Topic Models over Text Streams: A Study of Batch and Online Unsupervised Learning

Arindam Banerjee* Sugato Basu†

Abstract

Topic modeling techniques have widespread use in text data mining applications. Some applications use batch models, which perform clustering on the document collection in aggregate. In this paper, we analyze and compare the performance of three recently-proposed batch topic models—Latent Dirichlet Allocation (LDA), Dirichlet Compound Multinomial (DCM) mixtures and von-Mises Fisher (vMF) mixture models. In cases where offline clustering on complete document collections is infeasible due to resource and response-rate constraints, online unsupervised clustering methods that process incoming data incrementally are necessary. To this end, we propose online variants of vMF, EDCM and LDA. Experiments on large real-world document collections, in both the offline and online settings, demonstrate that though LDA is a good model for finding word-level topics, vMF finds better document-level topic clusters more efficiently, which is often important in text mining applications. Finally, we propose a practical heuristic for hybrid topic modeling, which learns online topic models on streaming text and intermittently runs batch topic models on aggregated documents offline. Such a hybrid model is useful for several applications (e.g., dynamic topic-based aggregation of user-generated content in social networks) that need a good tradeoff between the performance of batch offline algorithms and efficiency of incremental online algorithms.

1 Introduction

Automated unsupervised learning of latent topics from text documents has widespread application. In this paper, we first analyze three recently-proposed batch topic models—Latent Dirichlet Allocation (LDA), Dirichlet Compound Multinomial (DCM) mixtures and von-Mises Fisher (vMF) mixture models—using a common framework based on the particular assumptions made regarding the conditional distributions corresponding to each component and the topic priors. vMF and DCM are essentially mixture models, which model topics at the document-level, while LDA is a more complex Bayesian model that considers per-word topic distributions. Since scalable topic-based clustering at the document level is important in many text mining applications (e.g., news clustering), we compare the efficiency and performance tradeoffs of these batch models in the task of document clustering.

Many applications also need the ability to process large volumes of data arriving over time in a stream (e.g., news articles arriving continually over a newswire). There are various challenges in analyzing such data—the whole data cannot be fit into memory at once due to resource constraints and has to be processed incrementally, multiple scans of the data kept in secondary storage is not always possible due to real-time response rate requirements, etc. This necessitates the use of incremental topic models while performing unsupervised learning over streaming text, to efficiently handle the scale and response-rate requirements in unsupervised text mining applications on the web. To this end, we propose online variants of the three topic models—LDA, vMF and DCM.

Several recent Web 2.0 applications (e.g., Slashdot, Blogger, Digg) are facing the need to process large volumes of user-generated content incrementally during peak load, and doing offline processing on non-peak hours. This motivated us to create a practical hybrid topic model scheme: learning online topic models on streaming data, with intermittent background batch topic models on offline aggregated text documents. The online component is necessary for categorizing documents into topic-based clusters in real-time, whereas the intermittent batch processing is required for improved unsupervised mining on larger offline text collections.

The main contributions of the paper are: (1) Comparing the performance of different offline topic modeling algorithms, and demonstrating that while LDA is good at finding word-level topics, vMF is more effective and efficient at finding document-level clusters; (2) Proposing a new online vMF algorithm, that outperforms online versions of LDA and DCM in efficiency and performance; (3) Presenting a practical hybrid scheme for topic modeling over document streams, which provides a good tradeoff between speed and accuracy while performing unsupervised learning over large text data.

2 Batch Topic Models

Unsupervised text mining and topic modeling has been a focus of active research over the past few years. The popular generative clustering and topic models for text analysis can be broadly divided into a few categories, depending on the particular assumptions made regard-

*Department of CSE, University of Minnesota
†AI Center, SRI International

		Conditional		
		Multinomial		von-Mises Fisher
		Point	Bayesian	Point
Prior	Point	naive-Bayes [16]	DCM [11, 7]	movMF [3]
	NP Bayes	LDA [5]	Bayesian LDA [8]	-

Table 1: Unsupervised models for text. The conditional is typically from the multinomial or vMF distribution, and the observation probability can be computed from a point estimate or a Bayesian model. The prior is typically either a point estimate or a non-parametric Bayesian model.

ing the conditional distributions corresponding to each component and the cluster priors. The conditional assumptions are typically from one of two classes of distributions: multinomial distributions on the unit simplex [16], or the von-Mises Fisher distribution on the unit hypersphere [3]. Further, the probability of an observation can be computed from the conditional distribution using a point estimate of the distribution, as is typical in vMF models [3] and was used originally in multinomial models [16], or from a full Bayesian model, as is becoming increasingly common for multinomial distributions [8]. The cluster priors were traditionally modeled using a distribution that was fixed across all documents, leading to the mixture of unigrams model [3, 16]. Recent years have seen development of non-parametric Bayesian modeling of the priors [5, 8]. Table 1 summarizes the main unsupervised models for text analysis, based on the above discussion. In this paper, we focus on 3 representative models based on different assumptions on the conditional and prior. We now describe the details of each model briefly.

2.1 vMF Models. The first model is a classic example of a mixture model [3] that uses von Mises-Fisher distributions as the components. In the mixture of von Mises-Fisher (movMF) distributions model, a document is represented as an unit vector that is simply the L_2 normalized version of the TFIDF vector corresponding to the document. Thus, all documents lie on the surface of the unit hypersphere. A d-dimensional unit random vector $\mathbf{x}$ (i.e., $\mathbf{x} \in \mathbb{R}^d$ and $\|\mathbf{x}\| = 1$, or equivalently $\mathbf{x} \in \mathbb{S}^{d-1}$) is said to have d-variate von Mises-Fisher (vMF) distribution if its probability density function is given by $f(\mathbf{x}|\boldsymbol{\mu}, \kappa) = c_d(\kappa)e^{\kappa \boldsymbol{\mu}^T \mathbf{x}}$, where $\|\boldsymbol{\mu}\| = 1$, $\kappa \geq 0$ and $d \geq 2$. The normalizing constant $c_d(\kappa)$ is given by $c_d(\kappa) = \kappa^{d/2-1}/((2\pi)^{d/2} I_{d/2-1}(\kappa))$, where $I_r(\cdot)$ represents the modified Bessel function of the first kind and order r. The density $p(\mathbf{x}|\boldsymbol{\mu}, \kappa)$ is parameterized by the mean direction $\boldsymbol{\mu}$, and the *concentration* parameter is κ, so-called because it characterizes how strongly the unit vectors drawn according to $f(\mathbf{x}|\boldsymbol{\mu}, \kappa)$ are concentrated about the mean direction $\boldsymbol{\mu}$ [3]. Consider a mixture model over k vMF (movMF) distributions $p(\mathbf{x}|\Theta) = \sum_{h=1}^{k} \alpha_h p_h(\mathbf{x}|\theta_h)$, where $\Theta = \{\{\alpha_h\}_{h=1}^k, \{\theta_h\}_{h=1}^k\}$ and the α_h are non-negative and sum to one, $p_h(\mathbf{x}|\theta_h)$ is a vMF distribution with parameter $\theta_h = (\boldsymbol{\mu}_h, \kappa_h)$. Given a set $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^m$ of data points assumed to have been sampled i.i.d. following the mixture distribution, one can use the EM algorithm to estimate the parameters of the model. Following [3], the M-step involves the following parameter updates: $\alpha_h = \frac{1}{m}\sum_{i=1}^m p(h|\mathbf{x}_i, \Theta), \boldsymbol{r}_h = \sum_{i=1}^m \mathbf{x}_i p(h|\mathbf{x}_i, \Theta), \hat{\boldsymbol{\mu}}_h = \boldsymbol{r}_h/\|\boldsymbol{r}_h\|, \hat{\kappa}_h = (\bar{r}_h d - \bar{r}_h^3)/(1 - \bar{r}_h^2)$, where $\bar{r}_h = \|\mathbf{r}_h\|/(\sum_{i=1}^m p(h|\mathbf{x}_i))$. In the E-step, the distribution of the hidden variables is computed as $p(h|\mathbf{x}_i, \Theta) = (\alpha_h\ f_h(\mathbf{x}_i|\Theta))/(\sum_{l=1}^k \alpha_l\ f_l(\mathbf{x}_i|\Theta))$ [15]. It can be shown [6] that the incomplete data log-likelihood, $\log p(\mathcal{X}|\Theta)$, is non-decreasing at each iteration of the parameter and distribution updates. Iteration over these updates till convergence constitutes the `movMF` algorithm.

2.2 DM/DCM models. The second model we consider is the the mixture of Dirichlet compound multinomial (DCM) distributions [11, 7]. The model is similar to a mixture model, such as `movMF`, but uses a full Bayesian model on the conditional distribution corresponding to each cluster. In particular, the model uses a Dirichlet prior over multinomial conditionals, where the parameters of the Dirichlet are different for every cluster. The mixture of multinomial model [16] is one of the earlier models for text analysis and is an example of a naive-Bayes model. In the basic model, corresponding to each cluster, one assumes a probability distribution over all words, i.e., for a word w_j, ϕ_j is the probability of emitting w_j, so that $\sum_{j=1}^d \phi_j = 1$. Then, a document is generated by sampling repeatedly from the word distribution. The naive-Bayes assumption posits conditional independence of subsequent draws, so that for a document $\mathbf{x}$ with x_j occurrences of word w_j, the probability of observing $\mathbf{x}$ given the model is $p(\mathbf{x}|\boldsymbol{\phi}) = \frac{n!}{\prod_{j=1}^d x_j} \prod_{j=1}^d \phi_j^{x_j}$. Instead of using a single multinomial for each cluster, the DCM model assumes a Dirichlet prior over all multinomials. In particular, the prior probability of the multinomial with parameter $\boldsymbol{\phi}$ is $D(\boldsymbol{\phi}|\boldsymbol{\beta}) = \frac{\Gamma\left(\sum_{j=1}^d \beta_j\right)}{\prod_{j=1}^d \Gamma(\beta_j)} \prod_{j=1}^d \phi_j^{\beta_j - 1}$, where $\boldsymbol{\beta}$ is the parameter of the Dirichlet distribution. Then, the probability of observing document $\mathbf{x}$ is obtained by integrating the probability contributions of individual multinomials over the prior so that $p(\mathbf{x}|\boldsymbol{\beta}) = \int_{\boldsymbol{\phi}} p(\mathbf{x}|\boldsymbol{\phi}) p(\boldsymbol{\phi}|\boldsymbol{\beta}) d\boldsymbol{\phi}$. The DCM distribution does not belong to the exponential family [7] and the maximum likelihood parameters estimates need non-trivial iterative computations [13]. Motivated by such computational bottlenecks, Elkan recently proposed an

exponential family approximation of the DCM model, known as the EDCM model, for which the computations are comparatively reasonable [7]. In particular, the probability of the document $\mathbf{x}$ is given by $q(\mathbf{x}|\boldsymbol{\beta}) = n!\frac{\Gamma(s)}{\Gamma(s+n)}\prod_{j:x_j\geq 1}\frac{\beta_j}{x_j}$, where $\boldsymbol{\beta}$ is the parameter of the EDCM model, and $s = \sum_{j=1}^{d}\beta_j$. Given a set of m documents $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$, one can run an EM algorithm for the mixture of EDCM models in order to estimate the parameters as well as get a clustering of the documents. The E-step involves computing $p(h|\mathbf{x})$ and is similar to that of the mixture of vMF distributions. The M-step parameter estimates $(s_h, \boldsymbol{\beta}_h), h = 1, \ldots, k$ are given by $s_h = \frac{\sum_{j=1}^{d}\sum_{i=1}^{m} p(h|\mathbf{x}_i)\mathbb{I}(x_{ij}\geq 1)}{\sum_{i=1}^{m} p(h|\mathbf{x}_i)\Psi(s_h+n_i)-M\Psi(s_h)}$,and $\beta_{hj} = \frac{\sum_{i=1}^{m} p(h|\mathbf{x}_i)\mathbb{I}(x_{ij}\geq 1)}{\sum_{i=1}^{m} p(h|\mathbf{x}_i)\psi(s_h+n_i)-M\Psi(s_h)}$,where n_i is the number of words in document $\mathbf{x}_i$. Elkan [7] recommends several other practical heuristics to get high quality clustering results from the mixture of EDCM model.

2.3 LDA models. The third model is a full Bayesian version of latent Dirichlet allocation (LDA) [8, 5]. The fundamental difference between the LDA model and the vMF and DCM models is that LDA uses a non-parametric Bayesian model of the prior probability over all the clusters. In this paper, we focus on the full Bayesian version of LDA that uses a Bayesian model for computing the probability of an observation given each cluster [8]. Similar to the DCM model, the Bayesian LDA models assumes a Dirichlet prior over all multinomials, where the Dirichlet parameter is different for every cluster. However, unlike the previous models, LDA assumes a different topic distribution for every document. Note that LDA represents a document as a sequence $\mathbf{w}$ of words, rather than a feature vector $\mathbf{x}$ of word counts as used by vMF or DCM, the latter actually using a sequence representation that can be compiled into a feature vector. In order to generate the sequence of words $\mathbf{w}$ in a document, a topic distribution $\boldsymbol{\theta}$ is first sampled from a Dirichlet prior, with parameter $\boldsymbol{\alpha}$, on the topic simplex. To generate the ℓ^{th} word of the document, a topic z_ℓ is first sampled at random from the topic distribution $\boldsymbol{\theta}$. Then a topic-specific multinomial ϕ_{z_ℓ} for the word distribution is sampled from the Dirichlet prior with parameter $\boldsymbol{\beta}$, corresponding to the topic z_ℓ. Finally, the word w_ℓ is sampled according to the multinomial ϕ_{z_ℓ}. Thus, the probability of observing the document $\mathbf{w}$ is given by $p(\mathbf{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \int_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{\alpha})\left(\prod_{\ell=1}^{n}\sum_{z_\ell} p(z_\ell|\boldsymbol{\theta})\int_{\phi} p(w_\ell|\phi)p(\phi|\boldsymbol{\beta}_{z_\ell})d\phi\right)d\boldsymbol{\theta}$. For a k component topic model, each $z_\ell \in \{1, \ldots, k\}$. Consider a corpus $\mathbf{w} = \{w_1, \ldots, w_n\}$, where word w_ℓ is from document d_ℓ. Given particular values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the main inference problem to be solved involves estimating z_ℓ for each word w_ℓ. As [8] showed, the inference problem can be solved using Gibbs sampling. In particular, from Bayes rule, the conditional posterior distribution for z_ℓ is given by $p(z_\ell = h|\mathbf{z}_{-\ell}, \mathbf{w}) \propto p(w_\ell|z_\ell = h, z_{-\ell}, \mathbf{w}_{-\ell})p(z_\ell = h|\mathbf{z}_{-\ell})$. A careful calculation [8] shows that both terms in the right hand side can be computed in closed form so that $p(z_\ell = h|\mathbf{z}_{-\ell}, \mathbf{w}) = \frac{n_{-\ell,h}^{(w_\ell)}+\beta}{n_{-\ell,h}^{(\cdot)}+W\beta}\frac{n_{-\ell,h}^{(d_\ell)}+\alpha}{n_{-\ell,\cdot}^{(d_\ell)}+T\alpha}$, where $n_{-\ell,h}^{(w_\ell)}$ is the number of instances of word w_ℓ assigned to topic h not including the current word, $n_{-\ell,h}^{(\cdot)}$ is the total number of words assigned to topic h not including the current word, $n_{-\ell,h}^{(d_\ell)}$ is the number of words from document d_ℓ assigned to topic h, not including the current one, and $n_{-\ell,\cdot}^{(d_\ell)}$ is the number of words in document d_ℓ, not including the current one. A Markov Chain Monte Carlo algorithm based on the above equation can then be used to get samples from the topic distribution [8].

3 Online Topic Models

In this section, we present online versions of the three topic models discussed in Section 2. We also discuss a hybrid scheme of interleaving online topic modeling on streaming text with intermittent batch processing.

3.1 Online vMF. The mixture of vMF distributions model is the simplest of the three models discussed in Section 2. We focus on the spherical kmeans algorithm, which is a popular special case of the general vMF model, and propose a version that is fully online. Since the batch vMF model uses the EM algorithm, our extension is partly motivated by the analysis of [15], and an application of a similar analysis on frequency sensitive clustering due to [4].

Given an existing mixture of vMF model based on a stream of t documents, and given a new document $\mathbf{x}_{t+1}$, the document can be assigned to the best cluster, i.e., the cluster having highest posterior probability $p(h|\mathbf{x}_{t+1})$. Now, the parameters of the model need to be updated based on the new document. While there are several choices of doing such an update, we choose a simple approach of only updating the parameters of the mixture component to which the current document got assigned to. Such a choice is partly motivated by theoretical results on online learning of exponential family distributions [2]. In particular, for exponential family distributions, one can show a strong relative loss bound on streaming data based on the following simple recursive update of the mean parameter $\boldsymbol{\mu}^{(t+1)} = \boldsymbol{\mu}^{(t)} + \frac{1}{t+1}(\mathbf{x}_{t+1} - \boldsymbol{\mu}^{(t)})$. In practice, as t increases, the last term becomes vanishingly small, which may not be desirable particularly in non-stationary environments. A practical trade-off is to maintain an effective count

$c_{t+1} = (1 - 1/L)c_t + 1$, where L is the effective memory length [4]. Note that as $t \to \infty$, $c_t \to L$ from below. In case of von Mises-Fisher distributions, the estimate of the mean has to be normalized to lie on the unit hypersphere [4, 3], so that the recursive update becomes $\boldsymbol{\mu}^{(t+1)} = \frac{\boldsymbol{\mu}^{(t)} + \frac{1}{c_{t+1}}(\mathbf{x}_{t+1} - \boldsymbol{\mu}^{(t)})}{\|\boldsymbol{\mu}^{(t)} + \frac{1}{c_{t+1}}(\mathbf{x}_{t+1} - \boldsymbol{\mu}^{(t)})\|}$. Thus, the online vMF model is a truly online model that processes one point at a time and does not need to store any additional information other than the current set of parameters.

3.2 Online DCM. The DCM model is not an exponential family distribution, so the simple recursive update is not appropriate for the mixture of DCM model. In fact, the EDCM model, which is an exponential family approximation to DCM, is actually not an exponential family model in the form it appears in [7], since the cumulant function has not been determined exactly.

As a result, we resort to a more explicit windowed update that we describe next. Consider an existing mixture of EDCM models based on a stream of t documents. Given a new document $\mathbf{x}_{t+1}$, it is straightforward to compute $p(h|\mathbf{x}_{t+1})$ from the existing model components, by applying Bayes' Rule. After assigning the document to the most likely component, we update the component parameters as follows. Let M_h^L be the set of the last L documents that have been assigned to topic h. If the new document $\mathbf{x}_{t+1}$ is assigned to topic h, then M_h^L is updated by inserting $\mathbf{x}_{t+1}$ in, and deleting the oldest document in the set. Then, the documents in M_h^L is used to estimate a new sets of parameters following the update equations in Section 2.2. The parameters of the EDCM components are updated as a moving average of the new estimated parameters and the existing parameter values over the sliding window.

3.3 Online LDA. For learning the LDA model online, we use the incremental LDA model proposed in [17]. In the incremental LDA algorithm, batch LDA is initially run on a small window of the incoming data stream and the LDA parameters ϕ and θ are initialized using the MAP estimates $\hat{\phi}$ and $\hat{\theta}$ estimated from this window: $\phi_j^{w_l} = \frac{n_j^{(w_l)} + \beta}{n_j^{(\cdot)} + W\beta}$, $\theta_j^{d_l} = \frac{n_j^{(d_l)} + \alpha}{n^{(d_l)} + T\alpha}$, where $n_j^{w_l}$ is the number of times the word w_l is assigned to topic j, $n_j^{(\cdot)}$ is the sum of $n_j^{w_l}$ over all words, $n_j^{d_l}$ is the number of times a word from document d_l has been assigned to topic j, and $n_{(\cdot)}^{d_l}$ is the sum of $n_j^{d_l}$ over all topics. Henceforth, with the arrival of every new document d, the topic assignment of the i^{th} word in the document is estimated as: $P(z_i = j|z_{-i}, w) \propto \hat{\phi}_j^{w_i} \frac{n_{-i,j}^d + \alpha}{n_{-i,\cdot}^d + T\alpha}$. Subsequently, the MAP estimates $\hat{\phi}$ and $\hat{\theta}$ are updated using the expected assignments of words to topics in d. This process of assignment of incoming topics and update of the MAP estimates of the parameters is continued till the end of the document stream. Note that this is not the true online Bayesian version of the LDA algorithm, since it does not update the posterior distribution over the parameters ϕ and θ; instead, it works with their MAP estimates. Nonetheless, the incremental LDA algorithm is efficient, since the topic assignments and parameter updates with every new document depends only on the accumulated counts and the words in that document.

3.4 Hybrid Scheme. In the different motivating examples outlined in Section 1, online topic modeling is necessary for real-time topic analysis of an incoming document in the data stream. But at the same time it may be required to run offline topic models intermittently on the repository where the incoming data is stored, in order to get robust statistics of the overall topic model (and hence better clustering) from collective inference over a large text corpus. As a result, what we need in such applications is a hybrid topic modeling scheme that alternates between two phases: (i) STREAM phase: run an online topic algorithm on streaming data; and (ii) OFFLINE phase: intermittently run a batch algorithm on the accumulated offline repository data. A hybrid algorithm can operate on different schedules of alternation between the STREAM and OFFLINE phases. Similar schemes have been used successfully in clustering evolving data streams [1].

4 Experiments

This section describes the experiments, outlining the datasets, evaluation measures, and results.

4.1 Datasets. We used the *20 Newsgroups* collection (19941 documents in 25936 dimensions, 20 clusters), and four subsets derived from it: (i) subset-20 (1997 documents in 13341 dimensions, 20 clusters), (ii) rel-3 (2996 documents in 10091 dimensions, 3 clusters) , (iii) sim-3 (2980 documents in 5950 dimensions, 3 clusters), and (iv) diff-3 (2995 documents in 7670 dimensions, 3 clusters), which represent datasets in different levels of size and difficulty of clustering [3].

We also harvested news articles from the Slashdot website and created 2 new datasets: (i) slash-7: news articles posted to 7 Slashdot categories: Business, Education, Entertainment, Games, Music, Science and Internet (6714 documents in 5769 dimensions, 7 clusters); (ii) slash-6: articles posted to the 6 categories: Biotech, Microsoft, Privacy, Google, Security, Space (5182 documents in 4498 dimensions, 6 clusters).

All the datasets used the bag-of-words representation with word-level features, and were pre-processed using stop-word removal, TFIDF weighting (for vMF only, since LDA and DCM can handle only counts), and removal of very high-frequency and low-frequency words.

4.2 Evaluation. The following evaluation measures were used in the experiments: (i) Cluster quality: we used normalized mutual information (nMI), which measures how closely the cluster partitioning could reconstruct the underlying label distribution in the data [18]. (ii) Time: for the batch algorithms, we measured the system time taken to converge to the final clustering solution. In the online case, we report the average time to cluster each incoming document.

Dataset	nMI			Run Time (sec)		
	vMF	EDCM	LDA	vMF	EDCM	LDA
news-20	0.51	**0.54**	0.53	204	934	352
subset-20	0.41	0.36	**0.43**	14	25	34
sim-3	**0.27**	0.12	0.11	2	4	15
rel-3	**0.38**	0.30	0.28	3	9	17
diff-3	**0.82**	0.81	0.74	1	7	16
slash-7	**0.39**	0.22	0.31	15	40	47
slash-6	**0.65**	0.36	0.46	6	26	36

Table 2: Performance of batch algorithms averaged over 5 runs. Best nMI for every dataset is highlighted.

music	web	scientists	internet	games
apple	google	nasa	broadband	gaming
itunes	search	space	domain	game
riaa	yahoo	researchers	net	nintendo
ipod	site	science	network	sony
wikipedia	online	years	verisign	xbox
digital	sites	earth	bittorrent	gamers
napster	ebay	found	icann	wii
file	amazon	brain	service	console
drm	engine	university	access	video

Table 3: Five of the topics obtained by running batch vMF on slash-7.

Dataset	nMI			Time per doc (sec)		
	o-vMF	o-EDCM	o-LDA	o-vMF	o-EDCM	o-LDA
news-20	**0.54**	0.39	0.30	0.011	0.565	0.010
subset-20	**0.42**	0.27	0.29	0.032	0.361	0.041
sim-3	**0.17**	0.09	0.08	0.014	0.053	0.011
rel-3	**0.31**	0.16	0.19	0.019	0.092	0.012
diff-3	**0.72**	0.62	0.60	0.009	0.061	0.008
slash-7	**0.34**	0.16	0.12	0.007	0.048	0.006
slash-6	**0.54**	0.25	0.30	0.005	0.035	0.004

Table 4: Performance of online algorithms averaged over 5 epochs. Best nMI for every dataset is highlighted.

4.3 Results. *Experiment 1* compares the performance of the three batch algorithms—LDA, EDCM, and vMF—on the 7 datasets. Table 2 shows the nMI and run time results averaged across 5 runs, where vMF has the highest nMI accuracy and lowest run time for most of the datasets. Tables 3 shows the top 10 highest weighted words in 5 clusters obtained by vMF on the slash-7 dataset, illustrating that vMF is capable of finding good topics.

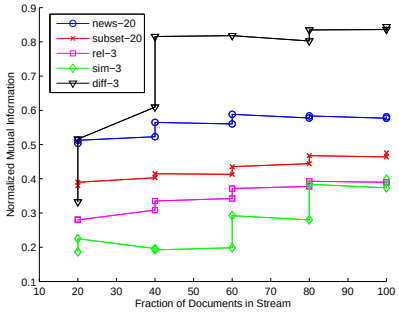

Figure 1: nMI results of h-vMF on news datasets. STREAM phase is run on the data incrementally, with one iteration of OFFLINE phase at every 20% of the stream.

Experiment 2 compares the online algorithms—o-LDA, o-EDCM, and o-vMF—on the 7 datasets. The nMI and time results shown in Table 4 are averaged across 5 epochs. o-vMF substantially outperforms o-LDA and o-EDCM for all datasets. In general, the online algorithms give worse nMI results than the corresponding batch algorithms, which is expected since the online algorithms can only update the cluster statistics incrementally.

Experiment 3 evaluated h-vMF on the news datasets. The STREAM phase was applied on the online data stream, and after every 20% of the stream, one iteration of the OFFLINE phase was run. Figure 1 shows how the nMI values improves with increasing fraction of the dataset being processed by h-vMF. As expected, there are sharp jumps in the plot where h-vMF switched to the OFFLINE phase from the STREAM phase, validating our claim that intermittent batch processing improves the clustering performance. Note that on more difficult datasets, e.g., sim-3, the STREAM phase can accumulate errors along the way, as previously noted by [1]—running intermittent OFFLINE phases can correct these errors and improve the overall performance.

5 Related Work

Our work is related to two different existing research directions: unsupervised models for text analysis, and online/streaming models for data analysis.

Among the first category of the models discussed in Section 2, vMF tends to perform marginally better than the basic naive-Bayes model [20], although the performance of naive-Bayes can be improved using techniques like annealing. In the second category of models, DCM

has also been studied by other researchers as Dirichlet Mixture (DM) models [19]. While probabilistic latent semantic indexing (PLSI) [9] was one of the first models in the third category, latent Dirichlet allocation (LDA) [5] as well as its full Bayesian variants [8] have become significantly more popular over time.

Extensive research on analysis of data streams has been done in the database, data mining and machine learning communities. A large part of the research has been motivated by specific problems and applications, including novelty detection [10], frequent pattern mining [12], and clustering [4]. One of the important earlier ideas on clustering evolving data streams suggested using a hybrid online-offline strategy, rather than a one pass algorithm, based on practical considerations [1]. The work on online spherical kmeans [4] is an example of an online extension of the first category of text models discussed earlier. Online extensions of the text models of the second and third category have also been proposed recently, e.g., online Dirichlet mixture model using a multinomial particle filter [14], online extension of the full Bayesian version of LDA [17].

6 Conclusion

This paper compares the performance of three popular topic models – LDA, vMF, EDCM – and demonstrates, via thorough experiments, that vMF provides the best overall performance for batch document clustering, discovering coherent underlying topics in the process. It also presents a new online algorithm for vMF, which outperforms corresponding online versions of LDA and EDCM. Finally, it proposes a practical hybrid scheme for topic modeling, which gives a good tradeoff of performance and efficiency for processing streaming text. In future work, we would like to investigate other hybrid topic model schemes, which can do a load-based switch between the online and batch algorithms depending on the rate of incoming documents in the text stream. We would also like to incorporate model-selection into the hybrid algorithm – this would enable new topics to be detected by the online algorithm in the Stream phase, following which the batch algorithm can get better statistics for the newly discovered topics in the Offline phase.

7 Acknowledgments

We would like to thank Jiye Yu for helping with data collection, Misha Bilenko for valuable feedback, Charles Elkan and Tom Griffiths for providing code for the batch EDCM and LDA models respectively, and the iLink team at SRI for very useful discussions. This work was supported partly by DARPA, Contract #NBCHD030010, Order #T310.

References

[1] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *VLDB*, 2003.

[2] K. S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.

[3] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra. Clustering on the unit hypersphere using von Mises-Fisher distributions. *JMLR*, 6:1345–1382, 2005.

[4] A. Banerjee and J. Ghosh. Frequency sensitive competitive learning for balanced clustering on high-dimensional hyperspheres. *IEEE Trans. on Neural Networks*, 15(3):702–719, May 2004.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *JRSS, Series B*, 39:1–38, 1977.

[7] C. Elkan. Clustering documents with an exponential-family approximation of the Dirichlet compund multinomial distribution. In *ICML*, 2006.

[8] T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101(1):5228–5225, 2004.

[9] T. Hoffman. Probabilistic latent semantic indexing. In *UAI*, 1999.

[10] J. Ma and S. Perkins. Online novelty detection on temporal sequences. In *KDD*, 2003.

[11] R. Madsen, D. Kauchak, and C. Elkan. Modeling word burstiness using the Dirichlet distribution. In *ICML*, 2005.

[12] G. S. Manku and R. Motawani. Approximate frequency counts over data streams. In *VLDB*, 2002.

[13] T. Minka. Estimating a Dirichlet distribution, 2003.

[14] D. Mochishashi and Y. Matsumoto. Context as filtering. In *NIPS*, 2006.

[15] R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. MIT Press, 1998.

[16] K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[17] X. Song, C.-Y. Lin, B. L. Tseng, and M.-T. Sun. Modeling and predicting personal information dissemination behavior. In *KDD*, 2005.

[18] A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *AAAI*, 2000.

[19] M. Yamamoto and K. Sadamitsu. Dirichlet mixtures in text modeling. Technical Report CS-TR-05-1, University of Tsukuba, 2005.

[20] S. Zhong and J. Ghosh. A comparative study of generative models for document clustering. In *Workshop on Clustering High Dimensional Data: SDM*, 2003.