Digital Signal Processing ••• (••••) •••–•••

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Bayesian topic model approaches to online and time-dependent clustering

M. Kharratzadeh, B. Renard, M.J. Coates *

Department of Electrical and Computer Engineering, McGill University, 3480 University St, Montreal, Quebec, H3A 0E9, Canada

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Online clustering
Probabilistic topic models
Dirichlet process mixture models
Streaming data
Sequential Monte Carlo sampling

ABSTRACT

Clustering algorithms strive to organize data into meaningful groups in an unsupervised fashion. For some datasets, these algorithms can provide important insights into the structure of the data and the relationships between the constituent items. Clustering analysis is applied in numerous fields, e.g., biology, economics, and computer vision. If the structure of the data changes over time, we need models and algorithms that can capture the time-varying characteristics and permit evolution of the clustering. Additional complications arise when we do not have the entire dataset but instead receive elements one-by-one. In the case of data streams, we would like to process the data online, sequentially maintaining an up-to-date clustering. In this paper, we focus on Bayesian topic models; although these were originally derived for processing collections of documents, they can be adapted to many kinds of data. The main purpose of the paper is to provide a tutorial description and survey of dynamic topic models that are suitable for online clustering algorithms, but we illustrate the modeling approach by introducing a novel algorithm that addresses the challenges of time-dependent clustering of streaming data.

1. Introduction

Clustering is an unsupervised learning technique, with the goal of finding a structure or pattern in a collection of unlabeled samples. It strives to identify groups, or clusters, of similar objects. The clusters may be distinct, in the sense that each object belongs to a single cluster, or they may overlap. In an unfortunate terminology overload in the literature, the word "clustering" is used to describe both the act of identifying the clusters (the algorithm) and the set of clusters identified by the algorithm. Most clustering algorithms employ some notion of distance between objects, and also have an explicit or implicit criterion defining what constitutes a "good" clustering. The algorithms then optimize (often heuristically) this criterion to determine the clustering.

In this paper, we focus on the task of online clustering, which involves clustering a series of data items that arrive sequentially. The goal is to provide a new clustering after the arrival of each data item; generally we also want to ensure that the identified clusterings evolve smoothly over time. We require that the clustering algorithm is capable of learning the number of clusters automatically, and that the use of computational and memory resources remains bounded over time. The algorithm must be capable of taking into account the order of the data arrivals; preferably the clustering should be dependent on the actual generation or arrival times associated with each data item.

In mathematical form, the input is a sequence of data items $\{x_1, x_2, \ldots\}$, which can either be a data stream or a sequence of limited size, as long as items are received one-by-one. Each data item x_i is associated with a timestamp t_i, whose value represents the time when the item was generated or received. In most cases, we will assume that each data item x_i is a set of a discrete elements that are members of a predefined "vocabulary", $\mathcal{V}$, and we assume that each element of x_i is essential to the meaning of the item. Our goal is to infer a clustering label z_i for each data item; we need to provide the label for x_i before data item x_{i+1} arrives.

We concentrate on clustering algorithms built upon probabilistic topic models. Although these have some limitations in terms of the types of data they can represent efficiently, they have advantages over many other clustering approaches. In particular, they specify a generative probabilistic model for the clustering, which permits application of principled inference procedures, including Bayesian methods. The primary purpose of this paper is to provide a tutorial description and survey of dynamic topic models that are suitable for online clustering algorithms, but we illustrate the modeling approach by introducing a novel algorithm that employs sequential Monte Carlo sampling to address the challenges of time-dependent clustering of streaming data.

* Corresponding author.
E-mail addresses: milad.kharratzadeh@mail.mcgill.ca (M. Kharratzadeh), benjamin.renard@melix.net (B. Renard), mark.coates@mcgill.ca (M.J. Coates).

http://dx.doi.org/10.1016/j.dsp.2015.03.010
1051-2004/

1.1. Probabilistic topic models

Probabilistic topic models were developed for the analysis of large collections of documents, with the goal of identifying common themes and topics. Excellent introductions are provided in [1,2]. One of the earliest topic models was *latent Dirichlet allocation* (LDA) [3]. The key idea in LDA is that each document addresses multiple topics, and the words comprising the document can thus be considered as samples from common words employed when discussing these topics. Mathematically, each topic is defined as a distribution over a fixed vocabulary. In the probabilistic LDA model, to generate each document, a distribution over the topics is drawn from a Dirichlet distribution. To generate each of the words that comprise the document, we first draw a topic from the topic distribution, and then draw a word from the topic. The LDA generative model assumes a fixed number of topics and fixed word probabilities within each topic.

There have been many extensions of the topic model employed in LDA. Of most interest to us are (i) the extension to Dirichlet process mixture models, which allow the number of topics to be learned from the data rather than requiring specification in the prior; and (ii) the incorporation of time-dependency in the models. In Section 2 of the paper we provide an introduction to Dirichlet distributions and processes, and Dirichlet process mixture models. In Section 3 we review some of the techniques that have been proposed for injecting temporal dependency into probabilistic topic models.

1.2. Dynamic/static and online/offline distinctions

Dynamic (as opposed to static) clustering incorporates the notion of time in the dataset. Data items can either have a timestamp associated with their arrival in the dataset (e.g., a data stream), or they can evolve dynamically (e.g., geographic position of mobile users over time). A dynamic clustering algorithm then identifies clusterings that change over time.

A dynamic clustering algorithm can be either online or offline. Online clustering means that the algorithm must provide a clustering for the data associated with timestamp t before seeing any data with timestamp $t' > t$ [4]. There are two main uses for online algorithms. The first case corresponds to data streams: we receive data items sequentially and we cannot afford to wait until we have all the items to perform processing. The second arises when we have access to the entire dataset, but the dataset is too big to be processed by offline methods, motivating sequential processing of elements or batches of elements. In offline clustering, the algorithm takes as an input the entire data stream or the complete history of the dataset.

When considering datasets where each data item is associated with a timestamp, we can ask ourselves whether the temporal distance between two consecutive data items (i.e., the difference between their timestamps) is of any importance for the analysis of the dataset. If not, then we can replace the timestamp by the index of the item in the ordered dataset. This setting is useful when the time difference between two consecutive items is always the same (for example when considering articles published in a yearly journal). An algorithm that considers only the order of the data items, rather than the actual times, is called *order-dependent*. If the algorithm explicitly takes into account the time difference between data item arrivals, we say that it is *time-dependent*.

Let us consider the example of clustering marathon runners by their performance in a race. An order-dependent algorithm would only consider the order in which the runners finished. A time-dependent algorithm, however, would process the actual completion times. If we wanted to identify the top-10 finishers, the order-dependent algorithm would suffice; if our goal were to identify a group of racers who all finished within 2 minutes of each other, a time-dependent algorithm is required.

1.3. Inference

Although topic models are well-matched to many data sets, and the prior is constructed so that there is conjugacy with the commonly-assumed likelihood function for the data, exact inference is in general infeasible. We must therefore turn our attention to approximate Bayesian inference approaches; the main candidates are Markov chain Monte Carlo (MCMC) [5], variational inference [6] and Sequential Monte Carlo (SMC) samplers [7].

One of the main challenges of a data stream setting is to keep the computational resources bounded as the number of processed items increases. For online clustering, we generally do not know the number of data items we will need to process ahead of time. Section 4 reviews methods that can be used to perform online posterior inference for the dynamic topic models. We focus on sequential Monte Carlo samplers, because they are naturally suited to online processing. We also highlight some of the recent work in streaming variational Bayes [8], which adapts variational approximation methods to make them more amenable to the online dynamic clustering task.

In Section 5 we present a sequential Monte Carlo (SMC) sampling approach for performing inference for a time-dependent dynamic topic model. We build upon the work in [9,10], where Ülker et al. developed an SMC sampling approach for *static* Dirichlet process mixture models. In the static models, the probability of assignment to a cluster does not depend on the time of arrival or generation of a data item. Our focus is on the streaming, online setting, where the importance of a data item is very much associated with when it arrived, so we incorporate explicit dependence on time, and we consider how to ensure that the memory requirements of the algorithm remain bounded. We introduce a novel sampling procedure that focuses on elements whose clustering labels are more uncertain; our numerical experiments and analysis of news data indicate that this procedure allows us to either improve inference performance or reduce computational overhead.

1.4. Example application

Privacy concerns have always existed for popular social networks such as Facebook, Twitter or Google+, due to their reliance on targeted advertising. These concerns led to the creation of several privacy-focused social networks such as Diaspora [11] and Friendica [12]. These alternative social networks are peer-to-peer networks of servers that distribute data throughout the network in an attempt to maintain a high level of privacy. Each user can remain in control of his/her data by selecting which server stores it. Although this distributed architecture protects the users' privacy, it generates several problems that centralized social networks do not face. Control of the network is more limited and performance problems can arise (primarily slow response time), because nodes of the network can be self-hosted web servers with limited computational resources. Search is more challenging, because each node has access only to a limited portion of the network.

Often users want to search for other users that share a similar interest, usually by providing a set of keywords related to that interest. Centralized networks have direct access to all users' data and hence can directly determine the users whose interests match the query. On the other hand, nodes in a distributed network have only access to the data corresponding to their own users and they only know a subset of entire network, composed of their neighbors in the peer-to-peer graph. To find users on other nodes, they need to forward the query in an efficient way.

Please cite this article in press as: M. Kharratzadeh et al., Bayesian topic model approaches to online and time-dependent clustering, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.010

One way we can improve the efficiency of the search is for nodes to maintain a forwarding table, indicating to which neighbor they should forward a query to improve the chance of success. It is impossible for nodes to maintain a forwarding table for every possible keyword (or group of keywords). An alternative is to dynamically cluster previous queries; each cluster identified by the algorithm can be considered as an "interest". Each *interest* is defined by a distribution over the keywords that have been seen by the server. With this process we can learn the current search trends in real-time. When a server receives a query, it maps the query to one or more interests, and uses its forwarding table that maps interests to neighboring nodes.

1.5. Outline

The rest of the paper is organized as follows. In Section 2, we introduce the necessary background material that is required to understand and employ probabilistic topic models. In Section 3, we survey dynamic, topic-model clustering algorithms that have been proposed in the literature. Section 4 describes approximate Bayesian inference techniques that can be employed for the dynamic topic models, focusing on sequential Monte Carlo samplers. Section 5 introduces a novel time-dependent algorithm that employs a sequential Monte Carlo (SMC) sampler to perform online Bayesian inference; and Section 6 presents examples of applying the algorithms to analyze synthetic and real-world datasets. Finally, we conclude and suggest possible future research directions in Section 7.

2. Background

In this section, we present relevant background material and provide an introduction to static probabilistic topic models. Numerous clustering algorithms use Dirichlet processes [13] for static and dynamic clustering [14–19]. These nonparametric processes are very useful for clustering because they eliminate the need for an assumption and specification of a fixed number of clusters. A good introduction to these processes in particular and to Bayesian nonparametric models in general can be found in [20]. In this section, we review the Dirichlet distribution, the Dirichlet process, and Dirichlet process mixture models that can be used for clustering.

2.1. Dirichlet distributions

The Dirichlet distribution is a distribution over probability mass functions (PMFs) of finite length. Let us consider a PMF with k components. This PMF lies in the $k-1$ simplex defined by $\Delta_k = \{q \in \mathbb{R}^k | \sum_{i=1}^k q_i = 1 \text{ and } \forall i, q_i \geq 0\}$.

Let $Q = [Q_1, \ldots, Q_k]$ be a random PMF and let $\alpha = [\alpha_1, \ldots, \alpha_k]$ be a k-dimensional vector with $\forall i, \alpha_i \geq 0$. Q is said to be generated from a Dirichlet distribution with parameter α if its density satisfies $f(q|\alpha) = 0$ if $q \notin \Delta_k$ and

$$f(q|\alpha) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^k \Gamma(\alpha_i)} \prod_{i=1}^k q_i^{\alpha_i - 1} \tag{1}$$

if $q \in \Delta_k$, where $\alpha_0 = \sum_{i=1}^k \alpha_i$ and Γ denotes the Gamma function. This distribution is denoted by $Q \sim \text{Dir}(\alpha)$. The mean of this Dirichlet distribution is the vector $m = \alpha/\alpha_0$.

In Bayesian probability theory, the prior distribution $p(\Theta)$ is called a conjugate prior to the likelihood $p(x|\Theta)$ if the prior distribution is of the same family of distributions as the posterior distribution $p(\Theta|x)$. Conjugate priors are of interest because their adoption makes it possible, in some cases, to derive analytical expressions for the posterior distribution, hence simplifying computation.

The multinomial distribution is parameterized by an integer n and a PMF $q = [q_1, \ldots, q_k]$. If $X \sim \text{Multinomial}_k(n, q)$, then its PMF is given by

$$f(x_1, \ldots, x_k | n, q) = \frac{n!}{x_1! \ldots x_k!} \prod_{i=1}^k q_i^{x_i}. \tag{2}$$

The Dirichlet distribution serves as a conjugate prior for the probability parameter q of the multinomial distribution: if $X|q \sim \text{Multinomial}_k(n, q)$ and $Q \sim \text{Dir}(\alpha)$, then $Q|(X = x) \sim \text{Dir}(\alpha + x)$. This property is one of the reasons why the Dirichlet distribution is often used for clustering text corpora, in conjunction with the *bag-of-words* model. This model assumes that texts are represented as unordered collections of words, disregarding grammar and word order: only the count of each word matters. Under this assumption, the likelihood of a text is often considered to be a multinomial distribution on the vocabulary, which is why the Dirichlet distribution becomes an attractive prior distribution.

2.2. Dirichlet processes

A Dirichlet process (DP) is an extension of the Dirichlet distribution that enables us to use infinite sets of events. It is a stochastic process over a set $\mathcal{X}$ such that its sample path is a Dirichlet distribution over $\mathcal{X}$. Written as $DP(H, \alpha)$, it is characterized by a base measure H and a concentration parameter α. Let $(\mathcal{X}, \mathcal{B})$ be a measurable space where $\mathcal{X}$ is a set and $\mathcal{B}$ is a σ-algebra on $\mathcal{X}$. Let H be a finite probability measure on $(\mathcal{X}, \mathcal{B})$ and $\alpha \in \mathbb{R}_+^*$ (the strictly positive reals). If P is a random distribution generated from a $DP(H, \alpha)$ (a sample path), then for any finite measurable partition $\{B_i\}_{i=1}^k$ of $\mathcal{X}$, the random vector $(P(B_1), \ldots, P(B_k))$ has a Dirichlet distribution with parameters $(\alpha \cdot H(B_1), \ldots, \alpha \cdot H(B_k))$.

The stick-breaking process, due to Sethuraman [21], defines the DP constructively as follows. Let $(\beta'_k)_{k=1}^\infty$ and $(\beta_k)_{k=1}^\infty$ be defined as:

$$\beta'_k \sim \text{Beta}(1, \alpha) \tag{3}$$

$$\beta_k = \beta'_k \prod_{l=1}^{k-1} (1 - \beta'_l) \tag{4}$$

where $\text{Beta}(1, \alpha)$ denotes the beta distribution. Let $(\psi_k)_{k=1}^\infty$ be samples from H. Let δ be the Dirac delta measure on $\mathcal{X}$, so that $\delta_{\psi_k}(\psi) = 1$ for $\psi = \psi_k$ and 0 otherwise. The distribution given by the density

$$P(\psi) = \sum_{k=1}^\infty \beta_k \delta_{\psi_k}(\psi) \tag{5}$$

is then a sample from the Dirichlet process $DP(H, \alpha)$. Note that the sequence $(\beta_k)_{k=1}^\infty$ satisfies $\sum_{k=1}^\infty \beta_k = 1$ with probability 1. Eqs. (3)–(5) are referred to as the *stick-breaking construction* for Dirichlet processes.

In measure theory, an *atom* of a measure μ on a σ-algebra $\mathcal{S}$ of subsets of a set X is an element $A \in \mathcal{S}$ that satisfies [22]:

- $\mu(A) > 0$
- for every $B \in \mathcal{S}$ such that $B \subset A$, either $\mu(B) = 0$ or $\mu(B) = \mu(A)$

μ is an *atomic measure* if there exists a countable partition of X where each element A is either an atom or verifies $\mu(A) = 0$. A realization drawn from $DP(H, \alpha)$ is, with probability 1, an atomic distribution with infinite atoms [13], as can be seen in Eq. (5).

Please cite this article in press as: M. Kharratzadeh et al., Bayesian topic model approaches to online and time-dependent clustering, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.010

Of critical importance in the context of clustering is the *clustering property* of the Dirichlet process: samples from a DP share repeated values with positive probability. Therefore if we use a DP to generate parameters of a data item, items with the same value belong to the same cluster.

2.3. Dirichlet process mixtures

Dirichlet process mixtures (DPMs) are generative models that use a Dirichlet process as a nonparametric prior on the parameters of a mixture. In the context of clustering, they define a procedure for generating clusters with associated parameters θ_i, and then associating a cluster label z_i with each data item x_i. The cluster parameters θ_i are drawn from a distribution G, which is generated from a base Dirichlet process $\mathrm{DP}(H,\alpha)$. As such, the model does not require us to specify a fixed number of clusters ahead of time.

The generative model determines the cluster parameters θ_i and the observation x_i as follows:

$$G \sim \mathrm{DP}(H,\alpha) \tag{6}$$

$$\theta_i \sim G \tag{7}$$

$$x_i \sim F(\theta_i) \tag{8}$$

where $F(\theta_i)$ represents the distribution of the observation x_i given the parameter θ_i, and is therefore problem-dependent. The components of the Dirichlet process, $\boldsymbol{\beta} = (\beta_k)_{k=1}^{\infty}$ and $\boldsymbol{\psi} = (\psi_k)_{k=1}^{\infty}$, are defined as above. We also introduce a cluster assignment variable z_i such that $z_i \sim \boldsymbol{\beta}$. With these notations, the DPM model is equivalent $\theta_i = \psi_{z_i}$ and $x_i \sim F(\psi_{z_i})$.

The DPM model can also be explained using the Chinese restaurant process (CRP) metaphor [23] where we have a restaurant with an infinite number of tables (clusters) with customers (data items) arriving one-by-one in the restaurant. Each customer chooses to sit at an existing table and share the dish (cluster parameters) already served at this table with a probability proportional to the number of customers seated at that table. The customer can also sit at a new table with some probability and choose a new dish from the menu (the menu being common to all the tables). An important property of this process is that data items are fully exchangeable; this simplifies inference when using models based on the process. Exchangeability implies that the probability distribution of the table (i.e., cluster assignment) is unchanged if the order in which the customers arrive is shuffled. A proof of this property can be found in [20].

3. Dynamic topic models

In this section, we describe how topic models based on Dirichlet processes have been extended to address time-varying structure in the data. We divide the discussion into two sections, describing first the clustering models that group the data into "epochs" or discrete time intervals, and then turning our attention to those that incorporate dependencies on real-valued time information.

3.1. The epoch approach — discrete time intervals

Incorporating unconstrained temporal dynamics directly into DPM models generally leads to inference problems that are very challenging. More tractable inference becomes possible if one focuses on the setting where time is indexed by a countable data set (e.g., $t \in \mathbb{N}$) and the data items are grouped by *epochs* (e.g., a year for analyzing scientific articles). The exchangeability property can then be preserved for each epoch (as opposed to the entire dataset). Since the actual timestamps of the data items are discarded, the epoch-based models can only support time- or order-dependent clustering in a limited sense, i.e., at the time-scale of the epochs.

In one of the earlier works adopting this approach [24], Blei and Lafferty extended the topic model approach introduced in [3], incorporating a state space model to capture time-variation of the topic mixture weights and the parameters of the multinomial distributions that describe the topics. This formulation allowed Blei and Lafferty to derive variational methods, based on the Kalman filter or wavelet regression, to conduct inference using the model.

In [16], Xu et al. consider the task of determining a suitable clustering for each epoch, while ensuring that the clustering parameters vary smoothly over time. Their approach is to build a time-varying model by extending the DPM model. For each epoch, they first use the stick-breaking construction to generate intermediate mixture weights $\boldsymbol{\beta}_t$, and then model the actual weights from which the topic mixture is drawn as $\boldsymbol{\pi}_t = \sum_{\tau=1}^{t} \exp\{-\eta(t-\tau)\}\boldsymbol{\beta}_\tau$, i.e., an exponentially smoothed averaging of historical mixture weights, with the constant η controlling the exponential decay.

In [18], Ahmed and Xing present the *Temporal Dirichlet process mixture* (TDPM) model. Instead of considering fully exchangeable data items, they assume that the data items are only exchangeable if they belong to the same epoch. The recurrent Chinese restaurant process (RCRP) metaphor is used to describe the framework. In this metaphor, customers (data items) enter on a given day (epoch) and leave the restaurant at the end of this day. When a customer arrives on day t, she joins a table (cluster) k that already existed on day $t-1$ with some probability. If she is the first to sit at that table on day t, then she chooses the dish (cluster parameters), drawing from a distribution that is parameterized by the previous day's parameters for the same table. The distribution is chosen in order to ensure a smooth evolution of the clusters over time. The customer can also pick a new empty table, or join an existing table created by a previous customer who arrived in the same epoch t. For these latter cases, the model behaves in exactly the same way as the DPM model described in Section 2.3.

In the generalized Polya urn (GPU) scheme introduced in [25], Caron et al. aim to develop a model which marginally preserves a Dirichlet process for each epoch. The model is built around a base Dirichlet process $DP(G_0,\alpha)$. A parameter $\rho \in [0,1]$ is introduced to control the "closeness" of the clusterings at epochs $t-1$ and t. During epoch t, the new data items are assigned to clusters according to a DPM, but evolution of the model is achieved by deletion of previous allocations of data items to clusters. The deletion may be uniform with probability $1-\rho$ across all data items; or size-biased (the number of items deleted from each cluster is proportional to the size of the cluster); or based on a sliding window, such that only the previous r epochs are considered (implying $\rho = 1 - \frac{1}{1+r}$). The model also introduces evolution of the parameters of any clusters that persist from epoch $t-1$ to t. This is achieved by sampling from a kernel $p(\psi_{i,t}|\psi_{i,t-1})$ which has invariant G_0, i.e.,

$$\int G_0(\psi_{i,t-1})\,p(\psi_{i,t}|\psi_{i,t-1})\,d\psi_{i,t-1} = G_0(\psi_{i,t}). \tag{9}$$

3.2. More general temporal dependence

In one of the earlier works addressing the discovery of the dynamic evolution of latent themes or topics inside a collection of texts, Mei and Zhai propose a generative model that used a hidden Markov model to capture the evolution of the topics [26]. In this model, however, the number of topics and their parameters were assumed known (in [26], they were learned using a separate procedure that divided the data into subcollections based on timestamps and matched topics across the subcollections).

Please cite this article in press as: M. Kharratzadeh et al., Bayesian topic model approaches to online and time-dependent clustering, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.010

3.2.1. Dependent Dirichlet processes

A preferable approach is to propose a model that allows one to jointly learn the number of topics, their evolution over time, and the parameters that describe them. Srebro and Roweis discuss in [27] how such topic models can be constructed using *dependent Dirichlet processes* (DDPs) [28]. A DDP is a process $G(t)$, defined over a set $t \in T$ such that for any t, $G(t)$ is marginally a Dirichlet process. With such a construction, it is possible to vary the nature of $G(t)$ over time to capture the evolution of topic distributions. If the Dirichlet process mixture is generated using $G(t)$, as opposed to the static G in (6), then we can introduce time-variation by evolving the weights in the topic mixture (to capture topic appearances, disappearances and popularity changes) and the weights in the word distributions for each topic (to capture evolution of the nature of the topics themselves). Srebro and Roweis describe how dynamic topic models can be constructed using the order-based dependent Dirichlet process introduced by Griffin and Steel [29], the stationary autoregressive model of Pitt et al. [30], and through a transformation of Gaussian processes [27]. In [31], Rao and Teh introduce a DDP-based model called a *spatial normalized Gamma process*. The proposed model constructs dependent Dirichlet processes by marginalizing and normalizing a single Gamma process over an extended space.

3.2.2. Time-sensitive Dirichlet process mixture model

In general, inference for DDPs is significantly more challenging than for classical Dirichlet Process mixtures, and Markov Chain Monte Carlo methods are the usual approach. This has motivated work towards models that do not provide all of the desirable theoretical properties encapsulated by DDPs, but are more amenable for practical, on-line inference. The time-sensitive Dirichlet process mixture (TS-DPM) model, proposed by Zhu et al. in [14], employs a temporal weight function for each cluster that depends on the cluster assignment history. Together, the time-varying weights specify a prior probability on the cluster assignment for each arriving data item and allow evolution of the prevalence of topics. We provide a more detailed description of the TS-DPM in Section 5, where we explain how it can be used in conjunction with a sequential Monte Carlo sampler to derive an online, time-dependent clustering algorithm.

4. Posterior inference

One of the challenges with DP-based generative models is that the computation of the posterior distribution of the parameters is usually intractable. Inference of the posterior allows us to perform clustering (determining the most probable cluster assignment) and prediction (predicting the attributes of the next data item). Since exact inference is not an option, approximate inference techniques are employed, and there are three main approaches: Markov chain Monte Carlo (MCMC) [5], variational inference [6] and Sequential Monte Carlo (SMC) samplers [7].

The majority of MCMC inference techniques are not well-suited to the online setting. There has been more work in adapting variational methods to sequential processing of the data. Several of these methods target the setting when the full data set is available, but due to its size, processing the entire data set as a batch is computationally infeasible [32–35]. These techniques rely on advance knowledge of the number of data items and it is challenging to introduce adaptations to make them suitable for processing data streams.

In [8], Broderick et al. introduce a framework to make streaming Bayesian updates to the estimated posterior using variational approximation methods. The data is divided into batches $C_1, \ldots, C_b$ and the posterior updated in the usual Bayesian way: $p(\Theta|C_1, \ldots, C_b) \propto p(C_b|\Theta)p(\Theta|C_1, \ldots, C_{b-1})$. In the dynamic topic models, the update is not computationally tractable, so Broderick et al. use an approximating algorithm, $\mathcal{A}$, to propagate an approximate posterior: $p(\Theta|C_1, \ldots, C_b) \approx q_b(\Theta) \propto \mathcal{A}(C_b, q_{b-1}(\Theta))$. For the variational approximation approach it is simplest if q_b is exponential, i.e., $q_b(\Theta) \propto \exp\{\zeta_b T(\Theta)\}$, for some parameter ζ_b and sufficient statistic $T(\Theta)$. This methodology is better suited to order-dependent clustering; the restriction to exponential distributions can limit the types of data that can be successfully modeled.

4.1. Sequential Monte Carlo methods

Sequential Monte Carlo (SMC) methods, such as the particle filter (PF), can be used to approximate a sequence of probabilities $\{\pi_n\}_{n\in\mathbb{T}}$ (e.g., $\mathbb{T} = \mathbb{N}$) sequentially, i.e., by inferring π_1, then π_2, and so on. We assume that $\pi_n(\Theta_n)$ is defined on a measurable space $(E_n, \mathcal{E}_n)$ for $\Theta_n \in E_n$.

The main idea behind the SMC methods is to obtain, at each time n, a large collection of N weighted random particles $\{\Theta_n^i, w_n^i\}_{i=1}^N$ with $w_n^i > 0$ and $\sum_i w_n^i = 1$, whose empirical weighted distribution converges asymptotically to π_n. To achieve this, we represent the target distribution in the form $\pi_n(\Theta_n) = \frac{\gamma_n(\Theta_n)}{Z_n}$, assuming that γ_n is known point-wise and Z_n is an unknown normalizing constant. To sample from this distribution, we introduce a known importance distribution $\eta_n(\Theta_n)$ whose exact definition is problem-specific. The unnormalized importance weight function $\widetilde{w}_n(\Theta_n)$ is defined by:

$$\widetilde{w}_n(\Theta_n) = \frac{\gamma_n(\Theta_n)}{\eta_n(\Theta_n)}. \tag{10}$$

We sample N particles $\{\Theta_n^i\}$ from η_n, calculate the unnormalized weights $\widetilde{w}_n$, and then normalize. The weighted particle set $\{\Theta_n^i, w_n^i\}_{i=1}^N$ then provides an approximation of the target distribution π_n.

In a sequential implementation, we assume that we have N particles at timestep $n-1$, $\{\Theta_{n-1}^i\}_{i=1}^N$, distributed according to $\eta_{n-1}(\Theta_{n-1})$. We then define a kernel function $K_n(\Theta_{n-1}, \Theta_n)$ that we can use to move the N particles obtained at time $n-1$ from $\{\Theta_{n-1}^i\}$ to construct an importance distribution at time n: $\{\Theta_n^i\}$. These particles are marginally distributed as $\eta_n(\Theta_n) = \int_E \eta_{n-1}(\Theta_{n-1})K_n(\Theta_{n-1}, \Theta_n)\mathrm{d}\Theta_{n-1}$. Several strategies exist to select the forward kernel sequence $\{K_n\}$; these include MCMC kernels or approximate Gibbs moves (see [7] for more examples and details).

A significant limitation of the traditional SMC approach is that closed-form expressions for $\eta_n(\Theta_n)$ cannot be derived for many kernels of practical interest, and this means that the importance weights cannot be calculated. Del Moral et al. circumvent this in [7], turning attention to the joint posterior $\widetilde{\pi}_n$ defined on $E^n = E_1 \times E_2 \times \cdots \times E_n$. We introduce a distribution $\widetilde{\gamma}_n$ such that:

$$\widetilde{\pi}_n(\boldsymbol{\Theta}_{1:n}) = \frac{\widetilde{\gamma}_n(\boldsymbol{\Theta}_{1:n})}{Z_n}, \tag{11}$$

where

$$\widetilde{\gamma}_n(\boldsymbol{\Theta}_{1:n}) = \gamma_n(\Theta_n)\prod_{k=1}^{n-1} L_k(\Theta_{k+1}, \Theta_k). \tag{12}$$

Here $L_k : E_k \times \mathcal{E}_k \to [0, 1]$ is an artificial backward Markov kernel. The sequential sampling framework then conducts importance sampling between the joint importance distribution $\eta_n(\boldsymbol{\Theta}_{1:n})$ and the target joint posterior $\widetilde{\pi}_n(\boldsymbol{\Theta}_{1:n})$.

The key advantage of this approach is that there is no longer a need to explicitly evaluate the importance sampling distribution. At time n, the path of each particle is extended using a Markov

Please cite this article in press as: M. Kharratzadeh et al., Bayesian topic model approaches to online and time-dependent clustering, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.010

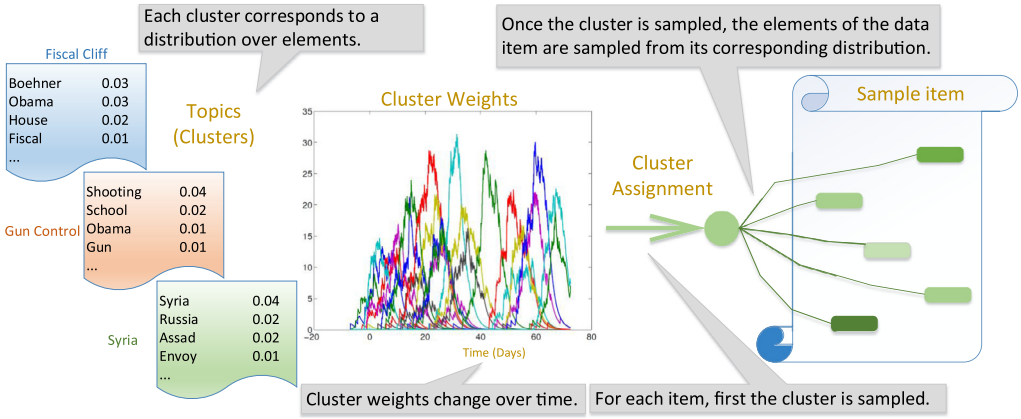


Fig. 1. The generative time-sensitive Dirichlet process mixture model [14]. Each cluster is associated with parameters that specify a probability distribution over the elements of a *vocabulary*. This distribution determines the probability of elements appearing in a data item belonging to that cluster. The probability of assigning an item to a specific cluster (the "weight" of that cluster) evolves over time.

kernel $K_n(\Theta_{n-1}, \Theta_n)$. The new expression of the unnormalized importance weights is:

$$\widetilde{w}_n(\boldsymbol{\Theta}_{1:n}) = \frac{\widetilde{\gamma}_n(\boldsymbol{\Theta}_{1:n})}{\eta_n(\boldsymbol{\Theta}_{1:n})} = w_{n-1}(\boldsymbol{\Theta}_{1:n-1}) v_n(\Theta_{n-1}, \Theta_n) \tag{13}$$

where the unnormalized incremental weight $v_n(\Theta_{n-1}, \Theta_n)$ is:

$$v_n(\Theta_{n-1}, \Theta_n) = \frac{\gamma_n(\Theta_n) L_{n-1}(\Theta_n, \Theta_{n-1})}{\gamma_{n-1}(\Theta_{n-1}) K_n(\Theta_{n-1}, \Theta_n)}. \tag{14}$$

The particles weights $\{w_n^{(i)}\}$ are then obtained by normalization. As we can see from (13) and (14), the weight update no longer involves direct calculation of η_n.

The performance of this approach depends critically on the choice of the backwards kernel L and how well it matches the forward kernel K. Del Moral et al. provide guidelines in [7] for identifying suitable backward kernels $\{L_n\}$. The optimal kernels (those minimizing the variance of the unnormalized importance weights) are given by [7]:

$$L_{n-1}^{\text{opt}}(\Theta_n, \Theta_{n-1}) = \frac{\eta_{n-1}(\Theta_{n-1}) K_n(\Theta_{n-1}, \Theta_n)}{\eta_n(\Theta_n)} \tag{15}$$

and these lead to weights $w_n(\boldsymbol{\Theta}_{1:n}) = \gamma_n(\Theta_n)/\eta_n(\Theta_n)$.

These optimal weights rarely admit a closed-form expression; an alternative sub-optimal approach is to replace η_{n-1} with π_{n-1}. This leads to backwards kernels of the form:

$$L_{n-1}(\Theta_n, \Theta_{n-1}) = \frac{\pi_{n-1}(\Theta_{n-1}) K_n(\Theta_{n-1}, \Theta_n)}{\pi_{n-1} K_n(\Theta_n)}. \tag{16}$$

The unnormalized incremental weights are then:

$$v_n(\Theta_{n-1}, \Theta_n) = \frac{\gamma_n(\Theta_n)}{\int_{E_{n-1}} \gamma_{n-1}(\Theta_{n-1}) K_n(\Theta_{n-1}, \Theta_n)\, d\Theta_{n-1}}. \tag{17}$$

As in conventional particle filtering, the difference between the target posterior and the sampling distribution may increase over time, so resampling is performed if the *effective sample size (ESS)*, $(\sum_{i=1}^N (w_n^i)^2)^{-1}$, is below a pre-defined threshold. The resampling process involves sampling N new particles with equal weights from the weighted empirical distribution of $\tilde{\pi}_n$: $\tilde{\pi}_n^N(\mathrm{d}\boldsymbol{\Theta}_{1:n}) = \sum_{i=1}^N w_n^i \delta_{\Theta_n^i}(\mathrm{d}\boldsymbol{\Theta}_{1:n})$.

5. Online time-dependent clustering

In this section, we illustrate how the sequential Monte Carlo sampler can be integrated with the TS-DPM model to develop a time-dependent, online clustering algorithm. We introduce modifications to the posterior inference procedure to ensure that memory requirements remain bounded over time and to improve the efficiency of the sampling process.

5.1. Generative model: TS-DPM

We now provide a more detailed description of the TS-DPM model [14] described in Section 3.2. Fig. 1 provides a pictorial representation of the generative model. Consider a sequence of N_d data items $x_{1:N_d}$, where each item x_i, $1 \le i \le N_d$ is associated with a time stamp $t_i \in \mathbb{R}$, and assume that data items are ordered in chronological order. Denote by $z_i \in \mathbb{N}$ the cluster index of item x_i. The TS-DPM assigns a temporal weight function $g(t,k)$ for each cluster index k that depends on the current time t and the collection of previous assignments $\{z_1, \ldots, z_{i-1}\}$:

$$g(t,k) = \sum_{j|t_j<t} \kappa(t - t_j) \cdot \delta(z_j, k)\,. \tag{18}$$

Here κ is a kernel function (e.g., $\kappa(\tau) = \exp(-\lambda\tau)$) and the Kronecker function $\delta(a,b) = 1$ if $a = b$ and 0 otherwise.

The prior probability of assigning x_i to cluster k given the history $\{z_1, \ldots, z_{i-1}\}$ is then defined as follows:

$$p(z_i = k | z_1, \ldots, z_{i-1}) = p(z_i = k | \{w(t_i, j)\}) = \begin{cases} \dfrac{g(t_i,k)}{\sum_{k'} g(t_i,k') + \alpha} & \text{if } k \in \{z_1, \ldots, z_{i-1}\} \\ \dfrac{\alpha}{\sum_{k'} g(t_i,k') + \alpha} & \text{if } k \text{ is a new cluster} \end{cases}. \tag{19}$$

We can observe the obvious link with the DPM model: instead of every past data item having a weight of 1 in determining the probability of cluster assignment, the weight changes depending on how long ago the data item arrived. The kernel function κ dictates how the weight is impacted by the time difference of the data items; a natural choice is a non-negative, decreasing function such that $\lim_{\tau\to\infty} \kappa(\tau) = 0$. We use $\kappa(\tau) = \exp(-c\tau)$ in the algorithmic implementations, with the constant c chosen to reflect how rapidly clusters change over time.

We now make the model more specific, and assume that each cluster index k is associated with a multinomial distribution θ_k over the set of all possible elements in a fixed vocabulary $\mathcal{V}$: $p(x_i | z_i = k) = \prod_{v\in\mathcal{V}} \theta_k(v)^{x_i(v)}$, where $x_i(v)$ is the number of time element v appears in item x_i. Assume that for each cluster k, the prior on θ_k is a Dirichlet distribution G_0 with parameter vector

Please cite this article in press as: M. Kharratzadeh et al., Bayesian topic model approaches to online and time-dependent clustering, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.010

$\beta \boldsymbol{m}$, where $\boldsymbol{m}$ is a vector of size V representing a base measure on the vocabulary and $\beta \in \mathbb{R}$ specifies the concentration.

5.2. Posterior inference

For $n \in \{1, \ldots, N_d\}$, let $y_n = (x_n, t_n)$ be the n-th observation. Let $\boldsymbol{z}_n$ denote the vector of cluster assignments at this time, so that $z_{n,j}$, with $n \geq j$, denotes the cluster assignment of item y_j after the n-th item has been seen. Thus $\boldsymbol{z}_n = \{z_{n,1}, \ldots, z_{n,n}\}$. Posterior inference in the TS-DPM model aims at inferring $\boldsymbol{z}_n$ given $\boldsymbol{x}_{1:n}$ and $\boldsymbol{t}_{1:n}$. Let $\pi_n(\boldsymbol{z}_n) = p(\boldsymbol{z}_n | \boldsymbol{y}_{1:n})$ denote the posterior distribution.

According to Bayes' rule, we have:

$$p(z_{n,i} = k | \boldsymbol{z}_{n,-i}, \boldsymbol{x}_{1:n}) \propto p(z_{n,i} = k | \boldsymbol{z}_{n,-i}) p(x_i | \boldsymbol{x}_{-i:\boldsymbol{z}_{n,-i}=k}) \tag{20}$$

where $\boldsymbol{z}_{n,-i} = \{z_{n,j} | j \neq i\}$ and $\boldsymbol{x}_{-i:\boldsymbol{z}_{n,-i}=k} = \{x_j | j \neq i, z_j = k\}$. The first term of the right side can be expressed as:

$$p(z_{n,i} = k | \boldsymbol{z}_{n,-i}) \propto p(z_{n,i} = k | \boldsymbol{z}_{n,1:i-1}) \left(\prod_{m=i+1}^{n} p(z_{n,m} | \boldsymbol{z}_{n,1:m-1}) \right)$$

which in turn can be expanded using Eq. (19).

We can integrate out the cluster parameters $\{\theta_k\}$ to derive the likelihood as in [14]:

$$\begin{aligned} p(x_i | \boldsymbol{x}_{-i:\boldsymbol{z}_{n,-i}=k}) &= \int p(x_i|\theta) p(\theta | \boldsymbol{x}_{-i:\boldsymbol{z}_{n,-i}=k}) d\theta \\ &= \frac{\Gamma(\sum_v f_v + \beta)}{\prod_v \Gamma(f_v + \beta m_v)} \frac{\prod_v \Gamma(x_i(v) + f_v + \beta m_v)}{\Gamma(\sum_v x_i(v) + \sum_v f_v + \beta)} \end{aligned} \tag{21}$$

where f_v denotes the counts of word with index $v \in \{1, \ldots, V\}$ in $\boldsymbol{x}_{-i:\boldsymbol{z}_{n,-i}=k}$.

We use an SMC sampler to approximate the posterior distribution. Ülker et al. present an SMC sampler for static Dirichlet process mixture models in [9,10], and we build on some of their methods to construct a sampler suitable for the TS-DPM model. An important difference arises because of the weight terms in (19). An unnormalized expression of $\pi_n(\boldsymbol{z}_n)$ can be obtained using Eq. (20). Let us denote by $\gamma_n(\boldsymbol{z}_n)$ this unnormalized posterior and let Z_n denote the normalizing constant: $\pi_n(\boldsymbol{z}_n) = \gamma_n(\boldsymbol{z}_n)/Z_n$. As $\gamma_n(\boldsymbol{z}_n)$ is known point-wise, we can apply importance sampling and hence we can use the SMC sampler framework introduced in [7].

The sampling procedure needs to update some of the previous cluster assignments to take account of the new information provided by the data item x_n. Following the approach proposed in [9], when processing the n-th data item, we partition the assignment vector $\boldsymbol{z}_n$ into three subsets $\boldsymbol{z}_{n,r}$, $\boldsymbol{z}_{n,d}$ and $\{z_{n,n}\}$. r is a subset of $\{1, \ldots, n-1\}$ and is called the *active set*. It contains the indices of the previous assignments that we re-evaluate when introducing the new observation y_n. The set d indicates the data items whose labels are not changed, i.e. $d = \{1, \ldots, n-1\} - r$. In [9], Ülker et al. initialize r as $\{1, \ldots, Q\}$ and then increase all the indices of r by Q (modulo n) every time step. In essence, this strategy is a superposition of a particle filter with a Gibbs sampler running in the background to improve its accuracy. We propose a more effective method to determine and update r later in this section.

The proposal distribution that minimizes the variance of the incremental importance weights in (17) is a Gibbs update [7]:

$$K_n(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n) = \delta_{\boldsymbol{z}_{n-1,d}}(\boldsymbol{z}_{n,d}) \pi_n(\boldsymbol{z}_{n,r}, z_n | \boldsymbol{z}_{n,d}). \tag{22}$$

This choice indicates that in transitioning from time $n-1$ to n, we hold the assignments in the subset d fixed, and draw a new value z_n and update $\boldsymbol{z}_r$ according to π_n (conditioned on these fixed values). Using the suboptimal backwards kernel of (16), we have:

$$L_n(\boldsymbol{z}_n, \boldsymbol{z}_{n-1}) = \delta_{\boldsymbol{z}_{n,d}}(\boldsymbol{z}_{n-1,d}) \pi_{n-1}(\boldsymbol{z}_{n-1,r} | \boldsymbol{z}_{n-1,d}) \ . \tag{23}$$

With these choices, the incremental importance weights in this model are given by:

$$v_n(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n) = \frac{\gamma_n(\boldsymbol{z}_{n,d})}{\gamma_{n-1}(\boldsymbol{z}_{n-1,d})}. \tag{24}$$

5.3. Algorithmic considerations for online operation

1. Annealing: In some cases, it is sensible to allow greater freedom for discovery of new clusters and to provide assistance to the inference procedure in exploring the space of candidate clusterings. We can achieve this through an annealing process, as outlined in [9,10]. Essentially, we replace the novelty parameter α in (19) with a value α_n that changes each time we process an item,

$$\alpha_n = \alpha_{n-1} + c_\alpha (\alpha - \alpha_{n-1}) \ . \tag{25}$$

We select the initial value α_1, the limit α, and the annealing update parameter c_α to achieve a balance between giving the algorithm freedom to discover new clusters and providing convergence to the rate at which we believe new clusters genuinely emerge in the dataset.

Our experience is that when the topics are evolving relatively rapidly, annealing is unnecessary, and provides little or no advantage. This is the case for the numerical experiments we perform in the next Section and for the news dataset we analyze. When the evolution is much slower, and approaches the static setting, then the improvement is more noticeable, as illustrated by the numerical experiments in [9,10].

2. Bounded memory and computation: The identification of an *active set* reduces the computational overhead of the algorithm (if we tried to resample all elements, the computational requirement would increase over time). But in [9,10], the active set is allowed to range over the entire data history, i.e., it is a subset of the elements $\{1, \ldots, n-1\}$. This makes sense for a static model, but is inappropriate for a streaming setting, because it means that memory requirements for the algorithm grow over time, and memory is dedicated to data items that have little relevance to existing clusters.

To achieve bounded memory requirements, we can limit the sampling set to $\{x_i | t - t_i < \tau_{\text{lim}}\}$, for a constant τ_{lim}. The choice of the kernel $\kappa(\tau)$ dictates a suitable value of τ_{lim}; it should be chosen such that an item with timestamp $t - \tau_{\text{lim}}$ has minimal impact on the cluster assignment of an item with timestamp t. Since we do not allow cluster assignments of the old items to change, we can delete them from memory as soon as their time of influence expires (i.e., after a period τ_{lim}). This reduces the memory requirements as well as the computation time, because we need to process fewer elements to evaluate the assignment probabilities for each new item.

If we use an exponential kernel, $\kappa(\tau) = \exp(-c\tau)$, we can still incorporate the combined effect of the old items in the assignment probability calculations. For each existing cluster k, we store the quantity $\widetilde{g}_k = \sum_{t_i < t_s | z_i = k} e^{-\lambda(t_s - t_i)}$, where t_s denotes the timestamp of the last deleted item. Then, we can compute the weight of cluster k at time $t > t_s$ as follows: $g(t,k) = \sum_{x_i | t_s < t_i < t} e^{-\lambda(t-t_i)} \delta(z_i, k) + e^{-\lambda(t-t_s)} \widetilde{g}_k$. Note that we only need t_s, $\{\widetilde{g}_k\}_k$, and the count of words in deleted items to compute the likelihoods, priors and consequently, weight updates. Thus, while preserving the principles of the SMC sampler, this enhancement reduces the required history size and computation time and keeps them bounded as new items arrive.

Please cite this article in press as: M. Kharratzadeh et al., Bayesian topic model approaches to online and time-dependent clustering, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.010

3. Targeted sampling: In [9], the choice of the cluster assignments to sample (the active set) is determined solely based on the arrival times of the items. The efficiency of the sampler can be improved if we can target those assignments where the uncertainty is greater. To quantify uncertainty in the cluster assignment, for each item, we introduce a sample uncertainty metric ρ, defined as $\rho(z_{n-1,j}) = 1/\sum_{k=1}^{K_{n-1}} \widehat{p}_{n-1,j}(k)^2$, where $\widehat{p}_{n-1,j}(k) = \sum_{i=1}^{N} w_{n-1}^i \delta(z_{n-1,j}^i = s_k)$ and $\mathcal{K}_{n-1} = \{s_1, \ldots, s_{K_{n-1}}\}$ is the set of all cluster labels identified by the particles at time $n-1$. We have $1 \leq \rho \leq K_{n-1}$; the lower bound is obtained when one of the probabilities is one and the upper bound is obtained when all probabilities are equal. A higher value of ρ means more uncertainty in the cluster assignment. We still identify an active set, but expand its size (the number of elements in r) from Q to Q'. We now resample assignment $z_{n-1,j}$ within the active set with a probability proportional to $\rho_{n-1,j}$. We only resample Q of the cluster assignments, so aside from the minor overhead involved in computing the uncertainty metric, there is no increase in the computational requirements. The proposed *targeted sampling* strategy has similarities with random scan Gibbs samplers [36] and adaptive Gibbs samplers [37].

6. Application to synthetic and real-world datasets

In this section, we evaluate the performance of the online, time-dependent clustering algorithm described in Section 5 using synthetic and real-world datasets.

6.1. Synthetic dataset

We build a synthetic dataset, which is a time-dependent extension of the dataset presented in [38]. We consider a fixed vocabulary size, $V = 128$, with a fixed number of clusters, $N_k = 15$. Each cluster is characterized as a uniform distribution over a set of vocabulary elements (ranging uniformly in size between 10 and 15). Each data item has between 3 and 7 elements drawn from its associated cluster. Items arrive according to a Poisson process with rate $\lambda = 30$ items per day. Unlike the dataset [38], we assume that the popularity of each cluster evolves over time and is specified by a weighted Gaussian, with weight drawn uniformly between 1 and 5, mean drawn uniformly over the time-interval of data item generation, and standard deviation uniform between 2.5 and 5 days. When an item is generated in the data set, the probability of its assignment to a given cluster is proportional to the current popularity of the cluster. We generate a dataset comprised of $N_d = 500$ elements. Fig. 2(a) shows how the weights (popularities) of the individual clusters vary over time.

For the simulated dataset, we have access to both the true cluster assignment (determined when we create the dataset) and the assignment provided by the algorithm. We can therefore employ clustering evaluation metrics such as the normalized mutual information (NMI) and the f-measure [39]. Let $\boldsymbol{c}$ denote the true clustering with cluster label c_i for data item i and let $\boldsymbol{z} = \{z_i\}$ denote the label assigned by the algorithm. The f-measure (in its most common form) is defined as:

$$F = \frac{2PR}{P+R} \tag{26}$$

where P is the precision and R is the recall, defined in this case as:

$$P = \frac{\sum_{i=1}^{N_d}\sum_{j=1}^{N_d} \mathbb{I}(c_i = c_j)\mathbb{I}(z_i = z_j)}{\sum_{i=1}^{N_d}\sum_{j=1}^{N_d} \mathbb{I}(z_i = z_j)} \quad ;$$

$$R = \frac{\sum_{i=1}^{N_d}\sum_{j=1}^{N_d} \mathbb{I}(c_i = c_j)\mathbb{I}(z_i = z_j)}{\sum_{i=1}^{N_d}\sum_{j=1}^{N_d} \mathbb{I}(c_i = c_j)} \tag{27}$$

Table 1
Performance comparison for synthetic data with non-annealed SMC sampler.

Model	Sampling scheme	NMI	f-measure
TS-DPM	non-targeted	0.81 (0.02)	0.69 (0.04)
	targeted	0.90 (0.01)	0.86 (0.02)
TDPM	non-targeted	0.74 (0.01)	0.51 (0.03)
	targeted	0.81 (0.01)	0.67 (0.02)
GPU	non-targeted	0.78 (0.01)	0.57 (0.02)
	targeted	0.82 (0.02)	0.70 (0.04)

where $\mathbb{I}$ denotes the indicator function. The normalized mutual information NMI between the true assignments $\boldsymbol{c}$ and the algorithm assignments $\boldsymbol{z}$ is defined as follows:

$$NMI(\boldsymbol{z}, \boldsymbol{c}) = \frac{2I(\boldsymbol{z}, \boldsymbol{c})}{H(\boldsymbol{c}) + H(\boldsymbol{z})} \tag{28}$$

where $I(\boldsymbol{z}, \boldsymbol{c})$ is the mutual information:

$$I(\boldsymbol{z}, \boldsymbol{c}) = \sum_{k=1}^{N_d}\sum_{j=1}^{N_d} \frac{|z_k \cap c_j|}{N_d} \log \frac{N_d |z_k \cap c_j|}{|z_k||c_j|} \tag{29}$$

$H(\boldsymbol{c})$ is the entropy of the clustering $\boldsymbol{c}$, defined by:

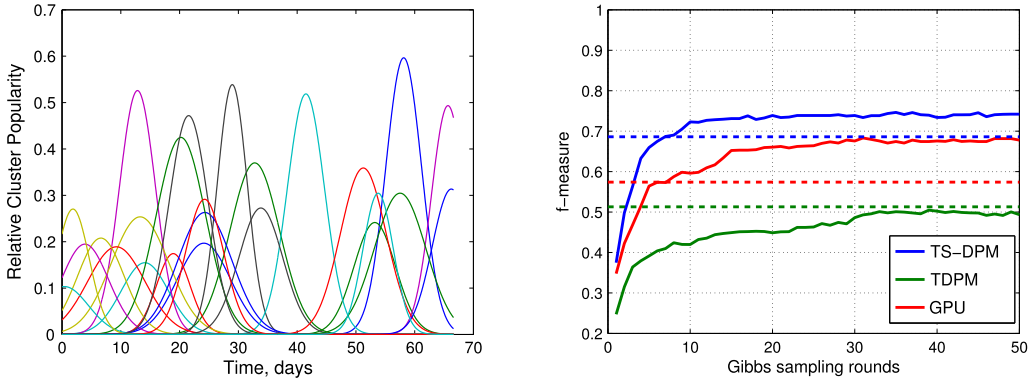
$$H(\boldsymbol{c}) = -\sum_j \frac{|c_j|}{N_d} \log \frac{|c_j|}{N_d}. \tag{30}$$

Both the f-measure and the NMI have a value in the range $[0, 1]$, and larger values indicate better clusterings (closer matches to the ground truth) in both cases.

We compare the results of three algorithms with different generative models (TS-DPM [14], TDPM [18] and GPU [25]) each paired with a SMC sampler. The results are presented for the non-annealed sampler in Table 1; the results for an annealed sampler are similar and hence not shown. Means and standard deviations are evaluated using multiple Monte Carlo runs. For the TDPM and GPU models, we use 1 day as the epoch. For the GPU model, we choose $\rho = 0.4$ (as the best value for performance evaluated over the set $\{0.2, 0.4, 0.6, 0.8\}$). For the TS-DPM model, we have $\lambda = 0.7$ (kernel parameter), $\beta = 1$ (prior parameter) and $\alpha = 1.25$ (novelty parameter). These values were chosen to approximately match the longevities of the popular clusters and the average rate of arrival of "new" clusters. The SMC samplers use $N_p = 100$ particles (chosen based on performance over the set $\{10, 20, 50, 100, 500\}$). For the non-targeted sampling, the active set is of size $Q = 8$. The active set size was chosen to achieve a reasonable compromise between computational efficiency and clustering accuracy. For the targeted sampling we increase the size of the active set to $Q' = 20$, but only resample $Q = 8$ elements from within the set. After each data item has been processed, we perform standard particle filter resampling, using a systematic resampling approach, if the effective sample size (ESS) drops below a threshold of $3N_p/4$. In the modification we introduce to bound memory requirements, we use a kernel limit of $\tau_{\text{lim}} = 3$ days, a value chosen based on the average duration of the popularity of a topic.

We see from Table 1 that the TS-DPM framework outperforms the other two generative models. This is probably due to its ability to take into account the actual time differences between data items, allowing it to better model the cluster popularity evolution. The proposed targeted sampling scheme improves the performance of all three algorithms. Fig. 2(b) compares the performance of the non-targeted SMC sampler to a block-based Gibbs sampler. We see that the block-based processing achieves a substantial improvement in estimation performance, although this improvement is eliminated when targeted sampling is incorporated (see Table 1).

Please cite this article in press as: M. Kharratzadeh et al., Bayesian topic model approaches to online and time-dependent clustering, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.010



(a) Weight evolution (b) Gibbs sampling comparison

Fig. 2. *(a)*: The evolution of the weights associated with individual clusters as used in the generation of the synthetic data set. *(b)*: A comparison between the achieved f-measures for the synthetic data for various algorithms and sampling approaches. Dashed lines indicate the non-targeted sequential Monte Carlo sampling. Solid lines indicate Gibbs sampling performance as more rounds of Gibbs sampling are incorporated.

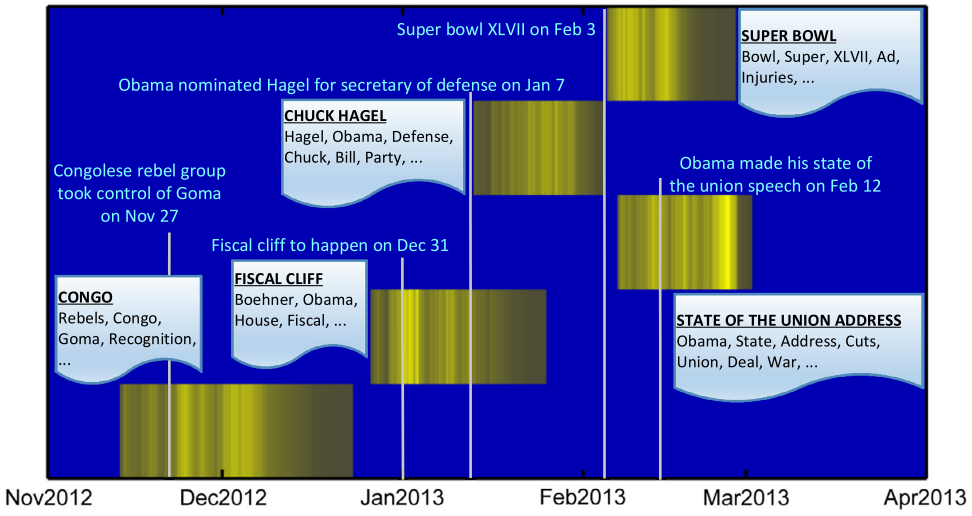


Fig. 3. Five sample clusters identified by the algorithm for the NYT dataset. Yellow bars indicate evolving cluster weights; we also indicate the most common words and the events that most likely inspired the articles about these topics. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The computational overhead of the two algorithms is very different. Our implementation of the SMC procedure requires 0.12 seconds per data item (AMD Phenom 9950 quad-core); the Gibbs sampling procedure with 30 rounds required 11.1 seconds per data item. Even allowing for inefficiencies in implementation, there is a clear difference in the amount of computation that must be performed for the respective sampling methods.

Although in [9,10], annealing improves the performance, our experiments indicated that there was no significant improvement for our synthetic dataset or the New York Times dataset evaluated in the next subsection, even when we explored a wide choice of possible annealing parameters. Our conjecture is that annealing is less important for the case when the weights of the clusters vary relatively quickly over time. For static settings as explored in [9,10], annealing provides an important mechanism for traversing low probability regions when transitioning from one candidate clustering to another. When topics are evolving, it is less likely that an inference algorithm will become trapped in a suboptimal solution, because the relative importance of the clusters identified by that solution change substantially over time, encouraging greater mixing during sampling.

6.2. Real-world dataset

We collected all articles from the New York Times (NYT) over the period from November 13th, 2012 to March 5th, 2013. To retrieve all the articles, we created a daemon that crawled the RSS feeds every 30 minutes and stored any new additions to the feed. Data items were created by extracting the titles and removing the stop-words. The data set consists of 4303 article titles where each title consists of 4–10 words selected from a vocabulary of size 3686.

We applied the sequential Monte Carlo sampler with the TS-DPM and TDPM generative models to the datasets. We used the same algorithm settings as presented in the previous section, except that we reduced α for the TS-DPM to 0.75, based on an inspection of article titles collected prior to November, which suggested that data items were less likely to be associated with new topics than in the synthetic data.

Fig. 3 depicts five of the strongest clusters that were identified using the TS-DPM model, and shows the most common words appearing in the items associated with each cluster. We have manually assigned labels to the clusters. We indicate in the figure relevant events (e.g. the fiscal cliff, the Superbowl) that prompted numerous articles to be written about the same topic.

Please cite this article in press as: M. Kharratzadeh et al., Bayesian topic model approaches to online and time-dependent clustering, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.010

Table 2
Performance comparison for NYT/CNN dataset.

Dataset	Model	Sampling scheme	DB Index
NYT	TS-DPM	non-targeted	1.30
		targeted	1.26
	TDPM	non-targeted	1.28
		targeted	1.15
Synthetic	TS-DPM	targeted	1.39
	TDPM	targeted	1.51

For this real-world dataset, we do not have access to a ground truth to evaluate clustering performance. Instead we use the Davies–Bouldin (DB) index [40] to provide a measure of the quality of a clustering. The index requires specification of a distance between data items that ranges between 0 and 1. We use a distance based on the cosine similarity $s(x_i, x_j)$, defined as:

$$d(x_i, x_j) = 1 - s(x_i, x_j) = 1 - \frac{x_i x_j}{||x_i||\,||x_j||}. \tag{31}$$

Here the data items x_i and x_j are represented in their vector form, i.e., a vector of length V (the size of the vocabulary) where the k-th element corresponds to the number of occurrences of the k-th word. Smaller values of the DB index indicate more coherent or "better" clusterings.

Table 2 presents the average DB index values achieved by the online clustering algorithms, with and without targeted sampling, on the NYT/CNN dataset. We observe that the TDPM model performs marginally better than TS-DPM model in this case, perhaps indicating that time scales and differences smaller than one day are not relevant for this dataset. The targeted sampling scheme improves the performance for both algorithms. For comparison, we also present the values of the DB index for TS-DPM and TDPM models with targeted sampling on the synthetic dataset. We observe that the index values for the NYT dataset are smaller than those for the synthetic dataset, indicating that the algorithm has identified meaningful clustering structure.

7. Conclusion

We have provided a tutorial description and survey of dynamic probabilistic topic models and indicated how they can be combined with sequential Bayesian inference procedures, particularly sequential Monte Carlo samplers, to derive online clustering algorithms. These algorithms are suitable for data streams and can take into account the order of the data items or their arrival times when detecting the evolving clusters in the data.

Much research has been dedicated to the development of clustering procedures, but the vast majority of the algorithms that are capable of processing truly large datasets are heuristic in nature. There is tremendous value in developing algorithms that are based on generative probabilistic models, which permits application of principled inference techniques. The past decade has seen significant advances in sequential Monte Carlo sampling techniques and variational inference approaches. These advances, together with the advent of increasingly powerful probabilistic models that can capture the dynamic structure of evolving datasets, provide fruitful territory for researchers who are striving to build algorithms that can scale to process hundreds of millions of data items arriving at very fast rates. Many research challenges remain, including how to decentralize the algorithms to address scenarios where data becomes available at distributed computers, and how to parallelize the algorithms to take advantage of multi-core and cluster computational capabilities.

Acknowledgments

This article was written in memory of Dr. William Fitzgerald, a wise and generous soul who imparted his enthusiasm for Bayesian inference to a host of students and colleagues, and made a difference in many academic lives. The authors gratefully acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] D. Blei, Probabilistic topic models, Commun. ACM 55 (4) (2012) 77–84.
[2] D. Blei, J. Lafferty, Text Mining: Classification, Clustering, and Applications, Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, 2009, Chap. Topic Models.
[3] D. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (1) (2003) 993–1022.
[4] D. Chakrabarti, R. Kumar, A. Tomkins, Evolutionary clustering, in: Proc. ACM Int. Conf. Knowledge Discovery and Data Mining, Philadelphia, PA, United States, 2006.
[5] R.M. Neal, Markov chain sampling methods for Dirichlet process mixture models, J. Comput. Stat. 9 (2) (2000) 249–265.
[6] M. Jordan, Z. Ghahramani, T. Jaakkola, L. Saul, An introduction to variational methods for graphical models, Mach. Learn. 37 (2) (1999) 183–233.
[7] P. Del Moral, A. Doucet, A. Jasra, Sequential Monte Carlo samplers, J. R. Stat. Soc., Ser. B, Stat. Methodol. 68 (3) (2006) 411–436.
[8] T. Broderick, N. Boyd, A. Wibisono, A.C. Wilson, M. Jordan, Streaming variational Bayes, in: Proc. Advances in Neural Information Processing Systems, Lake Tahoe, NV, United States, 2013.
[9] Y. Ülker, B. Günsel, A.T. Cemgil, Sequential Monte Carlo samplers for Dirichlet process mixtures, in: Proc. Int. Conf. Artificial Intelligence and Statistics, Chia Laguna Resort, Sardinia, Italy, 2010.
[10] Y. Ülker, B. Günsel, A.T. Cemgil, Annealed SMC samplers for nonparametric Bayesian mixture models, IEEE Signal Process. Lett. 18 (1) (2011) 3–6.
[11] Diaspora Project, https://diasporafoundation.org, last accessed: 5 Oct. 2014.
[12] Friendica, http://friendica.com, last accessed: 5 Oct. 2014.
[13] T. Ferguson, A Bayesian analysis of some nonparametric problems, Ann. Stat. 1 (2) (1973) 209–230.
[14] X. Zhu, Z. Ghahramani, J. Lafferty, Time-sensitive Dirichlet process mixture models, Tech. Rep. CMU-CALD-05-104, School of Computer Science, Carnegie Mellon University, Pittsburg, PA, United States, May 2005.
[15] N. Bouguila, D. Ziou, Online clustering via finite mixtures of Dirichlet and minimum message length, Eng. Appl. Artif. Intell. 19 (4) (2006) 371–379.
[16] T. Xu, Z. Zhang, P. Yu, B. Long, Dirichlet process based evolutionary clustering, in: Proc. IEEE Int. Conf. Data Mining, Pisa, Italy, 2008.
[17] T. Xu, Z. Zhang, P. Yu, B. Long, Evolutionary clustering by hierarchical Dirichlet process with hidden Markov state, in: Proc. IEEE Int. Conf. Data Mining, Pisa, Italy, 2008.
[18] A. Ahmed, E.P. Xing, Dynamic non-parametric mixture models and the recurrent Chinese restaurant process with application to evolutionary clustering, in: Proc. SIAM Int. Conf. Data Mining, Atlanta, GA, United States, 2008.
[19] A. Ahmed, E.P. Xing, Timeline: a dynamic hierarchical Dirichlet process model for recovering birth/death and evolution of topics in text stream, in: Proc. Conf. Uncertainty in Artificial Intelligence, Catalina Island, CA, United States, 2010.
[20] S.J. Gershman, D. Blei, A tutorial on Bayesian nonparametric models, J. Math. Psychol. 56 (1) (2012) 1–12.
[21] J. Sethuraman, A constructive definition of Dirichlet priors, Stat. Sin. 4 (2) (1994) 639–650.
[22] W. Feller, An Introduction to Probability Theory and Its Applications, Wiley, 1968.
[23] D. Aldous, Exchangeability and related topics, in: Ecole d'été des probabilités de Saint Flour, Springer, 1985.
[24] D. Blei, J.D. Lafferty, Dynamic topic models, in: Proc. ACM Int. Conf. Machine Learning, Pittsburgh, PA, United States, 2006.
[25] F. Caron, M. Davy, A. Doucet, Generalized Polya urn for time-varying Dirichlet process mixtures, in: Proc. Conf. Uncertainty in Artificial Intelligence, Vancouver, BC, Canada, 2007.
[26] Q. Mei, C. Zhai, Discovering evolutionary theme patterns from text: an exploration of temporal text mining, in: Proc. ACM Int. Conf. Knowledge Discovery and Data Mining, Chicago, IL, United States, 2005.
[27] N. Srebro, S. Roweis, Time-varying topic models using dependent Dirichlet processes, Tech. Rep. UTML TR 2005-003, University of Toronto, Mar. 2005.
[28] S.N. MacEachern, Decision theoretic aspects of dependent nonparametric processes, in: Proc. World Meeting of the International Society for Bayesian Analysis, Heraklion, Crete, 2000.
[29] J. Griffin, M. Steel, Order-based dependent Dirichlet processes, J. Am. Stat. Assoc. 101 (473) (2006) 179–194.

Please cite this article in press as: M. Kharratzadeh et al., Bayesian topic model approaches to online and time-dependent clustering, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.010

[30] M.K. Pitt, C. Chatfield, S.G. Walker, Constructing first order stationary autoregressive models via latent processes, Scand. J. Stat. 29 (4) (2002) 657–663.
[31] V. Rao, Y.W. Teh, Spatial normalized gamma processes, in: Proc. Neural Information Processing Systems, Vancouver, BC, Canada, 2009.
[32] M.-A. Sato, Online model selection based on the variational Bayes, Neural Comput. 13 (7) (2001) 1649–1681.
[33] W. Fan, N. Bouguila, Online variational finite Dirichlet mixture models and its applications, in: Proc. Int. Conf. Information Sciences, Signal Processing and their Applications, Montreal, QC, Canada, 2012.
[34] C. Wang, J. Paisley, D. Blei, Online variational inference for the hierarchical Dirichlet process, in: Proc. Int. Conf. Artificial Intelligence and Statistics, Ft. Lauderdale, FL, United States, 2011.
[35] M. Hoffman, D. Blei, F. Bach, Online learning for latent Dirichlet allocation, in: Proc. Int. Symp. Neural Information Processing Systems, Vancouver, Canada, vol. 23, 2010, pp. 856–864.
[36] R.A. Levine, Z. Yu, W.G. Hanley, J.J. Nitao, Implementing random scan Gibbs samplers, Comput. Stat. 20 (1) (2005) 177–196.
[37] K. Latuszynski, G.O. Roberts, J.S. Rosenthal, Adaptive Gibbs samplers and related MCMC methods, Ann. Appl. Probab. 23 (1) (2013) 66–98.
[38] J. He, D.J. Miller, G. Kesidis, Latent interest-group discovery and management by peer-to-peer online social networks, in: Proc. IEEE Int. Conf. Social Computing, DC, United States, 2013.
[39] E. Amigó, J. Gonzalo, J. Artiles, F. Verdejo, A comparison of extrinsic clustering evaluation metrics based on formal constraints, Inf. Retr. 12 (4) (2009) 461–486.
[40] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, Understanding of internal clustering validation measures, in: Proc. IEEE Int. Conf. Data Mining, Sydney, Australia, 2010.

Milad Kharratzadeh is currently a Ph.D. candidate in electrical and computer engineering at McGill university, Montreal, Canada. He received his B.Sc. degree in electrical engineering from Sharif university of technology, Tehran, Iran in 2010; and M.Eng. degree in electrical and computer engineering from McGill university, Montreal, Canada in 2012. His research interests include statistical machine learning, data science, and computational cognitive science.

Benjamin Renard received a Diplôme d'ingénieur from Ecole Polytechnique, France, in 2011 and a Masters of Engineering degree from McGill University, Montreal, Canada in 2013. During his Masters degree, his research focused on online clustering algorithms and its applications to search in distributed social networks. He currently works as a software engineer for Facebook.

Mark Coates received the B.E. degree in computer systems engineering from the University of Adelaide, Australia, in 1995, and a Ph.D. degree in information engineering from the University of Cambridge, UK, in 1999, under the supervision of Dr. William Fitzgerald. He joined McGill University (Montreal, Canada) in 2002, where he is currently an Associate Professor in the Department of Electrical and Computer Engineering. He was a research associate and lecturer at Rice University, Texas, from 1999–2001. In 2012–2013, he worked as a Senior Scientist at Winton Capital Management. His research interests include Bayesian inference, network analysis, and distributed estimation.

Please cite this article in press as: M. Kharratzadeh et al., Bayesian topic model approaches to online and time-dependent clustering, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.010