Steeler Nation, 12th Man, and Boo Birds: Classifying Twitter User Interests using Time Series

Tao Yang
The Pennsylvania State University
Email: tyang@ist.psu.edu

Dongwon Lee
The Pennsylvania State University
Email: dlee@ist.psu.edu

Su Yan
IBM Almaden Research Center
Email: syan@us.ibm.com

***Abstract*— The problem of Twitter *user classification* using the *contents* of tweets is studied. We generate time series from tweets by exploiting the latent temporal information and solve the classification problem in time series domain. Our approach is inspired by the fact that Twitter users sometimes exhibit the *periodicity* pattern when they share their activities or express their opinions. We apply our proposed methods to both *binary* and *multi-class* classification of sports and political interests of Twitter users and compare the performance against eight conventional classification methods using textual features. Experimental results using 2.56 million tweets show that our *best* binary and multi-class approaches improve the classification accuracy over the *best* baseline binary and multi-class approaches by 15% and 142%, respectively.**

I. Introduction

Twitter, one of the most popular microblog sites, has been used as a rich source of real-time information sharing in everyday life. When Twitter users express their opinions about organizations, companies, brands, or sports in tweets, it in turn provides important opportunities for businesses in improving their services such as targeted advertising and personalized services. Since the majority of Twitter users' basic demographic information (e.g., gender, age, ethnicity) is unknown or incomplete, being able to accurately identify the hidden information about users becomes an important and practical problem. As such, we study the problem of classifying Twitter users to a fixed set of categories based on the contents of their tweets. Formally, we define our research problem as:

Definition 1 *(Twitter User Classification) Given a set of Twitter users U, a stream of tweet messages $T_u = \{t_1, ..., t_{|T_u|}\}$ for each user $u \in U$, a pre-defined set of K class labels $C = \{c_1, ..., c_K\}$, and labeled samples such that $\langle u, c\rangle \in U \times C$, learn a classifier $\psi: U \rightarrow C$ to assign a class label to a unlabeled user.*□

Abundant relevant research on this problem exists (to be surveyed in Section II). However, the majority of existing solutions focused on using "textual" features of Twitter users (e.g., tweets messages) [1] or "network" features (e.g., follower/follwee network) [2] in classifying Twitter users. Despite their success, in this paper, we argue that modeling tweet features as *time series* to amplify its *periodicity* pattern can be more effective in solving certain types of Twitter user classification problems. Twitter users often exhibit a periodicity pattern when they post tweets to share their activities and statuses or express their opinions. This is because people tend to show interests in different activities during different time frames. For instance, Figure 1 (taken from [3]) shows that contents on microblogging platforms such as Twitter show patterns of temporal variation and there exists a recurring

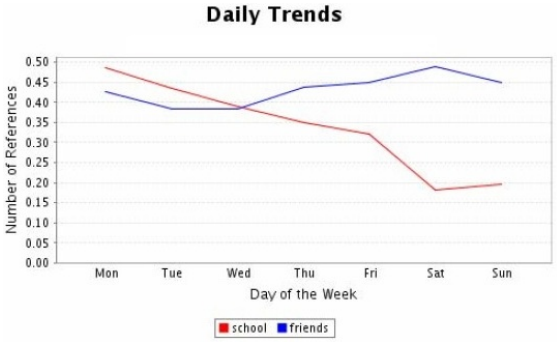


Fig. 1. Daily trends for the terms "friends" and "school," taken from [3].

pattern in word usage – i.e., the term "school" is more frequent during the early week and "friends" takes over during the weekend. Such patterns may be observed over a day or a week. As a result, instead of using tweet messages directly, one may leverage the *temporal* information derived from the word usage within tweet streams to boost the accuracy in classification. Therefore, in classifying Twitter users, we advocate to convert tweet contents into time series based on word usage patterns, and then perform time series based classification algorithms. The efficacy of our proposed approach is validated through extensive experiments in both sports and political interest domains.

Our contributions are as follows: (1) To the best of our knowledge, this is the first attempt to solve the Twitter *user classification* problem in time series domain. We formally propose our framework to map users to time series for classification; (2) We formulate the problem of user classification as a document categorization problem in the Twitter setting, and show the procedure of feature selection as well as the detailed evaluation of different classifiers; and (3) We validate our idea in both binary and multi-class Twitter user classification settings and successfully demonstrate that our proposal substantially outperforms eight competing methods in identifying Twitter users with certain sports and political interests.

II. Related Work

Recently, researchers tried to tackle the problem of short text classification from different perspectives [4], [5], [6]. [4] used a small set of domain-specific features extracted from the user's profile and text to effectively classify the text to a predefined set of generic classes such as news, events, opinions, deals and private messages. [5] tried to improve the classification accuracy by gaining external knowledge discov-

ered from search snippets on Web search results. [6] proposed a non-parametric approach for short text classification in an information retrieval framework. And the predicted category label is determined by the majority vote of the search results for better classification accuracy. [7] proposed a classification model of tweet streams that switches between two probability estimates of words, which can learn from stationary words and also respond to busty words. Note that these classification methods are carried over *tweets*. In contrast, in this paper, we focus on the problem of classification over *users*.

Several researchers have investigated the problem of detecting user attributes such as gender, age, regional origin, political orientation, sentiment, location, and spammer based on user communication streams. [8] investigated statistical models for identifying the gender of Twitter users as the binary classification problem. They adopted a number of text-based features through various basic classifier types. [2] presented a study of classification experiments about more latent user attributes such as gender, age, regional origin, and political orientation. They adopted various sociolinguistic features such as emoticons, ellipses, character repetition, etc., and used support vector machines to learn a binary classifier. Although the authors gave a general framework with classification techniques for various user attributes mining tasks, they employed a lot of domain knowledge in their experiments. [9] focused on classification problem on positive or negative feelings on tweet streams for opinion mining and sentiment analysis. [10], [11] studied user geo-location detection problem in the city level. Based purely on the contents of the user's tweets, the authors proposed a probabilistic framework to automatically identify words in tweets with a local geo-scope for estimating a Twitter user's city-level location. [12] further improved the prediction quality of a Twitter user's home location by estimating the spatial word usage probability with Gaussian Mixture models. Meanwhile, they also proposed unsupervised measurements to rank the local words to remove noises effectively. [13] used a number of characteristics features related to user social behavior as attributes of machine learning process for classifying users as either spammers or non-spammers on Twitter.

[14] proposed a temporal semantic analysis model to compute the degree of semantic relatedness of words by studying patterns of word usage over time. [15] proposed a time-aware clustering algorithm to uncover the common temporal patterns of online content popularity. [16] is closely related to ours. The authors developed a general machine learning approach to learn three binary classification models based on Decision Trees for identifying political affiliation, ethnicity, and business affinity from labeled data using a broad set of features such as profile, tweeting behavior, linguistic content and social network features. However, user profile information is typically missing or incomplete, and thus not a useful source for features [2]. Different from their work, therefore, we focus on classifying Twitter users based on the time series generated from the contents of tweet messages. When users' periodic pattern plays an important role (as in detecting sports fans), our method becomes more useful. In general, however, our work should be viewed as complementary to [16].

III. User Classification using Textual Features

We first present the baseline classification approach for classifying users based on the *textual features* extracted from tweets. Given a stream of tweets, we represent each user as a document with a bag of words and directly extract features from the document content.

A. Feature Selection

We select two types of features based on tweet contents: TF-IDF and Topic Vector generated from Latent Dirichlet Allocation (LDA).

TF-IDF. Term Frequency – Inverse Document Frequency (TF-IDF) is a classical term weighting method used in information retrieval. The idea is to find the important terms for the document within a corpus by assessing how often the word occurs within a document (TF) and how often in other documents (IDF). In our Twitter user setting, we have: $TF-IDF(t,u) = -\log \frac{df(t)}{U} \times tf(t,u)$, where $tf(t;u)$ is the term frequency of word t within the stream of tweets of user u, $df(t)$ is the document frequency within the corpus (i.e., how many users' tweets contain at least one instance of t), and U is the number of users in the corpus.

Topic Vector. The Latent Dirichlet Allocation (LDA) proposed by [17] models documents by assuming that a document is composed by a mixture of hidden topics and that each topic is characterized by a probability distribution over words. This model provides a more compressed format to represent documents. In Twitter user classification, we adapt the original LDA by replacing documents with users' tweet streams. While LDA represents documents as bags of words, we represent Twitter users as words of their tweets. Therefore, a Twitter user is represented as a multinomial distribution over hidden topics. Given a number U of Twitter users and a number T of topics, each user u is represented by a multinomial distribution θ_u over topics, which is drawn from a Dirichlet prior with parameter α. A topic is represented by a multinomial distribution ϕ_t, which is drawn from another Dirichlet prior with parameter β. The topic vector acts as a low-dimensional feature representation of users' tweet streams and can be used as input into any classification algorithm. In other words, for each user u, we can use LDA to learn θ_u for that user and then treat θ as the features in order to do classification. The next step is to correctly assign a class label to each user in the reduced dimensional space.

B. Classification Methods

We select two popular classifiers over text domain: Naive Bayes (NB) and Support Vector Machines (SVM).

Naive Bayes. The Naive Bayes is a simple model which works well on text categorization [18], and it is a successful classifier based on the principle of Maximum A Posteriori (MAP). In this paper, we adopt a multinomial Naive Bayes model. Given the user classification problem having K classes $\{c_1, c_2 ...,c_K\}$ with prior probabilities $P(c_1)$,...,$P(c_K)$, we assign a class label c to a Twitter user u with feature vector $\mathbf{f} = (f_1, f_2..., f_N)$, such that $c = \arg\max_c P(c = c_k|f_1, f_2..., f_N)$. That is to assign the class with the maximum a posterior probability given the observed data. This posterior probability can be formulated using Bayes theorem as follows:

$$P(c = c_k|f_1, f_2..., f_N) = \frac{P(c_k) \times \prod_{i=1}^{N} P(f_i|c_k)}{P(f_1, f_2..., f_N)}$$

where the objective is to assign a given user u having a feature vector $\mathbf{f}$ consisting of N features to the most probable

class. $P(f_i|c_k)$ denotes the conditional probability of feature f_i found in tweet streams of user u given the class label c_k. Typically the denominator $P(f_1, f_2..., f_N)$ is not computed explicitly as it remains constant for all c_k. $P(c_k)$ and $P(f_i|c_k)$ are obtained through maximum likelihood estimates (MLE).

Support Vector Machines. The Support Vector Machines is another popular classification technique [19]. While Naive Bayes is a generative classifier to form a statistical model for each class, SVM is a large-margin classifier. The basic idea of applying SVM on classification is to find the maximum-margin hyperplane to separate among classes in the feature space. Given a corpus of U Twitter users and class labels for training $\{(\mathbf{f}_u, c_u)|u = 1, ..., U\}$, where $\mathbf{f}_u$ is the feature vector of user u and c_u is the target class label, SVM maps these input feature vectors into a high dimensional reproducing kernel Hilbert space, where a linear machine is constructed by minimizing a regularized functional. The linear machine takes the form of $\varphi(\mathbf{f}) = \langle \mathbf{w} \cdot \phi(\mathbf{f}) \rangle + b$ where $\phi(\cdot)$ is the mapping function, b is the bias and the dot product $\langle \phi(\mathbf{f}) \cdot \phi(\mathbf{f}') \rangle$ is also the kernel $K(\mathbf{f}, \mathbf{f}')$. The regularized functional is defined as:

$$R(\mathbf{w}, b) = C \cdot \sum_{u=1}^{U} \ell(c_u, \varphi(\mathbf{f}_u)) + \frac{1}{2}\|\mathbf{w}\|^2$$

where the regularization parameter $C > 0$, the norm of $\mathbf{w}$ is the stabilizer and $\sum_{u=1}^{U} \ell(c_u, \varphi(\mathbf{f}_u))$ is empirical loss term.

IV. User Classification using Time Series

In this section, we introduce our novel time series approach to tackle the problem of Twitter user classification. In particular, we propose a new technique for feature selection in order to convert Twitter users to time series by incorporating temporal information into the stream of tweets. We also propose two classification algorithms such that the multi-class Twitter user classification problem can be solved effectively in the time series domain.

A. Feature Selection

In this subsection, we explore the impact of temporal information in classifying Twitter users. Our assumption is that Twitter users often exhibit *periodicity* patterns when they post tweets to share their activities and statuses or express their opinions. This is because people from various categories tend to do different activities during different time frames. For example, sports fans usually post more relevant tweets about their favorite teams or players on game days during the season instead of offseason. Female shoppers love to share more of their opinions on Twitter during weekends or holidays. Travel enthusiasts tend to share more about their journey during summer time. [3] has shown that users participate in online social communities which share similar interests and there are recurring daily or weekly patterns in word usages. Another recent study [15] has also indicated that contents on microblogging platforms such as Twitter show patterns of temporal variation and pieces of content become popular and fade away in different temporal scales. Thus, we aim at leveraging *temporal* information in generating features from contents of tweet streams for our classification task. Our feature extraction process consists of two stages as follows.

Given a set of Twitter users U and K class labels $C = \{c_1, c_2 ..., c_K\}$, first, we identify the *category-specific* keywords as a good source of relevant information of the entity in each class. In particular, we can harvest this kind of category related keywords from some external knowledge sources such as Wikipedia, or more directly, we can make use of online dictionaries such as WordNet, i.e. a network of words, to find all the related terms linked to the category keywords. For example, different sports have different Wikipedia pages consisting of rich corpora of sports-specific keywords which can be utilized to identify positive topics generated by sports fans in tweet streams. This keyword extraction process can be done manually or automatically depending on the scope of classification tasks and applications. The dictionary of these predefined keyword features serves as a rich representation of the entity in each category and contributes towards positive evidence of each class.

Second, given a stream of tweet messages $T_u = \{t_1, t_2, ..., t_{|T_u|}\}$ for each Twitter user u, we divide these tweets into segments based on predefined sliding time windows, e.g., daily or weekly time frames. We then record the number of word occurrences of category-specific keywords that appear in all the tweet messages within each sliding window. Based on these numbers, we convert each user into a numerical time series by calculating the frequency or percentage of keywords occurrences at different granularity levels, e.g., word or tweet levels. These time series reflect temporal fluctuations with respect to frequency changes of positive mentions of keywords in tweet messages from users in each class.

Example 1. As an illustration, consider Figure 2 that shows examples of using football-specific keywords (details shown in section V-A) to generate daily and weekly time series of two followers and two non-followers of the NFL team, New York Giants, during the month of September 2011. Figures 2(a) and 2(d) show daily and weekly time series based on frequencies of football-specific words. On a daily basis, we treat a user's daily tweet streams as a bag of words and count the frequency of football-specific words that appear in the daily tweet messages. On a weekly basis, we treat a user's tweet streams on game days (i.e., Sunday and Monday) vs. non-game days (i.e., Tuesday through Saturday) separately. That is, we count the frequency of football-specific words that appear in tweet messages on game days vs. non-game days in each week. We can easily see that both daily and weekly time series of the followers preserve similar shapes in real-value domain (with some shifting) while the time series of the non-followers have rather different shapes. Figures 2(b) and 2(e) show daily and weekly time series based on frequencies of football-specific tweets. On a daily basis, we count the frequency of tweets containing football-specific words that appear in daily tweet messages. On a weekly basis, we count the frequency of tweets containing football-specific words that appear in the tweet messages on game days vs. non-game days separately. Figures 2(c) and 2(f) show daily and weekly time series based on the percentage of football-specific tweets (i.e., fraction of the number of tweets containing football-specific keywords over the number of tweet messages within each time frame). Note that regardless of particular feature extraction methods to generate time series, there is a clear difference between time series of football followers and non-followers. □

Each time series serves as a feature vector of the corresponding Twitter user for further classification in the domain of numerical signals. The detailed feature extraction process is shown in Algorithm 1.

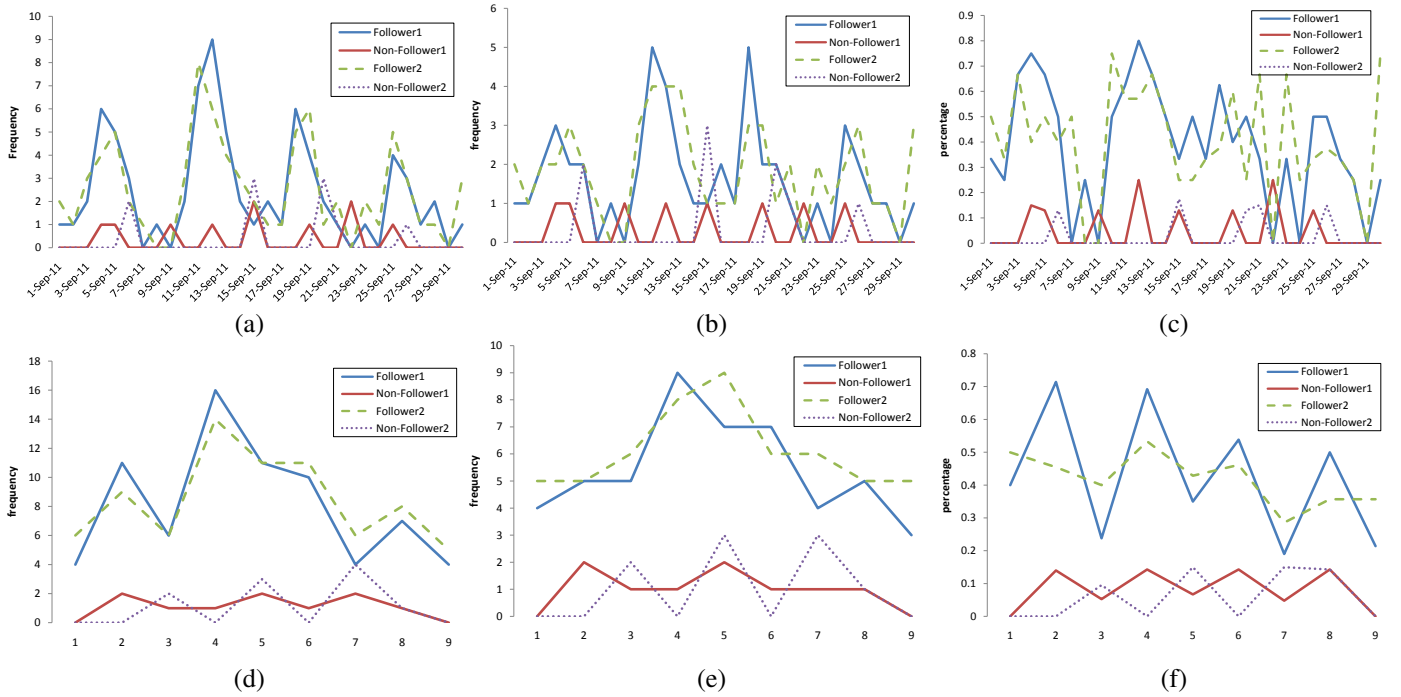


Fig. 2. (a)-(c): daily time series based on the *frequency* of relevant *words*, *frequency* of relevant *tweets*, and *percentage* of relevant *tweets*; (d)-(f): weekly time series based on the *frequency* of relevant *words*, *frequency* of relevant *tweets*, and *percentage* of relevant *tweets*.

Algorithm 1: **Time Series Feature Extraction.**

Input : A set of Twitter users U and a stream of tweet messages T_u = $\{t_1, t_2, ...,t_{|T_u|}\}$ for each user u with class label c from a predefined set of K class labels C = $\{c_1, c_2 ...,c_K\}$, a new user v and a stream of tweet messages T_v

Output: A set of time series TS = $\{TS_1, TS_2, ...,TS_{|U|}\}$where TS_u is a converted time series feature vector for each user u

```
/*stage1: preprocessing*/;
for class in classList do
    build category-specific keyword lists
    list[class] = preProcess(class);
endfor
/*stage2: transformation*/;
for class in classList do
    for u in userList[class] do
        break T_u into smaller segments S;
        for s in S do
            count the number of occurrences w_s of keywords from
            list[class] in s ;
        endfor
        convert user u into a time series TS_u from w; return(TS_u);
    endfor
endfor
```

B. Classification Methods

In time series classification, using feature-based methods as in section 3 is a challenging task because it is not easy to do feature enumeration on numerical time series data. Therefore, we use the common *distance-based* approach to classify time series. Previous research has shown that compared to commonly used classifiers such as SVM, k-nearest neighbor (kNN) classifier (especially 1NN) with dynamic time warping distance is usually superior in terms of classification accuracy [20].

kNN. The kNN is one of the simplest non-parametric classification algorithms, which does not need to pre-compute a classification model [21]. Given a labeled Twitter user set U, a positive integer k, and a new user u to be classified, the kNN classifier finds the k nearest neighbors of u in U, $kNN(u)$, and then returns the dominating class label in $kNN(u)$ as the label of user u. In particular, if k = 1, the 1NN classifier will return the class label of the nearest neighbor of user u in terms of distance in time series feature space.

C. Distance Functions

We select two types of distance functions for user classification in time series domain: Dynamic Time Warping (DTW) and Symbolic Aggregate approXimation (SAX).

DTW. The Dynamic Time Warping (DTW) is a well-known technique to find an optimal alignment between two time series [22]. Intuitively, the time series are warped in a nonlinear fashion to match each other. The idea of DTW is to align two time series in order to get the best distance by aligning. In data mining and information retrieval research, DTW has been successfully applied to automatically deal with time-dependent data. Given two Twitter Users' time series feature vectors $\mathbf{X} = (x_1, x_2..., x_{|X|})$ and $\mathbf{Y} = (y_1, y_2..., y_{|Y|})$, DTW is to construct a warping path $\mathbf{W} = (w_1, w_2..., w_K)$ with $\max(|X|, |Y|) \leq K < |X| + |Y|$ where K is the length of the warping path and w_k is the k^{th} element $(i, j)_k$ of the warping path. The optimal warping path is the path which minimizes the warping cost: $DTW(X, Y) = \min\{\sum_{k=1}^{K} d(w_k)\}$. The optimal path can be found very efficiently using dynamic programming to evaluate the following recurrence: $\gamma(i, j) = d(i, j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\}$, where $\gamma(i, j)$ denotes the cumulative distance as the distance $d(i, j)$ found in the current cell and the minimum of the cumulative distances of the adjacent elements.

SAX. The Symbolic Aggregate approXimation (SAX) is

Algorithm 2: **One-Vs-All User Classification**.

Input : A set of Twitter users U and a stream of tweet messages T_u = $\{t_1, t_2, ..., t_{|T_u|}\}$ for each user u with class label c from a predefined set of K class labels C = $\{c_1, c_2 ..., c_K\}$, a new user v and a stream of tweet messages T_v

Output: The class label for user v

for *class in classList* **do**
  for u *in userList* **do**
    TS_u = FeatureExtraction(u);
  endfor
  TS_v = FeatureExtraction(v);
  /*classification*/;
  learn a kNN classifier on time series TS_v and TS_u where $u \in U$;
endfor
/*pairwise comparison*/;
for *class in classList* **do**
  find the $class$ with the best kNN classifier ;
endfor
return(class);

known to provide good dimension reduction and indexing with a lower-bounding distance measure [23]. In many data mining applications, SAX has been reported to be as good as well-known representations such as Discrete Wavelet Transform (DWT) and Discrete Fourier Transform (DFT). However, SAX requires less storage space. In this paper, we adopt the same SAX technique in [23] for classifying Twitter users in time series domain.

D. Multi-class User Classification

We present two classification variations in time series domain for multi-class Twitter user classification.

One-Vs-All. The first approach is to reduce the problem of classifying among K classes into K binary problems and each problem discriminates a given class from the other $K-1$ classes. In this approach, we build K binary classifiers where the k^{th} classifier is trained with positive examples belonging to class k and negative examples belonging to the other $K-1$ classes. For the k^{th} binary classifier, we convert all users into time series using the category-specific keyword list of the k^{th} class. When classifying a new user v, the classifier with the nearest neighbor of the user is considered the winner, and the corresponding class label is assigned to the user v. The detailed classification algorithm is shown in Algorithm 2.

All-At-Once. The second approach is to convert Twitter users of each class c into time series simultaneously using the category-specific keyword list of the corresponding class. Given a new user v, we convert v using the combination of keyword lists of all classes. When classifying the new user v, the classifier returns the label of the nearest neighbor as the corresponding class label to be assigned to the user v. The detailed classification algorithm is shown in Algorithm 3.

V. Experiments on Sports Interests

In order to validate our classification approaches, we first apply them to both binary and multi-class classification problems with respect to identifying NFL football fans and team fans. Specifically, our experimental questions are the following: (1) *Binary*: How accurately can we predict if a Twitter user is a football fan or not? (2) *Multi-class*: How accurately can we predict the football team (1 out of 32 teams) of a Twitter user when she is known as a football fan?

Algorithm 3: **All-At-Once User Classification**.

Input : A set of Twitter users U and a stream of tweet messages T_u = $\{t_1, t_2, ..., t_{|T_u|}\}$ for each user u with class label c from a predefined set of K class labels C = $\{c_1, c_2 ..., c_K\}$, a new user v and a stream of tweet messages T_v

Output: The class label for user v

for *class in classList* **do**
  for u *in userList[class]* **do**
    TS_u = FeatureExtraction(u);
  endfor
  $listAll$ += $list[class]$
endfor
convert user v into a time series TS_v using $listAll$;
/*classification*/;
learn a kNN classifier to find the best $class$ on time series TS_v and TS_u where $u \in U$;
return(class);

A. Set-Up

Data Collection. We focused on the football season from Sep. 2011 to Dec. 2011 in the experiments. Starting from the 32 official Twitter accounts of NFL football teams, we first identified 1,000 followers per team (i.e., a total of 32,000 users) as the "fan" corpus. Similarly, we also identified a total of 32,000 users who do not follow any Twitter account of the football teams as the "non-fan" corpus. Each user has at least 3-4 tweets per day (i.e., about 400 tweets for 4-month period). For each tweet, we removed the external links, non-alphabetic characters such as "@" and "#", emoticons and stop words, and then filtered out tweets with less than five words. At the end, our data set included a total of 64,000 users and 2.56 million tweets. From the Wikipedia page of each of the 32 NFL teams, next, we automatically harvested the team and player names from the roster section, and manually identified football-specific keywords such as "nfl" and "quarterback" as well as team-specific keywords. This semi-automatic generation process of category-specific keywords resulted in a total of 2,330 unique terms at the end in our dictionary. Team-specific keywords serve as category-specific keywords for multi-class classification purpose while the combination of team-specific and football-specific keywords serve as category-specific keywords for binary classification purpose.

Evaluation Metrics. The binary classification task is to classify the users into two classes, i.e., one class which represents the users who are fans of NFL football (positive class) and the other class which represents user that are not fans of NFL football (negative class). Moreover, the multi-class classification task is to classify the users into 32 classes with each representing the fans of each individual NFL team. For evaluation purpose, all the users can be grouped into four categories, i.e., true positive (*TP*), true negatives (*TN*), false positives (*FP*) and false negatives (*FN*). For example, the true positives are the users that belong to positive class and are in fact classified to the positive class, and the false positives are the users not belonging to positive class but incorrectly classified to the positive class. Since we are interested in both positive and negative classes especially in multi-class classification, we use the *accuracy (ACC)* metric to measure the performance of our different classifiers as follows:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

In all subsequent experiments, we use the *10-fold cross validation* [1] to measure the accuracy.

Baseline Method. We use two types of baselines. First, the naive keyword-based (KB) classification uses the category-specific keywords when classifying a Twitter user. Given a stream of tweets from a user, we count the percentage of keywords from each category-specific keyword list present in the tweet corpus. If the percentage exceeds a predefined threshold, then the user is classified into the positive class. If there is a tie, then the class label with higher percentage ratio is returned. Second, the NB or SVM based classification using the textual features in Section III serves as the more sophisticated baseline. Finally, we compare the accuracy of our proposed time-series based classification against these two types of baselines.

B. Binary Classification

Given a stream of tweets from a user, the goal of binary classification is to predict whether the user is likely to follow NFL football teams, i.e., whether the user is a football fan or not. In this task, we combine all the tweets crawled for each of the 32 NFL teams and their fans as positive examples (i.e., 32,000 positive users) and similarly combine all the tweets from the users who do not follow any of the teams as negative examples (i.e., 32,000 negative users).

As to the baselines, we used a total of 8 approaches, all of which use features in *text* domain. First, we tested two approaches using the keyword-based classifier at word level (Word+KB) and tweet level (Tweet+KB). The Word+KB (resp. Tweet+KB) computes the percentage of words (resp. tweets) containing football-related keywords and uses a simple threshold (e.g., 10%) to classify a user into the positive class. Second, we prepared 6 baselines using three variations of features (i.e., TF-IDF and LDA with 20 and 100 topics) and two classifiers (i.e., NB and SVM).

As to our proposed methods, we first used football-specific keywords to convert each user's tweets into a time series on both daily and weekly time scales. On a daily scale, we treat a user's daily tweet streams as a bag of words and count the number of football-specific words (DW) or football-specific tweets (DT) that appear in the daily tweet messages. On a weekly scale, we treat a user's tweet streams on game days (i.e., Sunday and Monday) and non-game days (i.e., Tuesday through Saturday) separately within each week. Then, we prepared two variations – weekly words (WW) and weekly tweets (WT). Next, we used two distance functions (i.e., DTW and SAX) and kNN classifier to do classification in time series domain. Previous research has shown that compared to commonly used classifiers such as SVM, 1-nearest neighbor (1NN) classifier with the DTW distance usually yields superior classification accuracy [20]. Therefore, in this paper, we applied 1NN classifier for simplicity purpose.

Figure 3(a) shows the performance comparison of two types of baseline approaches. We can clearly see that TF-IDF or LDA based methods show much improvements over the keyword-based baseline. First, we can observe that keyword-based baseline at tweet level slightly outperforms the word-level baseline. This is reasonable because as long as a user's tweet contains a category-specific keyword, the classifier treats the entire tweet relevant to the positive class and this in turn increases the $relatedness$ of the user's tweet stream to the positive class. Second, regarding difference between features extracted from tweet contents, we can see that classifiers using the topic feature derived from topic models outperform classifiers using the TF-IDF feature. For example, SVM classifier using topic feature outperforms SVM classifier using TF-IDF and improves the classification accuracy by up to 25%. This is consistent with [16] as topic-based linguistic features are consistently reliable and more discriminative in user classification tasks. Third, using either TF-IDF feature or topic feature, SVM classifier generally outperforms NB classifier. This is also consistent with previous experimental results which show that SVM performs better than NB in general classification tasks [19].

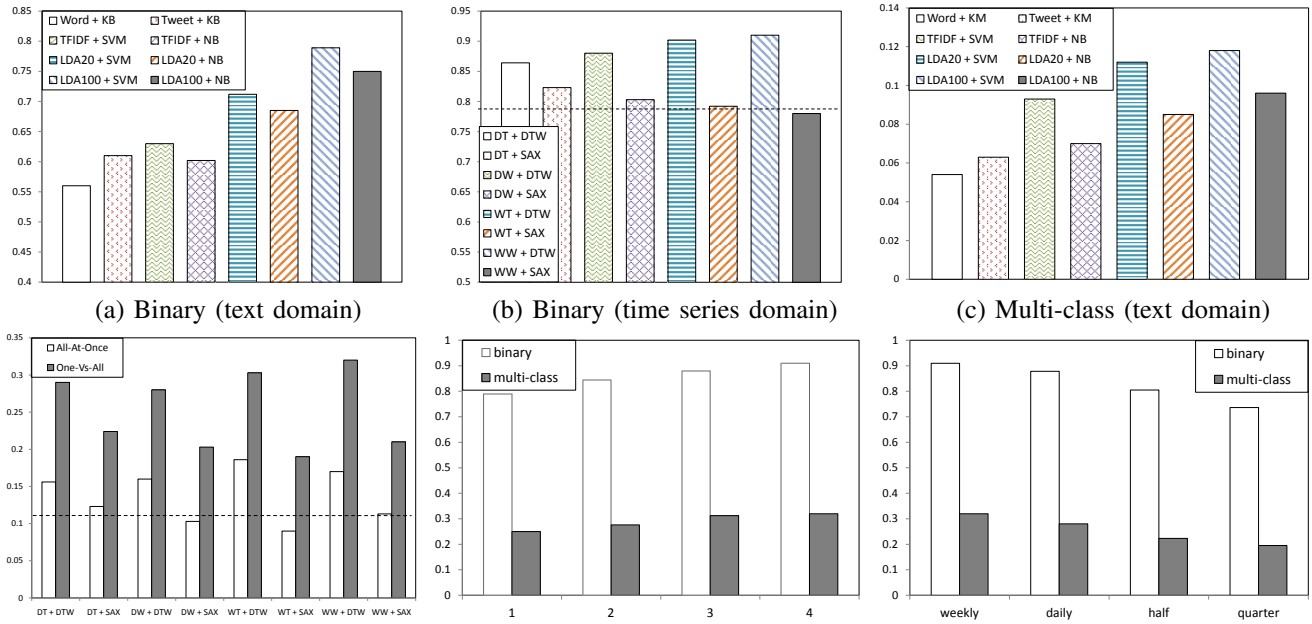
Figure 3(b) shows the performance of our proposed time series approach for user classification. First, we can see that our 1NN classifier using DTW or SAX as distance functions generally performs better than *all* baseline methods in Figure 3(a). For example, 1NN classifier using time series feature on a weekly basis and DTW as distance function outperforms SVM classifier using topic feature by improving the classification accuracy by around 15%, and outperforms NB classifier using topic feature by 22%. Second, regarding user classification in time series domain, 1NN classifier using DTW as distance function generally outperforms 1NN classifier using SAX as distance function. This is due to the fact that SAX is actually used as a symbolization technique for dimension reduction specifically in time series classification. Our time series approach consists of a transformation process to convert textual features to time series features, thus further symbolizing the time series may not be necessary and consequently results in some loss of information. However, the performance of 1NN classifier using SAX is still comparable to or slightly better than the performance of SVM classifier using topic feature.

C. Multi-class Classification

Next, the goal of multi-class classification is to predict which particular team (out of 32 NFL football teams) a given user is a fan of. In this task, we used the corpus with a total of 32,000 fans, i.e., 1,000 users per class. Similar to Section V-B, we applied 8 baselines using features in text domain and 8 variations of our proposal in time series domain. In addition, we adopted two alternatives to evaluate multi-class classification scenario, as illustrated in Algorithms 2 and 3.

Figure 3(c) compares the multi-class classification accuracy among 8 baseline methods in text domain. Again, NB or SVM based baseline methods outperform keyword-based heuristics. First, regarding different features extracted from tweet contents, it is shown that classifiers using the topic feature derived from topic models outperform classifiers using the TF-IDF feature. For example, SVM classifier using topic feature outperforms SVM classifier using TF-IDF and improves the classification accuracy by up to 27%, which is consistent with the binary classification case. This again confirms that topic-based linguistic features are consistently more reliable and discriminative in multi-class user classification tasks. Second, in terms of accuracy, SVM classifier outperforms NB classifier by 23% using either TF-IDF feature or topic feature, which again shows that SVM performs better than NB in multi-class classification tasks.

Figure 3(d) shows the performance of our proposed time-series based Algorithm 2 and Algorithm 3 for multi-class user classification. First, our 1NN classifier using DTW or SAX as distance functions show significant improvements over the



(a) Binary (text domain) (b) Binary (time series domain) (c) Multi-class (text domain)

(d) Multi-class (time series domain) (e) Varying length (X-axis in month) (f) Varying time scale

Fig. 3. (a) & (b): binary classification results, where dotted line in (b) denotes the accuracy of the "best" binary classifier in text domain in (a). Note that DT and DW (resp. WT and WW) are daily (resp. weekly) time series at tweet and word levels, respectively. (c) & (d): multi-class classification results, where dotted line in (d) denotes the accuracy of the "best" multi-class classifier in text domain in (c). (e) & (f): impact of temporal feature size for the "best" binary and multi-class classifiers in time series domain. Y-axis of all graphs denotes the classification accuracy of algorithms.

basic methods in Figure 3(c). For example, our proposed All-At-Once classification algorithm using time series feature on a weekly basis and DTW as distance function outperforms SVM classifier using topic feature by improving the classification accuracy by around 39%. Second, our proposed One-Vs-All classification algorithm further improves the accuracy by 67% over the All-At-Once classification algorithm in the same setting and hence improves by 142% over the baseline. This is because our One-Vs-All classification algorithm builds K binary classifiers when classifying a new user, and returns the classifier producing the best result as the winner. Moreover, during the training of k^{th} binary classifier, the algorithm uses the category-specific keywords of k^{th} class to convert all the users in the training set into time series such that the inter-class difference among users from different categories can be amplified in order to boost the accuracy of classifying the new user into the positive class.

D. Impact of Temporal Feature Size

Next, we select the "best" binary and multi-class classifiers in time series domain as shown in Figures 3(b) and (d), and further study the impact of temporal feature size in terms of classification accuracy. First, we choose different time periods ranging from 1, 2, 3, and 4 months. This represents different lengths of time series for classification. Second, in addition to daily and weekly time frames we used in converting tweet streams into time series, we further divide daily time frame into smaller segments, i.e., a half day and one quarter day. This represents different scales of time series generated for classification. Figure 3(e) compares the classification accuracy as a function of length of time series. We can clearly see that the performance of both binary and multi-class time series classifiers show similar patterns. First, as the length of time series increases, the accuracy of classification in time series domain increases accordingly. This is because the periodicity pattern in tweet streams tends to be steady in larger time periods. Second, the length of time series doesn't impact the accuracy of our time series classifiers too much even on shorter time periods, which demonstrates that our time series classifiers are robust. Figure 3(f) compares the classification accuracy as a function of time scale. First, as the time scale decreases, the accuracy of classification in time series domain decreases accordingly. This is because temporal variation in tweet streams can be aggregated in larger time scales, which in turn can amplify the inter-class difference. Second, using smaller time scale to convert into time series doesn't impact the accuracy of our time series classifiers too much, which again shows our time series classifiers are fairly reliable.

VI. Experiments on Political Interests

In order to corroborate our proposal, in this section, we further perform a classification task of user interests on a different data set. In this experiment, we aim at tackling the *binary* classification problem to identify users as either Democrats (i.e., left) or Republicans (i.e., right). Our experimental question is: How accurately can we predict if a Twitter user is a democrat or a republican?

We used the data set on political polarization from [24], which contains political communications during six-week period (Sep. 14 – Nov. 1, 2010) leading up to 2010 U.S. congressional midterm elections. A political communication is defined as any tweet containing at least one politically relevant *hashtag*. From the set of political tweets, two types of networks, i.e., mentions and retweets, are constructed among a set of Twitter users. Both networks represent public user interaction for political information flow. In this data set, each tweet has the timestamp and a set of hashtags available (no tweet messages available). Each user has the political affiliation

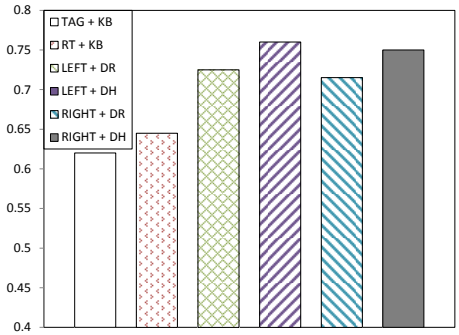


Fig. 4. Binary classification results on political interests. Note that DR and DH are daily time series at retweet and hashtag levels, respectively.

information available (i.e., ground truth). Using only users with at least 30 retweet (RT) activities during the time period, at the end, our data set included 200 Democrats and 200 Republicans, a total of 14,952 retweets with 1,829 unique hashtags. The data set also provides 678 left-leaning (i.e., democrats) political hashtags (e.g., `#p2`, `#dadt`, `#healthcare`, `#hollywood`, `#judaism`, `#capitalism`, `#recession`, `#security`, `#dreamact`, `#publicoption`) and 611 right-leaning (i.e., republicans) political hashtags (e.g., `#tcot`, `#gop`, `#twisters`, `#israel`, `#foxnews`, `#sgp`, `#constitution`, `#patriots`, `#rednov`, `#abortion`). Note that we used only hashtags in this experiment.

Since the data set does *not* contain textual messages but only hashtags, the textual feature based baselines in Section III are not applicable. Instead, therefore, we used two naive keyword-based (KB) classification as the baseline – i.e., the TAG+KB (resp. RT+KB) computes the percentage of hashtags (resp. retweets) containing category-related keywords and uses a simple threshold (e.g., 10%) to classify a user into the positive class. As to our proposed methods, we first used category-specific hashtags to convert each user's retweets into a time series on the daily time scale. We treat a user's daily retweet streams as a bag of hashtags and count the number of category-specific hashtags (DH) or category-specific retweets (DR) that appear in the daily retweets. We prepared two variations, using either democrats-specific (LEFT) and republicans-specific (RIGHT) hashtags to covert users into time series. Finally, we used the 1NN classifier with DTW as the distance function to do classification in time series domain. Same as Section V, we measured the classification accuracy with 10-fold cross validation.

Figure 4 shows the comparison result. Similar to the results for sport interests in Section V, our time series based classifiers outperform both heuristic baseline methods significantly. For instance, the best performing 1NN classifier using daily time series feature at hashtag level (LEFT+DH) increases the accuracy from the baselines using retweets (RT+KB) and hashtags (TAG+KB) by 16% and 22%, respectively. Next, regardless of using democrats-specific or republicans-specific hashtags, our time series classifiers at hashtag level (LEFT+DH or RIGHT+DH) outperforms the retweet-level classifiers (LEFT+DR or RIGHT+DR) . This is because there exist multiple category-specific hashtags in political retweets of a democrat or republican user such that the inter-class difference can be "amplified" when it is captured as time series based on the frequency of relevant political hashtags.

VII. Conclusion

In this paper, we presented a novel method to classify Twitter user interests using time series generated from the contents of tweet streams. By amplifying the latent *periodicity* pattern in tweets into time series, we showed the cases where both binary and multi-class classification accuracy can be improved significantly. Using real data sets on both sports and political interests, we validated our claim through comprehensive experiments by showing that our time series based classifiers outperform up to eight competing classification solutions significantly.

Acknowledgment

This research was in part supported by NSF awards of DUE-0817376, DUE-0937891, and SBIR-1214331.

References

[1] M. Pennacchiotti and A. M. Popescu, "Democrats, republicans and starbucks afficionados: User classification in twitter," in *KDD*, 2011.

[2] D. Rao, D. Yarowsky, A. Shreevats, and M. Gupta, "Classifying latent user attributes in twitter," in *SMUC*, 2010.

[3] A. Java, X. Song, T. Finin, and B. L. Tseng, "Why we twitter: An analysis of a microblogging community," in *WebKDD*, 2007.

[4] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, "Short text classification in twitter to improve information filtering," in *SIGIR*, 2010.

[5] X. H. Phan, M. L. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," in *WWW*, 2008.

[6] A. Sun, "Short text classification using very few words," in *SIGIR*, 2012.

[7] K. Nishida, T. Hoshide, and K. Fujimura, "Improving tweet stream classification by detecting changes in word probability," in *SIGIR*, 2012.

[8] J. D. Burger, J. C. Henderson, G. Kim, and G. Zarrella, "Discriminating gender on twitter," in *EMNLP*, 2011.

[9] A. Bifet and E. Frank, "Sentiment knowledge discovery in twitter streaming data," in *DS*, 2010.

[10] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: a content-based approach to geo-locating twitter users," in *CIKM*, 2010.

[11] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui, "Exploring millions of footprints in location sharing services," in *ICWSM*, 2011.

[12] H.-W. Chang, D. Lee, M. Eltaher, and J. Lee, "@phillies tweeting from philly? predicting twitter user locations with spatial word usage," in *ASONAM*, 2012.

[13] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on twitter," in *CEAS*, 2010.

[14] K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch, "A word at a time: computing word relatedness using temporal semantic analysis," in *WWW*, 2011.

[15] J. Yang and J. Leskovec, "Patterns of temporal variation in online media," in *WSDM*, 2011.

[16] M. Pennacchiotti and A.-M. Popescu, "A machine learning approach to twitter user classification," in *ICWSM*, 2011.

[17] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," in *Journal of Machine Learning Research*, 2003.

[18] C. D. Manning and H. Schutze, *Foundations of statistical natural language processing*. MIT Press, 1999.

[19] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[20] X. Xi, E. J. Keogh, C. R. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast time series classification using numerosity reduction," in *ICML*, 2006.

[21] S. D. Bay, "Combining nearest neighbor classifiers through multiple feature subsets," in *ICML*, 1998.

[22] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *KDD*, 1994.

[23] J. Shieh and E. J. Keogh, "*i*sax: indexing and mining terabyte sized time series," in *KDD*, 2008.

[24] M. Conover, J. Ratkiewicz, M. Francisco, B. Gonçalves, F. Menczer, and A. Flammini, "Political polarization on twitter," in *ICWSM*, 2011.