Computing and Informatics, Vol. 31, 2012, 1533–1555

DOCUMENT CLUSTERING WITH BURSTY INFORMATION

Apirak Hoonlor

Faculty of ICT, Mahidol University
Bangkok, Thailand
&
Rensselaer Polytechnic Institute
Troy, NY, USA
e-mail: apirak.hoo@mahidol.ac.th

Bolesław K. Szymanski

Społeczna Akademia Nauk
Łódź, Poland
&
Rensselaer Polytechnic Institute
Troy, NY, USA
e-mail: szymansk@cs.rpi.edu

Mohammed J. Zaki

Computer Science Department
Rensselaer Polytechnic Institute
Troy, NY, USA
e-mail: zaki@cs.rpi.edu

Vineet Chaoji

Yahoo! Labs
Bangalore, India
e-mail: chaojv@yahoo-inc.com

Communicated by Jacek Kitowski

Abstract. Nowadays, almost all text corpora, such as blogs, emails and RSS feeds, are a collection of text streams. The traditional vector space model (VSM), or bag-of-words representation, cannot capture the temporal aspect of these text streams. So far, only a few bursty features have been proposed to create text representations with temporal modeling for the text streams. We propose bursty feature representations that perform better than VSM on various text mining tasks, such as document retrieval, topic modeling and text categorization. For text clustering, we propose a novel framework to generate bursty distance measure. We evaluated it on UPGMA, Star and K-Medoids clustering algorithms. The bursty distance measure did not only perform equally well on various text collections, but it was also able to cluster the news articles related to specific events much better than other models.

Keywords: Document clustering, bursty model, web mining

1 INTRODUCTION

Given its reach, the Internet has become a popular medium for posting electronic documents on the World Wide Web. According to [15], the indexable web contains at least 23.72 billion pages. In 2008, Google reported in [7] that they processed at least 1 trillion unique URLs. Among those billions of pages, we are interested in a group of URLs relevant to news – the URLs of news media such as those from BBC, CNN, New York Times, and so on. With the RSS feed technology, the news are constantly updated on these websites. The news articles range from entertainment news, e.g. movies, to political news, e.g. presidential elections. These news are information sources that are text streams – sequences of chronologically ordered documents.

With the explosion of digital content, clustering tools have become a necessary aid in handling, retrieving and analyzing electronic documents. Either from online or offline data, clustering analysis has been used to help organize relevant documents into meaningful groups. However, traditional text representations for text mining tasks do not always account for the time component of text streams. Recently, models of stream of documents have been used to detect and track sudden burst in words and phrases [11, 14]. In another work in [9], He et al. used Kleinberg's algorithm to extract burst detection for topic clustering on text streams. Later in [10], He et al. proposed a bursty feature representation – a document representation to account for the temporal dimension in text streams. While the bursty feature representation on text stream clustering has shown improvement over bag-of-word representation, we believe that the bursty feature representation cannot fully utilize all the burstiness information.

In general, a word can be bursty in two different time periods. Even though the bursty score between two time periods are similar, they can correspond to two entirely separate events. For example, in online news in 2011, the word "death"

was very bursty in May, and July. In May, the burstiness was attributed to the high number of news articles regarding the death of Osama bin Laden on May 1st, 2011. In July, the burstiness of the term was attributed partly to the high number of articles honoring the death of Betty Ford on July 8th, 2011. While both sets of news articles are related to the death of well-known persons, they correspond to two separate events. The current bursty feature representations cannot distinguish between the two events.

Our paper makes the following contributions. We introduce a framework to create a *bursty distance measurement* that improves utilization of bursty period and bursty score for text streams with time dependent topics. The framework is based on our burst detection method that focuses on local metrics of burstiness using kernel density estimation. The paper is organized as follows. Related works are discussed in Section 2. Section 3 provides notations and definitions used in the rest of the paper. Our framework for bursty distance measurement, a word's burstiness and bursty period analysis are given in Section 4. Section 5 describes experimental evaluation while Section 6 provides our conclusion.

2 RELATED WORK

This work is influenced by multiple research areas. In information retrieval, detecting the sudden change of the signal, or a burst in data, is often referred to as outlier detection. [1] provided a detailed summary of the methods for detection of abrupt changes in data. In [4], Curry et al. introduced the change detection method for large data sets. They also discussed issues in developing a change detection system including:

1. detection models varies from one individual to another, and
2. the most important change identified by human expert turns out to have low statistical significance.

Recent studies of temporal dynamics of information stream defined the burstiness of a word in sequential datasets [11]. In [22], Vlachos et al. used information gained from compressed representations of periodic data to detect peaks of online search queries. Lappas et al. introduced a "burstiness-aware search framework" to integrate the burstiness concepts into search [13].

Topic modeling and topic detection research is another area related to this work. In [24], Wang et al. proposed the coordinated mixture model to detect the "correlated bursty topic patterns" in document sequences. The mixture model can detect the correlation between two patterns from two completely different languages, such as English and Chinese. In language modeling context [20, 3], "burstiness" was used to identify the case when a word appeared more than once in a document. In [14], Leskovec et al. introduced an application called "MemeTracker", which can trace pieces of information, termed "meme", in the document sequence. Using this application on social network and news media, one can identify the speed

of information flow from one source to another. In [23], an Latent Dirichlett Allocation based topic model used word co-occurrences and document's timestamp to detect how the topics changed over time. In later work [10], the time interval information was incorporated directly into document representation using Kleinberg's algorithm (discussed in Section 4.1). Such document representation showed improvement in text clustering. Other related work is in event clustering research areas. Papka [17] provided a good overview of on-line new event detection, event clustering and event tracking. Fung et al. [6] studied a hot bursty event detection problem. They proposed a parameter free probabilistic method to identify bursty events in text streams. Kuta and Kitowski applied various clustering methodologies on Polish newspaper articles in [12]. They found that partition-clustering algorithms achieved better results than agglomerative-clustering algorithms on small number of clusters.

3 PRELIMINARIES

For our work, we consider a text corpus, denoted $\mathcal{S}$, as a sequence of documents which appear in n consecutive time steps. $\mathcal{S}$ is defined as $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$, where s_i refers to the set of documents that appear in the i^{th} time step. t_d is the time step when document, d, appears in $\mathcal{S}$. In turn, we define a set of documents at i^{th} time step as $s_i = \{d_1^i, d_2^i, \ldots, d_m^i\}$, where d_j^i is the j^{th} document in s_i, and $m = |s_i|$ is the number of documents in s_i. Let $\mathcal{W}$ be set of all words found in $\mathcal{S}$. We define the appearance function of a word, w, in a document, d, denoted $app(w, d)$, as a binary function. $app(w, d) = 1$ if d contains w. Otherwise, $app(w, d) = 0$. $p_{[b,e]}$ represents the period from time step b to time step e. $app(w, d) \in p_{[b,e]})$ is the number of documents containing w in $p_{[b,e]}$. We denote $occ(w, d)$ as the number of times w appear in document, d. Finally, we denoted $tfidf(t, d)$, as the term frequency-inverse document frequency (TFIDF) value of a word w in document d. For each document d, its VSM (vector space model) with TFIDF value is the tuples $\langle w_1, tfidf(w_1, d)\rangle, \ldots, \langle w_{|d|}, tfidf(w_{|d|}, d)\rangle$, where $w_i \in \mathcal{W}$. $tfidf(w_i, d) = tf(w_i, d)idf(w_i, d)$, where $tf(w_i, d)$ (TF) equals $\frac{occ(w_i,d)}{\sum_{j=1}^{|d|} occ(w_j,d)}$ and $idf(w_i, d)$ (IDF) equals $\log\left(\frac{|\mathcal{S}|}{\sum_{d_j \in \mathcal{S}} app(w_i,d_j)}\right)$. VSM with binary value vector representation of d, is the tuples $\langle w_1, app(w_1, d)\rangle, \ldots, \langle w_{|d|}, app(w_{|d|}, d)\rangle$.

4 BURST DETECTION FOR DOCUMENT CLUSTERING

Burst detection is useful in recovering two pieces of important information: the period during which a word w is bursty, and its level of burstiness in that period. A word w is considered bursty at time step i, where $i = 1 \ldots n$, if the occurrence of w at time i^{th} is greater than a certain threshold. The burst period is then defined as the period where w is bursty over consecutive time steps. In previous works, the bursty thresholds were set to the mean over all time steps. Hence, the bursty period

of a given word was defined as the period when its occurrences in this period were higher than the average.

In general, there are several ways to group documents together. Online news articles are often grouped into major categories such as, business, entertainment, politics. They can also be assigned the pre-determined keywords related to each article. In the latter case, when an event occurs, the news regarding the event would be given the same tag. Such news articles are written over a period of time following the event. The length of the active period of the news and the number of articles in the media related to the event depend on the popularity of the story. If we plot the graph of the number of news articles related to an event on a timeline, it is likely to increase in the beginning of the active period of the event and to decay afterward.

The active period of the event often corresponds to the bursty period of the keywords related to the event. For example, the occurrence plot of documents containing keywords "victoria" and "death" in the San Francisco Call newspaper between January, 1900 and December, 1901 is shown in Figure 1. The highest peaks between the 12^{th} and the 13^{th} month – January 1901 and February 1901 – correspond to the death of Queen Victoria in late January of 1901. Note that from the document clustering perspective the documents are clustered according to the "similarity" scores between the documents. From this example, if we increase the similarity scores between the documents containing these two words, during such peaks, the documents related to this event would likely be clustered into the same group.

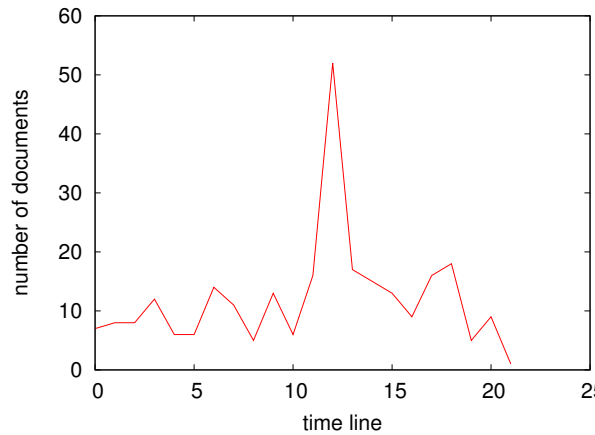


Fig. 1. Numbers of articles in San Francisco Call newspaper containing words "victoria" and "death" per month from Jan. 1900 to Dec. 1901

4.1 Bursty Features for Document Clustering

In [10], He et al. used bursty information to create the bursty features for document clustering. The bursty feature representation for a document $d = w_1, w_2, \ldots, w_n$ at time t_d is defined as follows.

$$w_i = \begin{cases} FP_i + \beta b_i & \text{if } w_i \text{ is bursty at time } t_d \\ FP_i & \text{otherwise} \end{cases} \quad (1)$$

where FP_i is the static feature weight such as binary weighting, b_i is the bursty score and $\beta > 0$ is the burst coefficient. There are two existing burst detection methods that can be applied to Equation (1): Kleinberg's algorithm introduced in [11], and Discrepancy model of Burstiness [13]. Both of them defined the bursty period of a word w as the period during which the documents containing w appeared more frequently than on average.

4.1.1 Kleinberg's Algorithm

Kleinberg [11] proposed to identify the bursty period of t using finite-state automaton. The Kleinberg's two-state finite automaton, $\mathcal{A}$, has q_0 and q_1 states. They represent the normal period and the bursty period, respectively. $\mathcal{A}$ is in q_1 state at time a, if the emission rate is $\frac{s \cdot |D_w|}{n}$, where $|D_w|$ is the number of documents containing w, n is the total number of time steps in S, and $s > 1$. $\mathcal{A}$ is in q_0 state otherwise. a is a bursty period if $\mathcal{A}$ is in q_1 state at time a. The cost of being in state q_i of a word w at time i is defined in Equation (2)

$$\sigma(i, app(w, s_i), |s_i|) = -ln\left[\binom{|s_i|}{app(w, s_i)} p_i^{|s_i|}(1 - p_i)^{|s_i| - app(w,s_i)}\right] \quad (2)$$

where $app(w, s_i)$ is the number of documents in s_i that contain w at time i, $p_0 = \frac{app(w,s_i)}{|\mathcal{S}|}$ and $p_1 = s \cdot p_0$. The bursty score of t in period $p_{[b,e]}$ is the cost increment if $\mathcal{A}$ is in q_1 rather than q_0 as defined in Equation (3). He et al. [10] adopted this two-state finite automaton and used it to create the bursty feature representation described in Section 4.1.

$$Burst(w, p_{[b,e]}) = \sum_{i=w_1}^{t_2} (\sigma(0, app(w, s_i), |s_i|)\sigma(1, app(w, s_i), |s_i|)) \quad (3)$$

4.1.2 Discrepancy Model of Burstiness (Max − 1)

In [13], the bursty period of word w was detected using the discrepancy model and the maximal segment problem. First, the discrepancy model identifies the time step in which the document appears more frequently than on average. Specifically, for a given word w, the burstiness score of each time step is defined by Equation (4).

$$Burst_{Max1}(w,i) = \frac{app(w,s_i)}{app(w,S)} - \frac{1}{n} \tag{4}$$

where $app(w,S)$ is the number of documents in the whole sequence that contain w. If $Burst(w,i) > 0$, then w is bursty at time i. Next, the maximal segments of burstiness scores in the sequence of documents are recovered using the linear-time maximum sum algorithm by Ruzzo and Tompa [19]. The bursty period is defined as the recovered maximum sum period that is strictly positive. Finally, the burstiness score of each maximum sum period is defined as

$$Burst_{Max1}(w,p_{[b,e]}) = \frac{app(w,p_{[b,e]})}{app(w,S)} - \frac{|p_{[b,e]}|}{n}. \tag{5}$$

4.2 Bursty Distance Measurement

Although the bursty feature representation shows improvement when applied to classical VSM, there are two distinct weaknesses in the bursty feature representation. First, according to [10], the same improvement cannot be seen as clearly on VSM with TFIDF value. The cause of marginal difference on VSM with the TFIDF value is the fact that TFIDF value of a "rare" word has high TF and IDF scores. Based on how often words appear in the corpus, words are partitioned into three separate categories: common, average, and rare words. The common words are words that appear commonly throughout the corpus such as "the" and "do". These common words are often considered irrelevant and dropped from the vector space. Rare words are words that appear in a small group of document such as "vampire" and "spaceship". Other words are considered as average words. Rare words are bursty simply because they appear in a small time span in the corpus. Noisy words are also part of rare words. By adding the bursty weight to the TFIDF value of the rare word, we also amplify the noise. On the other hand, average words appear in larger group of documents; so on average they will have lower IDF score. In order to put more emphasis on the bursty information, their TFIDF values have to be boosted. Since some rare features with low DF are already emphasized with high IDF value, the bursty feature effect on rare features on VSM with TFIDF value would not be as strong as the effect on VSM with binary value.

Second, the bursty models identify a period of word w as bursty if either there is an increase in volume of documents containing w or the burstiness score of w rises above the global average threshold. These definitions of bursty period will not work well in event detection because a small event may disappear as noise. For example, using SIGHT software, Baumes et al. [2] recovered that prior to the declaration of independence of Kosovo in 2008, there was a small group of bloggers discussing the issue. Although, the number of blogs was small, the discussion group was persistent up until the actual week of declaration of independence. If we used the mean as the threshold value to find the burstiness of "independence Kosovo", such a discussion

would never be considered due to high volumes of discussion of the event in the following weeks.

To address these issues, we introduce a Bursty Distance Measurement (BDM). For the first issue, given a word w, it is likely that the sharp increment in the number of occurrences of w at time b indicates the beginning of the event related to w. Also, the decay of the number of occurrences of w at time e indicates the ending of the event related to w. We make an assumption for BDM following these observations that if $w \in d_1$ and $w \in d_2$, w has a bursty period $[b, e]$, and both t_{d_1} and t_{d_2} are in $[b, e]$, then d_1 and d_2 are more similar to each other than when one document falls within the bursty period, while the other does not. For the latter case, BDM does not add the bursty weight of w to VSM value when distance between d_1 and d_2 is measured. For the second issue, we defined that a word w is bursty at t if the actual occurrence is higher than its probability density estimation. By looking at the probability density estimation, we allow detection of the small event such as those described above. Such an event would give a higher occurrence than the smooth one estimated by the kernel density estimation. BDM assigns the bursty score of a word w at time i as follows:

$$Burst_{kde}(w, i) = \frac{app(w, s_i)}{app(t, S)} - f_h(i) \tag{6}$$

where $f_h(i)$ is the kernel density estimated at time i. We used the Gaussian function as the kernel function for this work. h could be found following the fast bandwidth estimation in [18]. After finding the bursty score, we defined the burst period as the maximal segment of the burstiness similarly as Max1. Burstiness of word w on the bursty period $p_{[b,e]}$ is the average positive sum of $Burst_{kde}(w, i)$ for $i \in p_{[b,e]}$ as defined by Equation (7).

$$BBDM(w, p_{[b,e]}) = \frac{\sum_{j=b}^{e} \alpha_j \cdot Burst_{kde}(w, j)}{e - b + 1} \tag{7}$$

where $\alpha_j = 1$ if $Burst_{kde}(w, j) > 0$, otherwise it is zero. We define $B(t_1, t_2, w)$ as a bursty weight function where $B(t_1, t_2, w) = BBDM(w, p_{[b,e]})$ if times t_1 and t_2 are in the same maximal segment $p_{[b,e]}$. Otherwise, $B(t_1, t_2, w) = 0$.

The complete BDM algorithm is described below. The distance between two documents d_1 and d_2, with timestamps t_{d_1} and t_{d_2}, respectively, is defined using the following algorithm, where $DIST$ is the provided distance measurement between two vectors, ρ is the upper bound frequency for rare words. BDM uses this bound to control the noise amplification of rare words.

Bursty Distance Measurement
Require: d_1, d_2, β, ρ, and $DIST()$
Ensure: D_{d_1,d_2}
for $i = 1$ **to** m
  if $B(t_{d_1}, t_{d_2}, w) > 0$

$$\theta = \begin{cases} \beta^{-1} & \text{if } app(w_i) < \rho \\ 1 & \text{otherwise} \end{cases}$$

$v_i^1 = tfidf(w_i^1) + \theta \cdot B(t_{d_1}, t_{d_2}, w_i)$
$v_i^2 = tfidf(w_i^2) + \theta \cdot B(t_{d_1}, t_{d_2}, w_i)$
endif
endfor
$D_{d_1,d_2} = DIST(\{v_1^1, \ldots, v_n^1\}, \{v_1^2, \ldots, v_n^2\})$

5 EXPERIMENTS

We evaluated our model on three sets of experiments based on three groups of datasets: synthetic datasets, news article datasets and additional real-life datasets. We performed experiments on synthetic and news article datasets to illustrate that if there were a time-dependent storyline, the clusters recovered using the bursty distance measure would tend to cluster documents from the same storyline together. On additional real-life datasets, where time-dependent storyline was not apparent, our experiments show that our framework is more robust than other bursty feature frameworks.

For the experiments, we tested all three bursty periods and bursty score analysis methods discussed in Section 4. We used cosine distance as the provided distance function in the framework. In addition to the bursty distance measurement (BDM), we applied the cosine distance to the bursty feature representation for document clustering proposed in [10]. For the bursty feature representation, we used two bursty models:

1. Kleinberg's algorithm (KL), as used in [10],
2. discrepancy model (Max1), as proposed by [13].

We also used the normal VSM with TFIDF representation (TFIDF) for baseline comparison. We used three clustering methods, K-Medoids, hierarchical (UPGMA), and lower bound on similarity clustering (Star). In K-Medoids, the interpretation of its results was not sound as a center document did not necessarily reflect the time period in the cluster that it represented. Hence, we only used the K-Medoids in the synthetic dataset experiment to show that BDM was robust enough to be applied in K-Medoids. We implemented the K-Medoids algorithm. For UPGMA, we used the built-in code in Matlab. We chose Star algorithm for our experiment to evaluate BDM performance on clustering methods when true number of clusters was not known. The code for Star algorithm was acquired from clustering with lower bound algorithm package in [8]. The F-score was used to evaluate the clusters.

Since the input of Star algorithm required the similarity score, we calculated the similarity score by subtracting the cosine distance score from 1. Clustering using lower bound on similarity does not always provide the target number of clusters. We applied the following steps to find the clustering with the right number of clusters, k:

1. Find the similarity threshold that gives the smallest upper bound k_{tau} to the target number of clusters. (For these experiments, the precision of the threshold is on the order of 10^{-3}.)

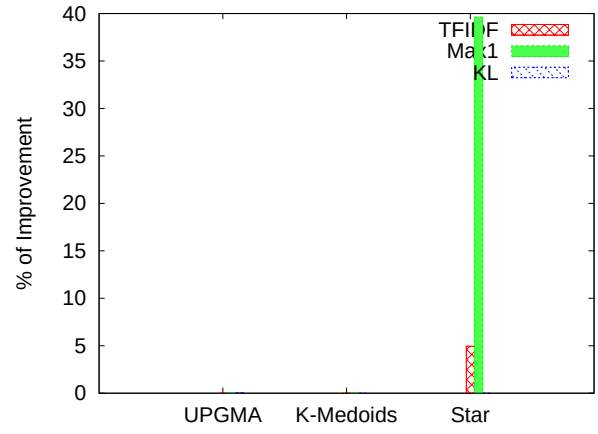
2. If $k_{tau} = k$, terminate, else find the smallest cluster and merge it with its closest cluster using average similarity.

3. Repeat step 2–3 until the number of remaining clusters is reduced to k.

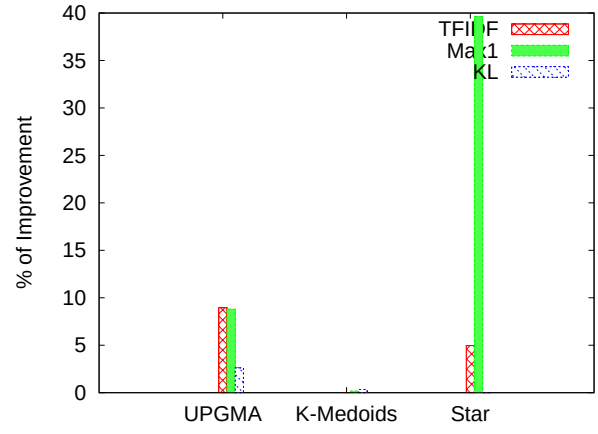
5.1 Synthetic Dataset

In the first experiment, four synthetic datasets were generated using the generative probabilistic model described in [21]. The first two synthetic datasets, Syn-1 and Syn-2, were generated using the generative models where each document contained two time dependent topics: the major topic and the minor topic. By time dependent topics we imply that if a document contains the major topic, then it can only contain a unique set of minor topics. Each major topic has five unique minor topics. Each topic has its active time span such that the topic can only appear if the document is generated in its active time span. Each topic is represented by 5 distinct keywords. There are a total of 20 major topics and 100 minor topics. Each document is 40 words long. For Syn-1, 20 words were drawn from the topic keywords (10 words each for the major and minor topic keywords). For Syn-2, 6 words were drawn from the topic keywords (3 words per each topic). The remaining words were randomly drawn from the vocabulary of 40 000 words. Syn-3 was generated using the same model as Syn-2, but each topic was time independent. Syn-4 was the combination of Syn-3 and Syn-2, where each document contained two time dependent topics and two time independent topics. Each document in Syn-4 is 40 words long with 3 words drawn from each of the topics in the document; the rest of the words were randomly drawn from the vocabulary of 2 000 words. Each data set contained 10 000 documents. The F-score was used to evaluate the clusters.

The results are shown in Figures 2 to 5. On time dependent datasets, Syn-1, Syn-2 and Syn-4, all three burstiness frameworks for clustering showed improvement over TFIDF for most of the clustering. Such results indicate that bursty information generally helps in documents clustering on text streams with time dependent topics. While Max1 performed worse than TFIDF on Star clustering, and KL performed worse than TFIDF on K-Medoids, BDM showed remarkable robustness on all three clustering methodologies as it had better F-scores than those of the baseline algorithm TFIDF. On the time independent dataset, Syn-3, KL appeared to boost the burstiness of random words more than other methods. On the other hand, since BDM did not increase the similarity score unless two documents were within the same bursty period of a word, it was able to avoid boosting noisy features.



a)

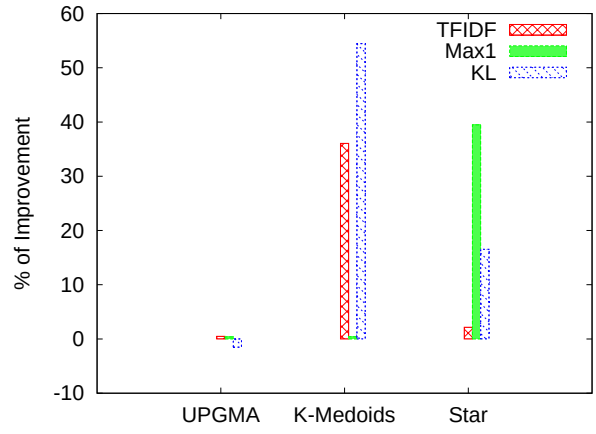


b)

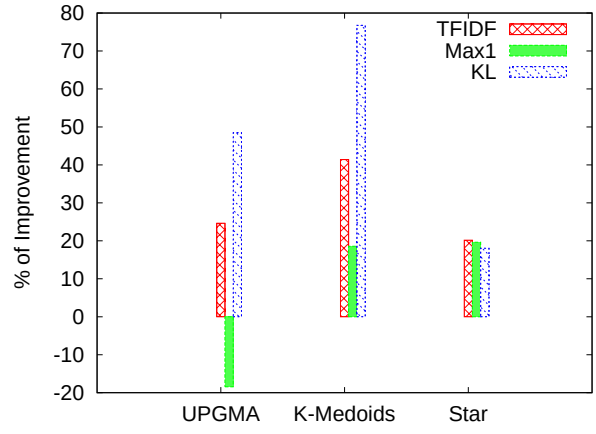
Fig. 2. Improvement of BDM over other methodologies on Syn-1 dataset

5.2 News Article Data Sets

In this set of experiments, our framework was applied to two datasets, namely Entertainment and Politics. Both datasets are news headlines collected by [2] from the following RSS feeds: `www.cnn.com/services/rss/` in 2009, `news.yahoo.com` between 2008 and 2009, and `today.reuters.com` between 2007 and 2008. The data was collected by simply checking the feed for new messages every hour. For entertainment news, we extracted a total of 9 087 news headlines from 03/01/2009 to 08/31/2009. For politics news, we extracted a total of 15 585 headlines from



a)

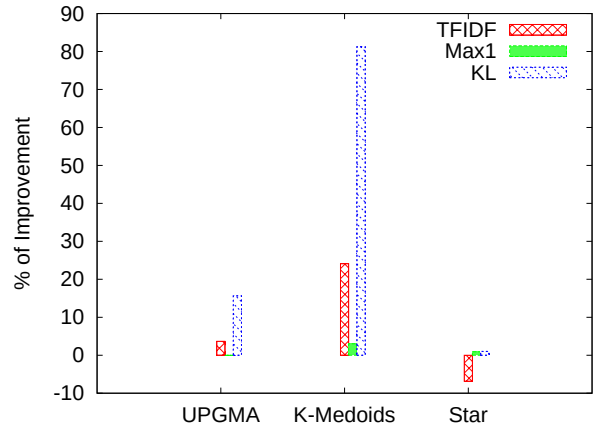


b)

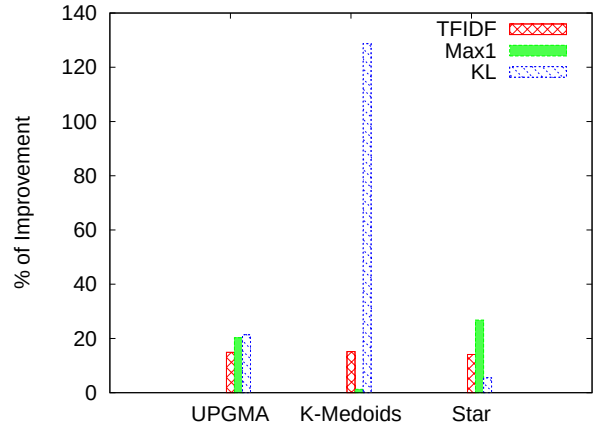
Fig. 3. Improvement of BDM over other methodologies on Syn-2 dataset

10/01/2008 to 12/31/2008. Although the sources of the news were different, we considered them as one single sequence of documents.

We tagged part of the documents related to four particular events, namely "Obama won 2008 presidential election" (Obama), "Clinton, H. is named the secretary of state" (Clinton), "the death of Natasha Richardson" (NR), and "the death of Michael Jackson" (MJ). We used the Star algorithm to cluster documents into roughly 1 700 clusters for entertainment news and roughly 4 000 clusters for political news. Our goal was to cluster the data so that each cluster had about 5 to 6 news articles on average. A good clustering should be able to put the similar news



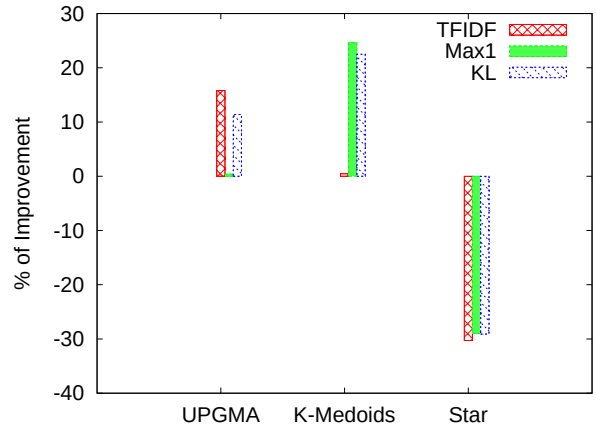
a)



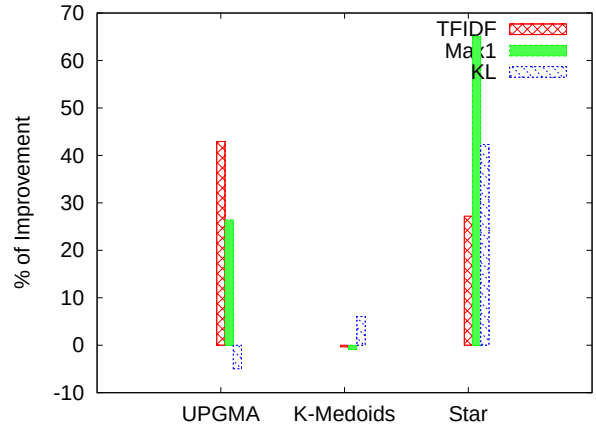
b)

Fig. 4. Improvement of BDM over other methodologies on Syn-3 dataset

together, while filtering out the non-related news. Then, the pair-tagged-document counts for each event were collected as follows. Given that documents A and B are tagged as Obama, the pair will produce the count of 1, if A and B appear in the same cluster. Otherwise, no count will be recorded. The recall was calculated from these counts as the percentage of recovered pair-tagged-document. We collected the number of clusters containing the documents, and used it to compute the precision as the number of tagged pairing over the number of total pairing recovered from the clusters containing tagged documents. Note that, if there exists one large cluster with every document, this setting will get the highest recall score. However, the



a)



b)

Fig. 5. Improvement of BDM over other methodologies on Syn-4 dataset

precision for such a clustering will be low. For a good clustering method, we expect the recall and precision to be high, while the number of clusters containing tagged documents is expected to be low. F-score was calculated based on these precision and recall.

The results of the experiment are shown in Figures 6 and 7. Figure 6 shows the number of clusters containing tagged document for each method on all four events. On three out of four events, BDM achieved the lowest number of clusters containing tagged document, and in the other case, it was the second lowest. This result showed that BDM was able to cluster documents from the same event into

the same cluster better than other method. Figure 7 shows the improvement of BDM over other model on Star clustering. Figure 7 shows that BDM had higher F-score than all other methods. Such results indicate that BDM was able to filter out documents that did not belong to the same event, from the clusters better than other methods.

TFIDF performed better than Max1 and KL on NR and MJ. Recalling the example in the introduction section, since bursty feature representations (Max1 and KL) consider all bursty periods as related to certain degrees, they cannot differentiate the deaths of two famous people from one another. Hence, Max1 and KL clustered related reports together instead of focusing on specific events. In particular, Max1 performed the poorest as it clustered the news articles on the wake for both Natasha Richardson and Michael Jackson together. In Obama and Clinton events, BDM had much higher F-score than other models because the other methods usually had one or two large clusters containing many documents with the words "obama" and "barack", and "hillary" and "clinton", respectively.

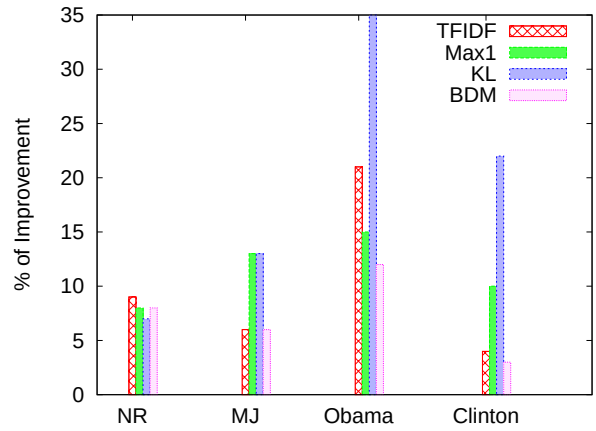
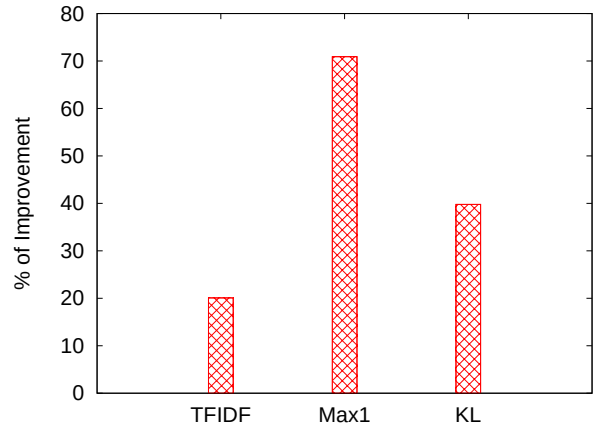


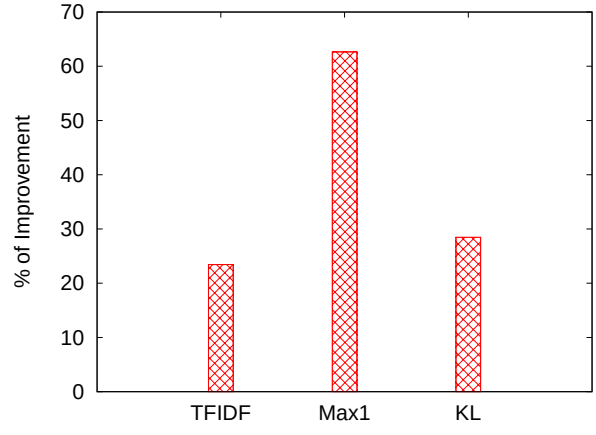
Fig. 6. Number of clusters containing tagged documents in each event

5.3 Additional Real-life Datasets

We used four other real-life datasets: 20 newsgroup, Reuters-21578, Nature and ACM datasets in the following experiments. The purpose of these experiments is to show that BDM is a robust framework that can improve clustering results on the same data even though it may have different feature space. For each dataset, we used the rainbow package [16] to prepare two indexes for the experiments. We generated two sets of indexing based on two feature selection quantifications: information gain and frequency. For information gain, we used the rainbow package to find 100 highest words in terms of information gain statistics. For frequency, we used the



a)

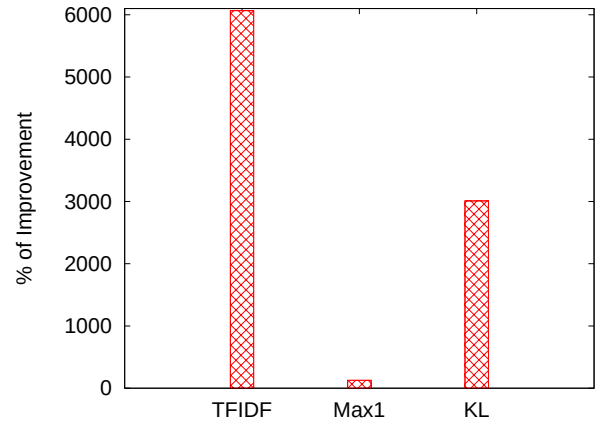


b)

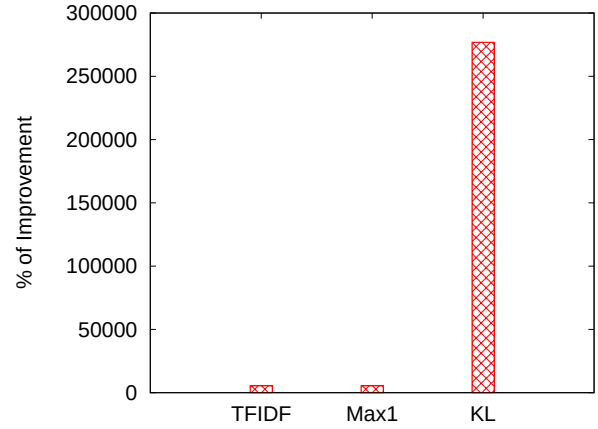
rainbow package to create the index of all words except the top and bottom 10 %. We ignored any document that did not contain any of the features.

20 newsgroup dataset: The dataset is available on the UCI Machine Learning Repository (see [5]). The dataset is a popular test bed for text clustering experiments. We automatically removed the headers and the reply/forward portions from each document. For multiple label documents, we selected the first listed label as the actual label. The results of the experiment on 20 newsgroup dataset are shown in Figure 8 a).

Reuters-21578 dataset: The Reuters-21578 dataset is made available on the UCI Machine Learning Repository (see [5]). We extracted the body from each docu-



c)

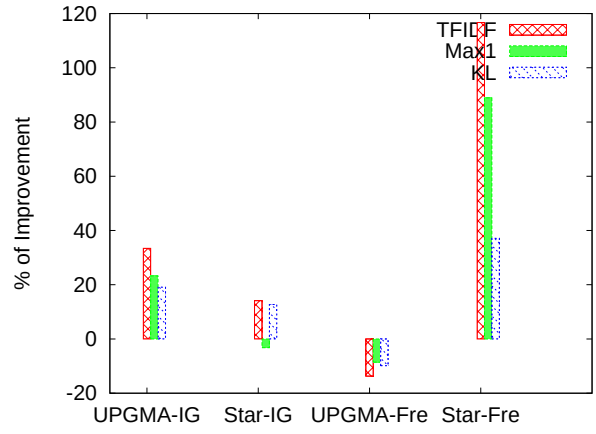


d)

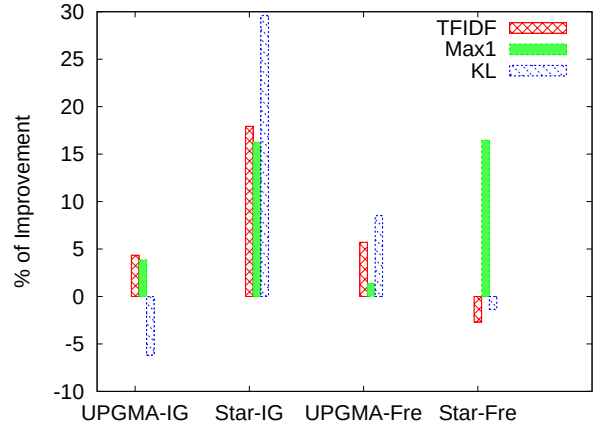
Fig. 7. Improvement of BDM over other methodologies on RSS feed dataset

ment and labeled each document with the first listed topic. We dropped any documents that did not contain a topic. We also dropped any topic that contained less than 5 documents. The results of the experiment on the Reuters-21578 dataset are shown in Figure 8 b).

Nature dataset: Nature dataset is the collection of Nature articles, published weekly and organized into five distinct classes. In this collection, there are 7 964 journal articles published between 01/01/2005 and 12/21/2006. The final dataset contains 7 437 journal articles. The results of the experiments with cosine distance are shown in Figure 8 c).



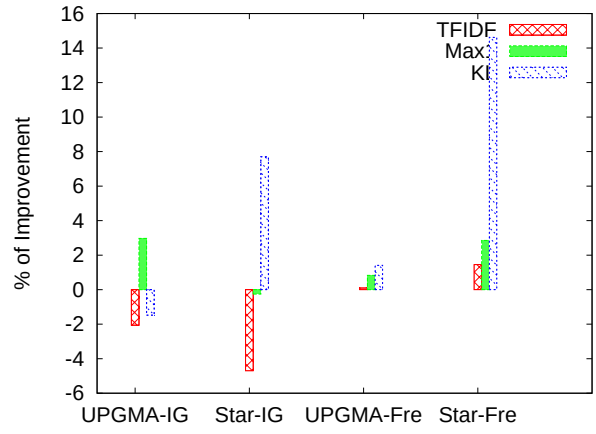
a)



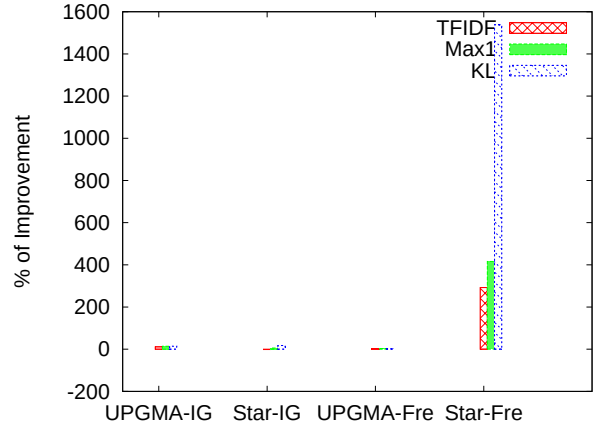
b)

ACM dataset: ACM dataset contains 24 897 Computer Science articles from the ACM digital library divided into 11 classes, published yearly from 1980 to 2001. The results are shown in Figures 8 d) and 9. As the y-scale for the Figure 8 d) was ill-proportioned, we generated Figure 9 to highlight the results of only UPGMA-IG, Star-IG and UPGMA-Fre on this dataset.

The results in Figure 8 indicated that on a one-on-one comparison BDM performed better than KL, Max1 and TFIDF. Again, BDM showed remarkable robustness as it was able to improve clustering on two types of features. Moreover, on 20 Newsgroup and Reuters-21578 datasets, where the text streams were likely to contain time dependent topics, frameworks that used bursty information for document clustering out performed TFIDF in general. Similar to Syn-3 dataset, KL and



c)



d)

Fig. 8. The improvement of BDM over other methodologies on 20 Newsgroup, Reuters-21578, Nature and ACM datasets

Max1 performed worse than BDM and TFIDF on Nature and ACM datasets, since these two datasets were not likely to have as much time dependent topics as the other two datasets.

6 CONCLUSION

In this paper, we introduced the bursty distance measurement framework that uses both the bursty period and burstiness score of each feature as the weight for dis-

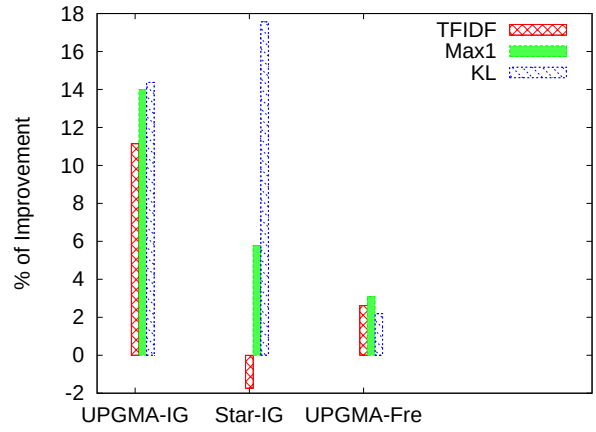


Fig. 9. The improvement of BDM over other methodologies on ACM dataset for UPGMA-IG, Star-IG and UPGMA-Fre

tance calculation between two documents. We showed that our framework excelled in clustering documents related to the same event both on Synthetic datasets and RSS feed datasets. It performed better than simply using VSM with existing bursty feature representation in such tasks. Our framework is a robust framework that can be applied to various clustering methodologies. We introduced the local burstiness score based on the local word occurrences. The local burstiness score leads to better event-related clustering on RSS feed dataset. However, it performs worse than the general bursty period detection methods on other datasets. These results suggest that our framework is likely to work well on event-related clustering on streaming text such as RSS feed, and Blogs. Also, we found that K-Medoids clustering algorithm is not performing well on clustering with bursty information in general.

Acknowledgments

We would like to thank Dr. Konstantin Mertsalov for making RSS feed datasets available to us and Dr. Meira for providing us with the Nature and ACM datasets.

REFERENCES

[1] Basseville, M.—Nikiforov, I. V.: Detection of Abrupt Changes – Theory and Application. Prentice-Hall Inc., Englewood Cliffs, NJ 1993.

[2] Baumes, J.—Goldberg, M.—Hayvanovych, M.—Kelley, S.—Magdon-Ismail, M.—Mertsalov, K.—Wallace, W.: Sights: A Software System for Finding Coalitions and Leaders in a Social Network. In IEEE International Conference on Intelligence and Security Informatics, New Brunswick, NJ 2007, pp. 193–199.

[3] Church, K. W.—Gale, W. A.: Poisson Mixtures. Natural Language Engineering, Vol. 1, 1995, No. 2, pp. 163–190.

[4] Curry, C.—Grossman, R.—Locke, D.—Vejcik, S.—Bugajski, J.: Detecting Changes in Large Data Sets of Payment Card Data: A Case Study. In ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 2007, pp. 1018–1022.

[5] Frank, A.—Asuncion, A.: UCI Repository. `http://archive.ics.uci.edu/ml`, 2010.

[6] Fung, G. P. C.—Yu, P. S.—Lu, H.: Parameter Free Bursty Events Detection in Text Streams. In Proceedings of the 31st International Conference on Very Large Data Bases, VLDB Endowment, 2005, pp. 181–192.

[7] Google. Google trends, `http://www.google.com/trends`, 2010.

[8] Hasan, M. A.—Salem, S.—Pupacdi, B.—Zaki, M. J.: Clustering with Lower Bound on Similarity. In The 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Bangkok, Thailand 2009, pp. 122–133.

[9] He, Q.—Chang, K.—Lim, E. P.: Using Burstiness to Improve Clustering of Topics in News Streams. In ICDM 2007, pp. 493–498.

[10] He, Q.—Chang, K.—Lim, E. P.—Zhang, J.: Bursty Feature Representation for Clustering Text Streams. In SIAM Conference on Data Mining, 2007, pp. 26–28.

[11] Kleinberg, J.: Temporal Dynamics of On-Line Information Streams. In In Data Stream Management: Processing HIGH-SPEED Data, Springer 2006.

[12] Kuta, M.—Kitowski, J.: Clustering Polish Texts with Latent Semantic Analysis. Lecture Notes in Computer Science, 6114, pp. 532–539, 2010.

[13] Lappas, T.—Arai, B.—Platakis, M.—Kotsakos, D.—Gunopulos, D.: On Burstiness-Aware Search for Document Sequences. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2009, pp. 477–486.

[14] Leskovec, J.—Backstrom, L.—Kleinberg, J.: Meme-Tracking and the Dynamics of the News Cycle. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France 2009, pp. 297–506.

[15] Kunder, M. D.: World Wide Web Size. `http://www.worldwidewebsize.com/`, 2008.

[16] McCallum, A.: Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering. `http://www.cs.cmu.edu/~mccallum/bow`, 1998.

[17] Papka, R.: On-Line New Event Detection, Clustering, and Tracking. Doctoral Dissertation, University of Massachusetts, Amherst 1999.

[18] Raykar, V. C.—Duraiswami, R.: Fast Optimal Bandwidth Selection for Kernel Density Estimation. In: J. Ghosh, D. Lambert, D. Skillicorn, J. Srivastava (Eds.): Proc. of the Sixth SIAM Int. Conf. on Data Mining,2006, pp. 524–528.

[19] Ruzzo, W. L.—Tompa, M.: A Linear Time Algorithm for Finding All Maximal Scoring Subsequences. In International Conference on Intelligent Systems for Molecular Biology, AAAI Press 1999, pp. 234–241.

[20] Serrano, M. A.–Flammini, A.—Menczer, F.: Modeling Statistical Properties of Written Text. PLoS ONE, Vol. 4, 2009, No. 4, e5372.

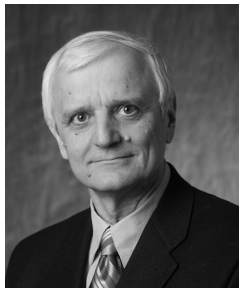
[21] Steyvers, M.—Griffiths, T.: Probabilistic Topic Models. In: T. Landauer, D. McNamara, S. Dennis, W. Kintsch (Eds.): Latent Semantic Analysis: A Road to Meaning. Lawrence Erlbaum Associates 2006.

[22] Vlachos, M.—Meek, C.—Vagena, Z.—Gunopulos, D.: Identifying Similarities, Periodicities and Bursts for Online Search Queries. In ACM SIGMOD International Conference on Management of Data, Paris (France) 2004, pp. 131–142.

[23] Wang, X.—McCallum, A.: Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mning, Philadelphia (USA) 2006, pp. 424–433.

[24] Wang, X.—Zhai, C.—Hu, X.—Sproat, R.: Mining Correlated Bursty Topic Patterns from Coordinated Text Streams. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2007, pp. 784–793.

Apirak Hoonlor received his Ph. D. degree in computer science from Rensselaer Polytechnic Institute, New York, USA in 2011. Since then, he has been a lecturer at the Faculty of Information and Communication Technology at Mahidol University, Thailand. He is also working as a researcher for Integrative Computational BioScience Center, Mahidol University. His research interests include artificial intelligence, data mining, text mining, outlier detection, and medical data mining, as well as applications in these areas.



Bolesław K. Szymanski received the Ph. D. degree in computer science from National Academy of Sciences in Warsaw, Poland, in 1976. He is the Claire and the Roland Schmitt distinguished professor of computer science and the director of the Social Cognitive Academic Research Center led by RPI. He is an author and coauthor of more than 300 publications and an editor of five books. He is also the Editor-in-Chief of Scientific Programming journal. His interests include parallel and distributed computing and networking. He is a fellow of the IEEE and a member of the ACM for which he was a national lecturer. In 2009, he was elected a foreign member of Polish Academy of Sciences.

Mohammed J. Zaki is a Professor of Computer Science at RPI. He received his Ph. D. degree in computer science from the University of Rochester in 1998. His research is focused on developing novel data mining techniques, especially in bioinformatics. He has over 200 publications. He is an Area Editor for Statistical Analysis and Data Mining, and an Associate Editor for Data Mining and Knowledge Discovery, ACM Transactions on Knowledge Discovery from Data, Knowledge and Information Systems, Social Networks and Mining, and International Journal of Knowledge Discovery in Bioinformatics. He was the program co-chair for SDM '08, SIGKDD '09, PAKDD '10, BIBM '11, CIKM '12, and ICDM '12. He is a senior member of the IEEE, and an ACM Distinguished Scientist.

Vineet Chaoji received a doctoral degree in computer science from Rensselaer Polytechnic Institute in 2009. His research interests span graph mining, social networks and clustering algorithms for large scale data. He has served on the program committees of many leading conferences such as KDD, CIKM, ICDM and SDM. He is currently working at Amazon as a Machine Learning Scientist. Prior to joining Amazon, he was a Scientist at Yahoo! Labs, Bangalore.