ONLINE TIME-DEPENDENT CLUSTERING USING PROBABILISTIC TOPIC MODELS

Benjamin Renard, Milad Kharratzadeh, Mark Coates

Department of Electrical and Computer Engineering, McGill University
Montreal, Quebec, Canada

ABSTRACT

We introduce an online, time-dependent clustering algorithm that employs a dynamic probabilistic topic model. The proposed algorithm can handle data that evolves over time and strives to capture the evolution of clusters in the dataset. It addresses the case where the entire dataset is not available at once (e.g., the case of data streams) but an up-to-date clustering of the data at any given time is required. One of the main challenges of the data stream setting is that the computational cost and memory overhead must stay bounded as the number of data points increases. Our proposed algorithm has a Dirichlet process-based generative component combined with a sequential Monte Carlo sampler for posterior inference. We also introduce a novel modification to the sampling process, called *targeted sampling*, which enhances the performance of the SMC sampler. We test the performance of our algorithm with both synthetic and real datasets.

Index Terms— Clustering, Dirichlet Process, Sequential Monte Carlo Sampling, Targeted Sampling

1. INTRODUCTION

We consider the problem of online, time-dependent clustering, where the goal is to cluster a series of data items arriving sequentially while explicitly taking into account their time stamps. More specifically, 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, denoting the time when it was generated or received. We also assume that each data item consists of elements belonging to a predefined set, $\mathcal{V}$, and that all of the elements are important to its meaning. The required output is an evolutionary clustering updated with each new data item. Besides being online and time-dependent, the algorithm must learn the number of clusters automatically and should use bounded computational resources as the number of data items increases.

Typical examples for the input include search engine queries, where we want to cluster queries in order to learn current search trends in real-time, or titles of news or scientific articles, where we want to automatically identify topics or themes. Online processing of data items is important in two settings: when the dataset is too big to be processed by offline methods or when we receive data items sequentially and do not want to wait until all the items are available.

Contributions: We propose a clustering algorithm based on probabilistic topic models which is both online and time-dependent. Topic models provide us with a generative framework to perform clustering in a principled analytical, as opposed to heuristic, fashion. We combine a time-dependent generative model, which is an extension of the Dirichlet Process Mixtures (DPM) model, with an online Sequential Monte Carlo (SMC) sampler. We also implement an adaptive sampling scheme which enhances the SMC sampler and improves the performance of the algorithm.

2. BACKGROUND AND RELATED WORK

Numerous clustering algorithms use Dirichlet processes (DPs) [1] for dynamic clustering [2–5]. DPs are nonparametric and thus eliminate the need to assume and specify a fixed number of clusters. A DP is an extension of the Dirichlet distribution that enables us to use infinite sets of events. The stick-breaking representation, due to Sethuraman [6], defines the DP(α, G_0) constructively as follows. Let $(p'_k)_{k=1}^{\infty}$, $(p_k)_{k=1}^{\infty}$, and $(\psi_k)_{k=1}^{\infty}$ be defined as: $p'_k \sim \text{Beta}(1, \alpha)$, $p_k = p'_k \prod_{l=1}^{k-1}(1 - p'_l)$, $\psi_k \sim G_0$, where Beta$(1, \alpha)$ denotes the beta distribution. Then, the Dirichlet Process P can be expressed as:

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

where δ_k is the Dirac measure. Note that $\sum_{k=1}^{\infty} p_k = 1$ with probability 1. Equation (1) is referred to as the *stick-breaking construction* for Dirichlet processes [7].

Dirichlet process mixtures (DPMs) are generative models using a DP as a nonparametric prior on the mixture parameters. The generative model determines the cluster parameters θ_i and the observation x_i as follows: $G \sim \text{DP}(\alpha, G_0)$, $\theta_i | G \sim G$, $x_i | \theta_i \sim F(\theta_i)$, where $F(\theta_i)$ is problem-dependent. $\boldsymbol{p} = (p_k)_{k=1}^{\infty}$ and $\boldsymbol{\psi} = (\psi_k)_{k=1}^{\infty}$ are defined as in the DP model and we also introduce a cluster assignment variable z_i such that $z_i \sim \boldsymbol{p}$. With these notations, the DPM model is equivalent to $x_i \sim F(\psi_{z_i})$ and $\theta_i = \psi_{z_i}$.

Since the computation of the posterior in DP-based generative models is intractable, exact inference is not possible.

There are three main approaches to the inference: Markov chain Monte Carlo (MCMC) [8], variational inference [9] and Sequential Monte Carlo (SMC) samplers [10]. Given that we want an online algorithm, we use SMC samplers since the other two approaches are either offline or assume that the number of items is known in advance [11–13]. SMC methods can be used to approximate a distribution evolving over time.

Almost all existing adaptive clustering algorithms based on DPs are either order-dependent (as opposed to time dependent) [14] or require a batch arrival or an epoch setting [3–5, 15]. Order dependence implies that cluster identification is dependent on the order-of-arrival as opposed to the actual timestamps of the data items. For example, the *time-dependent Dirichlet process mixture* model (TDPM), introduced in [5], requires that data is assembled into epochs, and cluster dependency is based on the time between epochs rather than individual timestamps. The exchangeability property enforced in most of the DPM models prevents the explicit inclusion of dependence of the data items on external elements or properties, such as time-of-arrival. One model that supports time-dependence and is amenable to online inference is the *time-sensitive Dirichlet process mixture* (TS-DPM) model introduced in [16]. Zhu et al. employ a temporal weight function for each cluster that depends on the cluster assignment history. This allows them to define a prior probability of cluster assignments which is similar to a DPM framework but evolves over time. MCMC methods are proposed in [16] to achieve offline inference.

With regard to posterior inference, SMC sampling strategies were applied to static DPM models in [17, 18] for offline inference; here we develop SMC samplers for the TDPM and TS-DPM models to obtain online time- or order-dependent clustering algorithms.

3. ONLINE TIME-DEPENDENT CLUSTERING

We propose a framework which uses TS-DPM or TDPM as the generative model and an SMC sampler as the inference technique. Due to space limitations, we only describe the details for the TS-DPM model here; the model derivation and posterior inference are similar for the TDPM model.

3.1. Generative model

Consider a sequence of N_d data items $x_{1:N_d}$, where each item $x_i, 1 \leq i \leq N_d$ is associated with a timestamp $t_i \in \mathbb{R}$. We assume that the items are ordered in chronological order: $t_i \leq t_j, \forall i \leq j$. We denote by $z_i \in \mathbb{N}$ the cluster index of item x_i. Each cluster index k is associated with a multinomial distribution θ_k over the set of all possible elements, $\mathcal{V}$: $p(x_i|z_i = k) = \prod_{v \in \mathcal{V}} \theta_k(v)^{x_i(v)}$, where $x_i(v)$ is the number of time element v appears in item x_i. We assume that the size of the $\mathcal{V}$ is fixed, say V. For each cluster k, we choose the prior on θ_k to be a Dirichlet distribution G_0 with parameters $(\beta, \boldsymbol{m})$, where $\boldsymbol{m}$ is a vector of size V representing a base measure on the vocabulary and $\beta \in \mathbb{R}$ expresses the strength of the prior.

Following [16], we introduce a temporal weight function $w(t, k)$ for each cluster index k that depends on the current time t and the collection of previous assignments $\{z_1, \ldots, z_{i-1}\}$: $w(t, k) = \sum_{j|t_j<t} \kappa(t - t_j) \cdot \delta(z_j, k)$, where κ is a kernel function (e.g., $\kappa(\tau) = \exp(-\lambda\tau)$) and where the Kronecker function $\delta(a, b) = 1$ if $a = b$ and 0 otherwise. The prior probability of assigning x_i to cluster k given previous cluster assignments, $\{z_1, \ldots, z_{i-1}\}$, is defined as:

$$p(z_i = k|z_1, \ldots, z_{i-1}) = p(z_i = k|w(t_i, \cdot)) \tag{2}$$

$$= \begin{cases} \dfrac{w(t_i, k)}{\sum_{k'} w(t_i, k') + \alpha} & \text{if } k \in \{z_1, \ldots, z_{i-1}\} \\ \dfrac{\alpha}{\sum_{k'} w(t_i, k') + \alpha} & \text{if } k \text{ is a new cluster} \end{cases} \tag{3}$$

This differs from the classical DPM framework as a temporal weight defined by the kernel function κ (instead of a fixed weight of 1) is assigned to past observations. The kernel κ is usually defined as a non-negative, decreasing function such that $\lim_{\tau \to \infty} \kappa(\tau) = 0$. This way, the very old data items will have a negligible effect on the prior of a new element. Approximating the weight of very old items as 0 gives:

$$\kappa_2(\tau) = \begin{cases} e^{-\lambda\tau} & \text{if } \tau < \tau_{\text{lim}} \\ 0 & \text{if } \tau \geq \tau_{\text{lim}} \end{cases} \tag{4}$$

Therefore, each data item impacts the prior probability of new items only for the duration of τ_{lim}. Using the approximation proposed in κ_2 enables us to delete data old data items as soon as their time expires (i.e., after τ_{lim}). This reduces the size of required memory as well as the computation time. However, by using a temporal weight function, we lose the exchangeability of the model and cannot use many typical inference techniques such as Gibbs sampling or variational inference.

3.2. Posterior inference

For $n \in \{1, \ldots, N_d\}$, let $y_n = (x_n, t_n)$ be the n-th observation. Let also $z_{n,j}$, with $n \geq j$, denote the cluster assignment of item y_j after the n-th item has been seen and let $\boldsymbol{z}_n = \{z_{n,1}, \ldots, z_{n,n}\}$. Posterior inference in the TS-DPM model aims at inferring $\boldsymbol{z}_n$ given $x_{1:n}$ and $t_{1:n}$. According to Bayes' rule, we have:

$$p(z_{n,i} = k|\boldsymbol{z}_{n,-i}, x_{1:n}) \propto p(z_{n,i} = k|\boldsymbol{z}_{n,-i}) \cdot p(x_i|\boldsymbol{x}_{-i:\boldsymbol{z}_{-i}=k}) \tag{5}$$

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

$$p(z_{n,i} = k|\boldsymbol{z}_{n,-i}) \propto p(z_{n,i} = k|\boldsymbol{z}_{n,1:i-1}) \times \left(\prod_{m=i+1}^{n} p(z_{n,m}|\boldsymbol{z}_{n,1:m-1}) \right) \tag{6}$$

which in turn can be expanded using equation (3).

Given the Dirichlet prior and the multinomial likelihood, we can integrate out the cluster parameters to get:

$$p(x_i|\boldsymbol{x}_{-i:\boldsymbol{z}_{-i}=k}) = \int p(x_i|\theta)p(\theta|\boldsymbol{x}_{-i:\boldsymbol{z}_{-i}=k})d\theta$$
$$= \frac{\Gamma(\sum_v f_v + \beta)}{\prod_v \Gamma(f_v + \beta m_v)} \frac{\prod_v \Gamma(x_i(v) + f_v + \beta m_v)}{\Gamma(\sum_v x_i(v) + \sum_v f_v + \beta)} \quad (7)$$

where f_v denotes the counts of word with index $v \in \{1, \ldots, V\}$ in $\boldsymbol{x}_{-i:\boldsymbol{z}_{-i}=k}$ and $\Gamma(\cdot)$ denotes the Gamma function. Thus, for each existing cluster, we only need to store two pieces of information: the number of deleted items that belonged to this cluster and the counts of their words.

We use an SMC sampler to approximate the posterior distribution. The SMC sampler allows us to revise previous cluster assignments in order to improve clustering performance. Let $\pi_n(\boldsymbol{z}_n) = p(\boldsymbol{z}_n|\boldsymbol{y}_{1:n})$ denote the posterior distribution of a sequence of observations. For the TS-DPM model, an unnormalized expression of $\pi_n(\boldsymbol{z}_n)$ can be obtained using equation (5). 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 [10].

After the n-th observation, 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. In [17], Ulker et al. initialize r as $\{1, \ldots, Q\}$ and then increase all the indices of r by Q in 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 choose r later in this section.

In the SMC sampler framework, the posterior distribution is approximated by a set of N_p weighted particles $\{\boldsymbol{z}_n^i, W_n^i\}_{1 \le i \le N_p}$ where i is the index of the particles and n is the number of observed items. Each particle $\boldsymbol{z}_n^i$ is a vector of all the cluster assignments. When a new observation y_{n+1} becomes available, the particles and their weights are updated to obtain a new set $\{\boldsymbol{z}_{n+1}^i, W_{n+1}^i\}_i$. This update can be done in a number of ways. In this work, we use the annealed SMC framework [18]. Annealing consists of replacing the fixed novelty parameter of the DPM model by a value α_n that changes at each iteration and that converges to a fixed value α. The weight update for the annealed target posterior is:

$$v_n(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n) \triangleq \frac{W_n(\boldsymbol{z}_n)}{W_{n-1}(\boldsymbol{z}_{n-1})}$$
$$= \frac{\gamma_n(\boldsymbol{z}_n)\pi_n(\boldsymbol{z}_{n-1,r}|\boldsymbol{z}_{n,d})}{\gamma_{n-1}(\boldsymbol{z}_{n-1})\pi_n(z_{n,n}, \boldsymbol{z}_{n,r}|\boldsymbol{z}_{n,d})}$$

3.3. Enhancements

To adapt our proposed inference methodology to an online, time-dependent setting, we introduce two enhancements. First, we keep the required memory bounded by limiting the resampling set to $\{x|t - t_x < \tau_{\lim}\}$. For each existing cluster k, we store the quantity $\widetilde{w}_k = \sum_{t_i<t_s|z_i=k} e^{-\lambda(t_s-t_i)}$, where t_s denotes the timestamp of the last deleted item, and . Then, we can compute the weight of cluster k at time $t > t_s$ as follows: $w(t,k) = \sum_{x_i|\tau \le t_i < t} e^{-\lambda(t-t_i)} \cdot \delta(z_i, k) + e^{-\lambda(t-t_s)} \cdot \widetilde{w}_k$. Also, note that we only need t_s, $\{\widetilde{w}_k\}_k$, and the count of words in deleted items to compute the likelihoods, priors and consequently, weight updates. Thus, while preserving the principles of SMC sampler, this enhancement reduces the required history size and computation time and keeps them bounded as new items arrive.

The second enhancement, which we call *targeted sampling*, strives to choose the active set more efficiently. The approach to sampling in the SMC framework in [17] is *blind*: the active set is defined solely based on the position of the items. In contrast, targeted sampling focuses the active set on those items whose cluster assignments are less concrete, i.e., items with two or more significant cluster weights. 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 $1 \le \rho \le K_{n-1}$; 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 now resample assignment $z_{n-1,j}$ within the set with a probability proportional to $\rho_{n-1,j}$. As we only resample a fraction of the previous assignments, we can increase the size of the active set and improve efficiency.

4. APPLICATION TO DATASETS

4.1. Synthetic dataset

We build a synthetic dataset which is a time-dependent extension of the dataset presented in [19]. We consider a fixed vocabulary size, $V = 128$, with a fixed number of clusters, $N_k = 15$. Each cluster is characterized by 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. 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, the

probability of its assignment to a given cluster is proportional to its current popularity.

Since we know the ground truth clusterings, we can use external clustering evaluation metrics such as the normalized mutual information (NMI) or the f-measure [20]. We compare the results of three algorithms with different generative models (TS-DPM [16] , TDPM [4] and GPU [21]) each paired with an SMC sampler. For the TDPM and GPU models, we use 1 day as the epoch. We also examine the impact of using the targeted sampling procedure. The results are presented in Table 1. We see that 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.

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)

Table 1. Performance comparison for synthetic data

4.2. Real-world dataset

We collected all Cable News Network (CNN) and New York Times (NYT) articles from Nov 13th, 2012 to Mar 5th, 2013 and extracted their titles. We removed the stop-words and infrequent words. Figure 1 shows the evolution of five of the clusters identified by the online clustering algorithm with the TS-DPM model. The figure indicates how the weight (popularity) changes over time and the most common words in each cluster. We identify relevant real-world events that prompted the publishing of many articles about the same topic.

We have no ground truth clustering, so we use an internal evaluation method called the Davies-Bouldin (DB) index [22], to assess the quality of our clustering. Lower values of the DB index correspond to better qualities of clustering. Table 2 represents the performance of different algorithms, with and without targeted sampling, on the NYT dataset. We observe that TDPM performs marginally better than TS-DPM. 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.

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

Table 2. Performance comparison for real data

5. CONCLUSION

We propose a novel algorithm for time-dependent clustering of streaming data based on probabilistic topic models. We combine a time-dependent DPM generative model with an SMC sampling scheme, and introduce modifications to bound computational and memory requirements and improve sampling performance. We show the effectiveness of our algorithm and proposed enhancements by experiments on both synthetic and real datasets.

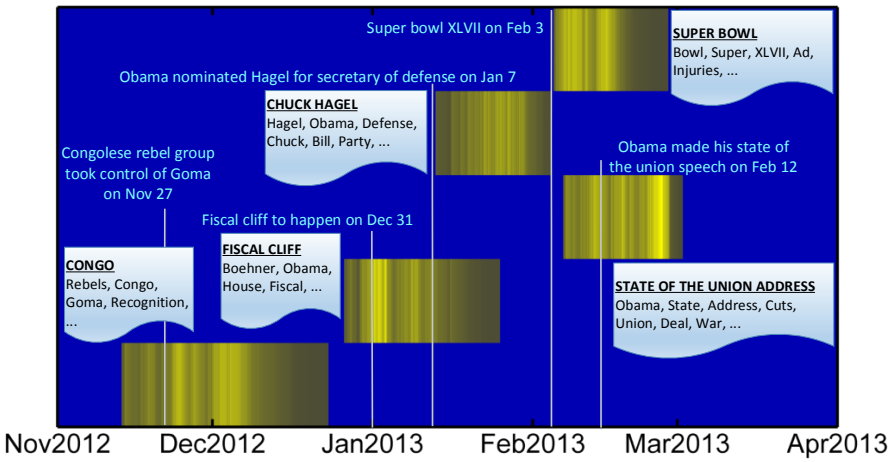

Fig. 1. Five sample clusters from NYT dataset and their weight evolution

6. REFERENCES

[1] T. Ferguson, "A Bayesian analysis of some nonparametric problems," *The Annals of Statistics*, vol. 1, no. 2, pp. 209–230, Mar. 1973.

[2] N. Bouguila and D. Ziou, "Online clustering via finite mixtures of Dirichlet and minimum message length," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 4, pp. 371–379, June 2006.

[3] T. Xu, Z. Zhang, P. Yu, and B. Long, "Dirichlet process based evolutionary clustering," in *Proc. IEEE Int. Conf. Data Mining*, Pisa, Italy, Dec. 2008.

[4] A. Ahmed and 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, Apr. 2008.

[5] A. Ahmed and 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, July 2010.

[6] J. Sethuraman, "A constructive definition of Dirichlet priors," *Statistica Sinica*, vol. 4, no. 2, pp. 639–650, July 1994.

[7] B. A. Frigyik, A. Kapila, and M. R. Gupta, "Introduction to the Dirichlet distribution and related processes," Tech. Rep. UWEETR-2010-0006, Departement of Electrical Engineering, University of Washington, Seattle, WA, 2010.

[8] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *Journal of Computational Statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[9] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, no. 2, pp. 183–233, Nov. 1999.

[10] P. Del Moral, A. Doucet, and A. Jasra, "Sequential Monte Carlo samplers," *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, vol. 68, no. 3, pp. 411–436, June 2006.

[11] M.-A. Sato, "Online model selection based on the variational Bayes," *Neural Computation*, vol. 13, no. 7, pp. 1649–1681, July 2001.

[12] W. Fan and 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, July 2012.

[13] C. Wang, J. Paisley, and D. M. Blei, "Online variational inference for the hierarchical Dirichlet process," in *Proc. Int. Conf. Artificial Intelligence and Statistics*, Ft. Lauderdale, FL, United States, Apr. 2011.

[14] T. Broderick, N. Boyd, A. Wibisono, A. C. Wilson, and M. Jordan, "Streaming variational Bayes," in *Proc. Advances in Neural Information Processing Systems*, Lake Tahoe, NV, Dec. 2013.

[15] A. Dubey, A. Hefny, S. Williamson, and E. P. Xing, "A nonparametric mixture model for topic modeling over time.," in *Proc. SIAM Int. Conf. Data Mining*, Austin, TX, May 2013.

[16] X. Zhu, Z. Ghahramani, and 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.

[17] Y. Ülker, B. Günsel, and A. T. Cemgil, "Sequential Monte Carlo samplers for Dirichlet process mixtures," in *Proc. Int. Conf. Artificial Intelligence and Statistics*, Chia Laguna Resort, Sardinia, Italy, May 2010.

[18] Y. Ülker, B. Günsel, and A. T. Cemgil, "Annealed SMC samplers for nonparametric Bayesian mixture models," *IEEE Signal Processing Letters*, vol. 18, no. 1, pp. 3–6, Jan. 2011.

[19] J. He, D. J. Miller, and G. Kesidis, "Latent interest-group discovery and management by peer-to-peer online social networks," in *Proc. IEEE Int. Conf. Social Computing*, Washington, DC, Sept. 2013, pp. 162–167.

[20] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information Retrieval*, vol. 12, no. 4, pp. 461–486, July 2009.

[21] F. Caron, M. Davy, and A. Doucet, "Generalized Polya urn for time-varying Dirichlet process mixtures," in *Proc. Conf. Uncertainty in Artificial Intelligence*, Vancouver, BC, Canada, July 2007.

[22] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *Proc. IEEE Int. Conf. Data Mining*, Sydney, Australia, Dec. 2010.