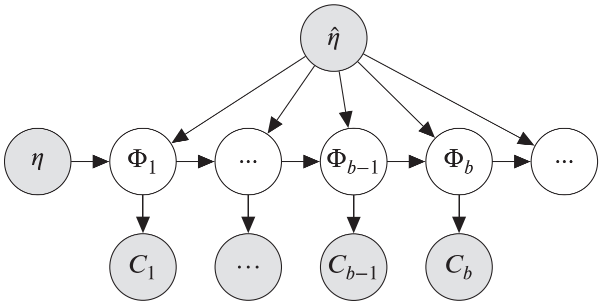


Fig. 5. Graphical representation for *BPS*.

An additional prior in the form of an exponential family $\hat{p}(\Phi|\hat{\eta}_b) \propto \exp(\hat{\eta}_b \cdot T(\Phi))$ is used in each minibatch. To emphasize the role of the new prior $\hat{\eta}_b$, we will optimize the following probability:

$$p(\Phi|\hat{\eta}_b)p(C_b|\Phi,\eta)p(\Phi|C_1,C_2,\ldots,C_{b-1},\eta) \tag{29}$$

$$=\frac{p(\Phi|\hat{\eta}_b)p(\Phi|C_b,\eta)}{p(\Phi|\eta)}p(\Phi|C_1,C_2,\ldots,C_{b-1},\eta) \tag{30}$$

With $\hat{\xi}_0^b \leftarrow \hat{\eta}_b$, using the additive property of prior for exponential family and the variational method, we have an approximation:

$$p(\Phi|\hat{\eta}_b)p(\Phi|C_b,\eta) \propto p(\Phi|C_b,\hat{\eta}_b+\eta) \tag{31}$$

$$\approx q(\Phi|\hat{\xi}_b) \tag{32}$$

Replacing Eqs. (32)–(30), we have the new update function:

$$\xi_b \leftarrow (\hat{\xi}_b - \xi_0) + \xi_{b-1} \tag{33}$$

Supposing that $\hat{\xi}_b$ can be decomposed into the sum of the learned information $\tilde{\xi}_b$ and the prior $\xi_0 + \hat{\xi}_0^b$, i.e., $\hat{\xi}_b = \tilde{\xi}_b + (\xi_0 + \hat{\xi}_0^b)$, then Eq. 33 becomes:

$$\xi_b = \tilde{\xi}_b + \hat{\xi}_0^b + \xi_{b-1} \tag{34}$$

The difference between the streaming update function in Eq. 34 of *BPS* and the one of *SVB* in Eq. 19 is that there is an additional element of the boosting prior $\hat{\xi}_0^b$. This property means that *BPS* boosts the knowledge from the initial prior through streaming learning process. If we set $\hat{\eta}_b = 0$, the *BPS* will become *SVB*.

4.2. Linear boosting function

In *BPS*, we introduce a boosting function in Eq. 28 that describes how the original prior impacts on the learned model. Now we propose a simple form of the boosting function in term of linear relationship with a *boosting rate* $r_b \in \mathbb{R}$:

$$\hat{\eta}_b = \hat{f}_b(\eta) = r_b\eta \tag{35}$$

Assuming η is the natural parameter in the exponential form of prior knowledge $p(\Phi|\eta) \propto \exp(\eta \cdot T(\Phi))$, we will show that with the linear function in Eq. 35, the new prior $p(\Phi|r\eta)$ will keep some main features of the original prior distribution with a level of importance.

Definition: *The feature points Φ_f of a probability distribution $p(\Phi|\eta)$ are defined by its local extrema.*

The feature points Φ_f can be used to describe the shape of the prior distribution, containing stationary points and boundary points. The local maximal points are positions that have high density, and the local minimal points have low density.

Taking derivative of the density functions of $p(\Phi|\eta)$ and $p(\Phi|r\eta)$ over Φ, we have:

$$g'_\eta = \frac{\partial(\exp(\eta \cdot T(\Phi))}{\partial(\Phi)} = \exp(\eta \cdot T(\Phi))\frac{\partial(\eta \cdot T(\Phi))}{\partial(\Phi)}$$

and

$$g'_{r\eta} = \frac{\partial(\exp(r\eta \cdot T(\Phi))}{\partial(\Phi)} = \exp(r\eta \cdot T(\Phi))\frac{r\partial(\eta \cdot T(\Phi))}{\partial(\Phi)}$$

Obviously, $g'_\eta = 0$ and $g'_{r\eta} = 0$ have the same set of solutions which means both $p(\Phi|\eta)$ and $p(\Phi|r\eta)$ have the *same feature points* Φ_f.

In other aspect, we interpret the value of $g\prime$ near a feature point as the changing of the concentrated density around the feature point. The big value of $g\prime$ means that there is a high concentrated density level or *a high important level*.

Assuming $\eta \cdot T(\Phi) > 0$, it is clear that if $r < 1$, then $|g\prime_{r\eta}| < |g'_\eta|$, we decrease the important level of feature points. Otherwise, if $r > 1$, then $|g\prime_{r\eta}| > |g\prime_\eta|$, we increase the important level of feature points. In another words, by changing the value of boosting rate in the linear boosting function, we can change the important level of feature points of the prior distribution.

To illustrate this property, we consider an example with Dirichlet distribution on the 2-simplex x_1, x_2, x_3:

$$p(x_1,x_2,x_3)$$

$$=\frac{1}{x_1x_2x_3}\exp(\sum_{i=1}^{3}\eta_i\ln(x_i) - \sum_{i=1}^{3}\ln\Gamma(\eta_i) - \ln\Gamma(\sum_{i=1}^{3}\eta_i))$$

In this example, the natural parameter $\eta = [4,5,6]$, and the boosting rates are tested with: 0.5, 1 and 1.5 respectively. The results are shown in Fig. 6. Intuitively, the figure shows that the bigger boosting rate, the more concentrated level of the density.

4.2.1. Relation to power prior

In BPS, we observe that $\exp(r\eta \cdot T(\Phi)) = [\exp(\eta \cdot T(\Phi))]^r$. This is equivalent to the use of the variational distribution of the following form

$$q(\Phi|\hat{\eta}) \propto q(\Phi|\eta)^r \tag{36}$$

In fact, linear boosting means that we use a power scale of the initial prior. This concept is related with the power prior method in [32,33,3], but *BPS* has some significant differences.

In power prior, presume the model's parameter Φ has the initial uninformative prior $p(\Phi|\eta)$, a past data C_0 and a power factor ρ. The power prior is defined by:

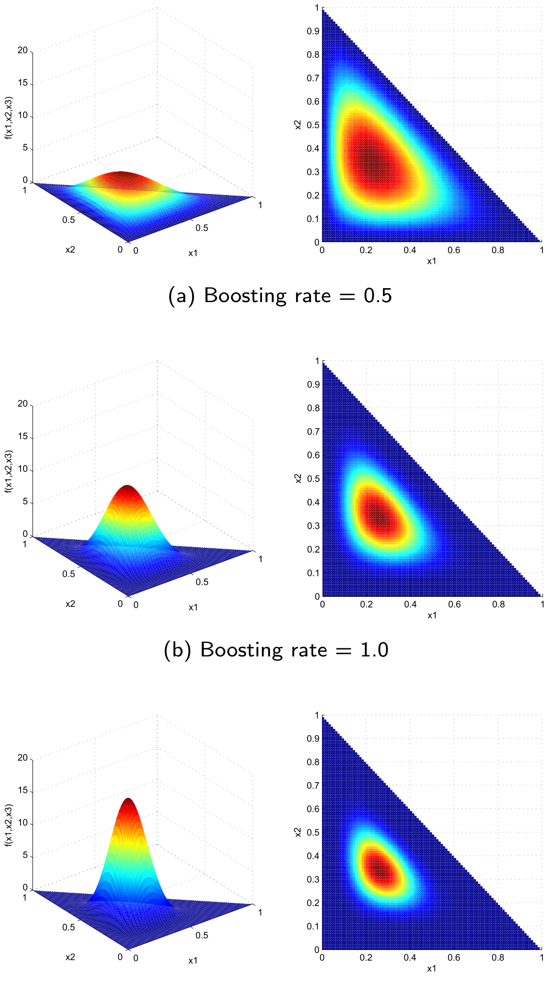
$$p(\Phi|C_0,\eta) \propto Likelihood(\Phi|C_0)^\rho.Prior(\Phi|\eta) \tag{37}$$

$$= p(C_0|\Phi)^\rho.p(\Phi|\eta). \tag{38}$$

Receiving new data C leads to the posterior:

$$\begin{aligned}p(\Phi|C,C_0,\eta) &\propto p(C|\Phi).p(\Phi|C_0,\eta)\\ &= p(C|\Phi).p(C_0|\Phi)^\rho.p(\Phi|\eta).\end{aligned} \tag{39}$$

The power prior depends on the past data and the power element is the likelihood of the past data C_0, while *BPS* straightforwardly powers the initial prior. On the other hand, the power prior method modifies the prior for the new data and replaces completely by the new prior specified in Eq. 38. By this way, the power prior provides a way to balance the old knowledge (learned from past data) and the new knowledge (just learned from C of the current step). The balancing constant ρ needs being chosen manually. Masegosa et al. [3] make a further progress by considering the balancing constant ρ to be a random variable which follows a prior distribution, allowing ρ to be adaptive with the data stream. In



(a) Boosting rate = 0.5

(b) Boosting rate = 1.0

(c) Boosting rate = 1.5

Fig. 6. Dirichlet distributions on the 2-simplex with different boosting rates of natural parameter $\eta = [4, 5, 6]$.

contrast, the viewpoint of *BPS* is to inject human/external knowledge into a model at each time step.

4.3. Keeping prior in BPS

We will show that in BPS with linear boosting functions, information from prior is not vanished. In BPS, the information learned from the data is: $\tilde{\xi}_1 + \ldots + \tilde{\xi}_b$ and the total information from prior is $\xi_0 + \tilde{\eta}_1 + \ldots + \tilde{\eta}_b$. By evaluating ratio of the norms of these two kinds of information, we have the following lemma:

Lemma 4.1 (*BPS*). *Assume that the learned information $\{\tilde{\xi}_h\}_{h=1}^{b}$ are i.i.d samples from a probability distribution with mean $\bar{\xi} \neq 0$, and BPS uses the linear boosting function $\tilde{\eta}_b = r_b \xi_0$, where coefficients r_b are lower bounded by some constant $c > 0$. We have the following property, with probability 1,*

$$\frac{||\tilde{\xi}_1 + \ldots + \tilde{\xi}_b||}{||\xi_0 + \tilde{\eta}_1 + \ldots + \tilde{\eta}_b||} \leqslant \frac{||\bar{\xi}||}{c||\xi_0||} \text{ as } b \to +\infty, \tag{40}$$

suggesting that as b increases, the prior information $\xi_0 + \tilde{\eta}_1 + \ldots + \tilde{\eta}_b$ is not vanished in comparison with the learned information $\tilde{\xi}_1 + \ldots + \tilde{\xi}_b$ from data.

Proof. Using the law of large numbers, we have:

$$\Pr\left(\lim_{b\to+\infty} \frac{\tilde{\xi}_1 + \ldots + \tilde{\xi}_b}{b} = \bar{\xi}\right) = 1,$$

and because $||\xi_0||$ is a constant:

$$\Pr\left(\lim_{b\to+\infty} \frac{||\tilde{\xi}_1 + \ldots + \tilde{\xi}_b||}{bc||\xi_0||} = \frac{||\bar{\xi}||}{c||\xi_0||}\right) = 1.$$

Note that $\tilde{\eta}_b = r_b \xi_0$ and $r_b \geqslant c$ for any b, we have:

$$\Pr\left(\lim_{b\to+\infty} \frac{||\tilde{\xi}_1 + \ldots + \tilde{\xi}_b||}{||\xi_0 + \tilde{\eta}_1 + \ldots + \tilde{\eta}_b||} \leqslant \frac{||\bar{\xi}||}{c||\xi_0||}\right) = 1.$$

This lemma shows that, in *BPS*, the information from external prior is not overwhelmed by the information learned from data. Hence, we can remain the valuable knowledge.

4.4. Balancing between learned information and boosting prior

BPS aims to inject the information from the initial prior into the new data. In Eq. 34, the learned information from the new data $\tilde{\xi}_b$ is independent of ξ_0 and $\hat{\xi}_0^b$. Because $\tilde{\xi}_b$ depends on the amount of data, overusing the boosting prior may leads to underfitting because it dominates the learned information. To tackle this problem, we propose an information scale ratio between $||\tilde{\xi}_b||$ and $||\hat{\xi}_0^b||$:

$$||\hat{\xi}_0^b|| = s||\tilde{\xi}_b|| \tag{41}$$

where s is a scale parameter. The purpose of this balancing is to let the information from the prior knowledge account for a certain ratio in the learned information of the new data.

Suppose we use the linear boosting function in Eq. 35. Note that $\xi_0 \leftarrow \eta$ and $\hat{\xi}_0^b \leftarrow \hat{\eta}^b$, combining with Eq. 41 leads to the boosting rate:

$$r_b = s\frac{||\tilde{\xi}_b||}{||\xi_0||} \tag{42}$$

and the boosting prior:

$$\hat{\xi}_0^b = s\frac{||\tilde{\xi}_b||}{||\xi_0||}.\xi_0 \tag{43}$$

In practice, at first we compute the learned information $\tilde{\xi}_b$, then we follow Eq. 43 to get the boosting prior. Finally, we update the parameter using Eq. 34. In *BPS*, s is a hyperparameter, we can choose a good value for s by grid search in the range $[0, 1]$.

Note that with scale factor $s = 0$, *BPS* turns out to be *SVB* without using boosting prior.

4.5. Discussion about BPS

BPS is a generalization of *SVB* with the ability to appropriately use prior knowledge. While a self-contained model to cover the data with high performance is not available and we often spend a high cost on deriving a good model, *BPS* is a considerable solution. In cases of having a good prior knowledge, *BPS* is a straightforward way with low cost to enhance a Bayesian model and reduce some bad effects from extreme sparsity and noise. One can also interpret that BPS uses external knowledge to regularize a model when dealing with ill-posed problems, e.g., sparsity and noise. We observe that external/human knowledge is available in various forms and easily accessible, including pre-trained machine learning models [22,37,38], ontology, Wikipedia, Zipf's law, Wordnet, etc. Furthermore, we believe that the idea of BPS can be easily employed in other streaming frameworks [2,19,3,5]. Those observations suggest the solution from BPS is significant.

Nonetheless, *BPS* requires a good prior knowledge to improve a model, otherwise, it may worsen the model. This consequence is explicitly shown in a case study of Section 5. In some situations, obtaining prior knowledge with good quality may be hard, and therefore *BPS* has some limitations.

5. Case studies

In this section, at first we conduct a case study with synthetic streaming data and then we build 3 other case studies with 3 models and different type of prior knowledge to evaluate the performance of *BPS* framework in comparison with *SVB*. The streaming data is simulated by dividing a dataset into continuous collections. For selecting models' hyperparameters, we do a grid search around their recommended settings in its original papers. For *BPS*, we first do a grid search to find a good range of scale factor *s* then evaluate how *BPS* responses to the change of *s*. We also do experiments on the sensitivity of *BPS* over different values of the batchsize parameter. Each experiment is run five times with random initialization of variables. The final result is the mean value of all runs. For simplicity, we only plot standard deviation of experiments in several cases. In all cases, higher values are better.

5.1. Case study 1: streaming with synthetic data

In this case study, we examine the performance of *BPS* in comparison with *SVB* in a synthetic streaming data, and investigate the impacts of the quality of prior knowledge and noise.

Consider a model where the binary data are all generated from a Bernoulli distribution: $x \sim Bernoulli(\theta)$. We fix $\theta = 0.2$ and generate a sequence of 300 data points.

Suppose that we use the following misspecified model to work with those data: $\theta = Beta(\eta^c, \eta^d); x \sim Bernoulli(\theta)$. Note $E[\theta] = \frac{\eta^c}{\eta^c+\eta^d}$, where η^c and η^d are hyperparameters.

This model is denoted as *BB*. The learning involves the following posterior: $p(\theta|\eta^c, \eta^d, C)$. Denote n_b and k_b be the number of data points and the number of 1's in the data C_b of minibatch *b*, respectively. Using variational inference with the variational distribution $q(\theta) \sim Beta(\lambda^c, \lambda^d)$, a lower bound of the log likelihood of the data is:

$$L = E_q \log p(\theta, C|\eta^c, \eta^d) - E_q \log q$$

Similar to the learning method of *LDA*, by taking the partial derivative of *L* with respect to λ^c and λ^d and setting them to zero, we have Algorithm 3 and Algorithm 4 for learning *BB* by *SVB* and *BPS* respectively.

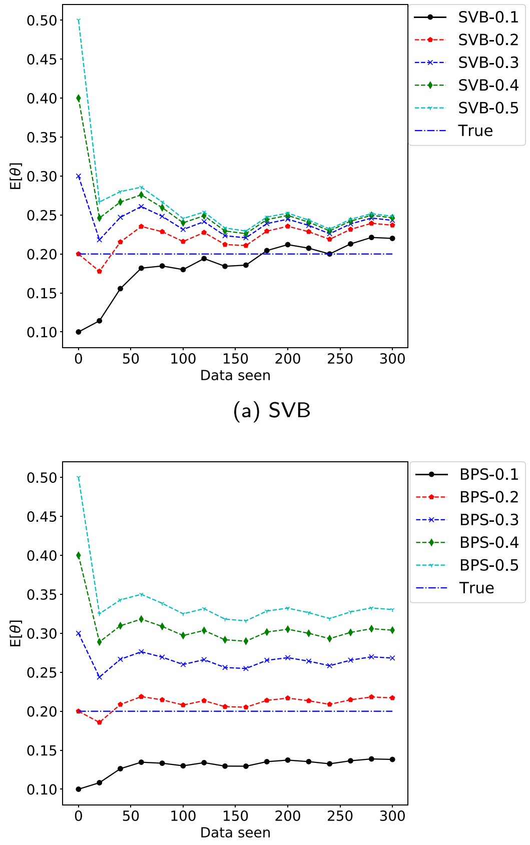
5.1.1. Impact of prior knowledge quality

We evaluate the impact of prior knowledge quality into the performance of *BPS* by setting different levels of prior knowledge. The knowledge to be used in *BPS* is encoded in (η^c, η^d). Note that $\frac{\eta^c}{\eta^c+\eta^d}$ being closer to $\theta = 0.2$ means that the prior quality is better.

Settings: The batchsize is fixed to 20, and the scale factor of *BPS* is $s = 0.3$.

Evaluation: We compare the expectation of θ in the learned model with its true value. The closer θ is to the true value, the better the model is.

Prior in use: We investigate different priors (η^c, η^d) so that $\hat{\theta} = \frac{\eta^c}{\eta^c+\eta^d} \in \{0.5, 0.4, 0.3, 0.2, 0.1\}$. The closer $\hat{\theta}$ to the truth $\theta = 0.2$, the better the quality of prior knowledge.



(a) SVB

(b) BPS

Fig. 7. Impact of prior knowledge quality. 'True' is the correct value (0.2) of model parameter θ in the BB model that generates the training data. Some guesses {0.1, 0.2, 0.3, 0.4, 0.5} about the true θ are used as priors for SVB and BPS. $E[\theta]$ shows how well a method recovers θ as learning from more data.

Result: The result is illustrated in Fig. 7a for *SVB* and in Fig. 7b for *BPS*. It shows that *SVB* has less impact of prior knowledge while *BPS* has a strong dependence with the quality of prior knowledge. A good prior knowledge will significantly enhance the quality of model. In contrast, a poor quality of prior knowledge misleads the model, and even makes it poorer than *SVB*. Therefore, the quality of prior knowledge has a real impact to *BPS*, and *BPS* only keeps its good performance in case of having a good prior.

5.1.2. Learning from noisy data

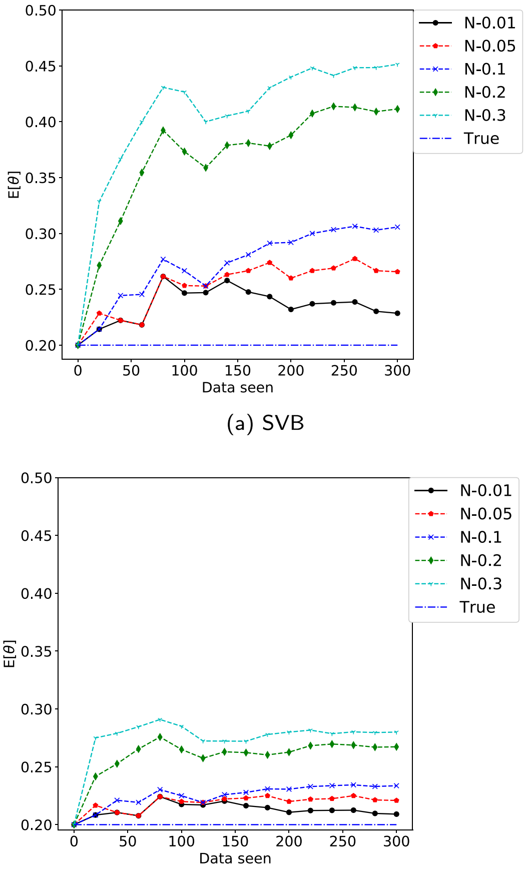
Suppose that some random noises change the results of Bernoulli trials. In more details, the noisy generative process for each data point is:

- Sample data point $x \sim Bernoulli(\theta)$
- Change the value of *x* with a noise $Bernoulli(\delta)$.[2]

With the same settings as sub-Section 5.1.1, but only using the good prior with the expectation $E[\theta] = 0.2$ and the noisy probability $\delta \in$ {0.01, 0.05, 0.1, 0.2, 0,3}, we have the results in Fig. 8a for *SVB* and in Fig. 8b for *BPS*.

It is clear that in noisy data, *BPS* with a good prior knowledge will drive the model into its true value. Meanwhile, *SVB* is impacted by wrong information from noise leading to a weak performance. In other words, *BPS* has ability to help the model cope with noisy information when providing a good prior knowledge.

[2] Note that the contaminated *x* turns out to be a trial from $Bernoulli(\theta + \delta - 2\theta\delta)$.



(a) SVB

(b) BPS

Fig. 8. Impact of noise. ‘True’ is the correct value of model parameter θ in the BB model that generates the training data. {0.01, 0.05, 0,1, 0.2, 0.3} show the level of noise that contaminates the training data. $E[\theta]$ shows how well a method recovers θ as learning from more noisy data.

```
Algorithm 3 SVB-BB
Input: Prior $\eta^c, \eta^d$, sequence of mini-batch $C_1, C_2, \ldots$
Output: a sequence $\lambda_b^c, \lambda_b^d$
  Initialize: $\lambda_0^c \leftarrow \eta^c$, $\lambda_0^d \leftarrow \eta^d$
  for each mini-batch $C_b$ in $C_1, C_2, \ldots$ do
    $\tilde{\lambda}_b^c \leftarrow k_b$
    $\tilde{\lambda}_b^c \leftarrow n_b - k_b$
    $\lambda_b^c \leftarrow \lambda_{b-1}^c + \tilde{\lambda}_b^d$
    $\lambda_b^d \leftarrow \lambda_{b-1}^d + \tilde{\lambda}_b^d$
  end for
```

```
Algorithm 4 BPS-BB
Input: Prior $\eta^c, \eta^d$, sequence of mini-batch $C_1, C_2, \ldots$
Output: a sequence $\lambda_b^c, \lambda_b^d$
  Initialize: $\lambda_0^c \leftarrow \eta^c$, $\lambda_0^d \leftarrow \eta^d$
  for each mini-batch $C_b$ in $C_1, C_2, \ldots$ do
    $\tilde{\lambda}_b^c \leftarrow f_b(\eta^c) + k_b$
    $\tilde{\lambda}_b^c \leftarrow f_b(\eta_d) + n_b - k_b$
    $\lambda_b^c \leftarrow \lambda_{b-1}^c + \tilde{\lambda}_b^d$
    $\lambda_b^d \leftarrow \lambda_{b-1}^d + \tilde{\lambda}_b^d$
  end for
```

5.2. Case study 2: streaming sentiment analysis

Sentiment analysis is an interesting area in text mining, with the aim to extract opinions from text data. One of the basic tasks in sentiment analysis is sentiment classification, in which opinions often need to be classified into positive, negative or neutral. However, a document can contains many aspects with different sentiments. Yo and Oh [17] proposed the *Aspect and Sentiment Unification model (ASUM)* for modeling the sentiments about aspects in an unsupervised manner. The graphical model of *ASUM* is presented in Fig. 9.

ASUM assumes that a corpus is talking about E sentiments and T aspects, represented by many Dirichlet distributions $\beta_{ez} \sim Dir(\eta_e)$, where $e \in \{1, \ldots, E\}$ is the index of a sentiment and $z \in \{1, \ldots, T\}$ is the index of an aspect. In each document, containing M sentences, the distribution of sentiments is $\pi_d \sim Dir(\gamma)$; and for each sentiment e, the aspect distribution is defined by $\theta_{de} \sim Dir(\alpha)$. We assume the mth sentence of length N is generated as follow: first choose the sentiment by a multinomial distribution $e_m \sim Mul(\pi_d)$, then generate the corresponding aspect $z_m \sim Mul(\theta_{de_m})$, generate each word $w \sim Mul(\beta_{e_m, z_m})$.

Learning algorithm: Given data C, the full posterior of interest is:

$$p(\beta, \pi, \theta, e, z | C, \gamma, \alpha, \eta) \tag{44}$$

The inference procedure infers the local variables π, θ, e, z. We use the method proposed by [39]. The Dirichlet parameters $\gamma = 1$ and $\alpha = 1$ are fixed so that the posterior in Eq. 44 is a convex function with respect to π and θ and can be efficiently inferred by Frank-Wolfe algorithm [40]. e and z can be directly approximated through $p(e|\pi, \theta, w, \beta$ and $p(z|\pi, \theta, w, \beta)$. The details of this procedure is presented in Algorithm 12 of the Appendix.

To update the global variable β, let $q(\beta|\lambda)$ be the variational distribution of $p(\beta|\eta)$. Using variational inference, we derive the two streaming learning algorithms for ASUM, *SVB* in Algorithm 5 and *BPS* in Algorithm 6.

Prior knowledge in use: We use the same way as in [17,39]. We use a list of positive (e.g., “good”, “excellent”) and negative words (e.g., “bad”, “poor”) as prior knowledge, often called *seed words.*[3] β_e contains information about the distribution of words over each sentiment. So for the seed words, the priors are set with higher value than other words. By this way, we form a human prior of sentiment over words: η_e. Therefore, the prior encoded in *BPS* is

$$\hat{f}_b(\eta) = s\frac{||\tilde{\lambda}_b||}{||\eta||}\eta.$$

We set $\eta_{ej} = 0.01$ for any sentiment seed word j and otherwise $\eta_{ei} = 0.0001$ for any other word i. η_e is used for both *SVB* at the first learning step and *BPS*.

Evaluation metric: The accuracy of sentiment classification is used to evaluate the quality.

Experimental setups: We use 4 datasets (Electronics, Yelp, Kitchen & Housewares, and Music)[4] with some statistics shown in Table 3.

To investigate the effect of scale ratio s, we fix batch-size to 1000 for the three first datasets, and to 2000 for Music as it’s size is large. We test $s \in \{0.2, 0.4, 0.6, 1.0\}$. To study the effect of batch-size, we fix the scale ratio to $s = 0.4$ and change batch-size in $\{500, 1500, 2000\}$ for the three small corpora, and in $\{4000, 6000, 8000\}$ for Music. We have $E = 2$ to represent

[3] The full list can be found in [39].

[4] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

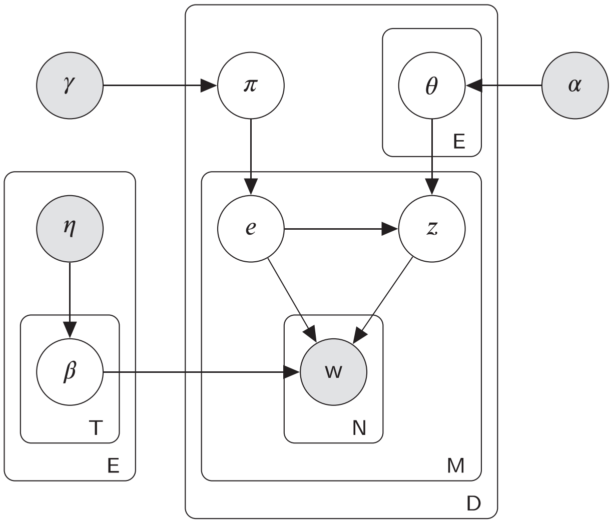


Fig. 9. The graphical representation of *ASUM*.

positive/negative sentiments, and fix $T = 50$ in all experiments for *ASUM*.[5]

```
Algorithm 5 SVB-ASUM
Input: Prior $\eta, \alpha, \gamma$, sequence of mini-batch $C_1, C_2, \ldots$
Output: a sequence $\lambda_1^{(l)}, \lambda_2^{(l)}, \ldots$
  Initialize: $\lambda_0 \leftarrow \eta$
  for each mini-batch $C$ in $C_1, C_2, \ldots$ do
    Inference:
    for each document $d$ in $C$ do
      Infer local variables $(\theta, \pi, e, z)$
    end for
    Update:
    $\tilde{\lambda}_{betj} = \sum_{d \in C_b} \sum_m \sum_n \tilde{e}_{dme} \tilde{z}_{dmt} d_j$
    $\lambda_b \leftarrow \lambda_{b-1} + \tilde{\lambda}_b$
    $\beta_b \propto \lambda_b$
  end for
```

```
Algorithm 6 BPS-ASUM
Input: Prior $\eta, \alpha, \gamma$, sequence of mini-batch $C_1, C_2, \ldots$
Output: a sequence $\lambda_1^{(l)}, \lambda_2^{(l)}, \ldots$
  Initialize: $\lambda_0 \leftarrow \eta$
  for each mini-batch $C$ in $C_1, C_2, \ldots$ do
    Inference:
    for each document $d$ in $C$ do
      Infer local variables $(\theta, \pi, e, z)$
    end for
    Update:
    $\tilde{\lambda}_{betj} = \sum_{d \in C_b} \sum_m \sum_n \tilde{e}_{dme} \tilde{z}_{dmt} d_j$
    $\lambda_b \leftarrow \lambda_{b-1} + \tilde{\lambda}_b + \hat{f}_b(\eta)$
    $\beta_b \propto \lambda_b$
  end for
```

[5] We found the same conclusion as investigating with different values of *T*. Hence for ease of discussion, we just report the results with $T = 50$.

Results:

The effect of prior: Fig. 10 shows the results of *BPS* with different scale ratios, in comparison with *SVB*. Higher value of the scale ratio means we are using more prior knowledge.

It is clear that the results of *BPS* are better than those of *SVB* even with different values for scale ratio. In addition, there are rises in the accuracy of *BPS* when the scale ratio increases. It means that the prior is meaningful, so that emphasizing it will enhance the quality of the learning process. Moreover, Fig. 10b and Fig. 10d give illustrations that ignoring the external prior in *SVB* may lead to a considerable decrease in accuracy, while by maintaining the prior, *BPS* improves the accuracy of the model. Those subfigures also suggest that by boosting the good prior, *BPS* can perform better by a large magin than *SVB*.

Fig. 11 summarizes the results when changing the batch-size. We observe that *BPS* works well with different settings of batch-size. In all 4 datasets, the accuracy of *BPS* with different batch-sizes is higher than that of *SVB*. The performance of *BPS* in Yelp and Music is particularly good, and significantly better than that of SVB. The improvement with a large margin in Music suggests that the prior is really meaningful for sentiment analysis. Losing such a good prior knowledge causes to quickly reduce the discriminative ability of SVB.

```
Algorithm 7 SVB-LDA
Input: Prior $\eta$, hyper-parameter $\alpha$, sequence of minibatches
  $C_1, C_2, \ldots$
Output: $\lambda$
  Initialize: $\lambda_0 \leftarrow \eta$
  for each minibatch $C$ in $C_1, C_2, \ldots$ do
    Inference:
    for each document $d$ in $C$ do
      Infer local variables $\phi, \gamma$
    end for
    Update:
    $\tilde{\lambda}_b \leftarrow \sum_{d \in C} \phi_{dvk} n_{dv}$
    $\lambda_b \leftarrow \lambda_{b-1} + \tilde{\lambda}_b$
    $\beta_b \propto \lambda_b$
  end for
```

```
Algorithm 8 BPS-LDA
Input: Prior $\eta$, hyper-parameter $\alpha$, sequence of minibatches
  $C_1, C_2, \ldots$
Output: $\lambda$
  Initialize: $\lambda_0 \leftarrow \eta$
  for each minibatch $C$ in $C_1, C_2, \ldots$ do
    Inference:
    for each document $d$ in $C$ do
      Infer local variables $\phi, \gamma$
    end for
    Update:
    $\tilde{\lambda}_b \leftarrow \sum_{d \in C} \phi_{dvk} n_{dv}$
    $\lambda_b \leftarrow \lambda_{b-1} + \tilde{\lambda}_b + \hat{f}_b(\eta)$
    $\beta_b \propto \lambda_b$
  end for
```

Table 3
Datasets for evaluating *ASUM*. l_d denotes the average number of words per review. l_s denotes the average number of words per sentence.

Dataset	Reviews	Positive reviews	Negative reviews	l_d	l_s
Electronics	23,009	17,961	5,048	38.6	6.1
Yelp restaurant	20,708	17,457	3,251	59.7	5.7
Kitchen & Housewares	19,856	15,737	4,119	31.9	5.8
Music	136,000	110,160	25,840	57.2	7.1

5.3. Case study 3: streaming learning for LDA

In this case study, we evaluate *BPS*'s performance with the *LDA* model presented in Section 3.2.

Streaming learning algorithm: We present more detailed versions of the learning algorithms for *LDA* than in Section 3.2. The posterior of the latent variables given a corpus C is: $p(\beta, \theta, z|C, \alpha, \eta)$.

We inherit the learning algorithm for *LDA* in Section 2.3 with the inference procedure for the local variables as in Algorithm 1. After applying *BPS* and *SVB* framework, we have two streaming learning algorithms for *LDA* in Algorithm 7 and 8. Note that Algorithm 7 is the same as the *SSU* algorithm in [1]. The only difference between the two algorithms is $\hat{f}_b(\eta)$.

Prior knowledge in use: Relating to the distribution of word in natural language, Zipf's law [27,28] gives us an interesting property that the frequencies of words in a specific language followed a power-law distribution of the form: $p(w) \propto r_w^{-l}$, in which $p(w)$ is the proportion of word w in the language. r_w is the rank of the word in the descending sorted frequencies. This means that the most frequency word has rank $r = 1$. Parameter l depends on the specific language. We use Zipf's law as the prior knowledge in *BPS* as follows:

$$\eta_w \propto r_w^{-l}. \tag{45}$$

Evaluation metric: We use log predictive probability [1] to evaluate the predictive capacity of the learned models. A testing corpus C_h is taken out from data. Each document of this test corpus is split into 2 parts: d_{test} and $d_{observed}$ (with a ratio of $1:4$ in this experiment). Suppose that the global parameters are given, the log predictive probability (*LPP*) is defined by:

$$LPP = \frac{\sum_{d\in C_h} \log p(d_{test}|C, d_{observed})}{\sum_{d\in C_h} |d_{test}|} \approx \frac{\sum_{d\in C_h} \sum_{w\in d_{test}} log \sum_{k=1}^{K} E_q[\theta_{dk}] E_q[\beta_{kw}]}{\sum_{d\in C_h} |d_{test}|}$$

Experimental setups: We use 3 datasets for evaluation: New York Times from the UCI Machine learning Repository[6], Yahoo and Twitter from [10]. Some statistics are presented in Table 4. Note that Yahoo and Twitter are two corpora of extremely short text, while New York Times is long text. The number of topics is set equally with $K = 100$, and the hyperparameter $\alpha = 0.01$. The prior η is taken from Eq. 45 with a heuristic parameter $l = 1.07$ [41] and the ranking of word's frequencies is downloaded from top 100,000 most frequently-used English words text,[7] only the words appeared in vocabulary of dataset are used.

For testing the effect of the scale ratio, we fix the batch-size = 5000 for large corpora and 1000 for New York Times, and change $s \in \{0.02, 0.05, 0.08, 0.1, 0.2\}$. With batch-size testing, we fix $s = 0.02$ and change batch-size in $\{500, 2000, 4000\}$ for New York Times. With Yahoo and Tweet we fix the scale ratio to $s = 0.08$ and change the batch-size in $\{5000, 10000, 20000\}$. We choose greater s and batch-size for short text corpora, because in those cases more information/knowledge should be needed.

The effect of prior: The results in this case study have a slightly different pattern with that in case study 2, and can be divided into two groups: long text (New York Times), and short text (Yahoo and Twitter). In general, a higher weight of the prior leads to worse performance of *BPS* in long text, however gives significant improvement for short text (Fig. 12).

In the experiments for testing boosting scale with the long text group, the smallest value of scale ratio for *BPS* in the experiments $s = 0.02$ gives the highest quality of the model and better than in *SVB*. However, when increasing the boosting scale, the predictive capacity of *LDA* using *BPS* may be lower than using *SVB*. In contrast, the performance of *BPS* for the short text increases as increasing the boosting scale and gets the highest quality with $s = 0.2$. The better performance of *BPS* over *SVB* is very clear and often with a large margin. The same behavior of *BPS* is reported in Fig. 13 when changing the batch-size and fixing the scale ratio with a suitable value (0.02 for short text, and 0.08 for long text).

We can explain such a behavior of *BPS* as follows. The short text often contains little information, and poses severe challenges for *SVB*. Further even a large number of short text could not overcome those challenges [11]. Therefore, exploiting external knowledge would be necessary in order to learn a good predictive model from short text. *BPS* seems to exploit considerably well the knowledge from Zipf's law, represented in (45), to surpass *SVB* with a large margin in Yahoo and Twitter. On the other hand, the long text itself contains much information and hence helps *SVB* perform well. Overusing prior knowledge with a unsuitable scale factor will lead to overwhelm the information from data. As a result, the role of prior is not very significant in this case.

5.4. Case study 4: streaming text classification

To solve text classification problem, *multi-view topic model* (*MviewLDA*) [23] was introduced. The idea of *MviewLDA* is that each document of D documents belongs to one of J classes with probability contributed by a multinominal distribution $\chi \sim Mul(\pi)$. Each class j contains K local topics $\{\beta_{jk}^{(l)}\}_{k=1}^{K}$ with the distribution over topics $\theta^{(l)}$ which are sampled from the Dirichlet distribution $Dir(\alpha_j^{(l)})$. Besides, they assume that there exists R global topics $\{\beta_r^{(g)}\}_{r=1}^{R}$ that are shared by all classes. The binary variable δ decides when a word belongs to global or local topics. The graphical representation of *MviewLDA* is presented in Fig. 14.

The generative process of *MviewLDA* is as follows:

- Generate topic distribution:
  - For local topics: $\beta^{(l)} \sim Dirichlet(\eta^{(l)})$
  - For global topics: $\beta^{(g)} \sim Dirichlet(\eta^{(g)})$
- Generate a document d of length N_d:
  - Draw a class: $\chi \sim Mul(\pi)$
  - Draw local topic proportion: $\theta_\chi^{(l)} \sim Dir(\alpha_\chi^{(l)})$
  - Draw global topic proportion: $\theta^{(g)} \sim Dir(\alpha^{(g)})$
  - Draw Bernoulli parameter $\omega \sim Beta(\gamma)$

[6] http://archive.ics.uci.edu/ml/machine-learning-databases/bag-of-words/

[7] https://gist.github.com/h3xx/1976236.

- For each word w of document d:
  - $*$ Draw a binary indicator $\delta \sim Bernoulli(\omega)$
  - $*$ If $\delta = 1$, word w belongs to local topic:
    - Draw a local topic $z^{(l)}_{\chi} \sim Mul(\theta^{(l)}_{\chi})$
    - Draw word $w \sim Mul(\beta_{z^{(l)}_{\chi}})$
  - $*$ If $\delta = 0$, word w belongs to global topic:
    - Draw a global topic: $z^{(g)} \sim Mul(\theta^{(g)})$
    - Draw word $w \sim Mul(\beta_{z^{(g)}})$

Let each local topic being contributed by a Dirichlet prior $\eta^{(l)}_j$. Note that local topics within each class contain feature words of its class. Therefore, if we set the weighted value of the feature words larger than the others and use it as a prior, we will provide more information into the model to differentiate classes. Such a technique can increase the quality of learning process.

Algorithm 9 SVB-MviewLDA

```
Input: Prior $\eta$, sequence of minibatches $C_1, C_2, \dots$
Output: $\lambda$
  Initialize $\lambda_0 \leftarrow \eta$
  for each minibatch $C$ in $C_1, C_2, \dots$ do
    Inference:
    for each document $d$ in $C$ do
      Infer local variables $(\zeta, \phi, \tau)$
    end for
    Update:
    $\tilde{\lambda}^{(l)}_b \leftarrow \sum_{d \in C} \sum_{i=1}^{N_d} I_{[\chi_d = j]} \tau_{di} \phi^{(l)}_{d,i,j,k} w_{d,i,j}$
    $\lambda_b \leftarrow \lambda_{b-1} + \tilde{\lambda}_b$
    $\beta_b \propto \lambda_b$
  end for
```

Algorithm 10 BPS-MviewLDA

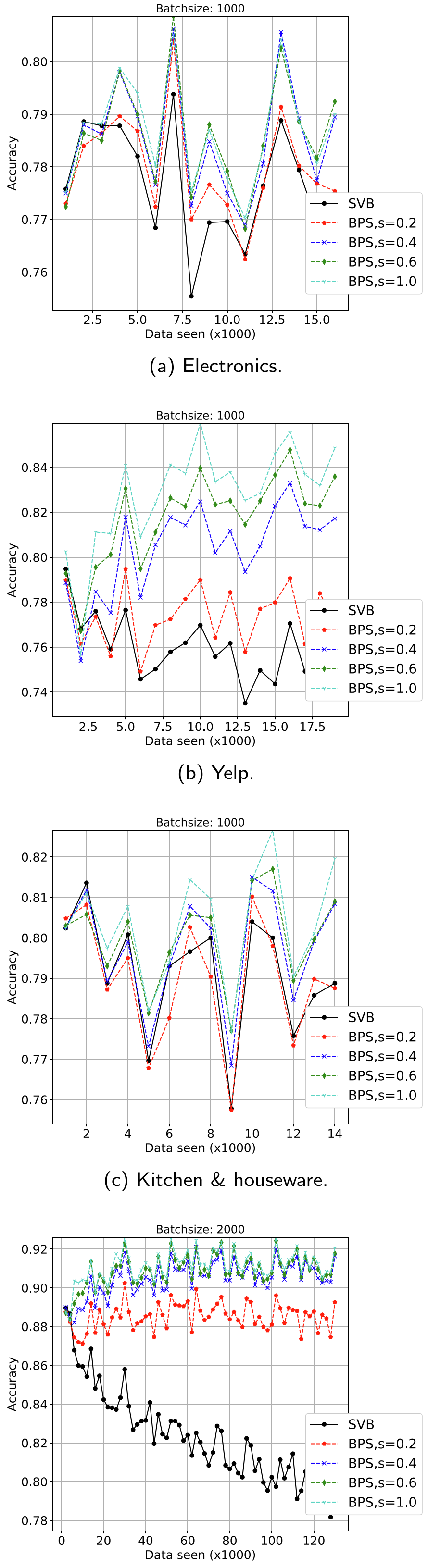
```
Input: Prior $\eta$, sequence of minibatches $C_1, C_2, \dots$
Output: $\lambda$
  Initialize $\lambda_0 \leftarrow \eta$
  for each minibatch $C$ in $C_1, C_2, \dots$ do
    Inference:
    for each document $d$ in $C$ do
      Infer local variables $(\zeta, \phi, \tau)$
    end for
    Update:
    $\tilde{\lambda}^{(l)}_b \leftarrow \sum_{d \in C} \sum_{i=1}^{N_d} I_{[\chi_d = j]} \tau_{di} \phi^{(l)}_{d,i,j,k} w_{d,i,j}$
    $\lambda_b \leftarrow \lambda_{b-1} + \tilde{\lambda}_b + f_b(\eta)$
    $\beta_b \propto \lambda_b$
  end for
```

Learning algorithm: Given corpus $C = \{C_i\}$, in case of supervised learning with known class label χ, the posterior of interest is:

$$p(\omega, \theta^{(l)}, \theta^{(g)}, \delta, z, \beta^{(l)}, \beta^{(g)} | \chi, C, \eta^{(l)}, \eta^{(g)}, \alpha^{(l)}, \alpha^{(g)}, \gamma, \pi). \qquad (46)$$

The learning algorithm in [23] is Gibbs sampling and designed for batch learning. In this case study, we use variational inference (as in [42]) and modify to streaming classification learning. The variational distribution is:

$$q(\omega, \theta^{(l)}, \theta^{(g)}, \delta, z^{(l)}, z^{(g)}, \beta^{(l)}, \beta^{(g)})$$
$$= q(\omega|v) q(\theta^{(l)}|\mu^{(l)}) q(\theta^{(g)}|\mu^{(g)}) q(\delta|\tau) q(z^{(l)}|\phi^{(l)})$$
$$q(z^{(g)}|\phi^{(g)}) q(\beta^{(l)}|\lambda^{(l)}) q(\beta^{(g)}|\lambda^{(g)})$$



(a) Electronics.

(b) Yelp.

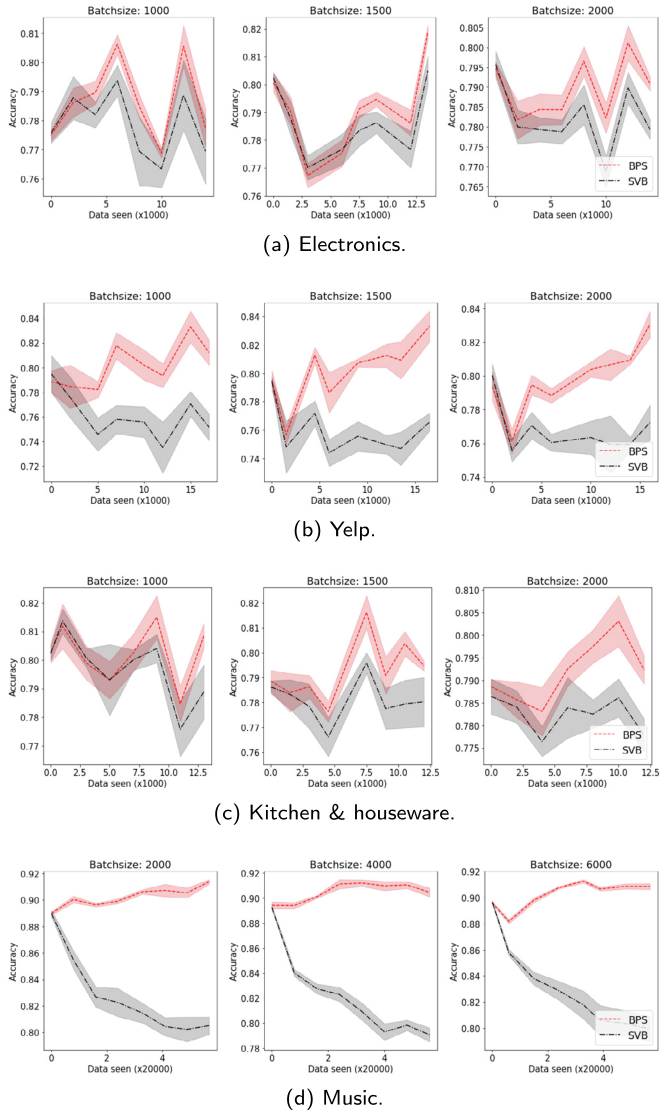
(c) Kitchen & houseware.

(d) Music.

Fig. 10. Streaming ASUM results with different scale ratios. The x-axis shows the amount of data received in the streaming learning process.

Table 4
Datasets for evaluation of LDA. l_d denotes the average number of words per document.

Dataset	Vocabulary size	Training size	Testing size	l_d
New York Times	102,660	200,000	10,000	228.8
Yahoo	24,419	500,000	10,000	4.7
Twitter	89,474	1,500,000	10,000	9.8



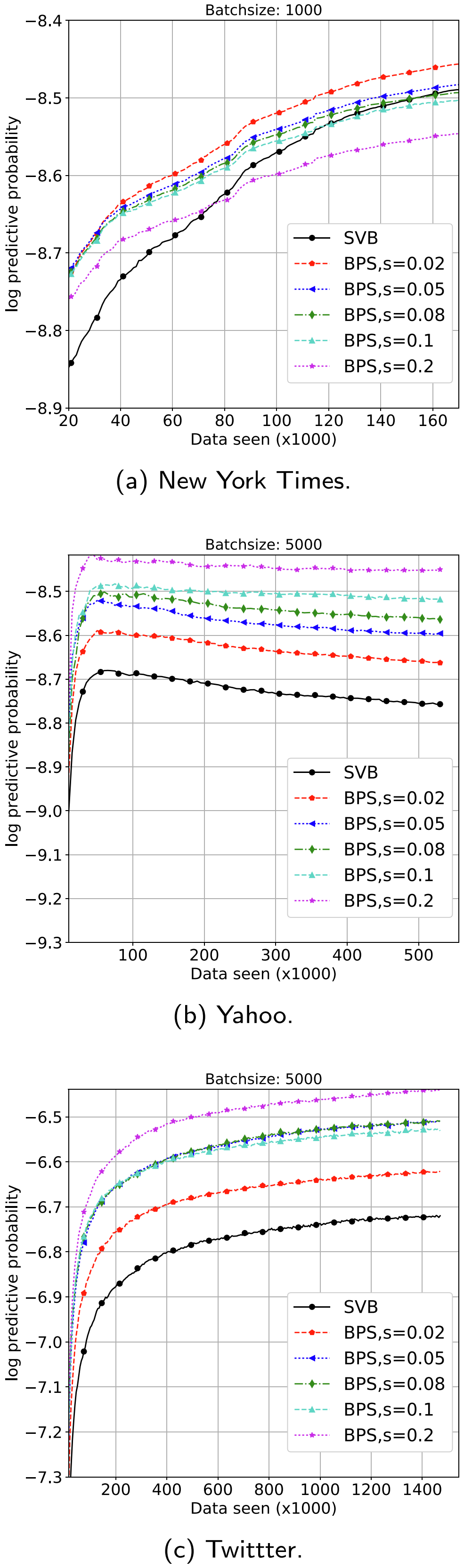
(a) Electronics.

(b) Yelp.

(c) Kitchen & houseware.

(d) Music.

Fig. 11. Streaming ASUM results with different batch-sizes.



(a) New York Times.

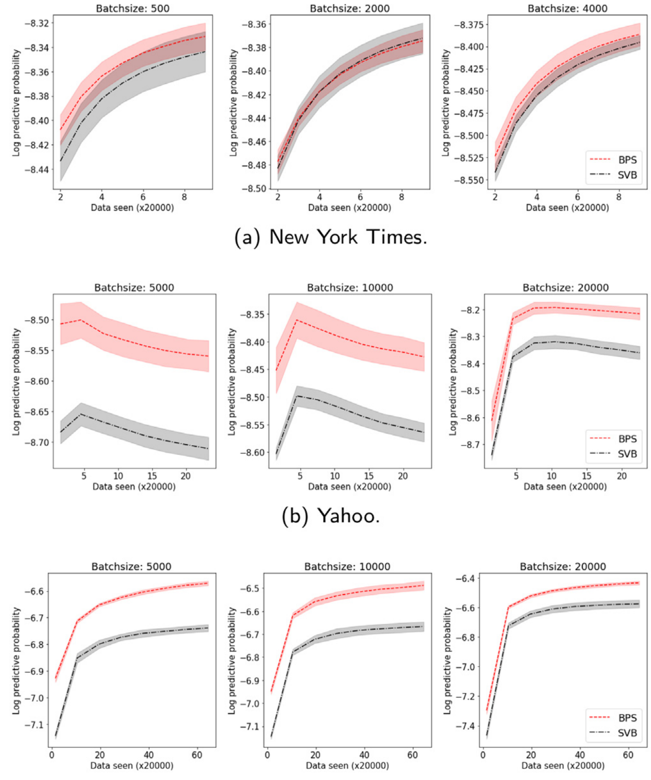
(b) Yahoo.

(c) Twittter.

Fig. 12. Streaming *LDA* results with different scale ratios.

where $\phi^{(l)}$ and $\phi^{(g)}$ are multinomial parameters, v is Beta parameter, τ is Bernouli parameter, $\phi^{(l)}, \phi^{(g)}, \lambda^{(l)}, \lambda^{(g)}$ are Dirichlet parameters. We denote the indicator $I^j_{\chi_d} = 1$ when document d belongs to class j and $I^j_{\chi_d} = 0$ otherwise. Similarly, $I^v_{w_{d,i}} = 1$ if the i^{th} word of document d is v. The detail of the inference procedure are shown in Algorithm 11 of the Appendix.

Because we aim to keep the information from prior $\eta^{(l)}$ of the local topics, we only apply *BPS* to $\beta^{(l)}$. Adopting *SVB* and *BPS* with the variational inference leads to 2 streaming versions for updating $\lambda^{(l)}$ in Algorithm 9 and Algorithm 10.



(a) New York Times.

(b) Yahoo.

(c) Twitter.

Fig. 13. Streaming LDA with different batch-sizes.

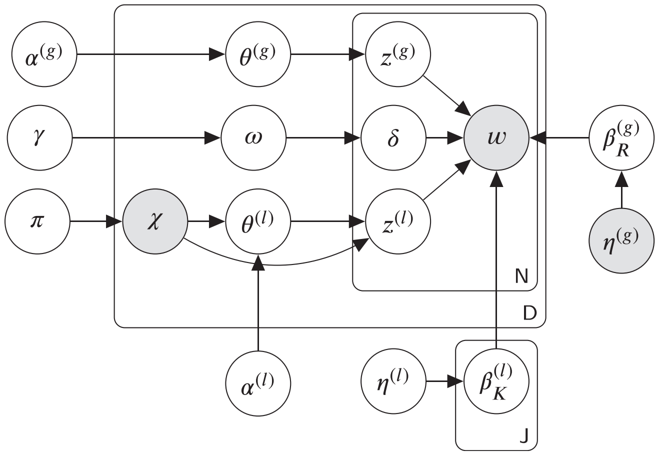


Fig. 14. Graphical representation for *MviewLDA*.

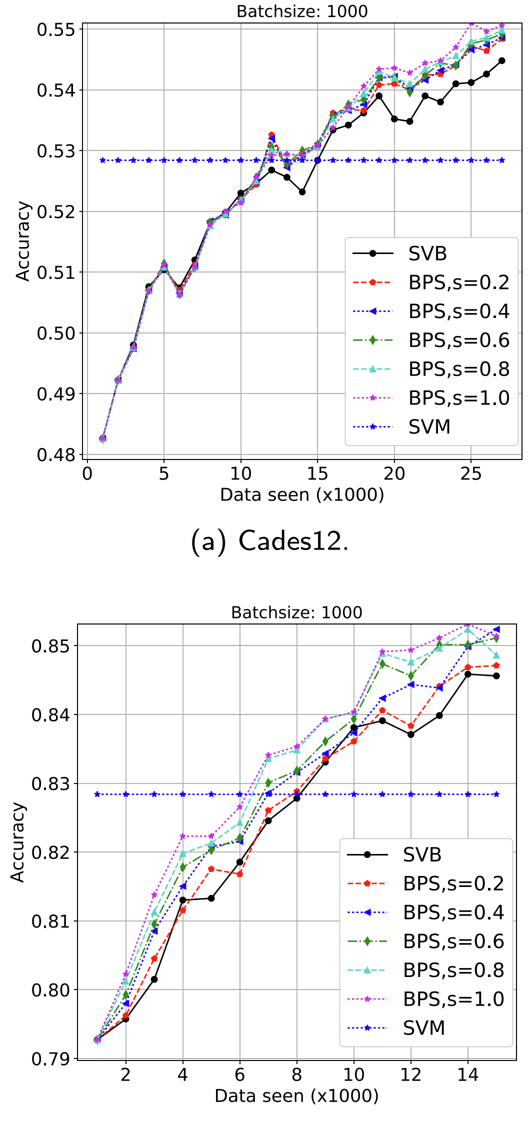
Prior knowledge in use: Borrowing the idea from [30], we extract the feature words of each class and use them as the prior knowledge. At first, we calculate TF.IDF for the words in each class then select top 5000 words with highest TF.IDF values as seed words. The seed words of class j are then used to initialize prior $\eta_j^{(l)}$ by assigning a value $v = 0.5$. The other values are set to a small value $\epsilon = 0.01$.

Evaluation metric: The classification accuracy is used in this evaluation.

Experimental setups: We use 2 labeled datasets (Cade 12 and News20).[8] with some statistics described in Table 5.

Table 5
Dataset for text classification.

Dataset	Num of classes	Vocabulary size	Training size	Testing size
News20	20	62,061	16,000	3,900
Cade12	12	193,997	27,322	4000



(a) Cades12.

(b) News20.

Fig. 15. Streaming MviewLDA result with different scale ratios.

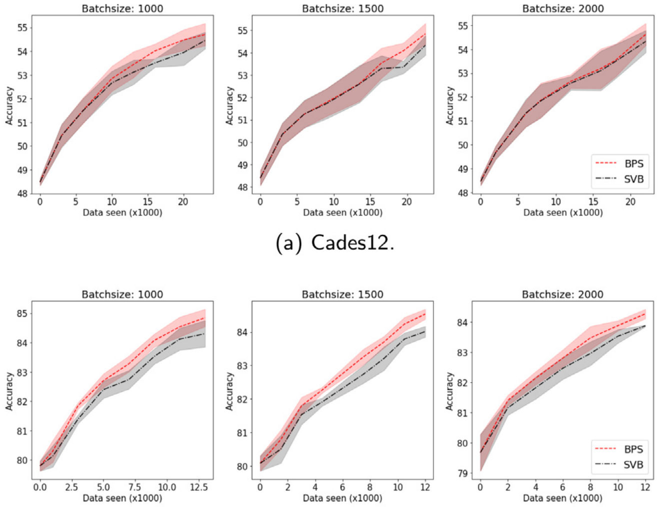
For News20, we set the number $K = 10$ of topics in each local class and the number $R = 8$ of global topics. For Cade 12, we use $K = 15$ and $R = 4$ because of being larger in size than News20. The other Dirichlet prior parameters are set equal to 0.01 as case study 3 on LDA. We take SVM into account as another baseline to evaluate the quality of classification.

For testing scale ratio, we set batchsize as 1000 and change scale factor $s \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. For testing batchsize, we fix scale factor $s = 0.4$ and change batchsize in $\{1000, 1500, 2000\}$.

Results: The results are summarized in Fig. 15 and Fig. 16. The results of *BPS* are comparable with those of *SVB*.

However, we can see that the accuracy of *BPS* tend to be higher than *SVB* in the latter minibatches, and significantly better in News20 dataset. Again, this pattern reflects that maintaining the

[8] http://ana.cachopo.org/datasets-for-single-label-text-categorization



(a) Cades12.

(b) News20.

Fig. 16. Streaming *MviewLDA* result with different batch-sizes.

prior knowledge is valuable in streaming learning. *SVB* will lose prior information while *BPS* still keep it, therefore, *BPS* is able to produce a better result than *SVB*.

From these experiments, we can see that the appropriate values of the scale ratio depend on the quality of the prior knowledge provided into models. Doing a pre-search to find a good range value of scale ratio will help to improve the result of the learned models. The result of *BPS* is better in comparison with *SVB* in most cases with the same batchsize settings.

6. Conclusion

In this paper, we discussed the problem of incorporating external knowledge as the prior for streaming Bayesian learning. We showed that *SVB* easily forgets prior knowledge as more data come. This may be problematic when we want to exploit valuable knowledge to deal with the challanges of sparsity and noise. We then proposed *BPS* to boost the role of the prior knowledge. Within *BPS*, the valuable information from prior is maintained through learning process, and hence *BPS* easily overcomes the vanishing prior problem. *BPS* provides a simple but effective solution for practice, and its idea can be easily employed in other streaming methods.

CRediT authorship contribution statement

Duc Anh Nguyen: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - original draft, Visualization, Investigation. **Van Linh Ngo:** Conceptualization, Methodology, Validation, Formal analysis, Writing - review & editing, Investigation. **Kim Anh Nguyen:** Conceptualization, Methodology, Validation, Formal analysis, Writing - review & editing, Supervision. **Canh Hao Nguyen:** Formal analysis, Writing - review & editing, Supervision. **Khoat Than:** Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

This section presents the details of the inference of local variables in ASUM (Algorithm 12) and MviewLDA (Algorithm 11).

Algorithm 11 Inference of local variables in ASUM

```
Input: Prior $\eta, \alpha = 1, \gamma = 1$, global variable $\beta$, document
  $d = \{w_1, \ldots, w_m\}$ with $M$ sentences
Output: Local variables: $\theta, \pi, \tilde{e}_d, \tilde{z}_d$
  Randomly initialize $\theta_d, \pi_d$.
  repeat
    Infer $\theta_d$ by using Frank-Wolfe algorithm [40] to maximize
$f(\theta_d) = \sum_{m=1}^{M} \log \sum_{e=1}^{E} \sum_{t=1}^{T} (\pi_{de}\theta_{det} \prod_{n=1}^{N} \beta_{etw_{mn}})$
    Infer $\pi_d$ by using Frank-Wolfe algorithm to maximize
$f(\pi_d) = \sum_{m=1}^{M} \log \sum_{e=1}^{E} \sum_{t=1}^{T} (\pi_{de}\theta_{det} \prod_{n=1}^{N} \beta_{etw_{mn}})$
  until convergence
  Estimate $\tilde{e}_d$, $\tilde{z}_d$ for each sentence $m$:
$\tilde{e}_{dme} = P(e_m = e|\pi_d, \theta_d, w_m, \phi) \propto \sum_{t=1}^{T} \pi_{de}\theta_{det} \prod_{n=1}^{N} \beta_{etw_{mn}}$
$\tilde{z}_{dmt} = P(z_m = t|\pi_d, \theta_d, w_m, \phi) \propto \sum_{e=1}^{E} \pi_{de}\theta_{det} \prod_{n=1}^{N} \beta_{etw_{mn}}$
```

Algorithm 12 Inference of local variables in MviewLDA

```
Input: Global variable $\eta^{(l)}$, $\eta^{(g)}$, document $d$ and label $\chi_d$
Output: $v, \mu, \tau, \phi$
  Randomly initialize $v, \mu, \tau, \phi$.
  repeat
$v_1 = \gamma_1 + \sum_{i=1}^{N} \tau_i, v_2 = \gamma_2 + \sum_{i=1}^{N} (1 - \tau_i)$
$\mu_{jk}^{(l)} = I_{\chi_d}^{j} \alpha_{jk}^{(l)} + \sum_{i=1}^{N} \tau_i I_{\chi_d}^{j} \phi_{i,j,k}^{(l)} + 1 - I_{\chi_d}^{j}, \mu_k^{(g)} = \alpha_k^{(g)} + \sum_{i=1}^{N} (1 - \tau_i)\phi_{ik}^{(g)}$
$\tau_i = \{1 + \exp\{-\Psi(\gamma_1) + \Psi(\gamma_2) - \sum_{j=1}^{J} \sum_{k=1}^{K} I_{\chi_d}^{j} \phi_{i,j,k}^{(l)} (\Psi(\mu_{jk}^{(l)}) - \Psi(\sum_{n=1}^{K} \mu_{jn}^{(l)}))\}\}$
$- \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{v=1}^{V} I_{\chi_d}^{j} \phi_{i,j,k}^{(l)} I_{w_i}^{v} (\Psi(\eta_{jkv}^{(l)}) - \Psi(\sum_{v=1}^{V} \eta_{jkv}^{(l)}))$
$+ \sum_{k=1}^{K} \phi_{ik}^{(g)} (\Psi(\mu_k^{(g)}) - \Psi(\sum_{i=1}^{K} \mu_j^{(g)})) + \sum_{k=1}^{K} \sum_{v=1}^{V} \phi_{ik}^{(g)} I_{w_i}^{v} (\Psi(\eta_{kv}^{(g)}) - \Psi(\sum_{v=1}^{V} \eta_{kv}^{(g)}))^{-1}$
$\phi_{i,j,k}^{(l)} \propto \exp\{I_{\chi_d}^{j} \tau_i (\Psi(\mu_{jk}^{(l)}) - \Psi(\sum_{n=1}^{K} \mu_{j,n}^{(l)}) + \sum_{v=1}^{V} I_{w_i}^{v} (\Psi(\eta_{jkv}^{(l)}) - \Psi(\sum_{v=1}^{V} \eta_{jkv}^{(l)}))\}$
$\phi_{ik}^{(g)} \propto \exp\{(1 - \tau_i)(\Psi(\mu_k^{(g)}) - \Psi(\sum_{j=1}^{K} \mu_j^{(g)}) + \sum_{v=1}^{V} I_{w_i}^{v} (\Psi(\eta_{kv}^{(g)}) - \Psi(\sum_{v=1}^{V} \eta_{kv}^{(g)}))\}$
  until convergence
```

References

[1] T. Broderick, N. Boyd, A. Wibisono, A.C. Wilson, M.I. Jordan, Streaming variational bayes, Adv. Neural Inf. Process. Syst. (2013) 1727–1735.

[2] J. McInerney, R. Ranganath, D.M. Blei, The population posterior and bayesian inference on streams, in: Advances in Neural Information Processing Systems (NIPS), 2015.

[3] A. Masegosa, D.N. Thomas, L. Helge, R. Darío, S. Antonio, L.M. Anders, Bayesian models of data streams with hierarchical power priors, in: Proceedings of the 34th International Conference on Machine Learning (ICML), 2017, pp. 2334–2343.

[4] T.D. Bui, C. Nguyen, R.E. Turner, Streaming sparse gaussian process approximations, Advances in Neural Information Processing Systems (2017) 3299–3307.

[5] M. Faraji, K. Preuschoff, W. Gerstner, Balancing new against old information: The role of puzzlement surprise in learning, Neural Comput. 30 (2018) 34–83.

[6] Z. Huang, H. Chen, D. Zeng, Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering, ACM Transactions on Information Systems (TOIS) 22 (2004) 116–142.

[7] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowl.-Based Syst. 46 (2013) 109–132.

[8] S. Banerjee, K. Ramanathan, A. Gupta, Clustering short texts using wikipedia, in, in: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2007, pp. 787–788.

[9] X. Cheng, X. Yan, Y. Lan, J. Guo, Btm: Topic modeling over short texts, IEEE Trans. Knowl. Data Eng. 26 (2014) 2928–2941.

[10] K. Mai, S. Mai, A. Nguyen, N. Van Linh, K. Than, Enabling hierarchical dirichlet processes to work better for short texts at large scale, in: Advances in Knowledge Discovery and Data Mining. Springer. volume 9652 of Lecture Notes in Computer Science, 2016, pp. 431–442.

[11] J. Tang, Z. Meng, X. Nguyen, Q. Mei, M. Zhang, Understanding the limiting factors of topic modeling via posterior contraction analysis, in, in: Proceedings of The 31st International Conference on Machine Learning (ICML), 2014, pp. 190–198.

[12] N. Oppermann, G. Robbers, T.A. Enßlin, Reconstructing signals from noisy data with unknown signal and noise covariance, Phys. Rev. E 84 (2011), 041118.

[13] L.M. Rickett, N. Pullen, M. Hartley, C. Zipfel, S. Kamoun, J. Baranyi, R.J. Morris, Incorporating prior knowledge improves detection of differences in bacterial growth rate, BMC systems biology 9 (2015) 60.

[14] J. Liang, L. Jiang, D. Meng, A. Hauptmann, Leveraging multi-modal prior knowledge for large-scale concept learning in noisy web data, in, in: Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, ACM, 2017, pp. 32–40.

[15] B. Luo, Y. Feng, Z. Wang, Z. Zhu, S. Huang, R. Yan, D. Zhao, Learning with noise: enhance distantly supervised relation extraction with dynamic transition matrix, 2017. arXiv preprint arXiv:1705.03995.

[16] C. Ha, V.D. Tran, V.L. Ngo, K. Than, Eliminating overfitting of probabilistic topic models on short and noisy text: The role of dropout, Int. J. Approximate Reasoning 112 (2019) 85–104.

[17] Y. Jo, A.H. Oh, Aspect and sentiment unification model for online review analysis, ACM International Conference on Web Search and Data Mining (2011) 815–824.

[18] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.

[19] L. Theis, M. Hoffman, A trust-region method for stochastic variational inference with applications to streaming data, in: Bach, F., Blei, D. (Eds.), Proceedings of the 32nd International Conference on Machine Learning, PMLR, Lille, France, 2015, pp. 2503–2511. URL:/http://proceedings.mlr.press/v37/theis15.html.

[20] M.D. Hoffman, D.M. Blei, C. Wang, J.W. Paisley, Stochastic variational inference, Journal of Machine Learning Research 14 (2013) 1303–1347.

[21] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the NAACL-HLT, Association for Computational Linguistics, 2019, pp. 384–394.

[22] D. Erhan, Y. Bengio, A. Courville, P.A. Manzagol, P. Vincent, S. Bengio, Why does unsupervised pre-training help deep learning?, Journal of Machine Learning Research 11 (2010) 625–660

[23] A. Ahmed, E.P. Xing, Staying informed: supervised and semi-supervised multi-view topical analysis of ideological perspective, Empirical Methods in Natural Language Processing (2010) 1140–1150.

[24] N. Van Linh, N.K. Anh, K. Than, C.N. Dang, An effective and interpretable method for document classification, Knowl. Inf. Syst. 50 (2017) 763–793, https://doi.org/10.1007/s10115-016-0956-6.

[25] G.A. Diamond, S. Kaul, Prior convictions: Bayesian approaches to the analysis and interpretation of clinical megatrials, J. Am. Coll. Cardiol. 43 (2004) 1929–1939.

[26] M.E. Alfaro, M.T. Holder, The posterior and the prior in bayesian phylogenetics, Annu. Rev. Ecol. Evol. Syst. 37 (2006) 19–42.

[27] Newman, E. Mark, Power laws, pareto distributions and zipf's law, Contemp. Phys. 46 (2005) 323–351.

[28] Piantadosi, T. Steven, Zipfs word frequency law in natural language: a critical review and future directions, Psychonomic Bull. Rev. 21 (2014) 1112–1130.

[29] I. Sato, H. Nakagawa, Topic models with power-law using pitman-yor process, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2010) 673–682.

[30] C. Lin, Y. He, C. Pedrinaci, J. Domingue, Feature lda: a supervised topic model for automatic detection of web api documentations from the web, International Semantic Web Conference, Springer. (2012) 328–343.

[31] C. Lin, Y. He, Joint sentiment/topic model for sentiment analysis, in: ACM Conference on Information and Knowledge Management, 2009, pp. 375–384.

[32] J.G. Ibrahim, M.H. Chen, Power prior distributions for regression models, Statistical Science (2000) 46–60.

[33] J.G. Ibrahim, M.H. Chen, Y. Gwon, F. Chen, The power prior: theory and applications, Stat. Med. 34 (2015) 3724–3749.

[34] M.C. Hughes, E. Sudderth, Memoized online variational inference for dirichlet process mixture models, Advances in Neural Information Processing Systems (2013) 1133–1141.

[35] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, L.K. Saul, An introduction to variational methods for graphical models, Machine learning 37 (1999) 183–233.

[36] S. Kullback, Information theory and statistics, Courier Corporation, 1997.

[37] J. Turian, L. Ratinov, Y. Bengio, Word representations: a simple and general method for semi-supervised learning, in, in: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL, 2010, pp. 384–394.

[38] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, science 313 (2006) 504–507.

[39] V. Le, C. Phung, C. Vu, L. Ngo, K. Than, Streaming aspect-sentiment analysis, in: IEEE RIVF International Conference on Computing Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2016, pp. 181–186.https://doi.org/10.1109/RIVF.2016.7800291.

[40] K.L. Clarkson, Coresets, sparse greedy approximation, and the frank-wolfe algorithm, ACM Trans. Algorithms 6 (2010) 63.

[41] C. Kluckhohn, Human behavior and the principle of least effort. george kingsley zipf, Am. Anthropol. 52 (1950) 268–270.

[42] P. Xie, E.P. Xing, Integrating document clustering and topic modeling, in: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2013, pp. 694–703.

Duc Anh Nguyen graduated from Hanoi University of Science and Technology with a master degree in Information Technology. He is currently a Ph.D student at Kyoto University, Bioinformatics Center. His interest focuses on representation learning and its application.

Van Linh Ngo is currently a PhD student at Hanoi University of Science and Technology (HUST), Vietnam. He also received B.S (2011) and M.S (2014) degrees from HUST. He is a member of Data Science Laboratory, HUST. His research interests include topic model, continual learning, recommender systems, and big data.

Kim Anh Nguyen received B.S (1988) and Ph.D. (1994) from Hanoi University of Science and Technology. Her recent research interests include topic modeling, opinion mining, deep learning, graph analytics with big data.

Canh Hao Nguyen received his B.S. degree in Computer Science from the University of New South Wales, Australia, M.S. and Ph.D. degrees from JAIST, Japan. He has been working in machine learning and bioinformatics. His current interests are machine learning for graph data, sparse modeling with applications in biological network analysis.

Khoat Than is currently an associate professor at Hanoi University of Science and Technology. He received Ph.D. degree from Japan Advanced Institute of Science and Technology in 2013. He joins the Program Committees of various leading international conferences, including ICML, NIPS, IJCAI, ICLR, PAKDD, ACML. His recent research interests include representation learning, deep generative models, topic modeling, continual learning.