Classifying a Stream of Infinite Concepts: A Bayesian Non-parametric Approach

Seyyed Abbas Hosseini, Hamid R. Rabiee, Hassan Hafez, and Ali Soltani-Farani

Sharif University of Technology, Tehran, Iran
{a_hosseini,hafez,a_soltani}@ce.sharif.edu
rabiee@sharif.edu

Abstract. Classifying streams of data, for instance financial transactions or emails, is an essential element in applications such as online advertising and spam or fraud detection. The data stream is often large or even unbounded; furthermore, the stream is in many instances non-stationary. Therefore, an adaptive approach is required that can manage concept drift in an online fashion. This paper presents a probabilistic non-parametric generative model for stream classification that can handle concept drift efficiently and adjust its complexity over time. Unlike recent methods, the proposed model handles concept drift by adapting data-concept association without unnecessary i.i.d. assumption among the data of a batch. This allows the model to efficiently classify data using fewer and simpler base classifiers. Moreover, an online algorithm for making inference on the proposed non-conjugate time-dependent non-parametric model is proposed. Extensive experimental results on several stream datasets demonstrate the effectiveness of the proposed model.

Keywords: Stream classification, Concept drift, Bayesian non-parametric, Online inference.

1 Introduction

The emergence of applications such as spam detection [29] and online advertising [1, 23] coupled with the dramatic growth of user-generated content [7, 35] has attracted more and more attention to stream classification.The data stream in such applications is large or even unbounded; moreover, the system is often required to respond in an online manner. Due to these constraints, a common scenario is usually used in stream classification: At each instant, a batch of data arrives at the system. The system is required to process the data and predict their labels before the next batch comes in. It is assumed that after prediction, the true labels of the data are revealed to the system. Also due to limited additional memory the system can only access one previous batch of data and their labels. For example, in online advertising, at each instant, a large number of requests arrive and the system is required to predict for each ad, the probability that

T. Calders et al. (Eds.): ECML PKDD 2014, Part I, LNCS 8724, pp. 1–16, 2014.

it will be clicked by each user. After a short time, the result is revealed to the system and the system can use it to adapt the model parameters.

One of the main challenges of stream classification is that often the process that generates the data is non-stationary. This phenomenon, known as concept drift, poses different challenges to the classification problem. For example, in a stationary classification task, one can model the underlying distribution of data and improve estimates of model parameters as more data become available; but this is not the case in a non-stationary environment. If we can not model the change of the underlying distribution of data, more data may even reduce the model's efficiency. Formally, concept drift between time t_1 and t_2 occurs when the posterior probability of an instance changes, that is [19]:

$$\exists x : p_{t_1}(y|x) \neq p_{t_2}(y|x) \tag{1}$$

When modeling change in the underlying distribution of data, a common assumption is that the data is generated by different sources and the underlying distribution of each source, which is called its concept, is constant over time [19]. If the classification algorithm can find the correct source of each data item, then the problem reduces to an online classification task with stationary distribution, because each concept can be modeled separately. While the main focus in classification literature is on stationary problems, recent methods have been introduced for classification in non-stationary environments [19]. However, these algorithms are often restricted to simple scenarios such as finite concepts, slow concept drift, or non-cyclical environment [16]. Furthermore, usually heuristic rules are applied to update the models and classifiers, which may cause overfitting.

Existing stream classification methods belong to one of two main categories. Uni-model methods use only one classifier to classify incoming data and hence need a forgetting mechanism to mitigate the effect of data that are not relevant to the current concept. These methods use two main approaches to handle concept drift: sample selection and sample weighting [30]. Sample selection methods keep a sliding window over the incoming data and only consider the most recent data that are relevant to the current concept. One of the challenges in these methods is determining the size of the window, since a very large window may cause non-relevant data to be included in the model and a small window may decrease the efficiency of the model by preventing the model from using all relevant data. Sample weighting methods weigh samples so that more recent data have more impact on the classifier parameters [13, 38]. In contrast to uni-model methods, ensemble methods keep a pool of classifiers and classify data either by choosing an appropriate classifier from the pool (model selection) or combining the answers of the classifiers (model combination) to find the correct label [31]. Inspired by the ability of ensemble methods to model different concepts, these models have been used in stream classification with encouraging results [16, 27, 29, 34]. The main problem is that many of these models update the pool of classifiers heuristically and hence may overfit to the data. Moreover, a common assumption among all existing ensemble methods is that all data of a batch are i.i.d. samples of a distribution that are generated from the same source and

hence have the same concept. This assumption may cause several problems. For example, since the data of a batch are not necessarily from the same source or may even belong to conflicting concepts, we may not be able to classify them with high accuracy even using complex base classifiers. Moreover, since the diversity of batches of data can be very high, the number of needed base classifiers may become very large.

In this paper, we propose a principled probabilistic framework for stream classification that is impervious to the aforementioned issues and is able to adapt the complexity of the model to the data over time. The idea is to model the data stream using a non-parametric generative model in which each concept is modeled by an incremental probabilistic classifier and concepts can emerge or die over time. Moreover, instead of the restrictive i.i.d. assumption among data of a batch, we assume that the data of a batch are exchangeable which is a much weaker assumption (refer to Section 3.1 for a detailed definition). This is realized by modeling each concept with an incremental probabilistic classifier and using the temporal Dirichlet process mixture model (TDPM) [3]. For inference, we propose a variation of forward Gibbs sampling.

To summarize, we make the following main contributions: (i) We propose a coherent generative model for stream classification. (ii) The model manages its complexity by adapting the size of the latent space and the number of classifiers over time. (iii) The proposed model handles concept drift by adapting data-concept association without unnecessary i.i.d. assumption among data of a batch. (iv) An online algorithm is proposed for inference on the non-conjugate non-parametric time-dependent model.

The remainder of this paper is organized as follows: Section 2 briefly discusses the prior art on this subject. The details of the proposed generative model are discussed in Section 3. To demonstrate the effectiveness of the proposed model, extensive experimental results on several stream datasets are reported and analyzed in Section 4. Finally, Section 5 concludes this paper and discusses paths for future research.

2 Review on Prior Art

Stream classification methods can be categorized based on different criteria. As is mentioned in [19], based on how concept drift is handled the different strategies can be categorized into informed adaptation and blind adaptation. In informed adaptation-based models, there is a separate building block that detects the drift allowing the system to act according to these triggers [8, 22]. However, blind adaptation models adapt the model without any explicit detection of concept drift [19]. In this paper, the focus is on blind adaptation.

Chu et al. proposed a probabilistic uni-model method for stream classification in [13] that uses sample weighting to handle concept drift. This method, uses a probit regression model as a classifier and adaptive density filtering (ADF) [32] to make inference on the model and update it. Probit regression, is a linear classifier with parameter w and prior distribution $N(w; \mu_0, \Sigma_0)$. After observing

each new data, the posterior of w is updated and approximated by a Gaussian distribution, that is:

$$w_t \sim N(w_t; \mu_t, \Sigma_t) \tag{2}$$

$$p(y_t|x_t, w_t) = \Phi(y_t w_t^T x_t) \tag{3}$$

$$p(w_{t+1}|x_t, y_t) \propto \Phi(y_t w_t^T x_t) N(w_t; \mu_t, \Sigma_t) \tag{4}$$

In order to decrease the effect of old data, this method introduces a memory loss factor and incorporates the prior of w_t with this factor in computing the posterior of w, that is:

$$p(w_{t+1}|x_t, y_t) \propto \Phi(y_t w_t^T x_t) N(w_t; \mu_t, \Sigma_t)^{\gamma} \quad 0 \ll \gamma < 1 \tag{5}$$

Using this method, the effect of out-of-date data is reduced as new data arrives into the system. As it is evident in (5), this method forgets old data gradually and hence can not handle abrupt changes in the distribution of data. On the other hand, since sample selection methods only consider the selected data, they easily can handle abrupt drift but they miss the information in the old data that are relevant to the current concept.

As mentioned in Section 1, ensemble methods can be categorized into model selection and model combination methods. Model combination methods assume that each data item is generated by a linear combination of base classifiers and thereby enrich the hypothesis space [15]. There have been different methods for stream classification based on model combination [16, 34]. These methods maintain a pool of classifiers and estimate the label of each datum of batch t by combining base classifiers using:

$$\hat{y}_i^t = \arg\max_c \sum_k W_k^t I_{[h_k(x_i^t)=c]} \tag{6}$$

where W_k^t is the weight of base classifier k for batch t, which is an estimate of its accuracy relative to other classifiers. After observing the true labels of a batch, these methods update the model by adding new classifiers or removing inefficient classifiers, or changing the weights of classifiers.

The main idea of model selection methods, is to find the concept of each data item, hence reducing the problem to an online classification task. The challenge is that finding the concepts of the data is an unsupervised task. There have been different methods to tackle this issue. The simplifying assumption that is common among almost all of these methods is that all of the data of a batch are i.i.d. and hence generated from one concept. For example, [29] uses this assumption and extracts some feature from each batch and finds their concept by clustering the extracted feature vectors. This method assumes that all of the batches that lie in the same cluster, can be classified using a single classifier. This assumption may not be true in many applications, which may decrease the efficiency. Since this method finds the concepts of data by clustering the features that are extracted from the whole batch and the diversity of batches may be very high, the number of clusters may become very large and hence the

model can become very complex. Hosseini et al. proposed an improved version of this method in [27]. This method introduces a new distance metric in the feature space together with pool management operations such as splitting or merging of concepts.

There is some prior work on classification using Dirichlet process mixture models [14,24,36]. All of these methods have been designed for classifying batch data and can't be applied to stream classification due to two main reasons. First, these methods does not model the temporal dependency among data and second, the inference algorithm in these methods is offline which is not suitable for stream classification. In order to solve these two issues, we proposed a time-dependent non-parametric generative model and an online inference algorithm based on forward collapsed Gibbs sampling which is an online version of Gibbs sampling [1].

3 Proposed Method

In this section, we introduce our proposed method for stream classification. In order to model concept drift, we propose a non-parametric mixture model with potentially infinite number of mixtures, in which each mixture represents a concept. This model uses a Bayesian model selection approach [31] and assumes that each data item is generated by one concept. Each concept is modeled with a generative classifier. In order to model the change in popularity of concepts over time and their emergence and death, we use TDPM. After observing the true labels of a batch, this model allows the number of concepts and the data-concept associations to be determined by inference, for which we propose an online inference algorithm based on Gibbs sampling.

For clarity, we define the problem setting and notations in Section 3.1. To make the presentation self-sufficient, TDPM is reviewed in Section 3.2. The proposed generative model is described in Section 3.3, followed by the details of the inference algorithm in Section 3.4.

3.1 Problem Setting and Notation

Consider a stream classification problem in which each data item consists of an $l-tuple$ feature vector and a label that associates it to one of C predefined classes. In this setting, data arrive in consecutive batches. The system is required to predict the labels for each batch, after which the true labels are revealed. Moreover due to limited memory, the system only has access to one previous batch of data. In summery, our goal is to classify a stream of data $(D_1, \ldots, D_T)$, where T denotes the number of batches and D_t is the batch of data that arrives at time t. Also, $D_t = (d_{ti})_{i=1}^{n_t}$ where d_{ti} is the ith data item in batch t and n_t is the number of data in that batch. Furthermore, each data item is denoted by (x_{ti}, y_{ti}), where x_{ti} is an $l-tuple$ vector with l_1 discrete features $x_{ti}^{1,\ldots,l_1}$ and l_2 continuous features $x_{ti}^{l_1+1,\ldots,l}$ and $y \in \{1, \ldots, C\}$.

For a general variable z, z_t denotes the set of all z values of batch t and $z_{t,i:j}$ denotes the corresponding z values of data i to j in batch t. Moreover, by z_t^{-i}, we mean all z values of batch t except the i'th one.

3.2 Temporal Dirichlet Process Mixture Model

Suppose that we assume that the data $(x_1, \ldots, x_N)$ are infinitely exchangeable, that is, the joint probability distribution underlying the data is invariant to permutation, then according to De Finetti theorem [26], the joint probability $p(x_1, \ldots, x_N)$ has a representation as a mixture:

$$p(x_1, \ldots, x_N) = \int \left(\prod_{i=1}^{N} p(x_i|G) \right) dP(G) \tag{7}$$

for some random variable G. The theorem needs G to range over measures, in which case $P(G)$ is a measure on measures. The Dirichlet process denoted by $DP(G_0, \alpha)$ is a measure on measures with base measure G_0 and concentration parameter α [17], and hence can be used to model exchangeable data . We write $G \sim DP(G_0, \alpha)$ if G is drawn from a Dirichlet process in which case G itself is a measure on the given parameter space θ. Integrating out G, the parameters θ follow the Chinese Restaurant Process (CRP) [9], in which the probability of redrawing a previously drawn value of θ is strictly positive which makes G a discrete probability measure with probability one; that is:

$$p(\theta_i|\theta_{1:i-1}) = \sum_k \frac{m_k}{i-1+\alpha}\delta(\phi_k) + \frac{\alpha}{i-1+\alpha}G_0 \tag{8}$$

where ϕ_ks are unique θ values and m_k is the number of θ_is having value ϕ_k. The CRP metaphor explains (8) clearly. In this metaphor, there is a Chinese restaurant with infinite number of tables. When a new customer x_i comes into the restaurant, she either sits on one of the previously occupied tables ϕ_k with probability $\frac{m_k}{i-1+\alpha}$ or sits on a new table with probability $\frac{\alpha}{i-1+\alpha}$. Using the Dirichlet process as the prior distribution of a hierarchical model, one obtains the Dirichlet process mixture model (DPM) [6]. As is evident from (8), DPM assumes the data are exchangeable. Since there are temporal dependencies among data in a stream, DPM is not appropriate for modeling.

There are several methods to incorporate temporal dependency in DPM [1,3, 11,37]. In this paper, we focus on TDPM introduced in [3], and use a variation of that in our proposed model for stream classification. TDPM assumes that the stream of data arrives in consecutive batches. Moreover, this model assumes that data are partially exchangeable, i.e., the data that belong to one batch are exchangeable but exchangeability does not hold among batches. A sample G_t drawn from $TDP(G_0, \alpha, \lambda, \Delta)$ is a time dependent probability measure over the parameter space θ:

$$G_t|\phi_{1:k}, G_0, \alpha \sim DP\left(\sum_k \frac{m'_{kt}}{\sum_l m'_{lt} + \alpha}\delta(\phi_k) + \frac{\alpha}{\sum_l m'_{lt} + \alpha}G_0, \alpha + \sum_k m'_{kt} \right) \tag{9}$$

where $\phi_{1:k}$ are the set of unique θ_i values used in recent Δ batches and m'_{kt} is the weighted number of θ_is having value ϕ_k. More formally, if m_{kt} denotes the number of θs in batch t with value ϕ_k, then:

$$m'_{kt} = \sum_{\tau=1}^{\Delta} e^{-\frac{\tau}{\lambda}} m_{k,t-\tau} \tag{10}$$

As it is evident from Eq. (9), the data in each batch are modeled by a DP and hence it is assumed that they are exchangeable. However, the parameters of these processes evolve over time and are dependent. By marginalizing over G_t, the parameters θ follow the Recurrent Chinese Restaurant Process (RCRP) introduced in [3]:

$$\theta_{t,i}|\theta_{t-\Delta}, \theta_t^{-i}, \alpha, G_0 \propto \sum_k (m'_{kt} + m_{kt})\delta(\phi_k) + \alpha G_0 \tag{11}$$

According to (11), when customer x_i comes to the restaurant, the probability of choosing table ϕ_k is proportional to the weighted number of customers in previous Δ batches that chose that table and the number of customers in the current batch that chose the same table. In fact, RCRP in (10), uses sample selection and sample weighting to model the evolution of the probability distribution over parameters. Moreover, this process assumes that the data in a batch are only exchangeable and doesn't force them to select the same mixture. Therefore, we use this process as the prior over the parameters of a classifier which uses model selection.

3.3 Infinite Concept Stream Classifier

In this section, we introduce our generative model for stream classification. In order to model concept drift, we propose a Bayesian model selection method [31]. Figure 1 depicts the graphical representation of the proposed generative model. In this graph, observed variables are depicted using shaded nodes and blank nodes represent latent variables. Moreover, arrows are used to represent the dependency among random variables. The plate structure is used to act as a for loop to represent repetition. As it can be seen, there are in total T batches of data, in which batch t, contains n_t data. Each data item consists of a feature vector x which is an observed variable and a latent variable label y which will be revealed to the system after it is estimated. In this model, there are potentially infinite number of concepts. Each concept is in fact a classifier with parameter set ϕ. Moreover, since it is based on model selection, it assumes that each (x_{ti}, y_{ti}) is generated by a single concept, where z_{ti} is the concept indicator. More formally, if data (x_{ti}, y_{ti}) is generated by a classifier with parameters θ_{ti}, then $\theta_{ti} = \phi_{z_{ti}}$. Since the size of data is very large in data streams, it is not possible to keep them all. Therefore, we need an incremental model for the base classifiers. The classifier model that we used in our model is a naive Bayes classifier. In this model, each continuous feature in each class is modeled by a Gaussian distribution and each discrete feature is modeled by a categorical distribution. In order to model the

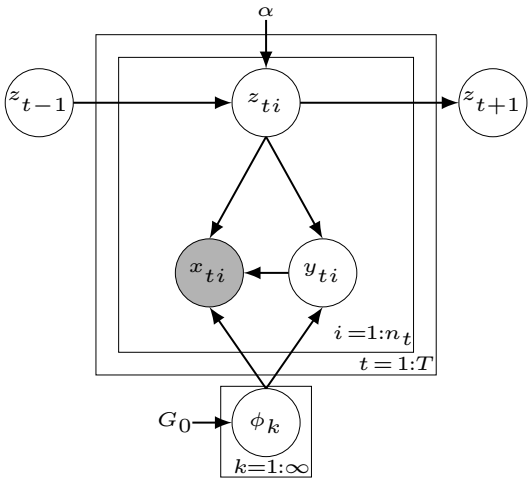

Fig. 1. The Graphical Model of the Proposed Method

temporal dependency among the data of the stream, we used $RCRP(G_0, \alpha, \lambda, \Delta)$ as the prior over concept indicators. The generative process of this model is described in Algorithm 1. According to this process, in order to generate the ith element of batch t, one may either choose one of the existing classifiers that have generated at least one data item in last Δ batches, or a new classifier. The probability of choosing each classifier is determined by RCRP. Furthermore, if a new classifier is needed to generate d_{ti}, then the parameters of the classifiers are obtained by drawing a sample from G_0. In order to make inference easier, we selected G_0 conjugate to the classifier's likelihoods. That is, Dirichlet distribution is used for the prior over class prior probabilities as well as the distribution of discrete features in each class, and Gaussian-Gamma distribution is used for the prior over continuous features in each class.

Indeed, this model assumes that the amount of activity of classifier k at batch t is proportional to the weighted number of data that this model has classified in the last Δ batches and hence uses sample selection and sample weighting concurrently to handle concept drift. Moreover, this model allows the data of a batch to select different concepts. This assumption increases the efficiency of the model in applications where the data of a batch are not necessarily i.i.d. Furthermore, unlike most existent ensemble methods that set the number and the weights of classifiers using heuristic rules, the number of classifiers and their corresponding weights are determined consistently in this method through Bayesian inference on the proposed model. The details of the inference algorithm are discussed next.

3.4 Inference

When a new batch of data arrives at the system, we need to find their labels by estimating the posterior probability of labels given all previously observed data, that is:

Algorithm 1. The proposed Generative Model

for all batch $t \in \{1, 2, \ldots, T\}$ **do**
  for all data $i \in \{1, \ldots, n_t\}$ **do**
    Draw $z_{t,i}|z_{1:t-1}, z_t^{-i} \sim RCRP(\alpha, \lambda, \Delta)$
    if $z_{t,i}$ is a new concept **then**
      Draw $\beta_{znew}|G_0 \sim Dir(\pi)$
      for all $c \in \{1, \ldots, C\}$ **do**
        for all $j \in \{1, \ldots, m_1\}$ **do**
          Draw $\rho_{znew,j,c}|G_0 \sim Dir(\gamma_{j,c})$
        for all $j \in \{1, \ldots, m_2\}$ **do**
          Draw $\lambda_{znew,j,c}|G_0 \sim Gam(a_{j,c}, b_{j,c})$
          Draw $\mu_{znew,j,c}|G_0 \sim N(\eta_{j,c}, (\nu_{j,c}\lambda_{znew,j,c})^{-1})$
    Draw $y_{t,i} \sim Cat(y_{t,i}; \beta_{z_{t,i}})$
    Draw $x_{t,i}^{1,\ldots,l_1}|y_{t,i} \sim \prod_{j=1}^{l_1} Cat(x_{t,i}^{j}; \rho_{z_{t,i},j,y_{t,i}})$
    Draw $x_{t,i}^{l_1+1,\ldots,l}|y_{t,i} \sim \prod_{j=1}^{l_2} N(x_{t,i}^{j+l_1}; \mu_{z_{t,i},j,y_{t,i}}, \lambda^{-1}_{z_{t,i},j,y_{t,i}})$

$$p(y_t|x_t, x_{1:t-1}, y_{1:t-1}, G_0, \alpha, \lambda, \Delta) \tag{12}$$

This can be done by marginalizing over concept indicators $z_{1:t}$ and concepts' parameters ϕ_ks. However, since the posterior of TDPM is not available in closed form, we need an algorithm to approximate it. Moreover, in stream classification, the algorithm only has access to one previous batch of data and hence, the inference algorithm must be online. Therefore, we approximate the posterior (12) in two phases. First, after observing the true labels of batch $t-1$, we update the model accordingly and then, after batch t is available, we approximate (12) using the updated model.

Several approximate algorithms have been introduced for inference on DPM models [21]. These methods either use Markov Chain Monte Carlo (MCMC) sampling methods [3,33] or Variational methods [10,28] to estimate the posterior distribution of desired latent variables. Moreover, online inference algorithms have been proposed for making inference on TDPM which are based on sequential Monte Carlo estimation [2] or Gibbs sampling [1]. The proposed algorithm for making inference on the proposed model is a variation of forward Gibbs sampling [1] which we explain next.

Generally, the main idea of MCMC estimation is to design a Markov Chain over the desired latent variables in which the equilibrium distribution of the Markov chain is the posterior of the variables [5]. By drawing samples from this Markov chain, one can obtain samples from the posterior of the desired random variables. Gibbs sampling is a widely used variation of MCMC sampling. If $p(z_{1:m})$ is the distribution that we want to draw samples from, then Gibbs chooses an initial value for $z_{1:m}$ and in each iteration, chooses one of the random variables z_i and replaces its value by the value drawn from $p(z_i|z_{1:m}^{-i})$. This process is repeated by iterating over z_is [20]. Gibbs sampling can not be applied to online applications such as stream classification where there is temporal dependency among latent variables. The reason is that in these models, the system

doesn't have access to old data and hence iterating over all latent variables is not practical.

After observing the labels of batch t, the set of latent variables in our model is $z_{1:t}$ and ϕ_ks. In order to use Gibbs sampling to draw samples from the posterior of these variables, it is necessary to access all previous batches of data. In order to solve this issue, we use forward Gibbs sampling, an online variation of Gibbs sampling [1]. In this method, at each time step t, we estimate the posterior of new random variables z_t using the estimates of concept indicators in previous batches. In order to do so, we run batch Gibbs sampling over newly added random variables given the state of the sampler in the last batch. In fact, in this method, the value of latent variables that are set in previous batches is no longer changed and the dependency of these variables on future data is not considered. Although this causes suboptimal estimates for initial batches, the estimates will improve over time.

Formally, for inference on the proposed model, we use a collapsed Gibbs sampler, the state of which at time t is $z_{1:t}$. In order to draw a sample at time t, we collapse the concepts' parameters ϕ_ks and compute the posterior of $z_{t,i}$ given the values assigned to $z_{1:t-1}$ in previous batches. To compute this conditional distribution, we use the exchangeability among data of a batch and assume that data i is the last data of the batch. Moreover, using the independency relations among random variables, which can be inferred from the graphical model in Fig. 1, we have:

$$\begin{aligned} p(z_{ti} = k | z_{1:t}^{-(ti)}, D_{1:t}, G_0, \alpha, \lambda, \Delta) \propto \\ p(z_{ti} = k | z_{1:t}^{-(ti)}, G_0, \alpha, \lambda, \Delta)\ p(d_{ti} | z_{ti} = k, z_{1:t}^{-(ti)}, d_{1:t}^{-(ti)}) \end{aligned} \tag{13}$$

According to Algorithm 1, the prior over z_{ti} obeys $RCRP(G_0, \alpha, \lambda, \Delta)$, i.e.:

$$p(z_{t,i} = k | z_{1:t}^{-(t,i)}, G_0, \alpha, \lambda, \Delta) \propto \begin{cases} m'_{k,t} + m_{k,t}, & \text{if } k \in I_{t-\Delta:t} \\ \alpha, & \text{if } k \text{ is a new concept} \end{cases} \tag{14}$$

where $I_{t-\Delta:t}$ is the set of all concept indices that generated at least one data in the last Δ batches. Since we chose G_0 conjugate to the likelihood functions of the base classifiers, the second term in (13) can be analytically computed by marginalizing over ϕ_k; that is:

$$p(d_{t,i} | z_{t,i} = k, z_{1:t}^{-(t,i)}, d_{1:t}^{-(t,i)}) = \int p(d_{t,i} | \phi_k) p(\phi_k | \{d_{\tau,j} : z_{\tau,j} = k\}) d\phi_k \tag{15}$$

where $p(\phi_k | \{d_{\tau,j} : z_{\tau,j} = k\})$ is the posterior of the parameters of classifier k given all data that was generated by this classifier. Since the base classifier is a naive Bayes classifier with normal likelihood for continuous features and categorical likelihood for discrete features and due to the conjugacy relationship between G_0 and classifier likelihoods, the posterior of the parameters of these classifiers can be easily computed using a few sufficient statistics.

When a new batch of data arrives to be classified, we find the labels by approximating their posteriors by:

$$p\ (y_{t+1,i}|x_{t+1}, d_{1:t}, z_{1:t}) \simeq p(y_{t+1,i}|x_{t+1,i}, d_{1:t}, z_{1:t}) \tag{16}$$

$$= \sum_k p(y_{t+1,i}|x_{t+1,i}, z_{t+1,i} = k, z_{1:t}, d_{1:t}) p(z_{t+1,i} = k|x_{t+1,i}, z_{1:t}, d_{1:t}) \tag{17}$$

In this approximation, we have discarded the information that x_{t+1}^{-i} may have about $z_{t+1,i}$. The first term of (17) can be calculated similar to (15) and the second term is calculated by:

$$p(z_{t+1,i}|x_{t+1,i}, z_{1:t}, d_{1:t}) \propto p(x_{t+1,i}|z_{t+1,i}, z_{1:t}, d_{1:t}) p(z_{t+1,i}|z_{1:t}) \tag{18}$$

where the first term is calculated using (15) and marginalizing over $y_{t+1,i}$.

4 Experimental Results

In this section, we provide experimental results and analysis regarding application of the proposed non-parametric generative model on real stream classification datasets, known as spam [29], weather [16], and electricity [25][1].

The spam dataset consists of 9324 emails extracted from the Spam Assassin Collection. Each email is represented by 500 binary features which indicate the existence of words derived using feature selection. The ratio of spam messages is approximately 20%, hence the classification problem is imbalanced. The emails are sorted according to their arrival date in to batches of 50 emails.

The weather dataset consists of 18,159 daily readings including features such as temperature, pressure, and wind speed. The data were collected by The U.S. National Oceanic and Atmospheric Administration from 1949 to 1999 in the Offutt Air Force Base in Bellevue, Nebraska which has diverse weather patterns making it a suitable dataset for evaluating concept drift. We use the same eight features as [16]. The samples belong to one of two classes: "rain" with 5698 (31%) and "no rain" with 12461 (69%) samples, and are sorted into 606 30-day batches. The model must predict the weather forecast for 30 days, after which the true forecast is revealed.

The electricity dataset consists of 45,312 samples from the Australian New South Wales Electricity Market. Each sample is described by 4 attributes, namely time stamp, day of the week, and 2 electricity demand values. The data was collected from May 1996 to December 1998, during which period the prices vary due to changes in demand and supply. The samples were taken every 30 minutes and sorted into batches of 48 samples each. The target is to predict whether the prices related to a moving average of the last 24 hours, increase or decrease.

In order to compare different stream classification methods on the above datasets, we use two well known measures, namely accuracy defined as the ratio of samples classified correctly and the κ coefficient [12] which is a robust measure of agreement that corrects for random classification. Furthermore, in order

[1] All codes for the experiments are available at http://ml.dml.ir/research/npsc

to evaluate accuracy over time, we use prequential accuracy with fading factor $\alpha = 0.95$ defined as [18]:

$$A_\alpha(t) = \frac{\sum_{\tau=1}^{t} \alpha^{t-\tau} a(\tau)}{\sum_{\tau=1}^{t} \alpha^{t-\tau}} \tag{19}$$

where $a(\tau)$ is the ratio of correctly classified samples in batch τ. The reason for choosing this measure is two fold: First, at time t, all previous accuracies contribute to $A_\alpha(t)$, providing an overall picture for evaluation. Second, the forgetting factor α mitigates the impact of older accuracies allowing us to observe how well the algorithm responds to concept drift.

We compare the proposed Non-Parametric Stream Classifier (NPSC) with Naive Bayes (NB) and Probit [13] as uni-model methods and with Conceptual Clustering and Prediction (CCP) [29] and Pool Management base Recurring Concepts Detection (PMRCD) [27] as state-of-the-art model selection algorithms that attempt to handle concept drift. The results are depicted in Table 1 and Fig. 2.

The parameters of Probit, CCP and PMRCD are set according to their corresponding publications. The proposed method has hyper-parameters that need to be set, namely $(G_0, \alpha, \lambda, \Delta)$. The baseline distribution G_0 can be treated as the expected distribution G_t, which is the prior distribution over the parameters of the base classifiers at time t. According to [37], it is unrealistic to assume that this parameter is constant over time. Therefore, we learn this parameter by training a single naive Bayes classifier on all observed data until time t. The precision parameter α, controls how much G_t can deviate from the baseline distribution G_0. Moreover, this parameter controls how often new classifiers emerge.This parameter was set equal to the batch size of each dataset. The parameter λ is the forgetting factor which determines how fast the effect of old data is mitigated. This parameter was set to 0.4 for all datasets. The parameter Δ can be safely set to any large value for which $e^{-\frac{\Delta}{\lambda}}$ is sufficiently small [4]; to incur less computation cost, we set Δ to 30 for all datasets.

The results show that although Probit is a uni-model method, it provides better results than CCP and PMRCD on the spam dataset. The reason is that CCP and PMRCD assume that all data in a batch belong to the same concept. This assumption coupled with the fact that initial batches in the dataset consist of data from a single class, causes their classifiers to overfit to a single class. Later batches in this dataset consist of data from different classes, which the classifiers of CCP and PMRCD can not classify correctly. We have observed that CCP and PMRCD tend to classify each batch with an accuracy similar to the ratio of the majority label. The single classifier of Probit can better handle this situation because it observes all the data and forgets older data gradually. NPSC Provides the best accuracy on this dataset, because it can use multiple classifiers without the unrealistic assumption that all data in a batch belong to the same concept.

The weather dataset exhibits recurrent and gradual concept drift, for which modeling with a finite number of concepts may be sufficient, but the assumption

Table 1. Classification Accuracy (%) And κ Measure For Different Classifiers For Different Methods

		NB	Probit	CCP	PMRCD	NPSC
Spam	Accuracy	90.7	92.4	91.6	89.7	94.5
	κ	0.8	0.8	0.76	0.7	0.85
Weather	Accuracy	73.8	73	73.2	73.0	75.5
	κ	0.31	0.41	0.37	0.32	0.41
Electricity	Accuracy	62.4	62.4	66.5	69.9	69.8
	κ	0.23	0.20	0.30	0.38	0.38

that all days in a month (one batch) belong to the same concept is still unrealistic. That may be the reason for the better performance of NPSC on this dataset (Table 1). According to Fig. 2, the ensemble methods (CCP, PMRCD, and NPSC) handle concept drift better than uni-model methods (NB and Probit), but it is hard to distinguish which performs better. This was expected due to the recurrent nature of weather which can be modeled by a finite number of concepts without the need for a complex management scheme for the pool of classifiers.

Finally, the results show that uni-model methods (NB and Probit) perform poorly on the Electricity dataset. The reason is that this dataset exhibits complex concept drift, due to the complex nature of demand and supply. Ensemble methods (CCP, PMRCD, and NPSC) perform better, because they can handle multiple concepts. On the other hand, CCP lacks a management scheme for the pool of classifiers which explains its poor performance in comparison to PMRCD and NPSC. Moreover, since each batch of data corresponds to a single day, the assumption that data in a batch belong to the same concept is not unrealistic. This explains the similar performance of PMRCD and NPSC.

5 Conclusions and Future Works

In this paper, we addressed the problem of stream classification and introduced a probabilistic framework. The proposed method handles concept drift using a non-parametric temporal model that builds a model selection based classifier via a mixture model with potentially infinite mixtures. This method finds the number of concepts and the data-concept association through inference on the proposed model. In order to make inference on the proposed model, we introduced an online algorithm which is based on Gibbs sampling.

Several directions of future research are possible. The proposed method yields accurate results using simple naive Bayes classifier. As it was mentioned in Section 4, more complex classifiers such as probit may provide better results. The challenge is that there are no conjugate priors for probit's parameters and hence, it may be necessary to use some approximate inference algorithms such as Expectation Propagation (EP) in each iteration of Gibbs sampling. Another direction is to use model combination instead of model selection. The assumption that each data is generated by one classifier may be a constraining assumption and

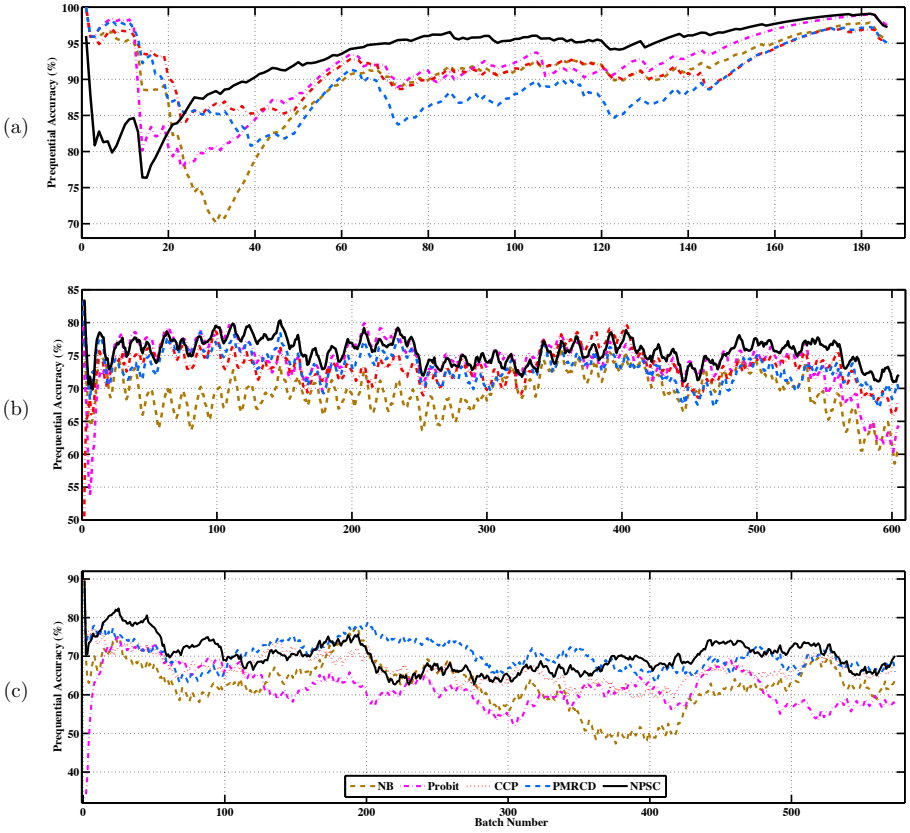

Fig. 2. Classification results of classifiers on (a) Spam, (b) Weather, and (c) Electricity

since model combination methods enrich the hypothesis space by combining different classifiers, they may increase the efficiency of the model. Furthermore, sampling based inference methods are non-deterministic and their convergence can not be verified easily. A direction we are currently pursuing is to develop an online variational inference algorithm based on the idea proposed in [28].

References

1. Ahmed, A., Low, Y., Aly, M., Josifovski, V., Smola, A.J.: Scalable distributed inference of dynamic user interests for behavioral targeting. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 114–122. ACM (2011)
2. Ahmed, A., Ho, Q., Eisenstein, J., Xing, E., Smola, A.J., Teo, C.H.: Unified analysis of streaming news. In: Proceedings of the 20th International Conference on World Wide Web, pp. 267–276. ACM (2011)

3. Ahmed, A., Xing, E.P.: Dynamic Non-Parametric Mixture Models and the Recurrent Chinese Restaurant Process: with Applications to Evolutionary Clustering. In: SDM, pp. 219–230 (2008)
4. Ahmed, A., Xing, E.P.: Timeline: A dynamic hierarchical Dirichlet process model for recovering birth/death and evolution of topics in text stream. arXiv preprint arXiv:1203.3463 (2012)
5. Andrieu, C., De Freitas, N., Doucet, A., Jordan, M.I.: An introduction to MCMC for machine learning. Machine Learning 50(1-2), 5–43 (2003)
6. Antoniak, C.E.: Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. The Annals of Statistics 2(6), 1152–1174 (1974)
7. Bifet, A., Frank, E.: Sentiment knowledge discovery in twitter streaming data. In: Pfahringer, B., Holmes, G., Hoffmann, A. (eds.) DS 2010. LNCS, vol. 6332, pp. 1–15. Springer, Heidelberg (2010)
8. Bifet, A., Pfahringer, B., Read, J., Holmes, G.: Efficient data stream classification via probabilistic adaptive windows. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, pp. 801–806. ACM (2013)
9. Blackwell, D., MacQueen, J.B.: Ferguson distributions via Plya urn schemes. The Annals of Statistics, 353–355 (1973)
10. Blei, D.M., Jordan, M.I.: Variational inference for Dirichlet process mixtures. Bayesian Analysis 1(1), 121–143 (2006)
11. Blei, D.M., Frazier, P.I.: Distance dependent Chinese restaurant processes. The Journal of Machine Learning Research 12, 2461–2488 (2011)
12. Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement 20(1), 37–46 (1960)
13. Chu, W., Zinkevich, M., Li, L., Thomas, A., Tseng, B.: Unbiased online active learning in data streams. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 195–203. ACM (2011)
14. Davy, M., Tourneret, J.Y.: Generative supervised classification using dirichlet process priors. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(10), 1781–1794 (2010)
15. Domingos, P.: Why Does Bagging Work? A Bayesian Account and its Implications. In: KDD, pp. 155–158 (1997)
16. Elwell, R., Polikar, R.: Incremental learning of concept drift in nonstationary environments. IEEE Transactions on Neural Networks 22(10), 1517–1531 (2011)
17. Ferguson, T.S.: A Bayesian analysis of some nonparametric problems. The Annals of Statistics, 209–230 (1973)
18. Gama, J., Sebastio, R., Rodrigues, P.P.: On evaluating stream learning algorithms. Machine Learning 90(3), 317–346 (2013)
19. Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A Survey on Concept Drift Adaptation. ACM Computing Surveys 46(4) (2014)
20. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence (6), 721–741 (1984)
21. Gershman, S.J., Blei, D.M.: A tutorial on Bayesian nonparametric models. Journal of Mathematical Psychology 56(1), 1–12 (2012)
22. Gomes, J.B., Menasalvas, E., Sousa, P.A.: Learning recurring concepts from data streams with a context-aware ensemble. In: Proceedings of the 2011 ACM Symposium on Applied Computing, pp. 994–999. ACM (2011)

23. Graepel, T., Candela, J.Q., Borchert, T., Herbrich, R.: Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In: Proceedings of the 27th International Conference on Machine Learning (ICML 2010), pp. 13–20 (2010)
24. Hannah, L.A., Blei, D.M., Powell, W.B.: Dirichlet process mixtures of generalized linear models. The Journal of Machine Learning Research 12, 1923–1953 (2011)
25. Harries, M.: Splice-2 comparative evaluation: Electricity pricing. Artificial Intelligence Group, School of Computer Science and Engineering, The University of New South Wales, Sidney, Tech.Rep. UNSW-CSE-TR-9905 (1999)
26. Heath, D., Sudderth, W.: De Finetti's theorem on exchangeable variables. The American Statistician 30(4), 188–189 (1976)
27. Hosseini, M.J., Ahmadi, Z., Beigy, H.: New management operations on classifiers pool to track recurring concepts. In: Cuzzocrea, A., Dayal, U. (eds.) DaWaK 2012. LNCS, vol. 7448, pp. 327–339. Springer, Heidelberg (2012)
28. Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.: Stochastic variational inference. The Journal of Machine Learning Research 14(1), 1303–1347 (2013)
29. Katakis, I., Tsoumakas, G., Vlahavas, I.: Tracking recurring contexts using ensemble classifiers: an application to email filtering. Knowledge and Information Systems 22(3), 371–391 (2010)
30. Klinkenberg, R.: Learning drifting concepts: Example selection vs. example weighting. Intelligent Data Analysis 8(3), 281–300 (2004)
31. Minka, T.P.: Bayesian model averaging is not model combination. Technical Report (2000)
32. Minka, T.P.: Expectation propagation for approximate Bayesian inference. In: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, pp. 362–369. Morgan Kaufmann Publishers Inc. (2001)
33. Neal, R.M.: Markov chain sampling methods for Dirichlet process mixture models. Journal of Computational and Graphical Statistics 9(2), 249–265 (2000)
34. Minku, L.L., Yao, X.: DDD: A new ensemble approach for dealing with concept drift. IEEE Transactions on Knowledge and Data Engineering 24(4), 619–633 (2012)
35. Paquet, U., Van Gael, J., Stern, D., Kasneci, G., Herbrich, R., Graepel, T.: Vuvuzelas & Active Learning for Online Classification. In: NIPS Workshop on Comp. Social Science and the Wisdom of Crowds (2010)
36. Shahbaba, B., Neal, R.: Nonlinear models using Dirichlet process mixtures. The Journal of Machine Learning Research 10, 1829–1850 (2009)
37. Zhang, J., Ghahramani, Z., Yang, Y.: A Probabilistic Model for Online Document Clustering with Application to Novelty Detection. In: NIPS, vol. 4, pp. 1617–1624 (2004)
38. Zhu, X., Zhang, P., Lin, X., Shi, Y.: Active learning from stream data using optimal weight classifier ensemble. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 40(6), 1607–1621 (2010)