Constructivism Learning: A Learning Paradigm for Transparent Predictive Analytics

Xiaoli Li and Jun Huan
{xiaoli.li,jhuan}@ku.edu
Department of Electrical Engineering and Computer Science,
University of Kansas

ABSTRACT

Developing transparent predictive analytics has attracted significant research attention recently. There have been multiple theories on how to model learning transparency but none of them aims to understand the internal and often complicated modeling processes. In this paper we adopt a contemporary philosophical concept called "constructivism", which is a theory regarding how human learns. We hypothesize that a critical aspect of transparent machine learning is to "reveal" model construction with two key process: (1) the assimilation process where we enhance our existing learning models and (2) the accommodation process where we create new learning models. With this intuition we propose a new learning paradigm, constructivism learning, using a Bayesian nonparametric model to dynamically handle the creation of new learning tasks. Our empirical study on both synthetic and real data sets demonstrate that the new learning algorithm is capable of delivering higher quality models (as compared to base lines and state-of-the-art) and at the same time increasing the transparency of the learning process.

CCS CONCEPTS

•Mathematics of computing →Bayesian nonparametric models; •Computing methodologies →*Online learning settings;*

KEYWORDS

Constructivisim Learning; Transparent Machine Learning; Sequential Dirichlet Process; Dynamic Task Construction

ACM Reference format:
Xiaoli Li and Jun Huan. 2017. Constructivism Learning: A Learning Paradigm for Transparent Predictive Analytics. In *Proceedings of KDD'17, August 13–17, 2017, Halifax, NS, Canada.,* , 10 pages.
DOI: http://dx.doi.org/10.1145/3097983.3097994

1 INTRODUCTION

Developing transparent predictive analytics has attracted significant research attention recently [1, 5]. There are many applications where transparent models are critical for the successful deployment of such systems. For example in the medical domain, it is hard for a physician to use results from predictive modeling without knowing how the results are derived. To better train robots, Thomaz and Breazeal showed that if a learning algorithm can reveal its uncertainty of an action in reinforcement learning, that information provides great help for human to better train robots [27]. In addition in legal system for recidivism prediction, using black box predictive analytics may lead to unfair treatment of minority groups and thus commit illegal discrimination [31].

There is intensive discussion on how to define transparency and how to introduce transparency in a predictive analytics. A large body of literature focuses on explaining the results of the prediction [8, 13]. The premise is that if we are able to explain the model results, we improve the transparency of the model and in this sense interpretability and transparency are two closely interleaved concepts. Exemplar work in this category includes sparse linear models [28], prototype based methods such as Bayesian case models [12], and approximating opaque models using local and interpretable models with low complexity [20], among others. Additional theoretic model includes Situated Learning Theory [27], regarding human learning in the context of social interactions and "black box in a glass box" [6, 9] where different levels of modeling transparency are discussed.

The critical limitation of existing discussion is that all the aforementioned works focus on either making sense of the produced model to or delivering models that are easily understandable by end users (i.e. *model transparency*). They do not aim to understand the internal and often complicated modeling process (i.e. *modeling transparency*). We view machine learning decision process as a process of "trade-off" between model fitness (usually evaluated by a loss function) and modeler's experience (usually encoded as the prior distribution in Bayesian learning or the regularization in PAC learning). We argue that in order to achieve transparency we have to at least reveal the internal trade-off process that involves features, hyper-parameters, learning machines, and key results statistics, to the end user as advocated for example in [32].

Our work is motivated by a much broader philosophical discussion called "constructivism", which has profound impact of modern viewpoint about the nature of knowledge. In the constructivism theory, the learner constructs new knowledge through her interaction with the world with two key processes *assimilation* and *accommodation*. Through assimilation, a learner incorporates new experience into an existing knowledge framework without changing that framework. Through accommodation, a learner changes her internal representation of the external world according to the new experience.

With this intuition, we propose a new learning paradigm where when we have new interactions with the world (i.e. through a new

KDD'17, August 13–17, 2017, Halifax, NS, Canada.
 ISBN 978-1-4503-4887-4/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3097983.3097994

training sample), we evaluate whether existing knowledge can generalize well to this new interaction with minor modifications. If not we conclude that new knowledge should be constructed. Specifically in the context of data analytics, we assume samples arrive sequentially. For each newly introduced sample we evaluate our trained models and decide whether we should simply update the existing models (assimilation) or we should create a new learning model if we have sufficient evidence to believe that there is a new learning task in our data sets (accommodation). Here "we" is a machine learning algorithm. Modeling transparency in constructivism learning hence is specifically defined as the algorithm's ability to recognize assimilation and accommodation.

This new learning paradigm poses two challenges in designing an algorithm. First, the algorithm must have the capability to dynamically create models when needed (dynamic task construction). Secondly, the algorithm must have a way to determine which model a newly arrived sample should belong to (sample task assignment). We formalized a Bayesian nonparametric approach using sequential Dirichlet Process Mixture Models (DPMM) to support constructivism learning. The advantage of Bayesian nonparametric is that we do not need to specify the total number of classification models up front and we use data driven approaches to explore a set with potentially infinite number of models. Such Bayesian nonparametric models naturally support the dynamic construction of learning tasks. For sample task assignment we introduced a technique called the *selection principle* to improve the fitness principle, commonly used in traditional Dirichlet Process Mixture of models, which demonstrate significant empirical improvement.

In summary the major contributions that we made in this paper towards transparent predictive analytics are highlighted below.

- We introduced the theory of constructivism in order to offer transparency in the learning process using two concepts: assimilation and accommodation. Based on the theory, we designed a principled approach called *constructivism learning*.
- We performed a systematic investigation and formalized the related learning problem as a novel sequential Dirichlet Process Mixture of Classification Model problem (a.k.a. sDPMCM) where with new training samples we may either update existing learning models or identify a new learning task.
- We introduced a novel and efficient variational inference method for sDPMCM with a technique that we call selection principle.
- Our experimental study demonstrated the improved classification performance of the new learning paradigm. We showed that the new paradigm improved modeling transparency by revealing insights on two key components in a reasoning process: assimilation and accommodation.

2 RELATED WORK

We review related work in three highly relevant categories: transparent machine learning, constructivism (human) learning, and task construction in functional data analysis.

2.1 Transparent Machine Learning

We notice that there are a few recent efforts aiming to reveal the underlying reasoning mechanics of a machine learning algorithm. For example Krause et al. [14] employed a visual analytics to depict input-output relationship by treating the algorithm as a black-box. In this way the user gets a sense of internal learning process by observing how the output may change according to the change of input. Zhou et al. [32] developed a technique to improve transparency by revealing the internal status of a hierarchical beta process. In their study they visualize how output statistics (e.g. precision, recall) change according to different hyperparameter settings. However they lack principled and systematic approaches to address the problem. To initialize the discussion in this paper we adopted the theory of constructivism in human learning and designed an approach with comprehensive experimental study.

2.2 Constructivism Learning

We briefly review the constructivism theory in order to provide further background information and motivation of our work. The full treatment of the concept is clearly beyond this technical discussion and useful references can be found in [18]. Constructivism aims to better understand the nature of knowledge and thus it belongs to epistemology, a branch of philosophy dated back to Aristotle. In that constructivism is not merely a pedagogy though it has been widely used in designing education methods. Following constructivism in education, the focus is to change the role of an educator from a supervisor to a facilitator. Constructivism thus promotes active learning where instructor provide all the necessary information aiming to help students acquire new knowledge.

2.3 Learning with Task Construction

Dynamically identifying learning tasks using Bayesian nonparametric models have been discussed in different context, primarily in functional data clustering [11]. In functional data analysis, we have n sets of samples (labeled data or functional data) from different subjects, experiments, or object, functional clustering tries to construct and learn m regression functions (i.e. tasks) for those n sets of samples by clustering them so that the samples in the same cluster can be described using the same regression function.

Most of existing methods in learning with task construction were designed for batch data [4, 16, 17, 19, 21, 22, 25]. For the scenario of streaming data, the only existing work was proposed by Bastani et al. [3] in a recent paper for object trajectory clustering. In that work, a framework is presented for incremental clustering of object trajectories using DP mixtures of Gaussian processes. It is worth noting that the application of that method is specific to activity mining and analysis from surveillance video.

3 PRELIMINARY

Before presenting our method, we give a brief introduction to Dirichlet Process in order to be self-contained.

3.1 Notation

For clarity, we introduce the following notations. We use lowercase letters to represent scalar values, lowercase letters with bold font to represent vectors (e.g. $\boldsymbol{u}$), uppercase bold letters to represent

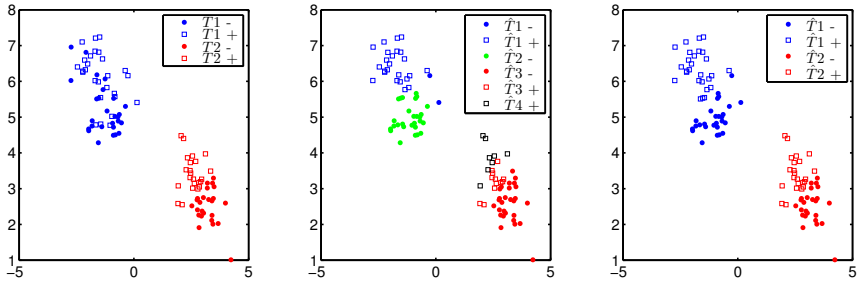


Figure 1: An Issue of the Fitness Principle: Left: Gound Truth; Middle: Tasks Constructed without Using the Selection Principle; Right: Tasks Constructed using the Selection Principle.

matrices (e.g. $\mathbf{A}$), Greek letters $\{\alpha, \lambda, \gamma, \dots\}$ to represent parameters. Given a matrix $\mathbf{A} = (a_{i,j}) \in \mathbb{R}^{p\times k}$, $|\mathbf{A}|$ is the determinant of $\mathbf{A}$. Unless stated otherwise, all vectors in this paper are column vectors. $\boldsymbol{u}^T$ is the transpose of the vector $\boldsymbol{u}$. We use $[1:N]$ to denote the set $\{1, 2, \dots, N\}$.

3.2 Dirichlet Process Mixture Models (DPMM)

The Dirichlet process is a random probability measure defined using an concentration parameter α and a base distribution H over a set Θ, denoted as $\mathrm{DP}(\alpha, H)$. It is a distribution over distributions. Each draw G from a DP is a discrete distribution consists of weighted sum of point masses with locations drawn from H. It has the property that, for any finite set of measurable partitions $A_1, A_2, \cdots, A_k$ of Θ,

$$(G(A_1), G(A_2), \cdots, G(A_k)) \sim \mathrm{Dir}(\alpha H(A_1), \alpha H(A_2), \cdots, \alpha H(A_k))$$

Where *Dir* denotes a Dirichlet distribution.

Consider drawing i.i.d sequence $\theta_1, \theta_2, \cdots \theta_n \sim G$, the predictive distribution of θ_i conditioned on other θ_j, $j < i$, written θ_{-i} is:

$$\theta_i|\theta_{-i} = \frac{1}{\alpha + n - 1}\sum_{j<i}\delta_{\theta_j} + \frac{\alpha}{\alpha + n - 1}H \tag{1}$$

Note that (1) implies clustering property of DP, i.e., θs in the same cluster have the same value, due to the positive probability that θ_i will take on the same value as other θ_{-i}.

Relying on this clustering property of DP, we formalize DP mixture of classification models (DPMCM) for data $(\boldsymbol{x}_i, y_i), i \in [1:N]$, where $\boldsymbol{x}_i$ is a feature vector and y_i is a label, as follows:

$$\begin{aligned} y_i|\boldsymbol{x}_i, \boldsymbol{\beta}_i &\sim F_y(\cdot|\boldsymbol{x}_i, \boldsymbol{\beta}_i) \\ \boldsymbol{x}_i|\boldsymbol{\theta}_i &\sim F_x(\cdot|\boldsymbol{\theta}_i) \\ (\boldsymbol{\theta}_i, \boldsymbol{\beta}_i)|G &\sim G \\ G &\sim \mathrm{DP}(\alpha, H_\theta \times H_\beta) \end{aligned} \tag{2}$$

Here we model the joint distribution of $\boldsymbol{x}_i$ and y_i and assume that the base distribution $H_\theta \times H_\beta$ is independent between parameters $\boldsymbol{\theta}_i$ and $\boldsymbol{\beta}_i$. Through this formulation, data are grouped into different clusters with each cluster k represented by a generative model parameterized by $(\boldsymbol{\theta}_k^*, \boldsymbol{\beta}_k^*)$. We have $\boldsymbol{\theta}_i = \boldsymbol{\theta}_k^*$ and $\boldsymbol{\beta}_i = \boldsymbol{\beta}_k^*$ if $(\boldsymbol{x}_i, y_i)$ is generated using the model of cluster k. F_x and F_y are generative probabilistic models.

Note that (2) can be directly applied to constructivism learning since it allows dynamic task construction and sample task assignment. Specially, due to the infinite-dimensional space of Bayesian nonparametric models, the complexity of models adapts to data so that new tasks can be constructed as needed. In addition, the clustering property of DPMCM provides us a solution for determining which task a sample should be assigned to. However, the implementation of constructivism learning by directly using (2) may lead to some issues, for which a detailed discussion will be given in section 4.1.

4 ALGORITHM

In this section, we describe our proposed algorithm for constructivism learning (CL). We begin by formalizing the problem setting of CL. This is followed by the description of our sequential DP mixture of classification model(sDPMCM). Then we propose an improved version of sDPMCM , sequential DP mixture of classification model with selection (sDPMCM-s), which is enhanced using an approach we call the selection principle. In the last section, we present the sequential variational inference algorithm for sDPMCM-s adapted from [15].

4.1 Problem Setting and Challenges

We aim to design an algorithm with modeling transparency, i.e. the ability to recognize assimilation and accommodation, by building a set of models incrementally through dynamic task construction and sample task assignment. Specially, suppose that we have data that arrives sequentially, we try to determine whether newly arrived sample $(\boldsymbol{x}_i, y_i)$ can be well classified using existing tasks constructed from previous data $(\boldsymbol{x}_j, y_j)$, $j = 1, 2, \dots, i-1$, or a new task needs to be constructed from scratch. Here $\boldsymbol{x}_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \{0, 1\}$ its corresponding label. Giving a sequence of training samples $(\boldsymbol{x}_i, y_i)$, indexed by i, the outcome of our learning algorithm is a set of classification models. For clarity, we summarize the important notations in Table 1.

As we mentioned before, we can use DPMCM directly for CL. However, this may lead to two issues, short-sighted task assignment and high computational cost, which we discuss in the following.

First, In DPMCM, the decision of assigning a sample to a task is based on the *fitness principle*, i.e., evaluating the likelihood that the sample $(\boldsymbol{x}, y)$ is generated from the task k, which is calculated using:

$$w\int_\Theta F(\boldsymbol{x}, y|\Theta)V(d\Theta) \tag{3}$$

Table 1: Notations for CL

$\boldsymbol{x}_i, y_i$	The feature vector and label for the ith sample
$\boldsymbol{\mu}_i^x, \Sigma_i^x$	The Parameters for the distribution of $\boldsymbol{x}_i$
$\boldsymbol{\beta}_i^y$	The parameter for the logistic function of the ith sample
α	The concentration parameter of DP
H	The base distribution of DP
ρ_i	Task assignment probability for the ith sample
$U(\boldsymbol{\beta}^y; \boldsymbol{x}, y)$	Utility of the sample $(\boldsymbol{x}, y)$ for the task with the parameter $\boldsymbol{\beta}^y$

where w is a weight determined by the parameter α and number of samples in the task k. F is the probability or density for $(\boldsymbol{x}, y)$ parameterized by Θ. V is a distribution for Θ. The drawback of this principle is that the contribution of a sample to a task is ignored. When DPMCM is applied to streaming data for classification models, this may lead to short-sighted task assignment and undesirable tasks since they are estimated in a single pass over the data.

We illustrate the problem of fitness principle using a synthetic data set as shown in Fig. 1. Here we have samples from two hidden tasks T_1 and T_2 (Left panel). We adopted a DPMCM, which outputted 4 tasks, rather than 2. In addition task $\hat{T}_4$ has only positive samples and $\hat{T}_2$ has only negative samples. Moreover the samples in $\hat{T}_1$ are highly unbalanced. We believe the reason why DPMCM produces large number of single-class tasks is that the samples in those tasks can be well classified and hence *well fitted*. However, those tasks constructed by DPMCM based only on fitness principle can hardly generalize well to unseen samples. Inspired by this observation, we develop a new DPMCM model enhanced with *selection principle* utilizing Kullback-Leibler divergence (KLD). It is a complement to the fitness principle. When making assignment decision, the contribution of a sample to a task is considered.

The second issue of employing DPMCM for CL is that we have to re-compute the inference for DPMCM when a new sample arrives, which is compute-intensive. To efficiently handle streaming data, we first adapt the sequential method proposed in [15] to DP mixture of classification models to achieve incrementally inference computation. Then we modify a variational approximation technique of logistic regression [10] for efficient task parameter updating.

4.2 Sequential DP Mixture of Classification Model

In this section, we first present a streaming inference method for Dirichlet process [15], which is the starting point of our formalization. Then we propose two new methods for CL, sequential DP mixture of classification models and its enhanced version by incorporating the selection principle.

4.2.1 Sequential DP Mixture Models (sDPMM). To handle streaming data for the following generic DP mixture model:

$$\begin{aligned} \boldsymbol{x}_i|\boldsymbol{\theta}_i &\sim F(\cdot|\boldsymbol{\theta}_i) \\ \boldsymbol{\theta}_i|G &\sim G \\ G &\sim \mathrm{DP}(\alpha, H) \end{aligned}$$

Lin proposed a sequential variational approximation method [15]. The advantage of Lin's method is that it is truncation free and a single pass over data can reliably estimate a DP mixture model. In this method suppose $\boldsymbol{\theta}$s are grouped into K clusters, and there is a cluster indicator c_i for each $\boldsymbol{\theta}_i, \forall i \in [1:N]$ such that $\boldsymbol{\theta}_i$ belongs to the kth cluster, i.e., $\boldsymbol{\theta}_i = \boldsymbol{\theta}_k^*$, if $c_i = k$. Lin proposed to approximate distribution:

$$p(G|\boldsymbol{x}_{1:N}) = \sum_{c_{1:N}} p(c_{1:N}|\boldsymbol{x}_{1:N}) p(G|c_{1:N}, \boldsymbol{x}_{1:N})$$

using a tractable distribution with the following form:

$$q(G|\boldsymbol{x}_{1:N}) = \sum_{c_{1:N}} \Big(\prod_{i=1}^{N} \rho(c_i)\Big) q_v^{(c)}(G|c_{1:N}).$$

Here $\boldsymbol{\theta}_k^* \sim \nu_k$, which is an independent distribution. The task of inference is to optimize two sets of parameters $\rho \triangleq (\rho_1, \rho_2, \cdots, \rho_i)$ and $\nu_i \triangleq (\nu_1^i, \nu_2^i, \cdots, \nu_K^i)$ so that $q(G|\boldsymbol{x}_{1:N})$ best approximates the true posterior $p(G|\boldsymbol{x}_{1:N})$.

By using variational approximation technique to minimize the Kullback-Leibler divergence between the true posterior and the approximate posterior, we have sequential approximation for the optimal settings of ρ_{i+1} and ν_k^{i+1} after processing i samples:

$$\rho_{i+1} \propto \begin{cases} w_k^i \int_{\boldsymbol{\theta}} F(\boldsymbol{x}_{i+1}|\boldsymbol{\theta}) \nu_k^i(d\theta) & k \le K \\ \alpha \int_{\boldsymbol{\theta}} F(\boldsymbol{x}_{i+1}|\boldsymbol{\theta}) H(d\theta) & k = K+1 \end{cases} \tag{4}$$

Where $w_k^i = \sum_{j=1}^{i} \rho_j(k)$.

$$\nu_k^{i+1}(d\boldsymbol{\theta}) \propto \begin{cases} H(d\boldsymbol{\theta}) \prod_{j=1}^{i+1} (F(\boldsymbol{x}_j|\boldsymbol{\theta}))^{\rho_j(k)} & k \le K \\ H(d\boldsymbol{\theta}) (F(\boldsymbol{x}_{i+1}|\boldsymbol{\theta}))^{\rho_{i+1}(k)} & k = K+1 \end{cases} \tag{5}$$

$\rho_i(k), \forall k \in [1:K+1]$, $i = 1, 2, \cdots$, are computed using:

$$\rho_i(k) = \frac{w_k^{i-1} \exp(h_i(k))}{\sum_{c=1}^{K+1} w_c^{i-1} \exp(h_i(c))} \tag{6}$$

Where $h_i(k)$ is marginal log-likelihood of $\boldsymbol{x}_i$ belonging to cluster k.

4.2.2 Sequential DP Mixture of Classification Models (sDPMCM). The DP Mixture of Classification Model (DPMCM) we use for CL is based on the general formulation (2). Here we model the joint distribution of $\boldsymbol{x}$ and y, $p(\boldsymbol{x}, y) = p(y|\boldsymbol{x})\, p(\boldsymbol{x})$, instead of only modeling the conditional distribution $p(y|\boldsymbol{x})$. This provide us the benefit of discovering hidden structure of data by clustering $\boldsymbol{x}$ [26]. Different from sDPMM model where only $\boldsymbol{x}$ is clustered, we cluster both $\boldsymbol{x}$ and y by modeling the joint distribution of $\boldsymbol{x}$ and y. Thus sDPMM cannot be directly applied to DP mixture of classification models. The main change required here is to modify the sequential approximation of ρ_{i+1} and ν_k^{i+1} as follows :

$$\rho_{i+1} \propto \begin{cases} w_k^i \int_{\Theta} F(\boldsymbol{x}_{i+1}, y_{i+1}|\Theta) \nu_k^i(d\Theta) & k \le K \\ \alpha \int_{\Theta} F(\boldsymbol{x}_{i+1}, y_{i+1}|\Theta) H(d\Theta) & k = K+1 \end{cases} \tag{7}$$

$$\nu_k^{i+1}(d\boldsymbol{\Theta}) \propto \begin{cases} H(d\Theta) \prod_{j=1}^{i+1} (F(\boldsymbol{x}_j|\boldsymbol{\mu}^x, \Sigma^x) F(y_j|\boldsymbol{\beta}^y))^{\rho_j(k)} & k \le K \\ H(d\Theta) (F(\boldsymbol{x}_{i+1}|\boldsymbol{\mu}^x, \Sigma^x) F(y_{i+1}|\boldsymbol{\beta}^y))^{\rho_{i+1}(k)} & k = K+1 \end{cases} \tag{8}$$

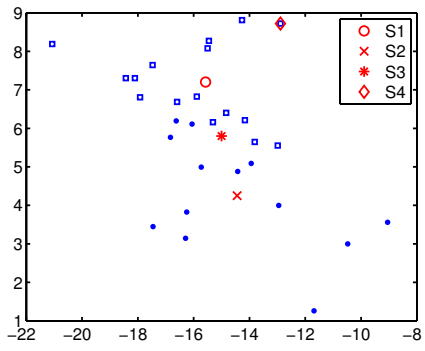


Figure 2: Function of Selection Principle: s_1: mean of positive samples; s_2: mean of negative samples; s_3: close to decision boundary; s_4: far away from decision boundary.

	s_1	s_2	s_3	s_4
KLD	4.485	4.486	4.547	4.491
Pr	0.972	0.967	0.729	0.953

Table 2: Comparison of KLD and Posterior Predictive Probability (Pr)

4.2.3 Sequential DP Mixture of Classification Model with Selection Principle (sDPMCM-s). In selection principle, we focus on evaluating the contribution of a sample to a task. Specially, we aim to measure the utility of an observation of a random variable, i.e. a sample, to other random variables, i.e., task parameters. To this end, we utilize Kullback-Leibler divergence (KLD). This technique has been used in Bayesian optimal experiment design (BOED) [24] to select the optimal design of experiment so that the expected utility of the experiment outcome can be maximized. Different from the general case in BOED, we focus on the utility of a given sample instead of the expected utility. Specifically, we assign a sample to a task so that KLD between the prior distribution of task parameters and the posterior given the sample is maximized. Suppose we have a sample $(\boldsymbol{x}, y)$ and task parameters $\boldsymbol{\beta}^y$, the utility is defined as:

$$U(\boldsymbol{\beta}^y; \boldsymbol{x}, y) = D_{KL}(p(\boldsymbol{\beta}^y|\boldsymbol{x}, y) \parallel p(\boldsymbol{\beta}^y))$$

The goal of selection principle is to assign a sample $(\boldsymbol{x}, y)$ to a task with parameter $\boldsymbol{\beta}^y$ such that $U(\boldsymbol{\beta}^y; \boldsymbol{x}, y)$ is maximized. To understand the function of selection principle, we illustrate it using Fig. 2 and Table 2. In this example we selected four representative samples in a task. s_1 and s_2 are mean of positive and negative samples separately. s_3 is close to decision boundary. s_4 is far away from decision boundary. We observe that although s_1, s_2, s_4 have higher posterior predictive probabilities, their utility to the task is lower than s_3. This example confirms our hypothesis that the selection principle can act as a regularization factor to regulate the range of a task to form more compact clusters.

To incorporate selection principle when making assignment decision, we modify the computation of task assignment probability ρ_{i+1} as follows:

$$\rho_{i+1} \propto \begin{cases} w_k^i s_k^{i+1} \int_\Theta F(\boldsymbol{x}_{i+1}, y_{i+1}|\Theta) v_k^i(d\Theta) & k \leq K \\ \alpha s_k^{i+1} \int_\Theta F(\boldsymbol{x}_{i+1}, y_{i+1}|\Theta) H(d\Theta) & k = K+1 \end{cases} \tag{9}$$

Where $s_k^{i+1} = \exp(\gamma U(\boldsymbol{\beta}^y; \boldsymbol{x}_{i+1}, y_{i+1}))$, where γ is a coefficient to regularize the effect of selection principle. Through s_k^{i+1}, the utility of $\boldsymbol{x}_{i+1}, y_{i+1}$ to the model parameter $\boldsymbol{\beta}^y$ is also considered when making task assignment decisions.

4.3 Inference

Several issues need to be addressed for the inference of the proposed sDPMCM-s for streaming data. First, posterior distribution for Θ and prediction distribution for $\boldsymbol{x}$ and y needs to be updated with relatively low complexity during streaming inference. Secondly, an efficient method is needed to compute $U(\boldsymbol{\beta}^y; (\boldsymbol{x}, y))$ when new samples arrive. Thirdly, the inference method, i.e. sDPMM, proposed by Lin was designed for DPMM instead of DPMCM. Thus some modification is needed to adapt it to DPMCM.

In the following, we first describe the details of classification models and variational approximation techniques used in sDPMCM-s for efficient model updating. Then we give the specific formula for utility computation. Lastly, we summarize the complete sequential inference for sDPMCM-s, and present the strategy we use for prediction.

4.3.1 Classification Model in sDPMCM. Within each task of the mixture, we assume that $\boldsymbol{x}$ follow a Multivariate Normal (MN) distribution with mean $\boldsymbol{m}^x$ and covariance matrix Σ^x. For simplicity of computation, we assume a conjugate prior, Normal-Inverse-Wishart(NIW), for $(\boldsymbol{m}^x, \Sigma^x)$. To model the relationship between $\boldsymbol{x}$ and y, we use logistic regression:

$$F_y(y|\boldsymbol{x}, \boldsymbol{\beta}) = \frac{\exp(y\boldsymbol{x}^T\boldsymbol{\beta})}{1 + \exp(\boldsymbol{x}^T\boldsymbol{\beta})} \tag{10}$$

Where $\boldsymbol{\beta} \in \mathbb{R}^d$. And we use a Multivariate Normal distribution for $\boldsymbol{\beta}^y$ with parameters $(\boldsymbol{\mu}_0^\beta, \boldsymbol{\Psi}_0^\beta)$. Note that we assume independence between $(\boldsymbol{m}^x, \Sigma^x)$ and $\boldsymbol{\beta}^y$, and the distribution of y does not depend on $(\boldsymbol{m}^x, \Sigma^x)$ given $\boldsymbol{x}$. This results in simplified computations.

The complete DPMCM for CL is summarized as follows:

$$G \sim \text{DP}(\alpha, \text{NIW}(\boldsymbol{\mu}_0^x, \kappa_0^x, \boldsymbol{\Psi}_0^x, \nu_0^x)\text{MN}(\boldsymbol{\mu}_0^\beta, \boldsymbol{\Psi}_0^\beta)) \tag{11}$$

For $i = 1, 2, \ldots,$ draw i.i.d using:

$$\begin{aligned} y_i|\boldsymbol{x}_i, \boldsymbol{\beta}_i^y &\sim F_y(\cdot|\boldsymbol{x}_i, \boldsymbol{\beta}_i^y) \\ \boldsymbol{x}_i|\boldsymbol{m}_i^x, \Sigma_i^x &\sim \text{MN}(\cdot|\boldsymbol{m}_i^x, \Sigma_i^x) \\ (\boldsymbol{m}_i^x, \Sigma_i^x, \boldsymbol{\beta}_i^y)|G &\sim G \end{aligned} \tag{12}$$

Due to the clustering property of DP, $\Theta = (\boldsymbol{m}_i^x, \Sigma_i^x, \boldsymbol{\beta}_i^y)$ will be grouped into different clusters. Θs in the same cluster have an identical value.

4.3.2 Variational Approximation of Logistic Regression. Using logistic regression to model the relationship between $\boldsymbol{x}$ and y poses challenges for computing posterior distribution of model parameters and predictive distribution since they are computationally intractable. To tackle this, we adopt the variational approximation

proposed in [10] to replace the logistic function with an adjustable lower bound that has a Gaussian form. This results in the Gaussian form of posterior due to the Gaussian prior of $\boldsymbol{\beta}^y$ and Gaussian variational form of $p(y|\boldsymbol{x}, \boldsymbol{\beta}^y, \xi)$. Here ξ is a variational parameter. To apply this variational approximation to DPMM in sequential setting, we need to modify the updating of parameters $\boldsymbol{\Psi}^\beta, \boldsymbol{\mu}^\beta, \xi$ in [10] so that the uncertainty of clustering brought by sequential inference can be incorporated. With straightforward calculation, we derive the following modification to the updating of parameters when $\rho_{i+1} > 0$:

$$\begin{aligned}\boldsymbol{\Psi}^\beta_{i+1} &= \left[(\boldsymbol{\Psi}^\beta_i)^{-1} + 2\rho_{i+1} f(\xi)\boldsymbol{x}_{i+1}\boldsymbol{x}^T_{i+1}\right]^{-1}\\ \boldsymbol{\mu}^\beta_{i+1} &= \boldsymbol{\Psi}^\beta_{i+1}\left[(\boldsymbol{\Psi}^\beta_i)^{-1}\boldsymbol{\mu}^\beta_i + (\rho_{i+1}y_{i+1} - \frac{1}{2})\boldsymbol{x}_{i+1}\right]\\ \xi^2 &= \rho_{i+1}\left[\boldsymbol{x}^T_{i+1}\boldsymbol{\Psi}^\beta_{i+1}\boldsymbol{x}_{i+1} + (\boldsymbol{x}^T_{i+1}\boldsymbol{\mu}^\beta_{i+1})^2\right]\end{aligned} \tag{13}$$

For the predictive lower bound for sample $\boldsymbol{x}_{i+1}, y_{i+1}$, it takes the form:

$$\begin{aligned}\ln q(y_{i+1}|\boldsymbol{x}_{i+1}, D) = \ln g(\xi_{i+1}) - \frac{\xi_{i+1}}{2} + f(\xi_{i+1})\xi^2_{i+1}\\ - \frac{1}{2}(\boldsymbol{\mu}^\beta_i)^T(\boldsymbol{\Psi}^\beta_i)^{-1}\boldsymbol{\mu}^\beta_i\\ + \frac{1}{2}(\boldsymbol{\mu}^\beta_{i+1})^T(\boldsymbol{\Psi}^\beta_{i+1})^{-1}\boldsymbol{\mu}^\beta_{i+1}\\ + \frac{1}{2}\ln\frac{|\boldsymbol{\Psi}^\beta_{i+1}|}{|\boldsymbol{\Psi}^\beta_i|}\end{aligned} \tag{14}$$

Where $D = (\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_i, y_i)$.

4.3.3 Utility Computation. The computation of $U(\boldsymbol{\beta}^y; \boldsymbol{x}_{i+1}, y_{i+1})$ is straightforward after the variational approximation is used for logistic regression. Let denote the distribution as $p_i(\boldsymbol{\beta}^y \mid \boldsymbol{\mu}^\beta_i, \boldsymbol{\Psi}^\beta_i)$ before $(\boldsymbol{x}_{i+1}, y_{i+1})$ is assigned to a model and $p_{i+1}(\boldsymbol{\beta}^y \mid \boldsymbol{\mu}^\beta_{i+1}, \boldsymbol{\Psi}^\beta_{i+1})$ after assignment, we can calculate the KLD of two MN distributions p_i and p_{i+1}:

$$\begin{aligned}U(\boldsymbol{\beta}^y; \boldsymbol{x}_{i+1}, y_{i+1}) = \frac{1}{2}\Bigg[\ln\frac{|\boldsymbol{\Psi}^\beta_i|}{|\boldsymbol{\Psi}^\beta_{i+1}|} - d + tr((\boldsymbol{\Psi}^\beta_i)^{-1}\boldsymbol{\Psi}^\beta_{i+1}) +\\ (\boldsymbol{\mu}^\beta_i - \boldsymbol{\mu}^\beta_{i+1})^T(\boldsymbol{\Psi}^\beta_i)^{-1}(\boldsymbol{\mu}^\beta_i - \boldsymbol{\mu}^\beta_{i+1})\Bigg]\end{aligned}$$

4.3.4 Sequential Inference for sDPMCM-s. To handle streaming data, we consider the method proposed in [15]. Specifically, when a new sample arrives, we first determine whether it will be assigned to an existing task or a new task. Then the model parameters of assigned task are updated. We describe these two steps separately in the following.

Task Assignment. To determine which task a newly arriving sample belongs to, we need to compute the assignment probability ρ_{i+1} based on (7) and (8). The probability of assigning newly arrived sample $(\boldsymbol{x}_{i+1}, y_{i+1})$ to task k, denoted as $\rho_{i+1}(k)$, is computed using:

$$\rho_{i+1}(k) = \frac{w^i_k s^{i+1}_k \exp(h_{i+1}(k))}{\sum_{c=1}^{K+1} w^i_c s^{i+1}_c \exp(h_{i+1}(c))} \tag{15}$$

Where $h_{i+1}(k)$ is the log posterior predictive of $(\boldsymbol{x}_{i+1}, y_{i+1})$ belonging to cluster k. It can be decomposed into two parts $h_{i+1}(k) = h^x_{i+1}(k) + h^y_{i+1}(k)$. Due to the conjugate property, the posterior predictive, i.e. $h^x_{i+1}(k)$, is a multivariate t distribution with density function:

$$\frac{\Gamma[(\delta_i + d)/2]}{\Gamma(\delta_i/2)\delta^{d/2}\pi^{d/2}|\boldsymbol{\Phi}_i|^{1/2}}\left[1 + \frac{1}{\delta_i}(\boldsymbol{x} - \boldsymbol{\upsilon}_i)^T\boldsymbol{\Phi}^{-1}_i(\boldsymbol{x} - \boldsymbol{\upsilon}_i)\right] \tag{16}$$

Where

$$\begin{aligned}\delta_i &= \nu^x_i(k) - d + 1\\ \boldsymbol{\upsilon}_i &= \boldsymbol{\mu}^x_i(k)\\ \boldsymbol{\Phi}_i &= \frac{\kappa^x_i(k) + 1}{\kappa^x_i(k)(\nu^x_i(k) - d + 1)}\boldsymbol{\Psi}^x_i(k)\end{aligned}$$

$\boldsymbol{\mu}^x_i(k), \boldsymbol{\Psi}^x_i(k), \kappa^x_i(k), \nu^x_i(k)$ are posterior parameters of a NIW distribution for task k after receiving i samples. With (16), $h^x_{i+1}(k)$ can be computed directly.

For the computation of $h^y_{i+1}(k)$, we use the lower bound specified in (14) derived from a variational approximation of logistic regression.

Updating Model parameters. Relying on the conjugate property, posterior parameters of the NIW distribution of task k have a closed-form updating when receiving a new sample $\boldsymbol{x}_{i+1}$. For simplicity, we update natural parameters $\Lambda = (\eta^1_i(k), \eta^2_i(k), \eta^3_i(k), \eta^4_i(k))$ of NIW distribution for task k at each step. They can be derived from $\boldsymbol{\mu}^x_i(k)$, $\boldsymbol{\Psi}^x_i(k)$, $\kappa^x_i(k)$, $\nu^x_i(k)$ using:

$$\begin{aligned}\eta^1_i(k) &= \kappa^x_i(k)\boldsymbol{\mu}^x_i(k)\\ \eta^2_i(k) &= \kappa^x_i(k)\\ \eta^3_i(k) &= \boldsymbol{\Psi}^x_i(k) + \kappa^x_i(k)\boldsymbol{\mu}^x_i(k)(\boldsymbol{\mu}^x_i(k))^T\\ \eta^4_i(k) &= \nu^x_i(k) + d + 2\end{aligned}$$

Modified from the sufficient statistics of the NIW distribution [7], the following form of updating for Λ with considering the uncertainty of task assignment are used:

$$\begin{aligned}\eta^1_{i+1}(k) &= \eta^1_i(k) + \rho_{i+1}(k)\boldsymbol{x}_{i+1}\\ \eta^2_{i+1}(k) &= \eta^2_i(k) + \rho_{i+1}(k)\\ \eta^3_{i+1}(k) &= \eta^3_i(k) + \rho_{i+1}(k)\boldsymbol{x}_{i+1}(\boldsymbol{x}_{i+1})^T\\ \eta^4_{i+1}(k) &= \eta^4_i(k) + \rho_{i+1}(k)\end{aligned}$$

For the updating of parameters $\boldsymbol{\mu}^\beta_i, \boldsymbol{\Psi}^\beta_i$ for $\boldsymbol{\beta}^y_i$, we use the variational approximation in (13).

4.3.5 Prediction. For predicting the label of a test sample $\boldsymbol{x}$, we use the following strategy. First, we use all the K tasks learned from training data to predict the label of $\boldsymbol{x}$ to get K labels, $\hat{y}_1$, $\hat{y}_2, \ldots, \hat{y}_K$. Then we compare the posterior predictive of $P(\boldsymbol{x}, \hat{y}_k)$, $\forall k \in [1 : K]$ and set the label of $\boldsymbol{x}$ to $\hat{y}_{k^*}$ so that $P(\boldsymbol{x}, \hat{y}_{k^*}) = max(P(\boldsymbol{x}, \hat{y}_1), P(\boldsymbol{x}, \hat{y}_2), \ldots, P(\boldsymbol{x}, \hat{y}_K))$.

5 EXPERIMENTS

In this section, we first introduce the data sets and experimental protocol used in our experiments. Then we evaluate the performance of our proposed methods, sDPMCM and its variation sDPMCM-s, by comparing them with base-line methods SVM, Random Forest, and a state-of-the-art classification model based on enriched Dirichlet Process Mixture model (EDPMM) on 4 synthetic data sets and 3 real-world data sets. Lastly, we demonstrate how modeling transparency

can be achieved by identifying assimilation and accommodation in a learning process.

5.1 Data Sets

For the experiments, we used both synthetic data sets and real-worlds data sets, as detailed below.

5.1.1 Synthetic Data Sets. we constructed 4 synthetic data sets with K hidden tasks each, where $K \in [2:5]$. For each task, we randomly draw its parameters from a NIW prior and a MN prior. For the NIW prior of $\boldsymbol{m}^x$ and Σ^x, we use zero mean and a diagonal scale matrix $\psi_0 I$, where $\psi_0 = 2$. We set $\kappa_0^x = 0.04$ and degree of freedom $v_0^x = d + 3$, where d is the dimension of $\boldsymbol{x}$. For the MN prior of $\boldsymbol{\beta}^y$, we use a MN distribution with zero mean and a unit diagonal covariance matrix. The data $\boldsymbol{x}, y$ are generated using the distribution described in (17). We summarize the statistics of 4 synthetic data sets in Table 3.

$$\begin{aligned} y_i|\boldsymbol{x}_i, \boldsymbol{\beta}_i^y &\sim F_y(\cdot|\boldsymbol{x}_i, \boldsymbol{\beta}_i^y) \\ \boldsymbol{x}_i|\boldsymbol{m}_i^x, \Sigma_i^x &\sim \text{MN}(\cdot|\boldsymbol{m}_i^x, \Sigma_i^x) \end{aligned} \tag{17}$$

Data Set	SDS1	SDS2	SDS3	SDS4
T	2	3	4	5
N	205	317	406	500

Table 3: Statistics of Synthetic Data Sets. T: Number of Hidden Tasks. N: Number of Samples.

5.1.2 Real-world Data Sets. We used 3 real-world data sets: WebKB, School Performance and Landmine.

WebKB. This data set contains a subset of the web pages collected from computer science departments of 4 universities in January 1997 by the World Wide Knowledge Base (WebKb) project of the CMU text learning group [1]. It is composed of 230 course pages and 821 non-course pages. For each web page, two types of representation are provided, text on the web page and anchor text of the hyperlinks to that page. We generate the features from text on the web pages using TF-IDF. Then we applied PCA to the features to keep the first 30 components. The classification goal here is to determine whether a web page is a course page or not.

School Performance. The school data set comes from the Inner London Education Authority (ILEA). It is composed of examination records from 140 secondary schools in years 1985,1986 and 1987. The original data includes the year of examination, 4 school-specific attributes and 3 student-specific attributes. In our experiments, we use the processed data set provided by [2], where categorical features are expressed as binary features. To use this data set for classification, we labeled those samples with examination scores larger than 30 as positive and others as negative. We use data from 123 schools by removing those schools with less than 5 positive or 5 negative samples.

LandMine. The land mine data set [23, 30] consists of 14,820 samples from 29 different geographical regions. The features are extracted from radar data, including four-moment based features, three correlation-based features, one energy-ratio feature, one spatial variance feature, and a bias term. The classification goal is to detect whether or not a land mine is present in an area. We used 20% of the data for our experiments.

For each data set, we randomly chose 50% of the data for training and the other 50% for testing. We applied bootstrap resampling to training data sets to create balanced data sets. The statistics about 3 data sets are summarized in Table 4.

Data Set	N	d
WebKB	1051	30
School	11966	17
LandMine	2972	9

Table 4: Statistics of Real Data Sets. N: number of samples d: number of features

5.2 Experiment Protocol

Baseline Methods. To the best of our knowledge, there is no previous work on learning with task construction for classification of streaming data. Thus we only compare our methods with two widely applied batch learning methods, SVM and Random Forest, and one state-of-the-art DP mixture of classification model, joint enriched Dirichlet process mixture model (EDPMM) proposed in [29]. For SVM, we used a SVM classifier with a RBF kernel provided in Matlab. For Random Forest, we used the algorithm implemented in scikit-learn python package. For EDPMM, we used the R code developed by the original authors. We implemented both sDPMCM and sDPMCM-s in Matlab.

Model Construction. We performed 10-fold cross validation to derive training and testing data.

Model Selection. We performed grid search to select optimal model parameters using 10-fold cross validation that was performed on the training data only.

Evaluation Metrics. We used AUC, the area under a ROC curve, calculated on testing data only, to compare the performance of different algorithms.

Significance Test. When we compared different methods, we made sure that these methods were trained using the same training data sets and were evaluated with the same testing data sets. We used paired student's t test to evaluate the statistical significance of the difference between different results.

5.3 Performance Evaluation Results

To evaluate the performance of our proposed methods, we compared them with 3 baseline methods on 4 synthetic data sets and 3 real-world data sets.

5.3.1 Comparison on Synthetic Data Sets. Table 5 presents the results of comparison on 4 synthetic data sets. Compared with SVM and RF, DP-based methods achieves competitive or better results on 4 data sets. As we expected, batch DP mixture model, EDPMM, outperforms sDPMCM and sDPMCM-s on the 3 synthetic data sets. However, the performance difference on SDS1 and SDS3 is not statistically significant according to the paired student's t test.

[1] http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/

Comparing sDPMCM and sDPMCM-s, we observe that sDPMCM-always outperforms sDPMCM. This demonstrates the effectiveness of selection principle on improving performance. It is worth noting that sDPMCM is comparable with EDPMM on SDS4 data set. And sDPMCM-s even performs significantly better than EDPMM on this data set. Our explanation is that the covariance structure may be more complicated with more hidden tasks. Compared with the Inverse-Gamma distribution adopted by EDPMM, the Inverse-Wishart distribution we used for sDPMCM and sDPMCM-s allows richer covariance structure.

DataSet	SVM	RF	EDPMM	sDPMCM	sDPMCM-s
SDS1	0.812	0.801	**0.860**	0.847	0.856
SDS2	0.787	0.748	**0.806***	0.788	0.798
SDS3	0.814	0.789	**0.823**	0.813	0.822
SDS4	0.823	0.814	0.839	0.838	**0.852***

Table 5: Comparison of Algorithms on Synthetic Data Sets. AUC is used for the performance metric.*: statistically significant with 5% significance level.

DataSet	SVM	RF	EDPMM	sDPMCM	sDPMCM-s
WebKB	0.873	0.896	0.894	0.897	**0.910***
School	**0.718**	**0.718**	0.676	0.715	0.717
LandMine	0.676	0.670	0.552	0.670	**0.687***

Table 6: Comparison of Algorithms on Real Data Sets. AUC is used for the performance metric.*: statistically significant with 5% significance level.

5.3.2 Comparison on Real Data Sets. We show the results of comparison of algorithms on real data sets in Table 6. Compared with base-line methods, EDPMM achieves similar performance on WebKB data set. However, its performance on LandMine data set is significantly worse than those of SVM and RF. There are two possible reasons. First, as we mentioned before, it is possible that the Inverse-Gamma prior adopted by EDPMM cannot explain the complicated covariance structure of data. Second, EDPMM used a nested structure to form hierarchical clusters, where X-clusters are nested into each y-cluster. This choice of ordering X and y may be inappropriate for this data set. Although it is possible to use a different ordering, this choice is problem specific and the work did not provide a way to guide this decision. For the school data set, EDPMM also has the worst performance among all algorithms. But note that we collect the result of EDPMM from one run of the experiment due to the high computation cost. For our proposed method, sDPMCM, it achieves comparable performance with random forest. Relying on selection principle, sDPMCM-s achieves statistically significant advantages over other algorithms on WebKB and Landmine data sets.

5.4 Transparency Evaluation

In this section, we conducted experiments to study whether our proposed methods sDPMCM and sDPMCM-s can achieve modeling transparency, i.e. the ability to recognize assimilation and accommodation in a learning process.

	SDS1		SDS5	
	Task 1	Task 2	Task 1	Task2
$\boldsymbol{m}$	[-1.394;5.673]	[2.899;2.818]	[0.851;7.978]	[1.155;1.234]
Σ	[0.568,-0.345; -0.3454,0.8279]	[0.257,-0.178; -0.178,0.513]	[6.099,0.034; 0.034,5.932]	[5.116,0.023; 0.023;5.199]

Table 7: Statistics of SDS1 and SDS5

5.4.1 Evaluation Data Sets and Methods. We evaluate the modeling transparency of each algorithm using both synthetic data sets and real word data sets. For synthetic data sets, we first picked the data set SDS1, which consists of two well separated tasks. To further investigate the issue, we generated another synthetic data set, SDS5, containing two tasks with overlapping samples. We summarize the mean and variance of each task in each data set in Table 7. For real-world data sets, we conducted a case study using a subset of school data set, which consists of data from 5 schools.

For the evaluation on synthetic data sets, we adopt the following learning process. The number of samples is sequentially introduced in such way that the first n samples are all from one task, then we have samples from the second task. *Modeling transparency* in this context is the capacity that a learning algorithm recognizes the second learning task. Quantitatively, we define *constructivism modeling transparency*, CMT, as

$$CMT = 1/N_2, \tag{18}$$

where N_2 is the number of samples from the newly introduced task that the learning algorithm correctly recognizes the first time that there is a new task. Following this definition, we notice that MT must be a positive number between 0 and 1 (inclusively). CMT may take the value 1, if it only takes one sample from the newly introduced task in order for the learning algorithm to recognize that there is a new task. For those algorithms that can not recognize the existence of new tasks, the CMT value is zero (since "can not" is equivalent to that the algorithm needs infinite number of samples from the new task to recognize that there is a new task). CMT is undefined if the algorithm does not correctly recognize the newly arrived task. Apparently the higher the CMT value is, the more sensitive the learning task is for detecting a new task.

5.4.2 Results on Synthetic Data Sets. The two algorithms sDPMCM and sDPMCM-s are designed specifically to be able to recognize newly introduced tasks. For base line algorithms, we are not aware of any discussion on how to use those algorithms to answer the question regarding the number of learning tasks exists in the data set. To fill the gap, we collect the number of support vectors, denoted as S, for SVM and the number of tress, R, for Random Forest. It is our hope that we may see significant changes in S or R when accommodation happens.

The experiments results on SDS1 are shown in Fig. 3. For SVM and Random Forest, we observe that there is no correspondence between the change in modeling processes and the number of tasks. For SVM, the number of support vectors shows a steady increasing with increasing number of samples. For Random Forest, the number of trees tends to fluctuate with different number of samples. The arriving of samples from a new task did not trigger substantial changes in the modeling processes of both SVM and

	Task 1	Task 2
N	197	131
N1	63	102
N2	124	29
P	26%	45%

Table 8: N:Total number of students in the task; N1: number of students from schools funded by a grant from the central government; N2: number of students from church of England schools; P: average of percentage of students eligible for free school meals.

Random Forest. Following the discussion, we conclude that the constructivism modeling transparency, CMT, for SVM and Random Forest is zero. For sDPMCM and sDMCM-s, they can immediately identify that a new task was needed (accommodation) when the first sample (the 55th sample) of the second task arrived. In this case, CMT for both sDPMCM and sDPMCM-s are 1. However sDPMCM failed to identify the correct number of tasks. With selection principle, sDPMCM-s constructed exactly two tasks.

An additional experiment was performed on SDS5. SDS5 is more challenging since the samples from the two learning tasks overlap significantly. The result is displayed in Fig. 4. For this data set, SVM and Random Forest still could not identify the emergence of a new task. In this harder case, 10 samples were needed by sDPMCM-s to trigger an accommodation and hence its *CMT* is 1/10=0.1. We notice that sDPMCM-s can detect the change in the data set and constructed a new task to accommodate the change correctly. sDPMCM is more sensitive comparing to sDPMCM-s. It requires only 2 samples to recognize the existence of new task but sDPMCM does not do so correctly since it identifies many non-existing tasks as well. In this case its *CMT* is undefined.

5.4.3 Results on Real-World Data Sets. For real data sets, it is difficult, if impossible, to know precisely the number of learning tasks in the data set. Therefore we did a case study where we try to understand the tasks that constructed by the sDPMCM-s algorithm. Using 5 school data sets, sDPMCM-s constructed 2 tasks which we summarize in Table 8. From the results, we observe that about one-third of the students in task 1 are from schools funded by the central government while the ratio is more than two-thirds in task 2. In contrast, the number of students who are eligible for free school meals in task 2 is almost twice the number in task 1. The comparison reveals an obvious socioeconomic difference between the students from two tasks. Since socioeconomic status can have an non-negligible impact on student academic achievement, this difference between task 1 and task 2 may validate the accommodation identified by sDPMCM-s .

6 CONCLUSION

In this paper we proposed a new learning paradigm for transparent predictive analytics where we incorporate a contemporary philosophical concept of constructivism in machine learning. We developed a model formalization using Dirichlet Process Mixture Models for streaming data with efficient inference. Our experimental study demonstrated the utility of the proposed methods. Our future work is to extend the current algorithm to other learning scenarios such as semi-supervised learning.

REFERENCES

[1] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35, 4 (2014), 105–120.

[2] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2008. Convex multi-task feature learning. *Machine Learning* 73, 3 (2008), 243–272.

[3] Vahid Bastani, Lucio Marcenaro, and Carlo S Regazzoni. 2016. Online Nonparametric Bayesian Activity Mining and Analysis From Surveillance Video. *IEEE Transactions on Image Processing* 25, 5 (2016), 2089–2102.

[4] J Bigelow and David B Dunson. 2005. Semiparametric classification in hierarchical functional data analysis. *Duke University ISDS Discussion paper* (2005), 05–18.

[5] Jenna Burrell. 2016. How the machine ‼thinks??: Understanding opacity in machine learning algorithms. *Big Data & Society* 3, 1 (2016), 2053951715622512.

[6] Laura Chiticariu, Yunyao Li, and Fred Reiss. 2015. Transparent Machine Learning for Information Extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

[7] Nicholas Foti, Jason Xu, Dillon Laird, and Emily Fox. 2014. Stochastic variational inference for hidden Markov models. In *Advances in Neural Information Processing Systems*. 3599–3607.

[8] Satoshi Hara and Kohei Hayashi. 2016. Making tree ensembles interpretable. In *ICML Workshop on Human Interpretability in Machine Learning (WHI 2016)*.

[9] Kristina Höök. 2000. Steps to take before intelligent user interfaces become real. *Interacting with computers* 12, 4 (2000), 409–426.

[10] Tommi S Jaakkola and Michael I Jordan. 2000. Bayesian parameter estimation via variational methods. *Statistics and Computing* 10, 1 (2000), 25–37.

[11] Julien Jacques and Cristian Preda. 2014. Functional data clustering: a survey. *Advances in Data Analysis and Classification* 8, 3 (2014), 231–255.

[12] Been Kim, Cynthia Rudin, and Julie A Shah. 2014. The Bayesian Case Model: A generative approach for case-based reasoning and prototype classification. In *Advances in Neural Information Processing Systems*. 1952–1960.

[13] Been Kim, Julie A Shah, and Finale Doshi-Velez. 2015. Mind the gap: A generative approach to interpretable feature selection and extraction. In *Advances in Neural Information Processing Systems*. 2260–2268.

[14] Josua Krause, Adam Perer, and Enrico Bertini. 2016. Using Visual Analytics to Interpret Predictive Machine Learning Models. In *ICML Workshop on Human Interpretability in Machine Learning (WHI 2016)*.

[15] Dahua Lin. 2013. Online learning of nonparametric mixture models via sequential variational approximation. In *Advances in Neural Information Processing Systems*. 395–403.

[16] Richard F MacLehose and David B Dunson. 2009. Nonparametric Bayes kernel-based priors for functional data analysis. *Statistica Sinica* (2009), 611–629.

[17] XuanLong Nguyen and Alan E Gelfand. 2011. The Dirichlet labeling process for clustering functional data. *Statistica Sinica* (2011), 1249–1289.

[18] Jean Piaget. 1985. *The equilibration of cognitive structures: The central problem of intellectual development*. University of Chicago Press.

[19] Shubhankar Ray and Bani Mallick. 2006. Functional clustering by Bayesian wavelet methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68, 2 (2006), 305–332.

[20] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why Should I Trust You? Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1135–1144.

[21] Abel Rodriguez, David B Dunson, and Alan E Gelfand. 2008. The nested Dirichlet process. *J. Amer. Statist. Assoc.* 103, 483 (2008), 1131–1154.

[22] James C Ross and Jennifer G Dy. 2013. Nonparametric Mixture of Gaussian Processes with Constraints. In *Proceedings of the 30 th International Conference on Machine Learning*, Vol. 28. 1346–1354.

[23] Paul Ruvolo and Eric Eaton. 2013. ELLA: An Efficient Lifelong Learning Algorithm. *ICML (1)* 28 (2013), 507–515.

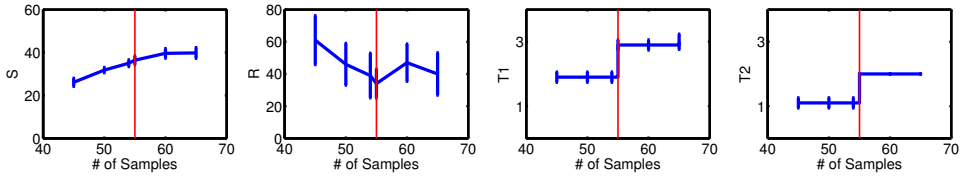
[24] Elizabeth G Ryan, Christopher C Drovandi, James M McGree, and Anthony N Pettitt. 2015. A review of modern computational algorithms for Bayesian optimal design. *International Statistical Review* (2015).

[25] Bruno Scarpa and David B Dunson. 2014. Enriched stick-breaking processes for functional data. *J. Amer. Statist. Assoc.* 109, 506 (2014), 647–660.

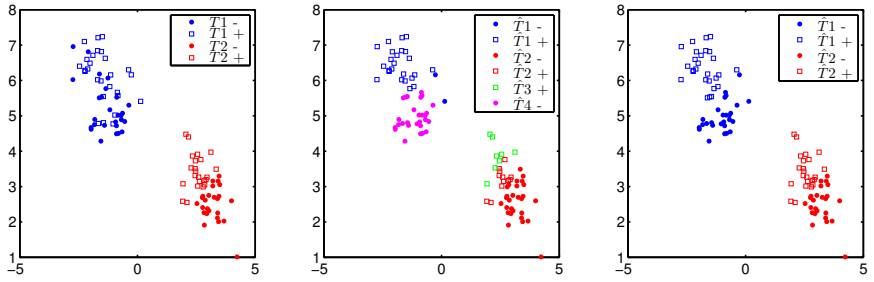
[26] Babak Shahbaba and Radford Neal. 2009. Nonlinear models using Dirichlet process mixtures. *Journal of Machine Learning Research* 10, Aug (2009), 1829–1850.

[27] Andrea L Thomaz and Cynthia Breazeal. 2006. Transparency and socially guided machine learning. In *Proceedings of 5th Intl. Conf. on Development and Learning (ICDL)*.

[28] Berk Ustun and Cynthia Rudin. 2016. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning* 102, 3 (2016), 349–391.

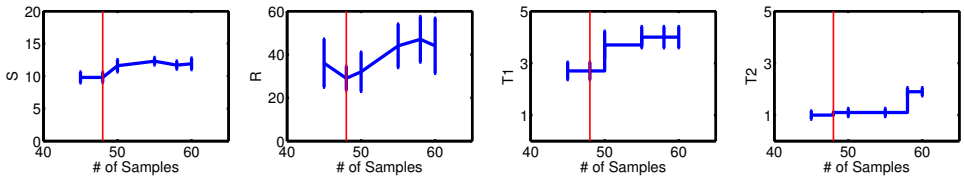


(a) Modeling Processes. S: number of support vectors; R: number of trees; T1,T2: number of tasks constructed by sDPMCM and sDPMCM-s. Red Line: the 1st sample from the 2nd task.

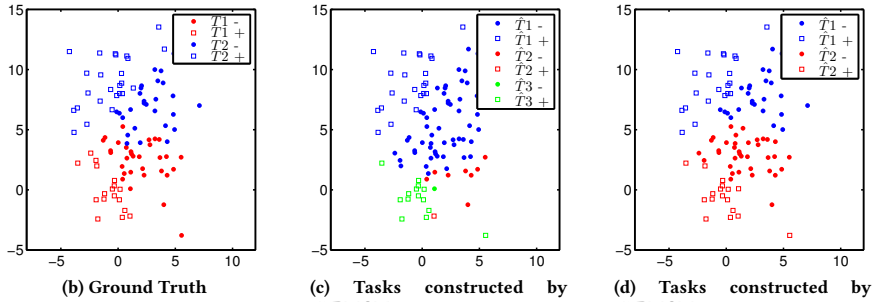


(b) Ground Truth (c) Tasks constructed by sDPMCM (d) Tasks constructed by sDPMCM-s

Figure 3: Transparency Evaluation with Non-overlapping Tasks



(a) Modeling Processes. S: number of support vectors; R: number of trees; T1,T2: number of tasks constructed by sDPMCM and sDPMCM-s. Red Line: the 1st sample from the 2nd task.



(b) Ground Truth (c) Tasks constructed by sDPMCM (d) Tasks constructed by sDPMCM-s

Figure 4: Transparency Evaluation with Overlapping Tasks

[29] Sara Wade, David B Dunson, Sonia Petrone, and Lorenzo Trippa. 2014. Improving prediction from dirichlet process mixtures via enrichment. *Journal of Machine Learning Research* 15, 1 (2014), 1041–1071.

[30] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. 2007. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research* 8, Jan (2007), 35–63.

[31] Jiaming Zeng, Berk Ustun, and Cynthia Rudin. 2016. Interpretable classification models for recidivism prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* (2016).

[32] Jianlong Zhou, Zhidong Li, Yang Wang, and Fang Chen. 2013. Transparent Machine Learning—Revealing Internal States of Machine Learning. In *Proceedings of IUI2013 Workshop on Interactive Machine Learning*. 1–3.