Infinite Dropout for training Bayesian models from data streams

Van-Son Nguyen[1], Duc-Tung Nguyen[1,2], Linh Ngo Van[1], Khoat Than[1,2]
[1]*Hanoi University of Science and Technology*, Hanoi, Vietnam
[2]*VinAI Research*, Hanoi, Vietnam
sonnguyenkstn@gmail.com, ductungnguyen1997@gmail.com, {linhnv, khoattq}@soict.hust.edu.vn

***Abstract*—The ability to continuously train Bayesian models in streaming environments is highly important in the era of big data. However, it has to face the famous stability-plasticity dilemma and the problem of noisy and sparse data. We propose a novel and easy-to-implement framework, called Infinite Dropout (iDropout), to address these challenges. iDropout has an easy mechanism to balance between old and new information, which allows models to trade off stability against plasticity. Thanks to the ability to reduce overfitting and the ensemble property of Dropout, our framework obtains better generalization, thus effectively handles undesirable effects of noise and sparsity. Further, iDropout is able to adapt quickly to abnormal changes in data streams. We theoretically analyze the equivalence of Dropout in iDropout to a regularizer, well applied to a much larger context than what was known before. Extensive experiments show that iDropout significantly outperforms the state-of-the-art baselines.**

***Index Terms*—Bayesian models, Data streams, Streaming learning, Dropout, Regularization**

I. Introduction

We are interested in how to efficiently train a Bayesian model in streaming conditions where the data comes continuously and infinitely. Learning from data streams requires to address the stability-plasticity dilemma [1], i.e., an algorithm should keep the learned knowledge stable while enabling the model to adapt well with sudden changes in the environments. Such a dilemma is present in most continual learning systems. Further, this learning process can be negatively affected by undesirable properties of data, including noise, which potentially causes overfitting, and sparsity, i.e., the situation when model does not have enough relevant information to make good predictions for unseen data. Our work focuses on these challenges.

Some recent studies [2]–[4] have provided excellent solutions for learning from data streams. Those methods enable Bayesian models, which are designed for static conditions, to work with data streams. However, those methods are limited when facing the above challenges. For example, we found that *streaming variational Bayes* (SVB) [2] becomes too stable after receiving a large enough amount of data. In other words, SVB makes models evolve slowly and have difficulties learning new information, thus fail to adapt sudden changes from the data stream. This is a serious problem and potentially happens in other methods, but has not been studied formally in any research before. Further, existing methods do not have any efficient way to deal with noisy and sparse data.

In this paper, we propose a novel framework called *Infinite Dropout* (iDropout) which enables a wide range of models to work in streaming environments. Our framework has several benefits. Firstly, iDropout has an easy mechanism to balance the information among old and new data throughout the data stream, which helps tackle the stability-plasticity dilemma. Secondly, we theoretically prove that Dropout in iDropout plays as a data-dependent regularizer, which allows our method to effectively overcome the overfitting issue. Moreover, with a fast approximation via a scaling factor, Dropout in our method works as an ensemble of an exponential number of learners, which is very useful in making good predictions for future data. These advantages help our method obtain better generalization. This is extremely important when data comes continuously with high uncertainty, which has the potential for sudden changes or undesirable properties such as noise and sparsity. Furthermore, our analysis about the role Dropout as regularization applies well to a large class of Bayesian models, extending existing works [5]–[10] to significantly wider contexts.

We did extensive experiments to compare iDropout with existing frameworks, using two base models: *latent Dirichlet allocation* (LDA) [11] for topic modeling and *Naive Bayes* (NB) [12] for classification. Empirical results show that our framework gives major improvements over existing state-of-the-art streaming methods on both learning tasks.

Roadmap: Section 2 briefly provides closely related work. We formally describe the iDropout framework and its applications in Section 3. Non-trivial findings about iDropout are described in Section 4. Finally, extensive evaluation appears in Section 5.

II. Related work

Recently, a lot of effort has been made to adapt Bayesian models from static conditions to streaming ones. Some work [2], [4], [13] propose recursive updating of the variational distribution. Streaming variational Bayes (SVB) [2] uses the variational parameter from preceding time step as the parameter for the prior distribution of current time step. However, this mechanism can be inappropriate in data streams, since the variational parameter learned from past data may not describe properly the property of current data. In particular, once given enough data, SVB becomes too stable and thus unable to learn new information from the data stream. To avoid this problem, HPP [4] is proposed to exponentially forget the

978-1-7281-0858-2/19/$31.00 ©2019 IEEE

variational parameters associated with old data, where the forgetting rate is considered as a hidden variable. Unfortunately, the introduction of this new latent variable makes the model no more conjugate, which requires non-trivial efforts to infer for the complicated Bayesian models (when the forgetting rate is considered a fixed hyperparameter, the method is called SVB-PP). The second direction is to cast the inference problem as a stochastic optimization problem. Stochastic variational inference (SVI) [14] is a typical example. However, SVI conditions on a fixed dataset to reveal the variational distribution, which isn't really appropriate in data streams. Population variational Bayes (PVB) [3], a closely related framework addresses this problem by assuming that the data stream is generated by sampling α data points from the population distribution F_α.

It is worth noting that none of these frameworks consider seriously the problem of noise and sparsity, which are pervasive in streaming environments. In order to address these problems, we consider using Dropout [15], which is a well-known stochastic regularization technique introduced in the context of feed forward neural networks. The idea of Dropout is to randomly omit a subset of features at each iteration of the training process. Dropout has two great advantages: it prevents models from overfitting by discouraging co-adaptation of features and more especially, Dropout provides an efficient way to approximately combine exponentially many models, working as a form of ensemble learning. Dropout works well for various machine learning methods, including neural networks [16], support vector machine [17], matrix factorization [18] and topic model [19]. The theory behind Dropout is considered by some recent researches [6], [8]–[10], [20]. Particularly, [6] shows that Dropout is equivalent to an L_2 regularizer when applied to generalized linear models.

III. Infinite Dropout

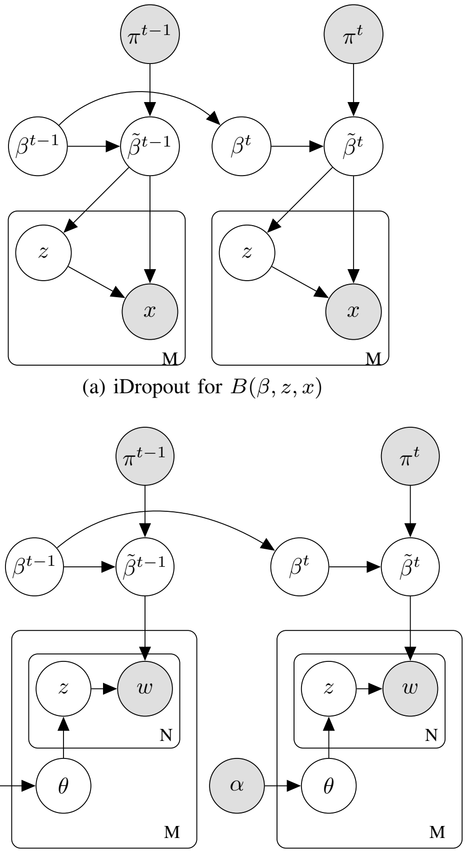
In this section, at first we present the framework for a general Bayesian model. After that, we explicitly describe applications to LDA and NB to clarify our framework.

A. The framework

We consider a general model $B(\beta, z, x)$ [3], [14] involving observations, global variables and local variables. The global variable is matrix β which has size $K \times V$, shared among data points $x_{1:M}$, while local variable z_i only governs ith data point x_i. In traditional Bayesian methods, we condition on a fixed dataset to reveal the posterior distribution of hidden variables $p(\beta, z|x)$. Undoubtedly, this can not work with data streams where the data come in an infinite sequence of minibatches $C = \{D^1, D^2, \cdots, D^t, \cdots\}$ and each minibatch t consists of M observed data points: $D^t = \{x_1^t, x_2^t, \cdots, x_M^t\}$.

We need to extend the model to also describe the dynamics of the data stream. Here we assume that only the global variable β evolves over time, which we indicate with superscript t, i.e., β^t. We introduce a transition model $p(\beta^t|\beta^{t-1})$ to describe the transformation between two consecutive time steps:

$$p(\beta_k^t|\beta_k^{t-1}) = \mathcal{N}(.|\beta_k^{t-1}, \sigma^2 I) \tag{1}$$



(a) iDropout for $B(\beta, z, x)$

(b) iDropout for LDA

Fig. 1: Graphical representation for $iDropout$.

where k is the row index of β^{t-1} and I is the identity matrix of size V. The variance σ^2 is a hyperparameter, which describes our assumption about the fluctuation of β_k between two consecutive time steps.

Dropout is utilized in our framework as follows. In each time step t, we drop randomly a number of elements of matrix β^t. This is implemented by using a hyperparameter called drop matrix π^t to make *element-wise product* with β^t, then going through a transformation: $\tilde{\beta}^t = f(\beta^t \odot \pi^t)$. Transformation f should be chosen to assure that $\tilde{\beta}^t$ can replace β in model $B(\beta, z, x)$ at each time step t (in the later subsections, we use *softmax* to be the transformation). Given the new global variable $\tilde{\beta}^t$ at each minibatch t, the generative process of all data points is similar to the original model B (Fig. 1a). In order to keep the randomness of Dropout, we use a different drop matrix at each minibatch. Each element π_{ij}^t of π^t is generated using one of two options:

1) Bernouli dropout: $p(\pi_{ij}^t = 1) = 1 - dr$, $p(\pi_{ij}^t = 0) = dr$
2) Inverted dropout:

$$p(\pi_{ij}^t = 1/(1 - dr)) = 1 - dr, p(\pi_{ij}^t = 0) = dr \tag{2}$$

in which dr is drop rate. Note that when β^t is used at test time, it has to be rescaled by $\mathbb{E}[\pi_{ij}^t]$. By doing this scaling, $2^{K \times V}$ models with shared parameters can be combined into a

Algorithm 1 Learning in iDropout

```
    Require: Drop rate $dr$, variance $\sigma^2$, data sequence $\{D^1, D^2, \cdots\}$
    Ensure: Global variable $\beta$
    Initialize $\beta^0$ randomly.
    for $t^{th}$ minibatch with data $D^t$ do
        Draw drop matrix $\pi^t$ randomly
        Do inference w.r.t. the local variables $z$ (e.g., by doing inference or sampling), given $\beta^t$ and $D^t$
        Estimate $\beta^t$ by using a gradient-based algorithm, given the statistics from $z, D^t$
    end for
```

Algorithm 2 iDropout training for LDA

```
    Require: Prior $\alpha$, drop rate $dr$, variance $\sigma^2$, data sequence $\{D^1, D^2, \cdots\}$
    Ensure: Global variable $\beta$
    Initialize $\beta^0$ randomly.
    for $t^{th}$ minibatch with data $D^t$ do
        Draw drop matrix $\pi^t$ randomly
        for each document $d$ in $D^t$ do
            Infer $(\gamma_d, \phi_d)$ by alternatively updating (4) and (5)
        end for
        Find each $\beta_k^t$ by maximizing (6)
    end for
```

single model to be used at test time, which works as a form of ensemble learning.

Learning in iDropout is done at each minibatch t by estimating β^t through the posterior $p(\beta^t|\beta^{t-1}, \pi^t, D^t)$, where β^{t-1} is learned from the previous minibatch:

$$\begin{aligned} &p(\beta^t|\beta^{t-1}, \pi^t, D^t) \propto p(\beta^t, D^t|\beta^{t-1}, \pi^t) \\ &\propto p(\beta^t|\beta^{t-1})p(D^t|\pi^t, \beta^t) \propto p(\beta^t|\beta^{t-1})p(D^t|\tilde{\beta}^t) \end{aligned}$$

In log form, we obtain:

$$\begin{aligned} F(\beta^t) &= \log p(\beta^t|\beta^{t-1}, \pi^t, D^t) \\ &= \log p(\beta^t|\beta^{t-1}) + \log p(D^t|\tilde{\beta}^t) + const \end{aligned} \quad (3)$$

The learning process is composed of two phases: infer local variables and update global variables, respectively. While the inference of local variables z is inherited from the original model B (e.g., by using variational inference or sampling from $p(z|x, \tilde{\beta}^t)$), we focus on optimizing F with respect to β^t. We extract the component $G(\beta^t)$, which contains β^t, from log-likelihood $\log p(D^t|\tilde{\beta}^t)$. Then, we obtain the objective function: $F(\beta^t) = \log p(\beta^t|\beta^{t-1}) + G(\beta^t)$ and maximize it by using a gradient-based method. Algorithm 1 briefly describes the learning process.

B. Case study 1: when LDA is the base model

In this subsection, we show an application of iDropout to latent Dirichlet allocation [11], which is used for document analysis. Suppose that each minibatch t consists of M documents and each document d contains N_d words. Hyperparameter α is the parameter of Dirichlet distribution (Dir) for topic mixture θ, and the matrix β of size $K \times V$ is the topic distribution over V words in the vocabulary.

The generative process of documents in each minibatch t^{th} is as follows (Fig. 1b).

1) Draw the global variable β^t: $\beta_k^t \sim \mathcal{N}(\beta_k^{t-1}, \sigma^2 I)$
2) Calculate the topic distribution matrix:

$$\tilde{\beta}_{kj}^t = \text{softmax}(\beta_k^t \odot \pi_k^t)_j = \frac{\exp(\beta_{kj}^t \pi_{kj}^t)}{\sum_{i=1}^V \exp(\beta_{ki}^t \pi_{ki}^t)}$$

3) For each document d in minibatch t:
   a) Draw topic mixture: $\theta_d \sim Dirichlet(\alpha)$
   b) For n^{th} word in document d:
      i) Draw topic index: $z_{dn} \sim Multinomial(\theta_d)$
      ii) Draw word: $w_{dn} \sim Multinomial(\tilde{\beta}_{z_{dn}}^t)$

Learning process: As in Algorithm 1, we estimate β^t through the log-posterior:

$$\begin{aligned} &\log p(\beta^t|\beta^{t-1}, \pi^t, \alpha, D^t) \\ &= \log p(\beta^t|\beta^{t-1}) + \log p(D^t|\tilde{\beta}^t, \alpha) + const \end{aligned}$$

As mentioned above, inference for local variables θ and z can be done by utilizing different inference methods, including variational inference and Gibbs sampling. In the experiments, we use mean-field variational inference as in the original paper [11]. For each document d: $q(\theta_d, z_d|\gamma_d, \phi_d) = q(\theta_d|\gamma_d)\prod_{n \in [N_d]} q(z_{dn}|\phi_{dn})$ with the variational distribution: $q(\theta_d|\gamma_d) = Dir(.|\gamma)$ and $q(z_{dn}|\phi_{dn}) = Mult(.|\phi_{dn})$, where γ_d and ϕ_d are variational parameters. According to [11], these parameters for each document d are updated until convergence as follow:

$$\gamma_{dk} \leftarrow \alpha_k + \sum_{n=1}^{N_d} \phi_{dnk} \text{ for } k = 1, \cdots, K \quad (4)$$

$$\phi_{dnk} \propto \exp(\mathbb{E}_q[\log \theta_{dk}] + \sum_{v=1}^{V} \mathbb{I}[w_{dn} = v] \log \beta_{kv}) \quad (5)$$

where $[V] = \{1, \cdots, V\}$, $\mathbb{I}[.]$ is the indicator function. Extracting $G(\beta^t)$ from $\log p(D^t|\tilde{\beta}^t, \alpha)$, we obtain the objective function. Since the topics are independent of each other, we only consider the objective function with respect to β_k^t:

$$\begin{aligned} F(\beta_k^t) &= \log p(\beta_k^t|\beta_k^{t-1}) + \sum_{d=1}^{M}\sum_{n=1}^{N_d} \log p(w_{dn}|z_{dn}, \tilde{\beta}^t) \\ &= -\frac{1}{2\sigma^2}||\beta_k^t - \beta_k^{t-1}||^2 + \sum_{d=1}^{M}\sum_{n,j=1}^{N_d,V} \phi_{dnk}\mathbb{I}[w_{dn} = j] \log \tilde{\beta}_{kj}^t \\ &= -\frac{1}{2\sigma^2}||\beta_k^t - \beta_{k-1}^t||^2 + \sum_{d=1}^{M}\sum_{n,j=1}^{N_d,V} \phi_{dnk} I[w_{dn} = j]\beta_{kj}^t \pi_{kj}^t \\ &\quad - \sum_{d=1}^{M}\sum_{n,j=1}^{N_d,V} \phi_{dnk} I[w_{dn} = j] \log(\sum_{i=1}^{V} \exp(\beta_{ki}^t \pi_{ki}^t)) \end{aligned} \quad (6)$$

The objective function F is guaranteed to be concave. Indeed, $-\frac{1}{2\sigma^2}||\beta_k^t - \beta_k^{t-1}||^2$ and $\beta_{kj}^t \pi_{kj}^t$ are obviously concave with respect to β_k^t, while the log-sum-exp is also a well-known convex function. Therefore, $F(\beta_k^t)$ is concave with respect to β_k^t, and we can find its maximum by applying gradient ascent on F. We sum up the learning algorithm of iDropout for LDA in Algorithm 2.

C. Case study 2: when NB is the base model

We use Multinomial Naive Bayes (NB) [12] for document classification. Suppose that each minibatch consists of M documents, each document d contains N_d words and belongs to a class $c_d \in \{1, 2, \cdots C\}$. Each c_d is generated by: $c_d \sim Mult(\alpha)$ in which α is a fixed symmetric vector, and finally β of size $C \times V$ is the class distribution over V words in the vocabulary.

The generative process for each minibatch t is as follows. Firstly, draw the global variable β^t: $\beta_c^t \sim \mathcal{N}(\beta_c^{t-1}, \sigma^2 I)$ and calculate the class matrix: $\tilde{\beta}_{cj}^t = \text{softmax}(\beta_c^t \odot \pi_c^t)_j$. Each document d is drawn by first choosing the class label $c_d \sim Mult(\alpha)$ and then drawing n^{th} word $w_{dn} \sim Mult(\tilde{\beta}_{c_d}^t)$.

Learning process: From (3), we extract the term associated with β^t for each class c:

$$\begin{aligned} F(\beta_c^t) &= \log p(\beta_c^t|\beta_c^{t-1}) + \sum_{d \in D_c^t} \sum_{n=1}^{N_d} \log p(w_{dn}|c_d, \tilde{\beta}^t) \\ &= -\frac{1}{2\sigma^2}||\beta_c^t - \beta_c^{t-1}||_2^2 + \sum_{d \in D_c^t} \sum_{n=1}^{N_d} \sum_{j=1}^{V} \mathbb{I}[w_{dn} = j] \log \tilde{\beta}_{cj}^t \\ &= -\frac{1}{2\sigma^2}||\beta_c^t - \beta_c^{t-1}||_2^2 + \sum_{d \in D_c^t} \sum_{n=1}^{N_d} \sum_{j=1}^{V} \mathbb{I}[w_{dn} = j] \beta_{cj}^t \pi_{cj}^t \\ &\quad - N_c \log(\sum_{i \in [V]} \exp(\beta_{ci}^t \pi_{ci}^t)) \end{aligned}$$

where D_c^t includes all documents which belong to class c, N_c is the total number of words in all documents belonging to class c. Learning for NB is very simple. At each minibatch t, we use gradient ascent to maximize $F(\beta_c^t)$ with respect to β_c^t.

IV. Discussions about iDropout

This section shows our non-trivial findings about iDropout. We compare the behavior of iDropout and other frameworks for data streams, and also consider the theory behind the effect of Dropout.

A. The stability-plasticity dilemma

In this subsection, we investigate how different streaming learning frameworks trade off stability against plasticity in models similar to LDA[1], i.e., how they balance between old and new information from data streams. In particular, SVB [2] uses the variational parameter of the global variable β^t at time step t, which we denote by λ^t, as the parameter in the Dirichlet prior distribution at time step $t + 1$. In other words, for each $k \in \{1, \cdots, K\}$, β_k^{t+1} has the prior distribution $Dir(\beta_k^{t+1}|\lambda_k^t)$. Then we have:

Theorem 1. *In SVB:* $\mathbb{E}_{Dir}[\beta_{kj}^{t+1}] = \beta_{kj}^t$ *and* $Var_{Dir}[\beta_{kj}^{t+1}] \to 0$ *as* $t \to \infty$.

Proof. SVB [2] proposes recursive updating of the variational distribution. For LDA (conjugate models, exponential family, i.i.d. data), the variational parameter λ^t of global variable β^t is updated by: $\lambda^t = \lambda^{t-1} + \tilde{\lambda}^t$, where λ^{t-1} is made available from the previous minibatch and $\tilde{\lambda}^t$ is the learned information from the current minibatch. In other words, λ^t is addition of the learned information from all previous steps:

$$\lambda^t = \tilde{\lambda}^t + \cdots + \tilde{\lambda}^1 + \lambda^0$$

where:

$$\begin{aligned} |||\tilde{\lambda}^t|||_1 &= \sum_{k=1}^{K} \sum_{j=1}^{V} \tilde{\lambda}_{kj}^t = \sum_{d \in D^t} \sum_{n=1}^{N_d} \sum_{k=1}^{K} \sum_{j=1}^{V} \phi_{dnk} \mathbb{I}[w_{dn} = j] \\ &= \sum_{d \in D^t} \sum_{n=1}^{N_d} \sum_{k=1}^{K} \phi_{dnk} = \sum_{d \in D^t} N_d \geq 1 \end{aligned}$$

Therefore, $||\lambda^t||_1 = \sum_{i=1}^{T} \sum_{d \in D^t} N_d \geq t$, which approaches infinity as t goes to infinity. When a new minibatch $t+1$ arrives, λ^t will be used as the parameter of the prior: $p(\beta_k^{t+1}|\lambda_k^t) = Dir(.|\lambda_k^t)$. This distribution has the expectation:

$$\mathbb{E}_{Dir}[\beta_k^{t+1}] \propto \lambda_k^t = \beta_k^t$$

and the variance:

$$\text{Var}[\beta_{kj}^{t+1}] = \frac{\lambda_{kj}^t (\sum_{i=1}^{V} \lambda_{ki}^t - \lambda_{kj}^t)}{(\sum_{i=1}^{V} \lambda_{ki}^t)^2 (\sum_{i=1}^{V} \lambda_{ki}^t + 1)}$$

which varies inversely with size of λ_k^t. As $t \to \infty$, leading to $||\lambda^t||_1 \to \infty$, we have $\text{Var}[\beta_{kj}^{t+1}] \to 0$. □

This problem is potentially present in SVB-PP [4], albeit λ^t takes longer to accumulate: $\lambda^t = \rho\lambda^{t-1} + (1 - \rho)\eta + \tilde{\lambda}^t$, where ρ is the forgetting factor $(0 < \rho < 1)$ and η is the uninformative prior.

When this happens, SVB and SVB-PP expect the model at time $t + 1$ to be nearly identical to the model at time t. This phenomenon essentially says that a model will evolve very slowly and have difficulties in learning new information, thus could not deal well with sudden changes in the environment.

iDropout does not encounter this problem. In iDropout, we have an easy mechanism to balance the information between old and new data. Indeed, to maximize the objective function $F(\beta_k^t) = -\frac{1}{2\sigma^2}||\beta_k^t - \beta_k^{t-1}||_2^2 + \log p(D^t|\tilde{\beta}_k^t)$ in (3), we need to consider both components. While the first term encourages new model β^t to fluctuate around the previously learned β^{t-1}, the latter allows model to accommodate information from new data D^t. In other words, iDropout helps model to flexibly learn new information, while retaining relevant information from historical observations to maintain the stability.

The balance ability of iDropout is easily controlled by the variance σ^2. The bigger σ^2 is, the more we focus on learning

[1] Such models require the global variable β to be in a simplex, e.g., NB.

new information, rather than keeping old information, and vice versa. This balance is unchanged throughout the learning process. Unlike iDropout, SVB and SVB-PP cannot control this balance. Particularly, in LDA, SVB and SVB-PP becomes too rigid and unable to learn new information after receiving a large amount of data, due to the reason mentioned above.

B. The role of Dropout in iDropout

In streaming environments, the problem of noisy and sparse data is unavoidable. Specifically, learning from noisy data potentially makes models become overfitting, while sparsity in data may not provide enough relevant information to make good predictions for unseen data, both leading to poor performance.

To overcome these challenges, we propose to utilize Dropout by omitting randomly a number of elements of the global variable β^t at each time step t. Dropout in our framework has two main roles. Firstly, we theoretically prove that it plays as a data-dependent regularizer, which makes iDropout more robust against overfitting. Moreover, in our framework, Dropout is used throughout the data stream, leading to a special effect, which is ensemble learning. Indeed, at each time step in training process, the use of Dropout is equivalent to sampling a single learner from a set of $2^{K\times V}$ possible learners. Then, by rescaling β^t with $\mathbb{E}[\pi^t]$, $2^{K\times V}$ learners with shared parameters can be combined into a single learner to be used at test time.

The ability to prevent overfitting and the ensemble property make iDropout have better generalization on future data, which is specially important in streaming learning, because data streams can be non-stationary and have high uncertainty.

C. Dropout as regularization

We examine the theory behind the effect of Dropout in iDropout for two models LDA and NB.

Theorem 2. *For LDA and NB, Dropout in iDropout is equivalent to a L2-regularization $R(\beta)$:*

$$R(\beta) = \frac{dr}{2(1-dr)} \sum_{k=1}^{K} \sum_{j=1}^{V} \left[\mu_{kj}(1-\mu_{kj}) \sum_{i=1}^{V} u_{kj} \right] \beta_{kj}^2$$

in which μ_{kj} is the model probability and:

$$u_{kj} = \begin{cases} \sum_{d=1}^{M} \sum_{n=1}^{N_d} \phi_{dnk} \mathbb{I}[w_{dn} = j] \text{ in LDA} \\ \sum_{d \in D_c^t} \sum_{n=1}^{N_d} \mathbb{I}[w_{dn} = j] \text{ in NB} \end{cases}$$

Proof. The learning process at each minibatch in iDropout for LDA and NB reduces to maximizing the objective function of the following form:

$$F = -\frac{1}{2\sigma^2} ||\beta_k - \beta_k^{prev}||_2^2 + \sum_{j=1}^{V} u_{kj} \log\left(\text{softmax}(\beta_k \odot \pi_k)_j\right)$$

where β_k^{prev} is made available from the previous minibatch (we omit superscript t for simplicity) and u_{kj} is defined as in the statement of the theorem.

Consider $x_1, \cdots, x_K$ as K-dimension one-hot vectors (x_k has only k^{th} element activated) and $\beta = [\beta_1 \beta_2 \cdots \beta_V]$ where β_j is j^{th} column of matrix β, then:

$$\text{softmax}(\beta_k)_j = \exp(s_{kj} - A(s_k))$$

with $s_{kj} = \beta_j^T x_k$ is a undropped score value and $A(s_k) = \log \sum_{i=1}^{V} \exp(s_{ki})$ is the log-partition function.

Assume π is drawn from the distribution ζ, corresponding the Inverted Dropout: $p(\pi_{ij} = 1/(1-dr)) = 1 - dr, p(\pi_{ij} = 0) = dr$, then $\mathbb{E}_\zeta[\pi_{kj}] = 1$, and:

$$\text{softmax}(\beta_k \odot \pi_k)_j = \exp(\tilde{s}_{kj} - A(\tilde{s}_k))$$

with $\tilde{s}_{kj} = (\beta_i \odot \pi_i)^T x_k$, $A(\tilde{s}_k) = \log \sum_{i=1}^{V} \exp(\tilde{s}_{ki})$.

Using this notation, we can write F as:

$$F = -\frac{1}{2\sigma^2} ||\beta_k - \beta_k^{prev}||_2^2 + \sum_{j=1}^{V} u_{kj} \mathbb{E}_\zeta[\tilde{s}_{kj} - A(\tilde{s}_k)]$$

Since $\mathbb{E}_\zeta[\pi_{kj}] = 1$ so the dropout technique preserves mean, leading to $\mathbb{E}_\zeta[\tilde{s}_{kj}] = s_{kj}$, then we have:

$$\begin{aligned} \mathbb{E}_\zeta[\tilde{s}_{kj} - A(\tilde{s}_k)] &= s_{kj} - A(s_k) - (\mathbb{E}_\zeta[A(\tilde{s}_k)] - A(s_k)) \\ &= \text{softmax}(\beta_k)_j - (\mathbb{E}_\zeta[A(\tilde{s}_k)] - A(s_k)) \end{aligned}$$

Then we can write:

$$\begin{aligned} F = &- \frac{1}{2\sigma^2} ||\beta_k - \beta_k^{prev}||_2^2 + \sum_{j=1}^{V} u_{kj} \log\left(\text{softmax}(\beta_k)_j\right) \\ &- (\mathbb{E}_\zeta[A(\tilde{s}_k)] - A(s_k)) \sum_{j=1}^{V} u_{kj} \end{aligned}$$

Since the log-partition function $A(.)$ is convex, $(\mathbb{E}_\zeta[A(\tilde{s}_k)] - A(s_k))$ is always positive by Jensen's inequality and can therefore be interpreted as a regularizer. Indeed, applying second-order Taylor approximation to $A(\tilde{s}_k)$ around the undropped score vector s_k, we have means and covariances of the dropout features:

$$\begin{aligned} A(\tilde{s}_k) = A(s_k) &+ \nabla A(s_k)^T (\tilde{s}_k - s_k) \\ &+ \frac{1}{2} (\tilde{s}_k - s_k)^T \nabla^2 A(s_k) (\tilde{s}_k - s_k) \end{aligned}$$

then we obtain a following regularizer:

$$\begin{aligned} \mathbb{E}_\zeta[A(\tilde{s}_k)] - A(s_k) &= \frac{1}{2} \mathbb{E}_\zeta[(\tilde{s}_k - s_k)^T \nabla^2 A(s_k) (\tilde{s}_k - s_k)] \\ &= \frac{1}{2} \text{Tr}[\nabla^2 A(s_k) \text{Cov}_\zeta(\tilde{s}_k)] = \frac{1}{2} \sum_{j=1}^{V} \mu_{kj}(1-\mu_{kj}) \text{Var}_\zeta[\tilde{s}_{kj}] \\ &= \frac{1}{2} \sum_{j=1}^{V} \mu_{kj}(1-\mu_{kj}) \beta_j^T \text{Cov}_\zeta(x_k) \beta_j \end{aligned}$$

where $\mu_{kj} = \text{softmax}(s_k)_j$ is the model probability, the variance $\mu_{kj}(1-\mu_{kj})$ measures model uncertainty, and

$$\beta_j^T \text{Cov}_\zeta(x_k) \beta_j = \sum_{m=1}^{K} \frac{dr}{1-dr} x_{km}^2 \beta_{mj}^2 = \frac{dr}{1-dr} \beta_{kj}^2$$

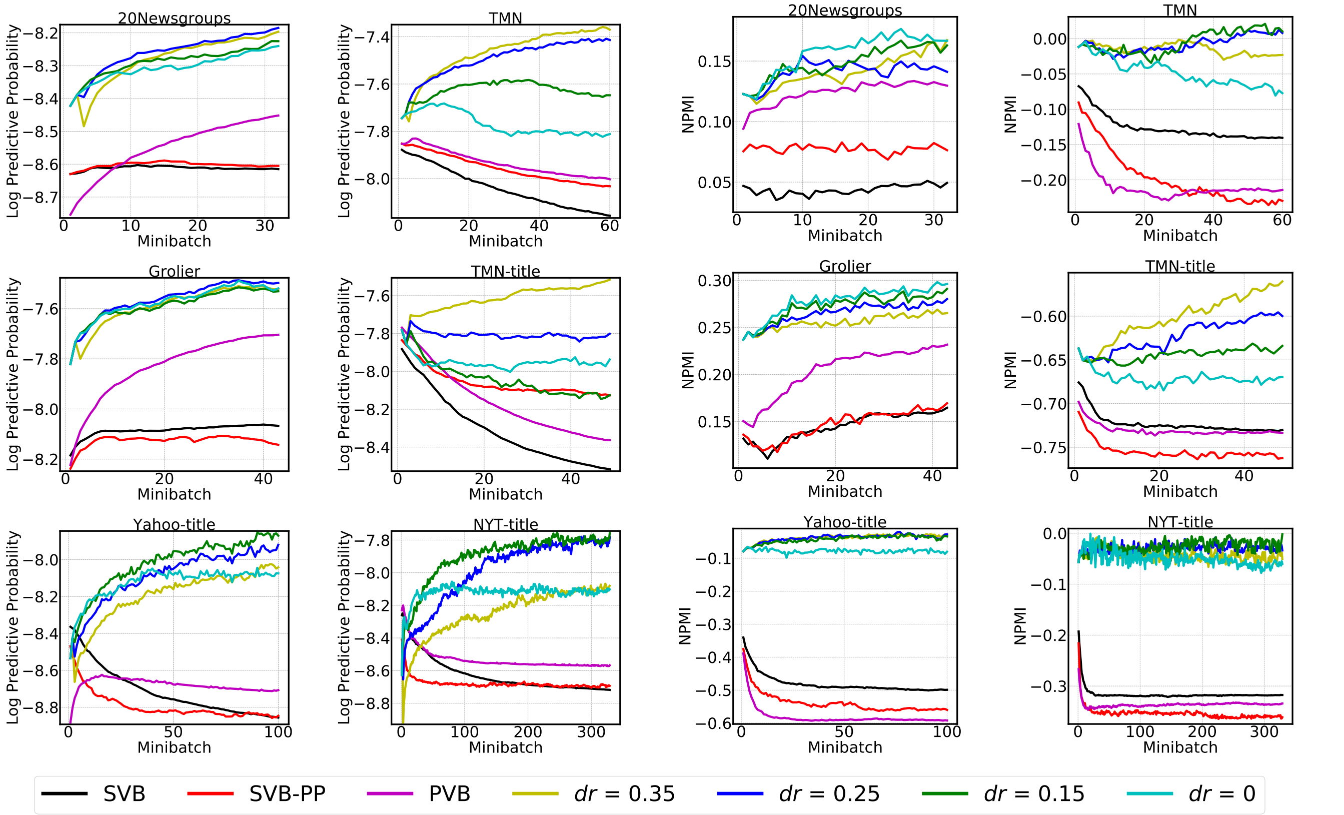


Fig. 2: Performance of 4 methods. iDropout uses drop rate $dr \in \{0.35, 0.25, 0.15, 0\}$ and $\sigma^2 = 100$. LDA is the base model. Higher is better.

Hence, $\mathbb{E}_\zeta[A(\tilde{s}_k)] - A(s_k) = \frac{dr}{2(1-dr)} \sum_{j=1}^{V} \mu_{kj}(1-\mu_{kj})\beta_{kj}^2$ has quadratic format w.r.t β_k. In other words, the effect of Dropout in iDropout is equivalent to a L2-regularization $R(\beta)$:

$$R(\beta) = (\mathbb{E}_\zeta[A(\tilde{s}_k)] - A(s_k)) \sum_{j=1}^{V} u_{kj}$$

$$= \frac{dr}{2(1-dr)} \sum_{j=1}^{V} \left[\mu_{kj}(1-\mu_{kj}) \sum_{j=1}^{V} u_{kj}\right] \beta_{kj}^2 .$$

□

This is a theoretical interpretation on the ability of iDropout to reduce overfitting. Unlike other regularization techniques, each β_{kj} in iDropout has a different regularization parameter $\frac{dr}{2(1-dr)}\mu_{kj}(1-\mu_{kj})\sum_{j=1}^{V} u_{kj}$, depending on the input data. This is interesting, since this data-dependent regularization allows each β_{kj} to have its own search space to catch the geometric property of data. With this property, dropout in our method is more effective than other standard computationally inexpensive regularizers, such as weight decay, filter norm constraints and sparse activity regularization [21].

V. Empirical evaluation

In this section, we conduct various experiments to evaluate the performance of iDropout. Firstly, we simulate the streaming environment using 6 non-chronologically ordered datasets to thoroughly investigate the behavior of different methods on two aspects: how they balance between old and new information from data streams, as well as their ability to deal with noise and sparsity. Additionally, we examine how these methods adapt with sudden changes from the data stream in two settings: (1) using a dataset with time stamp; (2) simulating concept drift.

A. *Baselines*

We compare iDropout with three state-of-the-art frameworks: **SVB** [2], **SVB-PP** [4][2] and **PVB** [3]. We use grid search to select the best version of each framework for each dataset. The range of each parameter is as follows: the forgetting factor $\rho \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.99\}$ for SVB-PP, the population size $\alpha \in \{10^3, 10^4, 10^5, 10^6, 5.10^3, 5.10^4, 5.10^5, 5.10^6\}$ and dimming factor $\kappa \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ for PVB, the variance $\sigma^2 \in \{0.01, 0.1, 1, 10, 100\}$ and the drop rate $dr \in \{0, 0.15, 0.25, 0.35\}$ for iDropout.

B. *Experiments on datasets without time stamp*

Base model and datasets: We use LDA to analyze 6 popular corporas including 2 regular text datasets (20News-

[2] SVB-HPP is not included since its application requires non-trivial efforts. Further, as observed by [4], SVB-HPP is often comparable to the best SVB-PP.

TABLE I: 6 datasets without time stamp

Dataset	Vocab size	Training size	Testing size	words/doc
20Newsgroups	24905	17846	1000	88.2
Grolier	15269	23044	1000	79.9
TMN	11599	31604	1000	24.3
TMN-title	2823	26251	1000	4.6
Yahoo-title	21439	517770	10000	4.6
NYT-title	46854	1664127	10000	5.0

Groups, Grolier[3]) and 4 short text ones (TagMyNews (TMN)[4], TagMyNews-title (TMN-title), Yahoo-title, NYT-title [5]) with some statistics in Table I.

Settings: Since all 6 datasets are not chronologically ordered, we simulate the streaming environment by dividing each dataset into a sequence of minibatches with batchsize: 500 for {Grolier, 20Newsgroups, TMN, TMN-title}, 5000 for {NYT-title, Yahoo-title}. We set prior of topic mixture $\alpha = 0.01$, the number of topic $K = 50$ for {Grolier, 20Newsgroups, TMN, TMN-title}, $K = 100$ for {NYT-title, Yahoo-title}.

Evaluation metric: Log predictive probability [14] (LPP) and Normalized pointwise mutual information (NPMI) [22] are used. While LPP measures the generalization of a model on unseen data, NPMI is used to examine the coherence and interpretability of the learned topics. The details of computing two metrics are given in the Appendix.

Result: The result is shown in Fig. 2. Overall, iDropout outperforms the baselines on all datasets, even when $dr = 0$. This observation essentially shows that iDropout has a more effective mechanism to balance information than other methods. More specifically, we figure out that methods have different behaviors on different types of datasets:

1) For two long text datasets 20NewsGroups and Grolier: $||\lambda^t||_1 = \sum_{i=1}^{T} \sum_{d \in D^t} N_d$ accumulates very fast over time, making SVB and SVB-PP become too stable with a very high rate, following Theorem 1. As a result, models have difficulties learning new information, explaining why the performance of these methods is roughly unchanged. PVB does not encounter this problem since it assumes the data stream is generated from a population distribution F_α, where the sample space is controlled by population size α, thus has a considerable evolution over time. It is also worth noting that training on these two datasets does not encounter seriously the problem of noise and sparsity, which explains why iDropout with different drop rate dr does not make a noticeable performance improvement.

2) In 4 short text datasets, there are two typical properties present in the data stream: noise and sparsity. While statistical noise potentially causes overfitting, sparsity leads to the lack of relevant information to make good predictions. Since the three baseline methods and iDropout with $dr = 0$ do not have an efficient way to tackle these serious problems, they all have poor performance over time. By contrast, iDropout with $dr \neq 0$ has a superior performance. This is achieved by two advantages of Dropout. Firstly, it plays as a data-dependent regularizer, which makes iDropout more robust against overfitting. Moreover, Dropout in our method can be regarded as an ensemble of exponentially many learners, which is a well-known solution to make better predictions for sparse data.

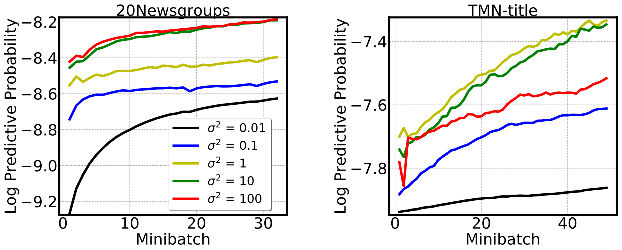


Fig. 3: Sensitivity of iDropout w.r.t variance σ^2.

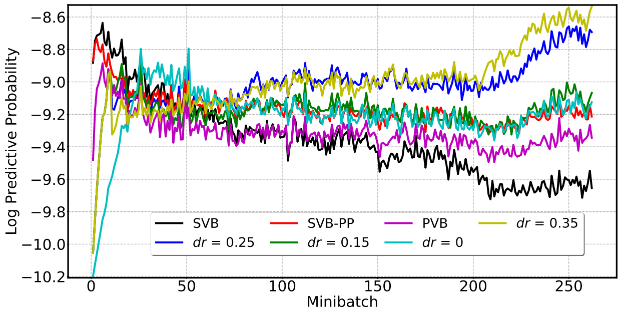


Fig. 4: Log predictive probability on The Irish Times dataset

Fig. 3 shows the sensitivity of iDropout w.r.t σ^2. 20Newsgroups (long text) and TMN-title (short text) are used for this evaluation, and the settings are the same as above. We can see that the variance affects the performance of iDropout significantly. Depending on the characteristics of data, there will be a trade-off between learning new information and keeping old information at early steps, corresponding to whether big or small values of σ^2 give better initial performance. In general, this hyperparameter needs to be tuned carefully in different datasets to obtain the best result. However, we suggest that $\sigma^2 = 100$ can be a good starting point, since this value gives fairly good performance on almost all datasets in our experiments.

C. *Experiments on datasets with time stamp*

Base model and datasets: We use the popular The Irish Times dataset[6], which is chronologically ordered to perform two different tasks: topic modeling using LDA and classification using Naive Bayes. In the LDA experiment, we simply throw away labels and use $K = 100$ and $\alpha = 0.01$. The Irish Times corpus contains 1376099 data instances from 02/01/1996 to 31/12/2017. There are 6 classes and vocab size is 25328.

Settings: Since the dataset is chronologically ordered, we divide the whole dataset into minibatches such that each minibatch D^t contains data of month t. We use documents

[3] https://cs.nyu.edu/~roweis/data.html

[4] http://acube.di.unipi.it/tmn-dataset/

[5] http://archive.ics.uci.edu/ml/datasets/Bag+of+Words

[6] https://www.kaggle.com/therohk/ireland-historical-news/

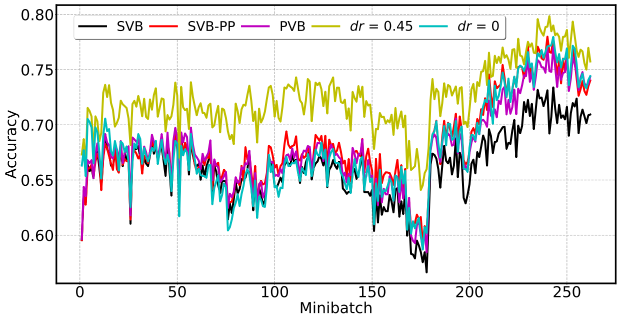


Fig. 5: Classification accuracy on The Irish Times

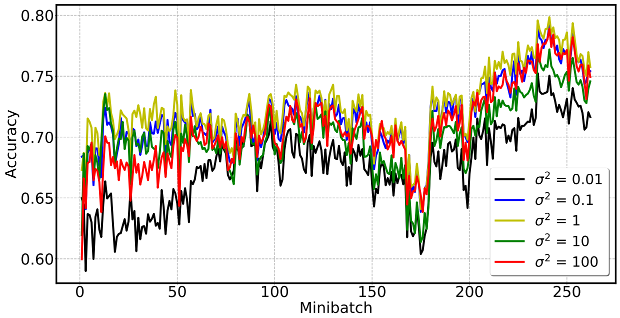


Fig. 6: Sensitivity of iDropout w.r.t variance σ^2 on the classification task

of the next minibatch (month) to evaluate the model at any minibatch.

Evaluation metric: We use LPP to evaluate the learned topic model in LDA and accuracy to evaluate the classification performance.

Result on LDA: The result is shown in Fig. 4, in which iDropout uses $\sigma^2 = 100$. It is easy to find that our framework has a significantly better performance in comparison to other streaming Bayesian learning methods. With $dr = 0$, iDropout is still slightly better than the baselines, which again demonstrates the effectiveness of the balance mechanism of iDropout. We can also see that SVB suffers from the serious overfitting problem and has the severe decline in performance later. This is explained by Theorem 1, SVB becomes too stable after receiving a large enough amount of data, which makes model not able to learn new information, therefore fail to adapt to the change of data. The Irish Times is a short-text dataset, which contains undesirable properties, esepecially noise and sparsity. Same as in the previous experiment, the three baseline methods and iDropout with $dr = 0$ encounter the overfitting problem and have a decrease in performance over time. Then, thanks to the ability to prevent overfitting and the ensemble property, Dropout helps our method to obtain better generalization and thus effectively handles negative effects of noisy and sparse data. More specific, iDropout with $dr = 0.25$ and $dr = 0.35$ has a significant improvement over the baselines. This result strengthens our argument about the efficiency of Dropout in our method.

Result on NB: Fig. 5 shows the classification performance of four methods. While iDropout with $\sigma^2 = 1$, $dr = 0.25$ achieves the highest result on nearly the whole data stream, better about $6 - 8\%$ than SVB and about $3 - 4\%$ than SVB-PP and PVB, iDropout with $dr = 0$ only has a similar performance compared to SVB-PP and PVB. This continues to strengthen our argument that Dropout plays an important role in our method. Furthermore, there is a period of time when the performance of all methods drops (about 175th minibatch) due to sudden changes in the data stream. Thanks to the balance ability and the effect of Dropout, iDropout (even with $dr = 0$) does not fall too deep and can recover quickly to keep leading on the remaining minibatches.

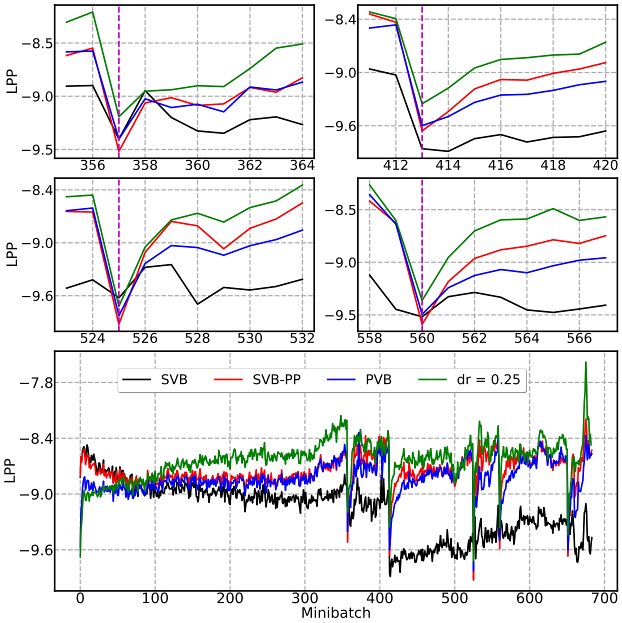


Fig. 7: Behavior on concept drift

Fig. 6 shows the impact of variance σ^2 on the performance of iDropout. More specific, we show the classification results of iDropout on The Irish Times dataset with different values of σ^2. Looking at Figure 6, we can see that $\sigma^2 = 1$ gives the best performance, better about 5% compared to $\sigma^2 = 0.01$. This a significant difference, suggesting that we need to tune this hyperparameter carefully.

D. *Evaluation on concept drift*

Concept drift [23] is the phenomenon when the underlying distribution of data changes suddenly. We conduct this experiment to examine our argument about the stability and plasticity dilemma in IV-A, especially the ability to adapt abrupt changes in the data stream.

Setting up: We simulate concept drift by using The Irish Times dataset as follow: data is divided into minibatches, each minibatch contains 2000 documents of a particular class and all minibatches of the same class are placed adjacent to each other. Therefore, the concept drift happens significantly when data transfers from one class to another. We then use LDA with $K = 100$ and $\alpha = 0.01$ to analyze these documents without information from labels. After learning on each minibatch, the model is evaluated by computing LPP on the next minibatch.

Result: The result is illustrated in Fig. 7, in which top four figures zoom in the first four drift points, i.e., where the data stream transfers from a class to another. SVB performs poorly when facing concept drift. The performance of SVB plunges after each drift point and recovers slowly due to its too much stability discussed in Theorem 1. SVB-PP can delay this problem by exponentially forgetting the information of old data, which allows it to adapt better new information from new data. PVB can also adapt to concept drift, since the variance of the variational posterior never decreases below a given threshold indirectly controlled by population size α. Finally, iDropout provides the best result (we continue to use variance $\sigma^2 = 100$). The ability to reduce overfitting and the ensemble property of Dropout allows iDropout to obtain better generalization, thus prevent the performance from falling too deeply when facing concept drift. Moreover, the balance mechanism enables iDropout to easily learn new underlying distribution of data, which incorporates with the ensemble learning to help our method adapt quickly to these new changes in data.

VI. Conclusion

We presented iDropout, a novel and straightforward framework to address many challenges of learning in streaming conditions. In particular, iDropout helps Bayesian models to tackle the stability-plasticity dilemma and handle noisy and sparse data. Further, iDropout is able to adapt quickly to abnormal changes in data streams. iDropout can be used for a wide range of models.

Acknowledgements

This research is supported by Vingroup Innovation Foundation (VINIF) in project code VINIF.2019.DA18. The first author is supported by Domestic Master Scholarship Programme code VINIF.2019.ThS.23.

Appendix

Evaluation metrics for the unsupervised task

Log Predictive Probability [14]: Predictive Probability measures the predictiveness and generalization of a model on new data. Assume that after learning from training data D_{train}, we obtain the model parameter β. For each document in testing D_{test} with more than or equal to 5 words, we divide randomly into two disjoint parts $\mathbf{w_{obs}}$ and $\mathbf{w_{ho}}$ with a ratio of 80:20. We next do inference for $\mathbf{w_{obs}}$ to estimate θ^{obs}. Then, we approximate the predictive probability $\mathbf{w_{ho}}$ as:

$$\begin{aligned} p(\mathbf{w_{ho}} \mid \mathbf{w_{obs}}, \beta) &= \prod_{w \in \mathbf{w_{ho}}} p(w \mid \mathbf{w_{obs}}, \beta) \\ &\approx \prod_{w \in \mathbf{w_{ho}}} p(w \mid \theta^{obs}, \beta) \\ &= \prod_{w \in \mathbf{w_{ho}}} \sum_{k=1}^{K} p(w \mid z = k, \beta) p(z = k \mid \theta^{obs}) \\ &= \prod_{w \in \mathbf{w_{ho}}} \sum_{k=1}^{K} \theta_k^{obs} \beta_{kw} \end{aligned}$$

Then Log Predictive Probability of each document d is:

$$LPP_d = \frac{\log p(\mathbf{w_{ho}} \mid \mathbf{w_{obs}}, \beta)}{|\mathbf{w_{ho}}|} \tag{7}$$

(with $|\mathbf{w_{ho}}|$ is the length of d in $\mathbf{w_{ho}}$) and on the whole testing D_{test} is:

$$\text{Log Predictive Probability} = \frac{\sum_{d \in D_{test}} LPP_d}{|D_{test}|} \tag{8}$$

Log Predictive Probability was averaged from 5 random splits, each was on 1000 documents.

Normalized Pointwise Mutual Information [22]: NPMI is the measure to help us see the coherence or semantic quality of individual topics. For each topic k, we pick a set $\mathbf{w^k} = \{w_1^k, w_2^k, ..., w_t^k\}$, including t words with the highest probabilities in topic distribution β_k. NPMI of one topic k is computed as follows:

$$\begin{aligned} \text{NPMI}(k, \mathbf{w^k}) &= \frac{2}{t(t-1)} \sum_{i=2}^{t} \sum_{j=1}^{i-1} \frac{\log \frac{p(w_i^k, w_j^k)}{p(w_i^k) p(w_j^k)}}{-\log p(w_i^k, w_j^k)} \\ &\approx \frac{2}{t(t-1)} \sum_{i=2}^{t} \sum_{j=1}^{i-1} \frac{\log \frac{D(w_i^k, w_j^k) + 10^{-2}}{D} - \log \frac{D(w_i^k) D(w_j^k)}{D^2}}{-\log \frac{D(w_i^k, w_j^k) + 10^{-2}}{D}} \\ &= \frac{2}{t(t-1)} \sum_{i=2}^{t} \sum_{j=1}^{i-1} -1 + \frac{2 \log D - \log D(w_i^k) - \log D(w_j^k)}{\log D - \log(D(w_i^k, w_j^k) + 10^{-2})} \end{aligned}$$

where D is the total number of documents, $D(w_i^k)$ is the number of docs containing w_i^k, $D(w_i^k, w_j^k)$ is the number of docs containing pair (w_i^k, w_j^k).

Overall, NPMI of a model with all K topics is:

$$NPMI = \frac{1}{K} \sum_{k=1}^{K} NPMI(k, t) \tag{9}$$

In the experiments, we choose $t = 20$ for each topic.

References

[1] M. Mermillod, A. Bugaiska, and P. Bonin, “The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects,” *Frontiers in psychology*, vol. 4, p. 504, 2013.

[2] T. Broderick, N. Boyd, A. Wibisono, A. C. Wilson, and M. I. Jordan, “Streaming variational bayes,” in *Advances in Neural Information Processing Systems*, pp. 1727–1735, 2013.

[3] J. McInerney, R. Ranganath, and D. Blei, “The population posterior and bayesian modeling on streams,” in *Advances in Neural Information Processing Systems*, pp. 1153–1161, 2015.

[4] A. Masegosa, T. D. Nielsen, H. Langseth, D. Ramos-López, A. Salmerón, and A. L. Madsen, “Bayesian models of data streams with hierarchical power priors,” in *International Conference on Machine Learning*, pp. 2334–2343, 2017.

[5] S. Rifai, X. Glorot, Y. Bengio, and P. Vincent, “Adding noise to the input of a model trained with a regularized objective,” *arXiv preprint arXiv:1104.3250*, 2011.

[6] S. Wager, S. Wang, and P. S. Liang, “Dropout training as adaptive regularization,” in *Advances in Neural Information Processing Systems*, pp. 351–359, 2013.

[7] S. Wang, M. Wang, S. Wager, P. Liang, and C. D. Manning, “Feature noising for log-linear structured prediction,” in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1170–1179, 2013.

[8] P. Baldi and P. Sadowski, “The dropout learning algorithm,” *Artificial intelligence*, vol. 210, pp. 78–122, 2014.

[9] D. P. Helmbold and P. M. Long, “On the inductive bias of dropout,” *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 3403–3454, 2015.

[10] P. Mianjy, R. Arora, and R. Vidal, “On the implicit bias of dropout,” in *International Conference on Machine Learning*, pp. 3537–3545, 2018.

[11] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[12] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[13] Z. Ghahramani and H. Attias, “Online variational bayesian learning,” in *Slides from talk presented at NIPS workshop on Online Learning*, 2000.

[14] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, “Stochastic variational inference,” *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[15] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv preprint arXiv:1207.0580*, 2012.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[17] N. Chen, J. Zhu, J. Chen, and B. Zhang, “Dropout training for support vector machines.,” in *AAAI*, pp. 1752–1759, 2014.

[18] S. Zhai and Z. Zhang, “Dropout training of matrix factorization and autoencoder for link prediction in sparse graphs,” in *Proceedings of the 2015 SIAM International Conference on Data Mining*, pp. 451–459, SIAM, 2015.

[19] C. Ha, V.-D. Tran, L. N. Van, and K. Than, “Eliminating overfitting of probabilistic topic models on short and noisy text: The role of dropout,” *International Journal of Approximate Reasoning*, vol. 112, pp. 85 – 104, 2019.

[20] D. P. Helmbold and P. M. Long, “Surprising properties of dropout in deep networks,” *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 7284–7311, 2017.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[22] G. Bouma, “Normalized (pointwise) mutual information in collocation extraction,” *Proceedings of GSCL*, pp. 31–40, 2009.

[23] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” *ACM computing surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014.

[24] L. Maaten, M. Chen, S. Tyree, and K. Weinberger, “Learning with marginalized corrupted features,” in *International Conference on Machine Learning*, pp. 410–418, 2013.

[25] H. Noh, T. You, J. Mun, and B. Han, “Regularizing deep neural networks by noise: Its interpretation and optimization,” in *Advances in Neural Information Processing Systems*, pp. 5109–5118, 2017.

[26] P. Baldi and P. J. Sadowski, “Understanding dropout,” in *Advances in neural information processing systems*, pp. 2814–2822, 2013.