Keep It Simple with Time: A Re-examination of Probabilistic Topic Detection Models

Qi He, *Member, IEEE,* Kuiyu Chang, Ee-Peng Lim, *Senior Member, IEEE,* and Arindam Banerjee,

Abstract—Topic detection (TD) is a fundamental research issue in the Topic Detection and Tracking (TDT) community with practical implications; TD helps analysts separate the wheat from the chaff among the thousands of incoming news streams. In this paper, we propose a simple and effective topic detection model called the temporal Discriminative Probabilistic Model (DPM), which is shown to be theoretically equivalent to the classic vector space model with feature selection and temporally discriminative weights. We compare DPM to its various probabilistic cousins ranging from mixture models like von-Mises Fisher (vMF) to mixed membership models like Latent Dirichlet Allocation (LDA). Benchmark results on the TDT3 dataset show that sophisticated models such as vMF and LDA do not necessarily lead to better results; in the case of LDA, notably worse performance was obtained under variational inference, which is likely due to the significantly larger number of LDA model parameters involved for document-level topic detection. On the contrary, using a relatively simple time-aware probabilistic model such as DPM suffices for both offline and online topic detection tasks, making DPM a theoretically elegant and effective model for practical topic detection.

Index Terms—topic detection, probabilistic model, time-aware, bursty feature, online, DPM, TFIDF

1 Introduction

TOPIC detection (TD) enables the automatic discovery of new topics from a news corpus and the subsequent assignment of news documents to discovered topics. A new topic typically corresponds to a newsworthy incident such as the 2008 US presidential elections. Relationships among the discovered topics can be flat or hierarchical. Moreover, since a topic is more specific than a news category such as sports or finance, most work on TD naturally assumes a simple flat topical structure. Other than topic structural differences, the TD process can be further divided into online (real-time) and offline (batch) modes, which are also known as new event detection [2] and retrospective event detection [40], respectively. Online TD incrementally examines each incoming news document to assess whether it belongs to an existing topic or if a new topic should be created based on it. Offline TD examines the entire corpus of news documents to simultaneously unravel topics and their associated news documents.

From a data-mining perspective, online and offline TD may seem no different from incremental and offline document clustering, respectively. However, there are a number of subtle TD characteristics, which if not taken into due consideration, can adversely affect practical clustering performances: 1) time plays a pivotal role, with every news document bearing a time stamp, 2) news topics are naturally bursty, i.e., new topics are constantly generated while old topics die off, 3) news documents with semantically similar content but disparate time-frames most likely originated from different topics, e.g., hurricane Mitch of October 1998 and hurricane Georges of September 1998 are two distinct topics that share many common words. TD can thus be viewed as a special case of stream clustering [17], with the clustering portfolio at any time point akin to a concept that drifts with time [38], [18].

Despite the fact that time plays an important role, the vast majority of existing TD solutions [40], [3], [35], [41], [12], [26] do not explicitly incorporate time into their formulations; each news document is represented as a vector with time-agnostic static weights, with just one minor procedural modification: news vectors are processed in time-stamp order, i.e., online TD, as opposed to batch TD, is used to handle the temporal factor. The entire setup smacks of ad hocism that builds upon the TFIDF (term frequency inverse document frequency) vector space model [34], which itself is lacking in mathematical formalism[1]. Fortunately, this simple procedural modification over static text representation model seems to work quite well in practice.

The question to ask is then this: can a concise text representation model be formulated to explicitly capture the temporal element in news streams, and yet remain practical and effective for both online and offline TD? In this paper, we seek to answer this question by evaluating several time-aware probabilistic formulations for TD, and proposing our temporal and discriminative probabilistic framework called DPM.

DPM is a Bayesian probabilistic framework that considers each news document as a point in discriminative word *and* time vector space. The posterior topic probability given a document *and* time is computed and subsequently used to

- *Q. He is with the College of Information Sciences and Technology, Pennsylvania State University, State College, PA 16802. E-mail: qhe@ist.psu.edu*
- *Kuiyu Chang is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798. E-mail: kuiyu.chang@pmail.ntu.edu.sg*
- *Ee-Peng Lim is with the School of Information Systems, Singapore Management University, Singapore 178902. E-mail: eplim@smu.edu.sg*
- *Arindam Banerjee is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455. E-mail: banerjee@cs.umn.edu*

1. The reformulation of TFIDF in a probabilistic framework by Joachims [24] does however help improve the standing of TFIDF.

TABLE 1
Probabilistic Topic Detection models.

Model	Document	Topic	TD approach
Deterministic	point	point	k-means
Discriminative Probabilistic	point	probability	**DPM**
Mixture	point	distribution	mixture of {Gaussian, vMF}
Mixed Membership	distribution	distribution	LDA

assign a document to multiple fitting topics. One unique feature of DPM is that it only needs to operate on a subset of topical and temporally discriminative words instead of the full vocabulary space, making DPM very efficient for practical implementation.

Disregarding the time component for now, DPM actually belongs to a more general class of probabilistic mixture framework that ranges from naive Bayesian models to generative mixture models like mixture of Gaussians [13] or von-Mises Fisher (vMF) [7] distributions, and finally to full Bayesian generative ones like Latent Dirichlet Allocation (LDA) [10] hierarchical topic models. Table 1 positions DPM among the different topic detection models. From Table 1, we can see that probabilistic models like DPM are simple extensions of deterministic models; it estimates the posterior probability of a topic given observed documents from the class-conditional document probabilities. Going one step further, mixture generative models treat each topic as a distribution over points (documents). Mixed membership generative models like LDA treat both documents and topics as distributions; documents are distributions over topics, topics are distributions over words, and words originate independently from documents.

Why did we choose to incorporate time into a simple probabilistic model instead of the more sophisticated generative models? In fact, we have tried adding the temporal element to mixture generative models like vMF, but that did not result in any significant improvements in TD performance. On the other hand, mixed membership generative models like LDA have already been extended to handle time [11]. In general, we found that simple probabilistic models seem to achieve the best balance in terms of model complexity and performance, validating the principal of Occam's Razor [6].

Formulating DPM using a probabilistic framework makes it more amenable to statistical analysis and practical deployment. For example, using an approach similar to Joachims [24], we show that DPM is equivalent to a temporal version of TFIDF with discriminative feature selection, which can be trivially implemented. Lastly, soft topic assignment is a natural and practical application of DPM that can be enabled at will; useful for TD in practice. In contrast, deterministic approaches like k-means hard-clustering assign all documents into a set of disjoint clusters, where each cluster represents one topic. The contributions of this paper are thus summarized below:

1) A temporal discriminative probabilistic model, DPM, is proposed and carefully evaluated for both offline and online topic detection performance.
2) The theoretical link between DPM and temporal TFIDF is shown, which helps explain why TFIDF has been fairly successful for topic detection.
3) An investigation of four types of TD models (cf. Table 1) is conducted, including offline (soft partition) and online (point assignment) topic detection benchmarks. To the best of our knowledge, the soft topic assignment benchmarks are the first of its kind for topic detection.
4) Bursty words, which are temporal in nature, are used as topic-discriminative features.

2 Related Work

We briefly review the plethora of TD research from the TDT research community, along with recent advances in probabilistic topic models.

2.1 Topic Detection in TDT and Clustering

An overview of TDT research can be found in [4]. The overall goal of TDT is to understand news content across different languages, and to develop a system to process news streams from a variety of sources. A formal definition of a *topic* was given in [36] as follows: *A topic is defined to be a seminal event or activity, along with all directly related events and activities*. The definition of a topic may be ambiguous at times. For example, "US attacked Iraq" can be a topic, which can be further divided into events or sub-topics like "US declared war with Iraq", "US sent Marine Corps to Iraq", etc. In fact, a hierarchical topic/event structure may gradually evolve [42], [18]. Detection of relationships among topics or events is beyond the scope of this paper, i.e., we will only consider a flat structure of topics, where every topic is equally important. In addition, we shall adopt the TDT definition of topic, and equate new event detection with online topic detection. This definition naturally puts classical document clustering into the batch or offline topic detection genre.

Non-probabilistic models like deterministic clustering have achieved significant success on topic detection in the past. For example, Yang et al. [40] successfully used hierarchical clustering for offline topic detection. Allan et al. [5] showed that Single-Link and Single-Pass clustering, which assigns the cluster label of the nearest neighbor (1-NN), achieved the best online topic detection performance. In fact, if we only consider a flat topic structure and ignore time, there exists extensive work on batch or offline topic detection, including the rich family of k-means clustering [14].

Our work is different from the vast majority of TDT approach in that we explicitly include time in our model, and we focus on the family of probabilistic topic detection models, which is surveyed in the next section.

2.2 Probabilistic Models for Topic Detection

There exists a plethora of literature on probabilistic models for unsupervised text clustering, most of which can be directly applied to offline topic detection, ignoring the time factor. Probabilistic models consider topics as distributions over either documents or words. Models that use the vector space word representation of documents typically model a topic as a distribution over documents, i.e., the Gaussian mixture model [13] and von-Mises Fisher (vMF) mixture model [8], which models a topic as a Gaussian or vMF distribution of documents in

word vector space, respectively. Other probabilistic models such as naive-Bayes [33], Dirichlet Compound Multinomial (DCM) mixtures [15], probabilistic Latent Semantic Indexing (pLSI) [22], and Latent Dirichlet Allocation (LDA) [10] treat each topic as a multinomial distribution of words. We can also simplify the topic distribution by modeling each topic as a discrete probability over documents, which is exactly what our DPM model assumes. We categorize the above work as "offline probabilistic models".

There are but few probabilistic models specifically geared for online topic detection, which as we emphasized before, depends strongly on time. Zhang et. al. [43] were the first to apply LDA to online topic detection. However, training on existing known topics is needed to model a new topic prior, which makes this approach supervised.

Simple online extensions of vMF mixture and LDA have been proposed in [8] and [11] respectively, where the focus has been on incrementally updating existing cluster topic parameters (assuming a fixed number of clusters), rather than detecting new topics. Wang and McCallum [39] presented an LDA-style document timestamp-aware topic model. However, their method was also used to track the evolution of existing topics in an offline manner. We categorize the above work into the task of "topic evolution", not online topic detection. In these work, the number of topics has to be predefined and no new topic would be found.

Recently, Dirichlet processes have been used to determine the number of topics automatically [32], [37], [29], with hyperparameters specifying the *rate* at which topics grows with data. These so-called non-parametric Dirichlet process mixture models are in fact variations of generative topic models (i.e., LDA) for offline topic detection, where document orderings are not utilized at all. More recently, an online version of Dirichlet process mixture model was proposed in [1], where the temporal order of documents is maintained across time and the number of topics at each time instance is unbounded. Similarly, He et. al. [21] used a simple cosine similarity comparison to determine the death/birth of old/new topics. However, these approaches still follow along the lines of "topic evolution" and are not suitable for online topic detection because: 1) topic evolution models were only assessed on pairwise sequential document sets (time t and $t-1$); and 2) In online topic detection, we have to decide if a new topic needs to be created upon the arrival of every new document. It is impractical to conduct offline probabilistic model evaluation for each incoming document (against all previous documents), or overkill to run the topic evolutionary model on each incoming document.

Our work has fundamental differences from the above probabilistic models: 1) we proposed a temporal discriminative probabilistic model DPM by removing the topic distribution assumptions made by other more sophisticated probabilistic models and incorporating the time element; 2) we comprehensively benchmarked various probabilistic models including DPM on both offline and online topic detection, making our work one of the most comprehensive TD benchmarks to date.

TABLE 2
Comparisons between offline and online TD models.

Type	Document ordering	Applications	# topics k	Topic assignment
offline	Irrelevant for all t	Post analysis of topics	Fixed	Distribution
online	Relevant for different t	Identify new topics in real time	Increasing	Point

3 Topic Detection Problem

Topic detection aims to group thematically-related documents from a temporal text stream into an unknown number of topics. Formally, let D be a news stream starting at time t_0, with varying number of documents $N(t)$ published at each discrete time point $t \geq t_0$ and total number of documents $N = \sum_t N(t)$ in D. The ordering of documents does not matter for documents published at the same discrete time point. Let Z be a set of document-level gold-standard topics for text stream D and $P(z|\mathbf{d}), z \in Z$ be the ground-truth posterior probability distribution for any document $d \in D$.

Our goal is to find a set of document-level clusters C as well as the hypothesis distribution $P(c|\mathbf{d}), c \in C$ to approximate the ground truth as closely as possible. If $P(c|\mathbf{d})$ is learnt at one go using all documents throughout the text stream D for all time $t \geq t_0$, the process is called *offline topic detection*. The objective is a soft partition that aims to recover the underlying true class labels Z for the complete corpus D. If $P(c|\mathbf{d})$ is learnt using only documents up till the time stamp of the newest incoming document d, the process is called *online topic detection*. The objective is a point assignment that aims to identify the cluster c for document d at time t, referred to as $\arg\max_c P(c|\mathbf{d}, t)$.

Online topic detection is modeled as a point assignment process (irrevocable once assigned) rather than a distribution because the total number of topics/clusters is unknown and increases over time; the distributional space is incomplete at any time. Table 2 summarizes the differences between offline and online topic detection models.

4 Topic Detection Models

Topic detection models can be broadly classified into two types in our context: non-probabilistic (Section 4.1) and probabilistic (Sections 4.2 and 4.3). Non-probabilistic topic detection models based on hard clustering have achieved decent offline topic detection performances in the past, yet probabilistic models have never been specifically applied to topic detection.

4.1 Non-probabilistic Models

Non-probabilistic models cluster documents directly, which are modeled as vectors in high-dimensional word space, where the relationship between documents and words are explicitly linear and independent[34].

4.1.1 Offline Topic Detection

We picked the spherical k-means (SPK) clustering [14] as the representative non-probabilistic topic detection model for offline topic detection. Suppose there are N document vectors

$\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N$ in $\mathbb{R}^n$ (n is the number of distinct words) from D, and each document vector has been normalized to lie on the unit hypersphere, i.e. $\|\mathbf{d}_i\| = 1 \ \forall i$. The TFIDF vector space model is used to generate each document vector as

$$\mathbf{d}_i = [w_{1i}, w_{2i}, \ldots, w_{ni}]^T, w_{ji} = \frac{1}{\|\mathbf{d}_i\|} f_{ji} \log(1 + \frac{N}{N_j}),$$

where f_{ji} is the term frequency of word x_j in document d_i and N_j is document frequency of word x_j, i.e., number of documents containing word x_j. SPK seeks a partitioning of D into k disjoint clusters $c_1, c_2, \ldots, c_k$ that maximizes the following objective function $\sum_{j=1}^{k} \sum_{d \in c_j} \mathbf{d}^T \mathbf{c}_j$, where $\mathbf{c}_j$ is the centroid of cluster c_j and defined as $\mathbf{c}_j = \sum_{d \in c_j} \mathbf{d}/N(c_j)$, and $N(c_j)$ is the number of documents in cluster c_j.

However, finding the optimal solution to the above objective function is NP-complete. SPK thus uses a k-means type of approximation algorithm, which is an iterative procedure:

1) Start with an initial clustering by arbitrarily partitioning documents. Compute the cluster centroids accordingly.
2) Update the posterior distribution by assigning each document vector $\mathbf{d}$ to the cluster with the nearest centroid,
$$P(c|\mathbf{d}) = \begin{cases} 1 & \text{if } c = \arg\max_{c_j} \mathbf{d}^T \mathbf{c}_j, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$
3) Update the new cluster centroids based on the new posterior topic distribution.
4) If the difference between the objective function of new clusters and old clusters is less than a threshold, new clusters are output as the solution. Otherwise, go to 2).

4.1.2 Online Topic Detection

One of the most common non-probabilistic models for online topic detection in TDT is the single-link and single-pass clustering [5]. Each document is represented as a bag-of-words vector based on cumulative TFIDF, which is given by [5]

$$w_{x,d} = \frac{f_{x,d} \cdot \log((0.5 + N(\tau))/N_x(\tau))}{\log(1.0 + N(\tau))}, \quad (2)$$

where τ means up to time $t \leq \tau$, $N_x(\tau)$ is the number of documents up to time τ in which word x appears, and $N(\tau)$ is the total number of documents seen to date. For each incoming document d at time t, compute its cosine similarity to every previous document ($\leq t$) in the collection. If its similarity to the nearest neighbor (1-NN) is above a threshold ϵ, assign d to the nearest cluster; otherwise, a new singleton cluster containing d is created. We shall call this non-probabilistic model *Single-Link-All*.

In fact, short of any domain knowledge such as the prior probability of generating a new topic, the threshold-based process of Single-Link-All is the most efficient and effective way for discovering new topics in an unsupervised manner. Throughout the paper, we will use this topic discovery process.

4.2 Mixture Generative Models

The vMF mixture model [7] is a mixture generative probabilistic model that has worked quite well for document clustering. More specifically, vMF mixture estimation can be viewed as a generalized version of SPK clustering. vMF mixtures assume that each topic/cluster is represented by a von-Mises Fisher distribution over all member documents, and the entire set of document distribution is a mixture of k vMF distributions.

4.2.1 vMF Mixture on Offline Topic Detection

Given a unit-length document vector $\mathbf{d} \in \mathbb{R}^n$ and $\|\mathbf{d}\| = 1$, we assume that $\mathbf{d}$ is generated by an n-variate vMF distribution with the following probability density function,

$$f(\mathbf{d}|c, \kappa) = R_n(\kappa) e^{\kappa \mathbf{d}^T \mathbf{c}}, \quad (3)$$

where $\mathbf{c}$ plays the role of the mean vector of a latent topic, $\|\mathbf{c}\| = 1$, $\kappa > 0$, and $R_n(\kappa)$ is a normalizing constant. Clearly, the probability of $\mathbf{d}$ given c will be high if they are similar, as shown in Eq. 3. The concentration parameter κ is similar to the classical variance parameter in Gaussian distributions, since it modulates the similarity between document vector $\mathbf{d}$ and topic centroid vector $\mathbf{c}$.

In the vMF mixture model, a document vector $\mathbf{d}$ is modeled by a mixture of k vMF distributions as

$$f(\mathbf{d}|\mathbf{\Theta}) = \sum_{j=1}^{k} \alpha_j f_j(\mathbf{d}|\theta_j), \quad (4)$$

where $\alpha_j > 0$ is the topic prior and $\sum_{j=1}^{k} \alpha_j = 1$, $\theta_j = (c_j, \kappa_j)$, and $\mathbf{\Theta} = \{\alpha_1, \ldots, \alpha_k, \theta_1, \ldots, \theta_k\}$.

Given a dataset D, assume that each document sample d_i is i.i.d. (independently and identically distributed) following the mixture distribution of Eq. 4, we can use the standard Expectation Maximization (EM) algorithm to estimate the parameters of the vMF mixtures. The M-step (update the parameters) can be found in [7]. The E-step, allocating the document, is given by

$$P(c_j|\mathbf{d}_i, \mathbf{\Theta}) = \frac{\alpha_j f_j(\mathbf{d}_i|\mathbf{\Theta})}{\sum_{l=1}^{k} \alpha_l f_l(\mathbf{d}_i|\mathbf{\Theta})}. \quad (5)$$

Eq. 5 computes the individual probability of assigning document d_j to cluster (topic) c_j, which depends on two factors: 1) the topic prior, i.e., larger clusters generally have a higher affinity compared to smaller ones; 2) the cosine similarity between document and cluster centroid (mean direction of vMF distributions).

4.2.2 vMF Mixture on Online Topic Detection

An online extension of vMF mixture was proposed in [8]. The basic idea is to incrementally update each cluster centroid as new documents are added while keeping the cluster concentration parameter κ and mixing proportions unchanged, as below,

$$\mathbf{c}^{(t+1)} = \mathbf{c}^{(t)} + \frac{1}{t+1}(\mathbf{d} - \mathbf{c}^{(t)}), \quad (6)$$

where $\mathbf{d}$ is the new document vector, $\mathbf{c}^{(t)}$ and $\mathbf{c}^{(t+1)}$ are centroid vectors of the cluster to which $\mathbf{d}$ is assigned before and after the arrival of document d.

Since we assume online topic detection to be a point assignment process, the new topic discovery procedure for both vMF mixtures and Single-Link-All is similar. The only difference between them is that vMF mixture compares a

new document with all existing vMF topic distributions, while Single-Link-All uses the 1-NN as the topic of reference.

We simply use the cosine similarity $\mathbf{d}^T\mathbf{c}$ (which is proportional to the generative probability of $\mathbf{d}$ given $\mathbf{c}$) to estimate a document's similarity to every generated cluster. If the maximum similarity is below a threshold ϵ, a new cluster centered at the document d is created. If there are more documents assigned to a newly created topic later, we update the centroid of this new topic by simply averaging all documents assigned to it. This simple strategy enables us compare various online topic detection methods using the same scale of threshold ϵ.

4.3 Mixed Membership Generative Models

Mixed membership generative models represent documents as mixtures of topics, where a topic is a probabilistic distribution over words. For each word in a document, a topic is sampled according to the document's topic distribution, and the word is drawn from this topic with a given probability. In this paper, LDA [10] is used to invert this process by inferring the set of topics responsible for generating a collection of documents.

4.3.1 LDA on Offline Topic Detection

Latent Dirichlet Allocation (LDA), a generalization of Probability Latent Semantic Indexing (pLSI), is a static generative topic model that represents each document as a mixture of topics. In general, the number of topics k for LDA must be prespecified, making it more suitable for offline topic detection. As we discussed in Section 2.2, although the number of topics can automatically grow with the data using Dirichlet processes, the Dirichlet process mixture models are still designed for offline topic detection/evolution, and not the online topic detection problem emphasized in this paper. Estimating the number of topics is not a crucial issue in offline topic detection, and thus for simplicity we will use LDA as the baseline.

Formally, the LDA generative process is described as below.

- draw k multinomials $\phi \sim Dirichlet(\beta)$, one for each topic c.
- for each document d:
  - draw a topic distribution $\theta \sim Dirichlet(\alpha)$ for d.
  - for each word $w \in d$:
    - draw a topic $\mathbf{c} \sim Multinomial(\theta)$.
    - draw a word w from topic $\mathbf{c}$, $w \sim Multinomial(\phi_c)$.

In LDA, two hyperparameter vectors α and β are used to initialize the parameters of the document-topic mix θ and topic-word weights ϕ. The key part of learning the topic model hypothesis is to compute the posterior probabilities of topics given $\mathbf{d}$, which is in turn the inference of LDA, as follows:

$$P(\theta, \phi|\mathbf{d}, \alpha, \beta) = \frac{P(\theta, \phi, \mathbf{d}|\alpha, \beta)}{P(\mathbf{d}|\alpha, \beta)}.$$

Variational inference or Gibbs sampling can be used to approximate the above intractable posterior probabilities, which are beyond the scope of this paper. We will simply adopt the variational inference previously elaborated in [10].

4.3.2 LDA on Online Topic Detection

There exists no simple extension of LDA for online topic detection. As discussed in Section 2.2, both Dynamic Topic Models (DTM) [11] and non-parametric Dirichlet process mixture models [32], [37], [29], [1] were designed for topic evolution. In topic evolution, documents are grouped into discrete epochs. Documents in the current epoch are used to train a topic model. A topic evolution model will gradually evolve from the current topic space into the next newly generated topic space. For example, DTM conducts k-component LDA analysis at each time slice t sequentially, and conditionally defines the natural parameters of each topic $\phi_j(t+1)$ to be a Gaussian distribution centered upon the previous value $\phi_j(t+1)|\phi_j(t) \sim N(\phi_j(t), \sigma^2\mathbf{I})$, where $\phi_{ij}(t) = P(x_i|c_j, t)$ and $\mathbf{I}$ is the identity matrix.

In online topic detection, it is impractical to conduct either k-component LDA or unbounded Dirichlet process mixture for each incoming document. What we can do is to periodically rerun the topic model after observing/processing a bunch of new documents.

In the irrevocable online detection phase, we have to assign a topic label for each incoming document only based on the historical topic space. Since there are no similar approaches to update the topic model parameters for each incoming document as in vMF [8], we will preserve the existing topic model parameters until the next round of global optimization. There are a number of methods to assign an existing topic label to a new document. First, for each word in the new document, we can model it as a mixture of existing topics since the latter will likely have non-zero generative probability with respect to the word. The standard EM algorithm can be used to compute the mixture weights for every word while maximizing the likelihood for the document. Subsequently, by summing over all words, we can assign to the document the topic that has the largest influence as the final document label. Second, we can simply use the maximum likelihood method to assign initial topic labels for every word and then choose the topic assigned to the largest number of words. The first method is impractical since it can take several minutes to make a decision for each new document while the second one is fast. However, neither of them is suitable for creating a new topic. Given a new document, the weights of existing topics would sum to 1 using either of the two methods, which means using an absolute threshold to determine if a new topic should be created is infeasible.

The simplest solution to tackle the above problem might be to use some global measures like cosine similarity. This is consistent with our previous choices for other online topic detection methods. We can easily map both topics and documents from the the $n-1$ dimensional simplex to the n dimensional Euclidean space using the natural parameterization approach [11], so that the cosine similarity can be calculated properly. If the maximum similarity falls below a user defined threshold ϵ, a new topic is created and centered at $\mathbf{d}$. A new topic can be simply centered at all documents assigned to it while leaving the LDA parameters of each old topic unchanged.

5 Discriminative Probabilistic Models

In this section, we shall propose our discriminative probabilistic model (DPM) for both offline and online topic detection. There are a number of driving factors that led to the

formulation of the DPM model. First, existing probabilistic models, especially LDA, seem to be overly complex for the problem of topic detection. Second, although non-probabilistic models have worked fairly well for topic detection in practice, up till now there has been no corresponding mathematical justification. For example, the TFIDF weighting formula (Eq. 2) adopted by Single-Link-All is tuned empirically, without any theoretical basis or mathematical insight.

We therefore come up with the simple discriminative models (DPM) for topic detection, with two goals: 1) simplify the overly complicated probabilistic models by removing the distribution assumption, and 2) provide a compelling theoretical framework to support the non-probabilistic models. In fact we will show in later sections that the posterior topic probability given document is a variation on the classical TFIDF formulation for both online and offline topic detection, given the condition that a set of discriminative words can be found.

5.1 Offline Discriminative Model

We assume that there exists a feature set $X \in F$ in the text collection D, with which all documents can be topically discriminated from each other, where F is the full word/vocubulary feature space. Any feature $x \in X$ can become a discriminative feature for a latent topic. For example, for the "Hurricane Mitch" topic, the words *wind*, *hurricane*, *Mitch* and *storm* might be a set of discriminative features. However, stop words like *the* will not contribute to any latent topic and are thus treated as non-discriminative features. We use non-discriminative features instead of stop words because given a data collection D, there might exist word features that are neither in the common stop word list nor relevant to any latent topic in D. Defining discriminative features is thus data-dependent.

Given a new document vector $\mathbf{d}$ and an existing topic label c_j, we are effectively assuming c_j and $\mathbf{d}$ are conditionally independent given $x \in X$. The conditional independence implies $p(c_j, \mathbf{d}|x) = p(c_j|x)p(\mathbf{d}|x)$ so that

$$p(c_j|x, \mathbf{d}) = \frac{p(c_j, \mathbf{d}|x)}{p(\mathbf{d}|x)} = \frac{p(c_j|x)p(\mathbf{d}|x)}{p(\mathbf{d}|x)} = p(c_j|x).$$

As a result

$$\sum_{x \in X} p(c_j|x, \mathbf{d})p(x|\mathbf{d}) = \sum_{x \in X} p(c_j|x)p(x|\mathbf{d}).$$

The objective discriminative probability is then given by

$$\begin{aligned} P(c_j|\mathbf{d}) &= \sum_{x \in F} P(c_j|x, \mathbf{d})P(x|\mathbf{d}) \\ &= \sum_{x \in X} P(c_j|x, \mathbf{d})P(x|\mathbf{d}) + \sum_{x \in F \setminus X} P(c_j|x, \mathbf{d})P(x|\mathbf{d}) \\ &= \sum_{x \in X} P(c_j|x)P(x|\mathbf{d}) + P(c_j|\mathbf{d}) \cdot \sum_{x \in F \setminus X} P(x|\mathbf{d}), \end{aligned}$$

where we assume that $P(c_j|x, \mathbf{d}) = P(c_j|\mathbf{d})$ for those non-discriminative features that do not contribute to any latent topic. We further assume that for any document $\mathbf{d}$, $\sum_{x \in F \setminus X} p(x|\mathbf{d}) = R$, which is a constant. In other words, the total probability mass on the non-discriminative words is a constant for all documents. With this, we have

$$P(c_j|\mathbf{d}) = \frac{1}{1-R} \sum_{x \in X} P(c_j|x)P(x|\mathbf{d}). \tag{7}$$

Accordingly, the point assignment (assigning the most likely topic) of document d is given by $\arg\max_{c_j} P(c_j|\mathbf{d})$.

Property 1: Eq. 7 defines a valid probability distribution over all offline topics so that we can directly use it as the posterior discriminative probability.

Proof:

$$\begin{aligned} \sum_j P(c_j|\mathbf{d}) &= \frac{1}{1-R} \sum_j \sum_{x \in X} P(c_j|x)P(x|\mathbf{d}) \\ &= \frac{1}{1-R} \sum_{x \in X} \left(\sum_j P(c_j|x) \right) P(x|\mathbf{d}) \\ &= \frac{\sum_{x \in X} P(x|d)}{1-R} = \frac{1 - \sum_{x \in F \setminus X} P(x|\mathbf{d})}{1-R} = 1, \end{aligned}$$

where $\sum_j P(c_j|x) = 1$ for all offline topics. □

Finally, note that the above calculation does not require R to be a global constant across all documents. The value R is document dependent, $R(d) = \sum_{x \in F \setminus X} P(x|d)$. For topic assignment, we simply use R instead of $R(d)$ in the following.

5.1.1 Estimation of Offline Discriminative Model

In the work of [24], the right-hand-side of Eq. 7 has been shown to be equivalent to the Rocchio classifier by allowing for reasonable variations on the popular TFIDF document representation. We re-apply this seminal result onto a clustering framework as follows.

The probability distribution defined in Eq. 7 could be rewritten as

$$\begin{aligned} P(c_j|\mathbf{d}) &= \frac{1}{1-R} \sum_{x \in X} P(c_j|x)P(x|\mathbf{d}) \\ &= \frac{1}{1-R} \sum_{x \in X} \frac{P(x|c_j)P(c_j)}{\sum_{c_l \in C} P(x|c_l)P(c_l)} \cdot P(x|\mathbf{d}), \end{aligned}$$

where $P(x|\mathbf{d})$ is estimated as $f_{x,d}/|d|$, $P(c_j)$ is estimated as $N(c_j)/N$ and $P(x|c_j)$ is estimated as $1/N(c_j) \cdot \sum_{d' \in c_j} P(x|\mathbf{d}')$. Accordingly, $P(c_j|\mathbf{d})$ is reformulated as

$$P(c_j|\mathbf{d}) = \frac{1}{1-R} \sum_{x \in X} \frac{\frac{1}{N} \sum_{d' \in c_j} \frac{f_{x,d'}}{|d'|}}{\sum_{c_l \in C} \frac{1}{N} \sum_{d'' \in c_l} \frac{f_{x,d''}}{|d''|}} \cdot \frac{f_{x,d}}{|d|}.$$

By defining the *term frequency* and *inverse document frequency* of x as

$$TF'(x,d) = \frac{f_{x,d}}{|d|}, \quad IDF'(x) = \sqrt{\frac{N}{\sum_{d \in D} \frac{f_{x,d}}{|d|}}}, \tag{8}$$

$P(c_j|\mathbf{d})$ is further reformulated as

$$P(c_j|\mathbf{d}) = \frac{1}{1-R} \cdot \frac{N(c_j)}{N} \sum_{x \in X} \Big(\frac{1}{N(c_j)} \sum_{d' \in c_j} TF'(x,d') \cdot IDF'(x) \Big) \cdot \Big(TF'(x,d) \cdot IDF'(x) \Big) = \frac{1}{1-R} \cdot \frac{N(c_j)}{N} \cdot \mathbf{d}^T \mathbf{c}_j, \quad (9)$$

where

$$\mathbf{c}_j = \frac{1}{N(c_j)} \cdot \sum_{d' \in c_j} \mathbf{d}',$$

and we can express each word in the document vector as

$$w_i = TF'(x_i, d) \cdot IDF'(x_i).$$

From Eq. 9, after embedding documents into the discriminative feature space X, we see that the topic probability distribution of a document is proportional to both the absolute size of the topic and the distance (inner product) from the document to the topic's centroid. This process is equivalent to each iteration of the k-component centroid-based soft clustering based on the TFIDF vector space model, except that in our case, a larger cluster has a higher affinity compared to a smaller one.

It is interesting to take a closer look at Eq. 8, which redefines TF and IDF. Traditionally, IDF only counts term presence/absence in documents. Here, we factor the term frequency of a word for the redefined IDF. Thus, rare common words (appearing infrequently within a document but across many documents) will be assigned a reasonably high IDF value, as opposed to their traditional IDF values, which are typically low. Considering that topical words (discriminative features) are often rare common words, our model can thus effectively enhance their weights. Lebanon [28] had a similar conclusion while explaining the Riemannian metric as TFIDF-like score on the multinomial simplex. The Riemannian metric outperformed TFIDF in general text classification. In essence, our IDF variation is suitable for offline topic detection in text corpus as well as online topic detection in text streams, as shown later.

5.1.2 Applying to Offline Topic Detection

An iterative process is needed to locally optimize the clustering process for the discriminative model. For hard partitioning, the process can exactly follow the SPK algorithm. During each iteration, point assignment of each document is used to determine the cluster label. For soft partitioning, the process can mimic the vMF algorithm, where the topic probability distribution in Eq. 7 is used to assign documents, and the clusters are updated as a mixture of all documents weighed by the posterior cluster probability $P(c_j|\mathbf{d})$. The constant R is estimated using a bag of well-known stop words.

5.2 Online Discriminative Model

In this section, we shall extend the static discriminative model to a dynamic version, where documents at different times are not *exchangeable*. In an online model, both topics and discriminators (word feature) are time-dependent. That is to say, the topical meaning of discriminator shifts over time, and a topic has different representations at different times.

We first compute the posterior probability of assigning a new document vector $\mathbf{d}$ to class c_j as

$$P(c_j|\mathbf{d},t) = \sum_{x \in F} P(c_j|x,\mathbf{d},t) P(x|\mathbf{d},t) = \sum_{x \in X} P(c_j|x,\mathbf{d},t) P(x|\mathbf{d},t) + \sum_{x \in F \setminus X} P(c_j|x,\mathbf{d},t) P(x|\mathbf{d},t) = \sum_{x \in X} P(c_j|x,t) P(x|\mathbf{d},t) + P(c_j|\mathbf{d},t) \cdot \sum_{x \in F \setminus X} P(x|\mathbf{d},t),$$

where we, as in offline DPM, we assume that c_j and $\mathbf{d}$ are conditionally independent given $x \in X$ and time t, and $P(c_j|x,\mathbf{d},t) = P(c_j|\mathbf{d},t)$ for non-discriminative features. Similar to the offline model, we further assume that for any document $\mathbf{d}$, $\sum_{x \in F \setminus X} p(x|\mathbf{d},t) = R$ is a constant. As a result

$$P(c_j|\mathbf{d},t) = \frac{1}{1-R} \sum_{x \in X} P(c_j|x,t) P(x|\mathbf{d},t). \quad (10)$$

The point assignment of document vector $\mathbf{d}$ is given by $\arg\max_{c_j} P(c_j|\mathbf{d},t)$.

Property 2: Eq. 10 defines a valid probability distribution over all topics, assuming that topics are global variables until the current time, but they could have *zero* probabilities at birth.

Proof:

$$\sum_j P(c_j|\mathbf{d},t) = \frac{1}{1-R} \sum_j \sum_{x \in X} P(c_j|x,t) P(x|\mathbf{d},t) = \frac{1}{1-R} \sum_{x \in X} \left(\sum_j P(c_j|x,t) \right) P(x|\mathbf{d},t).$$

Note that only when all topics including the future ones are global variables, we can have $\sum_j P(c_j|x,t) = 1$. As a result

$$\sum_j P(c_j|\mathbf{d},t) = \frac{1 - \sum_{x \in F \setminus X} P(x|\mathbf{d},t)}{1-R} = 1.$$

For online topic detection, let C_o be the set of old topics and C_n be the set of unseen (new) topics. Suppose that $\mathbf{d}$ belongs to some new topic, we have $\sum_{c_j \in C_o} P(c_j|x,t) + \sum_{c_j \in C_n} P(c_j|x,t) = 1$ for those discriminative features belonging to the new topic. Apparently, now $\sum_{c_j \in C_o} P(c_j|\mathbf{d},t) < 1$. Although we cannot directly calculate the posterior probability of $\mathbf{d}$ belonging to its new topic, we can use

$$1 - \sum_{c_j \in C_o} P(c_j|\mathbf{d},t)$$

to estimate its probability of belonging to any new topic. □

5.2.1 Estimation of Online Discriminative Model

We assume there is no explicit dependency between document d and time t given the temporal descriptor x.

This conditional independence between d and t is reasonable because given d, its publication time t is also given

explicitly, and its generative probability only relies on the bag of words, which leads to $P(\mathbf{d}, t|x) = P(\mathbf{d}|x)P(t|x)$.

In the following, we use all seen documents assigned to c_j as its representation,

$$P(c_j|x,t) = \frac{P(c_j,x,t)}{P(x,t)} = \frac{P(x|c_j,t)P(c_j|t)}{P(x|t)} = \frac{\frac{1}{N(c_j,\tau)}\sum_{d'\in c_j(\tau)} P(x|\mathbf{d}',t)P(c_j|t)}{P(x|t)}, \quad (11)$$

where τ indicates the time period $t' \leq t$, $N(c_j,\tau)$ is the number of documents in cluster c_j up to time t.

Inserting Eq. 11 to Eq. 10, we have

$$P(c_j|\mathbf{d},t) = \frac{1}{1-R} \cdot \sum_{x\in X} \frac{\frac{1}{N(c_j,\tau)}\sum_{d'\in c_j(\tau)} P(x|\mathbf{d}',t)P(c_j|t)}{P(x|t)} P(x|\mathbf{d},t).$$

It is easy to estimate both $P(x|t)$ and $P(c_j|t)$ as follows:

$$P(x|t) = \frac{\sum_{d'\in D(t)} TF'(x,d')}{N(t)}, \quad P(c_j|t) = \frac{N(c_j,\tau)}{N(\tau)},$$

where $TF'(x,d')$ is the normalized term frequency of word x in document d' as given in Equation 8, $N(t)$ is the number of documents at time t, $N(\tau)$ is the number of documents up to time t. Apparently, $1/\sqrt{P(x|t)}$ works as the IDF and $P(c_j|t)$ could be used to normalized the topic size.

The remaining task is to derive $P(x|\mathbf{d},t)$ or $P(x|\mathbf{d}',t)$, both of which has the same form. Intuitively, we can treat $P(x|\mathbf{d},t)$ as dynamic term frequency which incorporates the temporal information into the static term frequency, $P(x|\mathbf{d})$. Following Bayes' rule, we have

$$\begin{aligned} P(x|\mathbf{d},t) &= \frac{P(x,\mathbf{d},t)}{P(\mathbf{d},t)} = \frac{P(\mathbf{d},t|x)P(x)}{\sum_{x'\in F} P(\mathbf{d},t|x')P(x')} \\ &= \frac{P(\mathbf{d}|x)P(t|x)P(x)}{\sum_{x'\in F} P(\mathbf{d}|x')P(t|x')P(x')} \\ &= \frac{\frac{P(x|\mathbf{d})P(\mathbf{d})}{P(x)}\frac{P(x|t)P(t)}{P(x)}P(x)}{\sum_{x'\in F}\frac{P(x'|\mathbf{d})P(\mathbf{d})}{P(x')}\frac{P(x'|t)P(t)}{P(x')}P(x')} \\ &= \frac{P(x|\mathbf{d})P(x|t)\frac{1}{P(x)}}{\sum_{x'\in F} P(x'|\mathbf{d})P(x'|t)\frac{1}{P(x')}}, \end{aligned}$$

where $P(x|\mathbf{d})$ is the static TF, $P(x|t)$ is the inverse IDF at time t, and $P(x)$ is the normalized cumulative inverse IDF up to time t, defined as follows,

$$P(x) = \frac{\sum_{d''\in D(\tau)} TF'(x,d'')}{N(\tau)}.$$

We can further simplify $P(x|\mathbf{d},t)$ to be

$$P(x|\mathbf{d},t) = \frac{P(x|\mathbf{d})P(x|t)\frac{1}{P(x)}}{\sum_{x'\in X} P(x'|\mathbf{d})P(x'|t)\frac{1}{P(x')} + R_S}, \quad (12)$$

since $R_S = \sum_{x'\in F\setminus X} P(x'|\mathbf{d})P(x'|t)\frac{1}{P(x')}$ can be a constant.

Similarly, by defining the *dynamic term frequency* and *dynamic inverse document frequency* of x as

$$TF''(x,d) = P(x|\mathbf{d},t), \quad IDF''(x) = \frac{1}{\sqrt{P(x|t)}}, \quad (13)$$

$P(c_j|\mathbf{d},t)$ can be reformulated as

$$\begin{aligned} P(c_j|\mathbf{d},t) &= \frac{1}{1-R}\cdot\sum_{x\in X}\frac{1}{N(\tau)}\cdot\Big(\sum_{d'\in c_j(\tau)} TF''(x,d')\cdot IDF''(x)\Big)\cdot\Big(TF''(x,d)\cdot IDF''(x)\Big) = \frac{1}{1-R}\cdot \frac{N(c_j,\tau)}{N(\tau)}\cdot\sum_{x\in X}\Big(\frac{\sum_{d'\in c_j(\tau)} TF''(x,d')\cdot IDF''(x)}{N(c_j,\tau)}\Big) \\ &\cdot\Big(TF''(x,d)\cdot IDF''(x)\Big) \\ &= \frac{1}{1-R}\cdot\frac{N(c_j,\tau)}{N(\tau)}\cdot\mathbf{d}^T\mathbf{c}_j(\tau), \end{aligned} \quad (14)$$

where $\mathbf{d}$ is the document vector and $\mathbf{c}_j(\tau)$ is the topic mean vector up to time t as shown below,

$$\mathbf{c}_j(\tau) = \frac{1}{N(c_j,\tau)}\sum_{d'\in c_j(\tau)}\mathbf{d}', \quad (15)$$

and each word of the document vector is expressed as

$$w_i = TF''(x_i,d)\cdot IDF''(x_i).$$

From Eq. 14, we see that by only using discriminative features, the online discriminative model is equivalent to a variation of incremental TFIDF clustering, with the following observations:

1) Both TF and IDF are time-dependent. The TF part considers the generative probability of a given document and the IDF part accounts for the document frequency (DF), both at the current time;
2) Topics must be global variables until the current time, but they could have *zero* probabilities initially. Once a topic is created, it will always be valid;
3) Larger clusters have a higher affinity to new documents compared to smaller ones.

5.2.2 Applying to Online Topic Detection

Property 2 defines an effective way to directly estimate the probability of announcing a new topic for each incoming document. However, for a consistent comparison with other methods like vMF and LDA, we compare the maximum value from $\{P(c_j|\mathbf{d},t),\ c_j \in C_o\}$ with the user defined threshold ϵ. Note that with fixed number of existing topics, this maximum value is inversely proportional to the probability that $\mathbf{d}$ indicates any new topic. If this maximum value is below ϵ, a new topic is created and its topic vector is set to be the new document's vector. Finally, Eq. 15 defines an efficient way for updating the topic vectors for both existing topics and newly created topics.

5.3 Exploring Discriminative Features

Both the online and offline DPM models make one key assumption, that documents can be represented by topic-discriminative features. However, there is no general definition for a discriminative word. We can use Linear Discriminative Analysis [31] or other techniques to find discriminators, and compare their performances on topic detection accordingly. However, this is beyond the scope of the paper. For simplicity, we select bursty words as discriminative words in this paper.

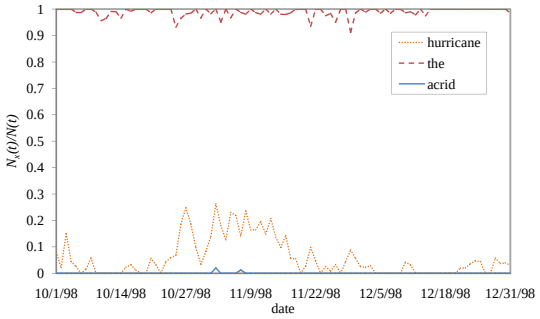


Fig. 1. Document frequency signal for three word examples. The x-axis shows the publication date, and the y-axis gives the normalized document frequency at each day.

5.3.1 Definition of Bursty Words

Definition 1: (burstiness) A word in a news stream is bursty if it appears in a large number of documents over a finite time window.

In practice, a bursty word exhibits high document frequency over a finite time window, which is distinct from rare words (low document frequency) and stop words (consistently high document frequency).

To illustrate typical "burstiness" behavior, consider the following time series plots of document frequency for three words in the TDT3 dataset as shown in Figure 1: common word "the", rare word "acrid", and topical word "hurricane" that is highly related to the topic "Hurricane Mitch". We see that only the topical word "hurricane" has the burstiness behavior as expected, and its bursts is related to two Hurricane topics: *Hurricane George* in Oct, 1998 and *Hurricane Mitch* in Nov, 1998. We thus hypothesize that, there exists three categories of words in a text stream: common words, rare words, and bursty words, of which only bursty words are *potentially* discriminative with respect to latent topics.

5.3.2 Incorporating Bursty Words

Bursty word identification from text streams have recently been investigated by a number of researchers [25], [44], [16], [20]. Since our goal is to utilize bursty words and not to develop a new bursty word identification algorithm, we simply adopt the word trajectory energy approach proposed in our previous work [20] to identify the bursty weight of each word.

We treat each and every word as a document frequency trajectory, i.e., $y_x = [y_x(1), y_x(2), \ldots, y_x(T)]$, where each element $y_x(t)$ is a measure of word feature x at time t, which could be defined using the normalized DF score

$$y_x(t) = \frac{N_x(t)}{N(t)},$$

where $N_x(t)$ is the number of documents containing word x at day t, and $N(t)$ is the number of documents for day t.

We decompose the word trajectory $y_x = [y_x(1), y_x(2), ..., y_x(T)]$ into a sequence of T complex numbers $[X(1), \ldots, X(T)]$ via the discrete Fourier transform:

$$X(k) = \sum_{t=1}^{T} y_x(t) e^{-\frac{2\pi i}{T}(k-1)t},\ k = 1, 2, \ldots, T.$$

TABLE 3
Bursty score examples.

word	$b_x(T)$
hurricane	9.1608
the	0.0990
acrid	0.0011

We define the word trajectory energy as the bursty weight until time T, by using the dominant power spectrum of a given word feature x

$$b_x(T) = \|X(k)\|^2,\ \ with\ \ \|X(k)\|^2 \geq \|X(j)\|^2,\ \ \forall j \neq k. \tag{16}$$

Given the word corpus F, where μ_b and σ_b are the mean and standard deviation of bursty weights of all words respectively, the normalized bursty weight $b'_x(T)$ is written as,

$$b'_x(T) = \frac{1 + \frac{2}{\pi} \arctan \frac{b_x(T) - \mu_b}{\sigma_b}}{2}.$$

Table 3 lists the raw bursty scores for the three sample words: *hurricane*, *the*, and *acrid*. Compared to their trajectories as shown in Figure 1, we see that the bursty word *hurricane* has a significantly larger bursty score, while both common (e.g.,*the*) and rare words (e.g.,*acrid*) have very low bursty scores.

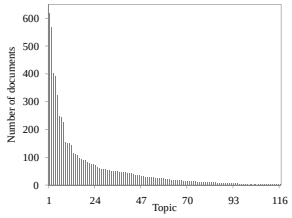
For offline topic detection, the simplest way of simulating the discriminative features is to set a normalized bursty score threshold. Words with normalized bursty scores above the threshold are chosen as discriminative features. Such a threshold can be empirically determined. For online topic detection, in the early stage the bursty scores of words are not accurate. We thus use a simple heuristic to enhance discriminative words by incorporating the bursty score into the original cumulative TFIDF score as $TFIDF + \lambda \times b'_x(T)$, where the optimal parameter λ can be estimated via cross-validation on a subset of the seen data.

6 Experiments

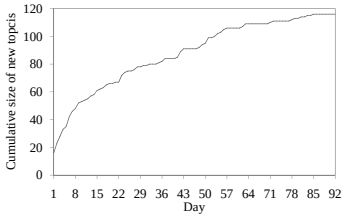
6.1 Dataset and Data Preprocessing

We use the standard TDT3 dataset, one of the few news datasets with both class labels and timestamps, released by the TDT community as the testbed. The TDT3 dataset includes 51,183 multilingual news documents collected during the three month period (92 days) of October through December 1998. We extracted *all* on-topic English news documents first. Among these, 6,502 documents covering 116 topics consist of **TDT3-Single**, with each document labeled with a single topic. The other 928 on-topic English news documents are treated as **TDT3-Multiple** covering 73 topics where each document belongs to multiple topics.

We used **TDT3-Single** as the testbed for online topic detection, and **TDT3-Multiple** as the testbed for offline topic detection. The distribution of document count in each topic for **TDT3-Single** is shown in Figure 2 (a). **TDT3-Single** is rather unbalanced, wherein only 60 topics contain more than 20 documents, and 15 topics contain more than 100 documents. In **TDT3-Single**, new topics are created continuously from the



(a) Topic distribution of documents.



(b) The cumulative count of topics versus day.

Fig. 2. Analysis of **TDT3-Single**.

TABLE 4
The distribution of # topics for documents in **TDT3-Multiple**.

# topics	2	3	4	5	6
# documents	837	84	6	0	1

first day until the 85th day. The cumulative count of topics over time is depicted in Figure 2 (b). In **TDT3-Multiple**, the majority of the documents (90%) belong to exactly two topics and the remaining documents are labeled with up to 6 topics. The topic distribution is illustrated in Table 4.

After stemming, 36,521 distinct words (set F) are retained in **TDT3-Single** and **TDT3-Multiple** without removing stop words. We did not remove the stop words because stop word removal is one of the functionalities of our discriminative word identifying method. A total of 1,473 discriminative words (set X) are identified over the whole corpus by setting the normalized bursty score threshold to 0.1; this is the discriminative feature that will be used in our DPM model for offline topic detection. For online topic detection, the parameter λ was estimated to be around 1. For parameter settings of vMF and LDA, $\kappa = 1$, $\alpha = 0.1$, $\beta = 0.01$. Version 2 of the open source indexing software Lucene was used to tokenize/index each document into a document-word vector.

6.2 Offline Topic Detection - Soft Partition

6.2.1 Methodology

In many cases a document could belong to multiple topics. Given a document d, we would like to generate a set of probabilities $P(c_j|\mathbf{d})$ for various c_j. In general, only the top m probabilities are meaningful. We shall compare the soft partitioning performances between vMF[2], LDA and DPM. Many methods have been proposed to decide the optimal k for clustering [23], and we can also apply non-parametric Dirichlet process mixture models to automatically grow k with the data. However, this is not the focus of this paper. For simplicity, here we only examine the external performances of soft partition by varying m and fixing $k = 73$ to be the correct number of clusters for **TDT3-Multiple**. In the extended version of the paper, we tested a simple yet popular method, which produces the optimal k by identifying a "knee" in the plot of MSE (mean squared error) vs. k [23].

2. vMF can be viewed as a generalized version of SPK for soft-clustering.

6.2.2 Evaluation Metrics

Since a document can be assigned to all clusters, the traditional contingency table is thus meaningless. Even if we only select the top m ($m > 1$) clusters for each document, the classical purity/entropy measures is unable to capture the ordering. For example, a low class entropy does not necessarily indicate a high recall, since documents originating from this class may belong to other classes as well. We thus need to devise new external evaluation measures.

Given a document d, we cannot find one-to-one relationship between its clusters and category labels. Alternatively, we consider pairwise scores given to a pair of documents. In [9], the pairwise F-measure was defined where each document can only belong to one cluster. Here we introduce an extension to the metric where each document can belong to a subset of clusters, and define the pairwise score similarly to Rand index as follows.

Given a pair of documents d_i and d_j, there are three types of class/cluster membership counts:

- a: number of class/cluster containing both d_i and d_j;
- b: number of class/cluster containing only d_i or only d_j;
- c: number of class/cluster containing neither d_i nor d_j.

Accordingly, for each document pair we have the ground truth vector $z(d_i, d_j) =< a_z, b_z, c_z >$ tallying class memberships, and the clustering result vector $c(d_i, d_j) =< a_c, b_c, c_c >$ tallying cluster memberships. We can define a general pairwise metric by weighing different types of class/cluster membership counts as

$$y(d_i, d_j) = \frac{w_a \min(a_z, a_c) + w_b \min(b_z, b_c) + w_c \min(c_z, c_c)}{\max(w_a a_z + w_b b_z + w_c c_z, w_a a_c + w_b b_c + w_c c_c)},$$

where $w_a, w_b, w_c \geq 0$ are weights on the three types of similarities. It is easy to see that $y(d_i, d_j) \in [0, 1]$, and $y = 1$ means a perfect match and $y = 0$ indicates the worst case. The advantage of this general weighted metric lies in the flexibility of setting the weights (w_a,w_b,w_c).

In this paper, for simplicity we set $w_a = w_b = w_c = 1$ and have

$$y(d_i, d_j) = \frac{\min(a_z, a_c) + \min(b_z, b_c) + \min(c_z, c_c)}{k},$$

where $a_z + b_z + c_z = a_c + b_c + c_c = k$. The pairwise score not only considers the "right" classes/clusters where d_i and d_j should behave identically (a and c), but also takes those classes/clusters where they repel each other into account (b). If we only consider one number like a, for each cluster where d_i and d_j appear together, the other documents in this cluster would have nothing to do with the ground truth labels of d_i and d_j. However, after collectively considering all three numbers, if the class memberships of the other documents do not match the ground truth labels of d_i and d_j, putting them together in the same cluster would reduce the value of b_c (compared to b_z) and increase the value of c_c (compared to c_z), although a_z and a_c might still be the same. Therefore, the final performance would be penalized by b and c. Hence, only putting documents sharing the same class label into the same cluster and finding out the correct number of clusters k could satisfy $a_z = a_c$, $b_z = b_c$ and $c_z = c_c$ at the same time, which leads to the

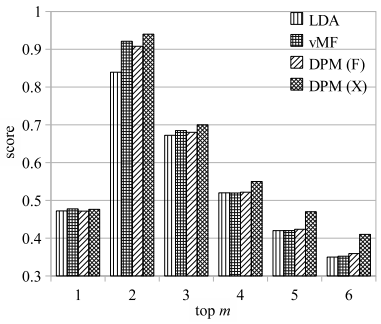


Fig. 3. Average pairwise scores across various top m soft clusters.

optimal score $y = 1$. The pairwise score thus seems fairly intuitive and reasonable, though a theoretical proof will be nice, and is left as future work.

In our experiment, m is meaningful only at small values because the average number of topics assigned to any document is small (cf. Table 4). Since $k = 73$ is much larger than m, the value of c would dominate the composition of k for both ground truth and clustering results, which results in a consistently large y. For a better comparison across various m, we set $w_a = w_b = 1, w_c = 0$ to obtain

$$y(d_i, d_j) = \frac{\min(a_z, a_c) + \min(b_z, b_c)}{\max(a_z + b_z, a_c + b_c)}.$$

Note that this new pairwise score definition still maintains the properties discussed above as $a_z + b_z + c_z = a_c + b_c + c_c = k$.

For the document corpus, we define the overall measure as the average pairwise score over all pairs of documents as

$$y(D) = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} y(d_i, d_j).$$

6.2.3 Result Analysis

The running time of all models except LDA grows supralinearly with k. Since the time efficiency of offline topic detection is not as crucial as online topic detection, the details are omitted. Figure 3 illustrates the average pairwise scores by varying m from 1 to 6 (the maximum possible number of topics assigned to one document). All results were averaged over 10 runs. To evaluate the bursty feature selection aspect of DPM separately, we created a DPM version based on all word features (set F) and denoted it by DPM (F). The DPM based only on selected bursty word features is denoted by DPM (X). Not surprisingly, all models achieved the best performance at $m = 2$ since about 90% of the documents have 2 assigned topics according to Table 4. For $m > 2$, the performance drops gradually, as expected.

Overall, LDA yielded the worst performance. For example, it has the lowest average pairwise score of 0.84 at $m = 2$ (the most important position; also the optimal value after using the internal similarity to automatically determine m) while every other models scored more than 0.9. It is not hard to understand why DPM and vMF showed vast improvements over LDA on soft clustering; since we evaluate the soft clustering performance based on the multiple categorical labels of documents, not words. On average, vMF slightly outperformed DPM (F) (0.92 versus 0.91 at $m = 2$). This is not surprising because vMF has been shown to be an effective document-level soft clustering algorithm in the past [8]. After working on the discriminative features, DPM (X) in turn outperformed vMF (0.94 versus 0.92 at $m = 2$). We also tested vMF on the same set of discriminative features of DPM (X), with slightly worse results (i.e., 0.9 at $m = 2$). This means that discriminative features are more suitable for discriminative probabilistic models like DPM. For a generative model like vMF, due to its generative smoothing process, the utility of discriminative features is attenuated. We also note that while using bursty words helped, DPM (X) did not enjoy a remarkable improvement, probably because bursty words goes only so far as universal discriminative words. We believe that a more systematic selection of discriminative features based on class labels could further improve soft clustering performance.

Moreover, from Figure 3 we observe that as m increases, DPM (X) starts to break away from the pack including DPM (F). This is interesting because with increasing m, the quality of the late topic assignments (remaining clusters are equally far away for high m) tend to deteriorate quickly. It is precisely for these late topic assignments that bursty words start to play an important role; related documents are pulled closer together by the common bursty words in the farther (high m) clusters [19].

6.3 Online Topic Detection - Point Assignment

6.3.1 Methodology

Compared to offline topic detection, online topic detection is a more challenging problem. Our goal here is to devise an efficient and effective algorithm for online topic detection. We shall adopt the non-probabilistic model, Single-Link-All, as our topic-detection strategy [4]. The major problem of deploying Single-Link-All in practice [5] is that the size of detected cluster is frequently either too large or too small. In other words, cluster quality is typically very poor. Allan et al. [5] suggested two possible remedies: *hierarchical clustering* and *increasing the threshold* ϵ. Both solutions could break large clusters into pieces. However, to avoid producing too many small clusters, additional strategies should be considered simultaneously.

We focus on the second remedy since we deal only with a flat topic structure in this paper. Increasing the threshold ϵ will increase the difficulty of inserting a new document into an existing cluster. Accordingly, it is easier to create new clusters, as well as false alarms. Our target is then very clear: reduce the false alarms. Allan et al. [5] further considered two engineering solutions for reducing the false alarms. First, the clusters have "age" so that it is difficult to add new documents to an "aged" cluster. Second, each existing cluster is represented by its averaged centroid (not corresponding to any real document) rather than the single most representative document. Accordingly, we define two more baseline models, *Single-Link* and *Incre-Mean*, for online topic detection. In Single-Link, every new incoming document is compared to documents published within the last 7 days. We simply used the sliding window of 7-day because most of topics in the

real news don't last for more than one week. In Incre-mean, each cluster is represented by its centroid, which is continually updated with new member documents. Clusters that have not been updated for a certain time period, i.e., 7 days, will be effectively discarded and archived, and not considered during new topic detection. Every new document is thus compared to the centroid of a valid existing cluster.

All three baseline models incrementally increase the number of clusters k, but none of them perform any global optimizations at any time. For practical online topic detection, some modifications are necessary. First, global clustering is periodically conducted on the latest set of documents. Second, the number of clusters k has to be reduced to remove false alarms. These modifications are summarized in Algorithm 1, which will be used for the different online topic detection models.

Algorithm 1 (Online Topic Detection) (D, ϵ)

Input: News stream D, and threshold ϵ;
Output: detected (new) clusters on the fly;

1: create the first topic ($k = 1$, $c_k \leftarrow d_1$) and announce it;
2: **repeat**
3: **if** a new day begins **then**
4: remove those obsolete clusters which have not been updated for the past 7 days (reduce k at the same time);
5: run k-component topic detection model on the past 7-day data: a divide and conquer strategy is used to optimize k by comparing the internal similarity of clustering results;
6: **end if**
7: Compute the similarity of new document d with the k existing cluster centroids (or, compute the posterior probabilities of k existing topics given d);
8: **if** the largest similarity is greater than ϵ **then**
9: assign d to the nearest cluster, and update this cluster's parameters if allowed by model;
10: **else**
11: create a new topic ($k = k + 1$, $c_k \leftarrow d$) and announce it;
12: **end if**
13: **until** no new document comes.

Algorithm 1 is indeed an extension of Single-Link and Incre-mean, with two major changes: 1) a global clustering is conducted every day; 2) a simple "divide and conquer" strategy is adopted to reduce the number of clusters k.

6.3.2 Evaluation Metrics

The performance of online topic detection largely relies on the threshold ϵ. We adopt the Detection Error Tradeoff (DET) curve [30] which has been widely used in TDT [26] to measure the miss and false alarm values at each threshold. Tracing the DET curve, TDT defines the official evaluation measure as a cost function, which is a weighted combination of miss and false alarm values, as follows [5],

$$Cost = C_{miss}P(miss)P(target) + C_{fa}P(fa)P(offtarget), \quad (17)$$

where $P(target) = 1 - P(offtarget)$ is the prior probability that a document will be a new topic (0.02, derived from the TDT training data), $C_{miss} = 10$ and $C_{fa} = 1$ are user-specified penalty factors, and $P(miss)$ and $P(fa)$ are the empirical miss and false alarm probabilities by comparing clusters announced by Algorithm 1 with the ground truth. The cost function is further normalized because the system would get a default score of 0.2 if it fails to detect any new topic, and a score of 0.98 if it accepts each and every new document as the new topic. The final cost is divided by 0.2, indicating the system with a detection cost of 1.0 is no better than a system that does not detect any new topic.

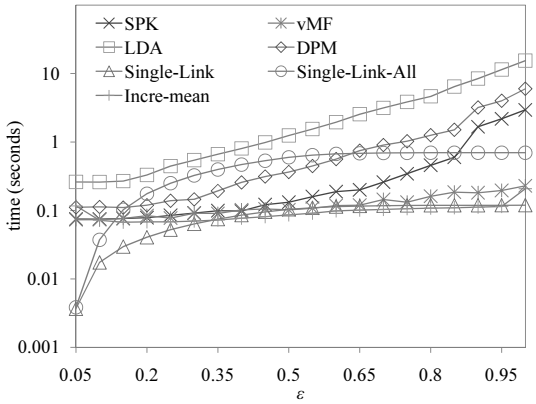


Fig. 4. Online topic detection time (log-scaled).

The above TDT evaluation metric actually still allows for a large proportion of misses, despite it already penalizing misses more than false alarms. We have discussed before that increasing the threshold ϵ could help remove those overly large clusters. In fact, increasing the threshold ϵ could also reduce the miss rate, at the cost of producing more false alarms. In practice, we are more interested in whether topic detection models could reduce the false alarms under a very low miss rate, i.e., $P(miss) = 0$.

6.3.3 Result Analysis

We only use **TDT3-Single** as the testbed for online topic detection as it is neither straightforward nor meaningful to evaluate the miss/false alarm for a document with multiple topic labels. Figure 4 shows the running time of the various online topic detection models.

As the threshold ϵ increases, the non-probabilistic models tend to be constant because: 1) all documents are detected as new topics (false alarm rate is maximized) after a certain value of ϵ, i.e., for Single-Link-All; 2) the number of documents used for comparison maintains stable after introducing the sliding window (size of 7 days), i.e., for Single-Link and Incre-mean. Single-Link and Incre-mean are the fastest methods among all models because no iterations are involved and the fewest number of documents (within the sliding window) need to be compared. On the contrary, the running time of probabilistic models grows significantly as the number of clusters k increases *largely* (induced by the increment of ϵ), except for vMF which converges so quickly that it is comparable to non-probabilistic models. More specifically, the online running time of LDA increases the most dramatically. Without feature selection for online topic detection, DPM also needs considerable time. Its discrete probabilistic topic assignment required a bit more time compared to the point topic assignment adopted by SPK. However, considering that

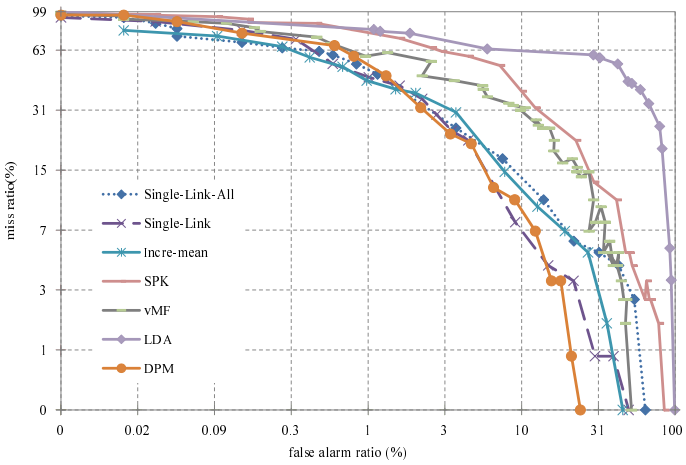


Fig. 5. The DET curve for online topic detection.

TABLE 5
Minimal DET cost values of all online models.

	Single-Link-All	Single-Link	Incre-mean	
cost	0.4326	0.4133	0.4702	
	SPK	vMF	LDA	DPM
cost	0.7680	0.7069	0.8428	0.4008

a reasonably small value of ϵ is often selected in practice, both DPM and SPK are even faster than the traditional Single-Link-All method. Therefore, we conclude that our proposed DPM probabilistic model combined with Algorithm 1 to be highly efficient for online topic detection.

Figure 5 illustrates the log-scaled DET curve for all models on **TDT3-Single**. The DET curves in Figure 5 are a bit complicated. In the upper left region, Single-Link-All and Incre-mean have the best miss/false alarm balance. Afterwards, DPM and Single-Link lead the pack respectively. Finally, DPM has the lowest false alarms under very low miss rates. That is to say, different models have their own bias/edges towards different DET regions. For example, Incre-mean works well under a small threshold ϵ (very few false alarms) by producing the largest number of correct topics, and DPM has the best performance under a large threshold ϵ (very low miss rate), where the fewest number of false alarms are produced. The other three models, SPK, vMF and LDA, always perform not well along the whole DET curve.

Table 5 lists the minimal DET cost values of all seven models. Not surprisingly, DPM achieved the minimal[3] cost value of 0.4008, followed by Single-Link which achieved a close second at 0.4133. Comparing Single-Link with Single-Link-All, we see that the sliding window did not increase the cost (even a slight improvement), yet contributed a lot in speeding up the probabilistic models (largely reduced the number of documents on clustering). Incre-mean did not improve the performance, which is consistent to the early finding that representing clusters of documents by their centroid was not effective [4]. This further verifies our hypothesis that a global clustering process is necessary from time to time. Although

3. The best TDT system achieves about a 0.3 cost value, with a 28% miss rate and a 0.3% false alarm rate on average [5].

TABLE 6
Compare false alarms under the *zero* miss rate.

	Single-Link-All	Single-Link	Incre-mean	
false alarm (%)	63.97	49.83	62.72	
	SPK	vMF	LDA	DPM
false alarm (%)	85.08	61.42	94.31	24.15

DPM has not improved a lot on the cost value, it achieved a much smaller false alarm rate (less than half of others) under the zero miss rate, as shown in Table 6.

All in all, Algorithm 1 can significantly reduce false alarms by merging small clusters, and periodic global clustering based on flat topic detection models can enhance the overall clustering for online topic detection. However, partially due to the simplicity of Algorithm 1, not all topic detection models could achieve the goal. For example, if we only consider the cumulative TFIDF without paying attention to the DF or the bursty properties of words/topics, SPK and vMF cannot cluster documents well on-the-fly, which will often lead to huge clusters. The LDA topic model, on the other hand, performed poorly for online topic detection since it was primarily designed for offline word clustering.

7 Conclusions

In this paper we studied a set of topic detection models on both offline and online topic detection problems for news streams. We first investigated the traditional non-probabilistic models, along with their limitations on topic detection, i.e, no theoretical explanation, documents cannot belong to multiple topics, and it is hard to tune the parameters of online topic detection, etc. We then proposed a discriminative model (DPM) for topic detection in news streams, which is a simple and effective probabilistic model without the assumptions made by more complicated generative models like vMF mixture and LDA. We show the equivalence of DPM to the clustering process of a variation of TFIDF under the condition that only discriminative words are used. A simple heuristic of utilizing the bursty phenomenon of words is used to extract discriminative features. DPM in fact provides a theoretical explanation to the classical non-probabilistic models for topic detection. Moreover, we also benchmarked DPM soft-clustering performance on offline topic detection. The experimental results show that DPM is surprisingly good in assigning multiple topics to a document (offline topic detection), and reducing the overall false alarm rate (online topic detection). Our results thus lead to the main conclusion of this paper. Sophisticated models like vMF or LDA may shine when there are enough training data for accurate parameter estimation, but for the problem of topic detection, a simple and mathematically elegant model like DPM can be surprisingly effective and practical (fast). As future work, we will explore using non-parametric Dirichlet process mixture models from topic evolution. We will also consider adopting some supervised dimensionality reduction algorithm like discriminative LDA [27] for extracting discriminative features for our online topic detection model.

ACKNOWLEDGEMENT

We thank the anonymous reviewers for their keen insight and valuable feedback. This research was supported in part by NSF grant IIS-0812183.

REFERENCES

[1] A. Ahmed and E. Xing, *Dynamic Non-Parametric Mixture Models and The Recurrent Chinese Restaurant Process: with Applications to Evolutionary Clustering*, In SDM'08.
[2] James Allan, Carbonell, George Doddington, Jonathan Yamron and Yiming Yang, *Topic Detection and Tracking Pilot Study: Final Report*, In Proceedings of the Broadcast News Understanding and Transcription Workshop, 1998.
[3] James Allan, Victor Lavrenko, Hubert Jin, *First story detection in TDT is hard*, In CIKM'00.
[4] James Allan, *Topic Detection and Tracking. Event-based Information Organization*, Kluwer Academic Publishers, 2002.
[5] James Allan, Stephen Harding, David Fisher, Alvaro Bolivar, Sergio Guzman-Lara and Peter Amstutz, *Taking Topic Detection From Evaluation to Practice*, In HICSS 2005.
[6] Roger Ariew, *Ockham's Razor: A Historical and Philosophical Analysis of Ockham's Principle of Parsimony*, University of Illinois, 1976.
[7] Arindam Banerjee, Inderjit Dhillon, Joydeep Ghosh and Suvrit Sra, *Generative Model-based Clustering of Directional Data*, In SIGKDD'03.
[8] Arindam Banerjee and Sugato Basuy, *Topic Models over Text Streams: A Study of Batch and Online Unsupervised Learning*, In SDM'07.
[9] Sugato Basu, Arindam Banerjee and Raymond J. Mooney, *Active Semi-Supervision for Pairwise Constrained Clustering*, In SDM'04.
[10] Blei, D. M., Ng, A. Y., and Jordan, M. I, *Latent dirichlet allocation*, J. Machine Learning Research 2003.
[11] David M. Blei and John D. Lafferty, *Dynamic topic models*, In ICML'06.
[12] T. Brants, F. Chen and A. Farahat, *A system for New Event Detection*, In SIGIR'03.
[13] S. Dasgupta, *Learning mixtures of Gaussians*, In IEEE Symposium on Foundations of Computer Science, 1999.
[14] I. S. Dhillon and D. S. Modha, *Concept Decompositions for Large Sparse Text Data Using Clustering*, J. Machine Learning 2001.
[15] C. Elkan, *Clustering documents with an exponentialfamily approximation of the Dirichlet compund multinomial distribution*, In ICML'06.
[16] G. P. C. Fung, Jeffrey X. Yu, Philip S. Yu and H. Lu, *Parameter free bursty events detection in text streams* , In VLDB'05.
[17] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani and Liadan O'Callaghan, *Clustering data streams: Theory and practice*, J. TKDE 2003.
[18] Qi He, Kuiyu Chang and Ee-Peng Lim, *A Model for Anticipatory Event Detection* , In ER'06.
[19] Qi He, Kuiyu Chang, Ee-Peng Lim and Jun Zhang, *Bursty Feature Reprensentation for Clustering Text Streams* , In SDM, 2007.
[20] Qi He, Kuiyu Chang and Ee-Peng Lim, *Analyzing Feature Trajectories for Event Detection* , In SIGIR'07.
[21] Qi He, Bi Chen, Jian Pei, Baojun Qiu, Prasenjit Mitra and C. Lee Giles, *Detecting Topic Evolution in Scientific Literature: How Can Citations Help?* , In CIKM'09.
[22] Thomas Hofmann, *Probabilistic Latent Semantic Indexing*, In SIGIR'99.
[23] A. K. Jain, M. N. Murty and P. J. Flynn, *Data clustering: a review*, ACM Computing Surveys 1999.
[24] Thorsten Joachims, *A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization*, In ICML'97.
[25] J. Kleinberg, *Bursty and hierarchical structure in streams*, In SIGKDD'02.
[26] G. Kumaran and J. Allan, *Text classification and named entities for new event detection*, In SIGIR'04.
[27] Simon Lacoste-Julien, Fei Sha and Michael I. Jordan, *DiscLDA: Discriminative Learning for Dimensionality Reduction and Classification*, In NIPS'08.
[28] Guy Lebanon, *Learning Riemannian Metrics*, In UAI'03.
[29] W. Li, D. Blei and A. McCallum, *Nonparametric bayes pachinko allocation*, In UAI'07.
[30] Alvin Martin, George Doddington, Terri Kamm, Mark Ordowski and Mark Przybocki, *The DET Curve in Assessment of Detection Task Performance*, In Proc. Eurospeech, 1997.
[31] G.J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*, Wiley-Interscience, New Ed edition, 2004.
[32] R. Neal, *Markov chain sampling methods for Dirichlet process mixture models*, Technical Report 9815, Uni- versity of Toronto, 1998.
[33] K. Nigam, A. K. McCallum, S. Thrun and T. M. Mitchell, *Text classification from labeled and unlabeled documents using EM*, J. Machine Learning 2000.
[34] G. Salton and C. Buckley, *Term-weighting approaches in automatic text retrieval*, Information Processing and Management 1988.
[35] Nicola Stokes and Joe Carthy, *Combining semantic and syntactic document classifiers to improve first story detection*, In SIGIR'01.
[36] TDT: Annotation Manual Version 1.2, August 4 2004, http://www.ldc.upenn.edu/Projects/TDT2004.
[37] Y. Teh, M. Jordan, M. Beal and D. Blei, *Hierarchical dirichlet processes*, Technical report, UC Berkeley Statistics TR-653, 2004.
[38] Alexey Tsymbal, *The problem of concept drift: Definitions and related work*, Technical report, Department of Computer Science, Trinity College, 2004.
[39] Xuerui Wang, Andrew McCallum, *Topics over time: a non-Markov continuous-time model of topical trends*, In SIGKDD'06.
[40] Y. Yang, T. Pierce and J. Carbonell, *A Study of Retrospective and On-Line Event Detection*, In SIGIR'98.
[41] Y. Yang, J. Zhang, J. Carbonell and C. Jin, *Topic-conditioned Novelty Detection*, In SIGKDD'02.
[42] Christopher C. Yang and Xiaodong Shi, *Discovering event evolution graphs from newswires*, In WWW'06.
[43] Jian Zhang, Zoubin Ghahramani and Yiming Yang, *A Probabilistic Model for Online Document Clustering with Application to Novelty Detection*, In NIPS'05.
[44] Y. Zhu and D. Shasha, *Efficient elastic burst detection in data streams*, In SIGKDD'03.

Qi He is currently a postdoctoral researcher at Penn State working in the CiteSeerX project. He received his Ph.D. degree from Nanyang Technological University in 2008. He is interested in applications of machine learning and text/data mining techniques to Web Search, Information Retrieval and Social Networks. He was one of two recipients in Singapore to receive a Microsoft Research Fellowship in 2006, and recipient of the best application paper award in SIGKDD 2008. His work currently focuses on modeling and exploring topic evolution and author influence from the citation network and author social network, and studying citation contexts in research papers with the goal of recommending citations for new manuscripts.

Kuiyu Chang is an assistant professor of computer engineering at Nanyang Technological University, Singapore. Prior to that, he served as senior risk management analyst for ClearCommerce, USA. From 2000 to 2002 Kuiyu was a member of technical staff at Interwoven, USA. He has served as program co-chair for Intelligence and Security Informatics workshops (PAISI 2006,2007,2008), and publications chair for Pacific Asia conference on Knowledge and Data Discovery (PAKDD 2006). Kuiyu is recipient of two best paper awards (IEEE/ISI 2005 and Motorola 1996). His current research interest includes open standards and free software, information retrieval, and emotion/sentiment analysis from social networks. Kuiyu received his Ph.D. from the University of Texas at Austin.

Ee-Peng Lim is currently a professor at the School of Information Systems of the Singapore Management University (SMU). He received Ph.D. from the University of Minnesota, Minneapolis. His research interests include social network/web mining, information integration, and digital libraries. He is currently an Associate Editor of the ACM Transactions on Information Systems (TOIS), Journal of Web Engineering (JWE), IEEE Intelligent Systems, International Journal of Digital Libraries (IJDL) and International Journal of Data Warehousing and Mining (IJDWM). He is a member of the ACM Publications Board, and is a member of the Steering Committees of the Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD) and International Conference on Asian Digital Libraries (ICADL).

Arindam Banerjee is an Assistant Professor and a McKnight Land Grant Professor in the Department of Computer Science and Engineering at the University of Minnesota, Twin Cities. He received his Ph.D. degree from the University of Texas at Austin in 2005. His research interests are in Data Mining and Machine Learning, and their applications to real world problems. His work currently focuses on statistical and graphical models for learning and predictive modeling with large scale data. His research interests also include Information Theory and Convex Analysis, and applications in complex real world learning problems including problems in Text and Web Mining, Bioinformatics and Social Network Analysis.