Towards Cross-Domain Learning for Social Video Popularity Prediction

Suman Deb Roy, Tao Mei, *Senior Member, IEEE*, Wenjun Zeng, *Fellow, IEEE*, and Shipeng Li, *Fellow, IEEE*

***Abstract*—Previous research on online media popularity prediction concluded that the rise in popularity of online videos maintains a conventional logarithmic distribution. However, recent studies have shown that a significant portion of online videos exhibit bursty/sudden rise in popularity, which cannot be accounted for by video domain features alone. In this paper, we propose a novel transfer learning framework that utilizes knowledge from social streams (e.g., Twitter) to grasp sudden popularity bursts in online content. We develop a transfer learning algorithm that can learn topics from social streams allowing us to model the *social prominence of video content* and improve popularity predictions in the video domain. Our transfer learning framework has the ability to scale with incoming stream of tweets, harnessing physical world event information in real-time. Using data comprising of 10.2 million tweets and 3.5 million YouTube videos, we show that social prominence of the video topic (context) is responsible for the sudden rise in its popularity where social trends have a ripple effect as they spread from the Twitter domain to the video domain. We envision that our cross-domain popularity prediction model will be substantially useful for various media applications that could not be previously solved by traditional multimedia techniques alone.**

***Index Terms*—Cross-domain media retrieval, social media, transfer learning, Twitter, video popularity.**

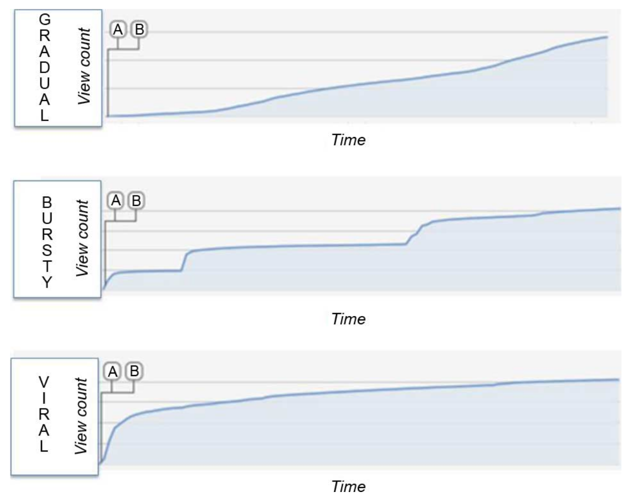

Fig. 1. Common video popularity growth patterns seen in YouTube. Established models are unable to account for bursty video growth, consisting of several sharp spikes of view counts. Note that unlike bursty videos, viral videos contain only a single spike near origin. Bursty videos enjoy several sharp rises in view counts over their life time.

I. Introduction

PREDICTING the popularity of online content has been a subject of great interest over the years due to its ever-growing importance in applications, such as network content caching and advertising, as well as its enormous impact on opinions, culture, policy and profits [1], [24], [25], [28]. Video view count and web traffic are distinctive measures of popularity. However, in spite of brave efforts by researchers, dynamics that drive popularity of online videos in social video portals still remains largely unexplained. In this paper, we aim to throw light upon the main causes that affect social video popularity, explain the diverse popularity growth patterns and build a model that can predict to-be popular videos.

Manuscript received September 12, 2012; revised January 26, 2013 and April 23, 2013; accepted April 29, 2013. Date of publication May 29, 2013; date of current version September 13, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Daniel Gatica-Perez.

S. D. Roy and W. Zeng are with the Department of Computer Science, University of Missouri, Columbia, MO 65211 USA (e-mail: sumandebroy9@gmail.com; zengw@missouri.edu).

T. Mei and S. Li are with Microsoft Research Asia, Beijing 100080, China (e-mail: tmei@microsoft.com; spli@microsoft.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2013.2265079

Previous research on online content popularity prediction concluded that the growth in popularity often maintains a logarithmic distribution [20], [28]. This theory asserts that videos have a gradual rise in view count with no sudden bursts. Fig. 1 shows some distinguishing popularity growth patterns exhibited by YouTube videos. However, recent research has discovered that a significant portion of online videos in video portals do not in fact display the gradual rise in popularity [25], [26]. Instead, they exhibit sudden bursts of popularity [29]; an effect not captured by the established models for video popularity prediction [20], [27]. Bursty videos are lucrative to detect computationally, since the sudden rise in popularity of such videos provides an unique opportunity for advertising and caching. We define bursty videos as those videos for which the slope of view count growth changed drastically (slope>45 degrees) over a period of one day not considering the day of upload (see Fig. 1).

The shift in online activity patterns in recent times can be deeply attributed to the rise of social media [24], [32]. Consider the video portal YouTube, which recently reported that over 700 YouTube videos are shared in Twitter (a social micro-blogging site with 500 million users) each minute. Eventually, view counts are credited to user activity and a significant portion of what users watch is being increasingly referred by social media [20]. Moreover, search query logs of video portals are indicative of what users are looking for. Search in video portals is often

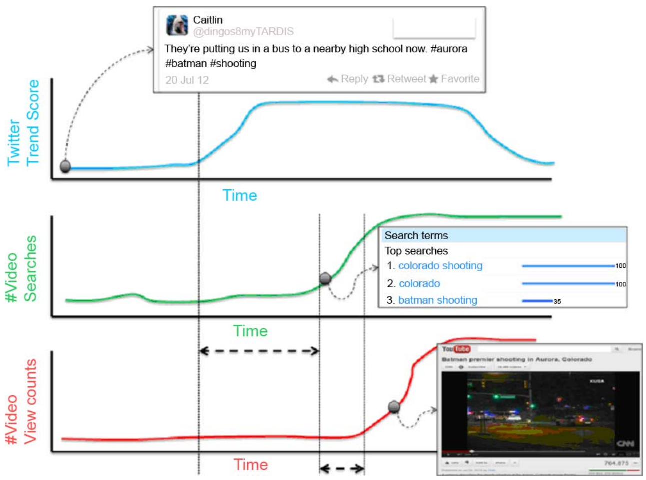

Fig. 2. In the chain of digitization, a physical event is first detected using Twitter trends (blue/top). It is then searched online (green/middle), finally leading to increased views for a video related to the event (red/bottom). The x and y axes are not to scale. There is a certain time gap between trend appearing on Twitter and rising popularity in video portals. This time span can be used to make popularity predictions.

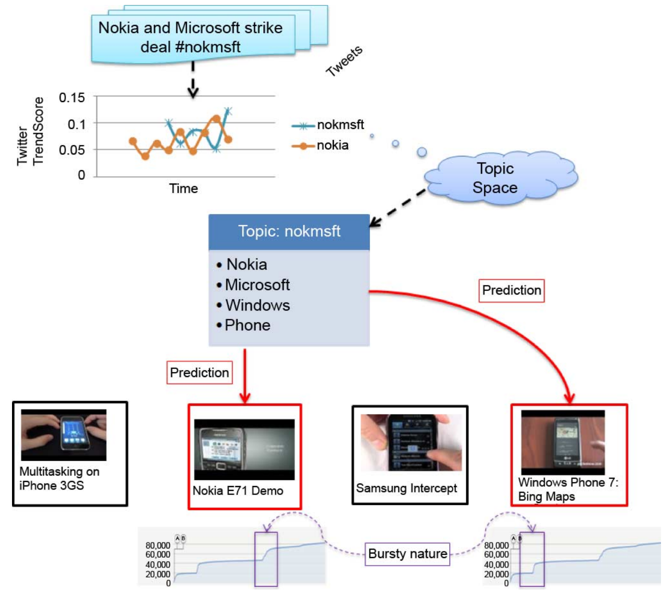

Fig. 3. Example of using social topics in building social trend aware multimedia applications. In this example, we show that bursty video popularity can be predicted by using topics learned from the domain of social streams. This cross-domain transfer of knowledge is accomplished through a mutual topic space (e.g., the space includes the topics like "nokmsft" containing words like "Nokia", "Microsoft", "Phone" etc.).

motivated by real world events. Such events are captured foremost by social micro blogs like Twitter [16]. This was validated in our preliminary study [13], which showed a remarkable correlation between terms trending on Twitter and keywords comprising video search queries in a commercial video portal. This effect, in light of the recent "Aurora shootings", is visualized in Fig. 2. It would thus seem logical that social trends have a surprisingly huge impact on what we watch online and subsequently on video popularity growth patterns.

The information in social streams like Twitter is a good indicator of crowd sourcing activity of a social community and can be used to learn about real life events quickly. Consider the recent shootings in Aurora Colorado which were first reported on Twitter (Fig. 2), closely followed by Reddit and Facebook and finally in news websites and video portals [30]. Thus, "the chain of digitization of a real world event", especially breaking news usually originates in micro-blogs like Twitter and then spreads to other web sites. In addition to its enormous sensitivity to real world breaking news, Twitter data are also being utilized to solve a lot of multimedia problems that could not be elegantly solved before, including semantic video indexing, video context annotation, visualizing political activity, Olympics sentiment analysis and flu-outbreaks [16], [23], [32].

However, social streams (like Twitter) and traditional media (like video publishing sites) exist across disparate domains on the Internet. When dealing with various Internet media (like Twitter and YouTube), the word 'domain' usually indicates the media platform in which it is generated. The domain greatly affects how fast the data is generated and updated (i.e., the data distribution), reflecting freshness and usability in real-time applications. Thus, the potential of these two resources (tweets vs. videos) is constrained within the domain where it resides. In order to predict which videos will gain sudden/bursty popularity, we need to incorporate social knowledge from Twitter domain to video domain. Incorporating social knowledge into traditional media applications requires cross-domain information transfer, which bestows the *wisdom of the crowds*. It is therefore important to develop a cross-domain knowledge transfer mechanism from the crowd-sourced social domain to the traditional media (video) domain.

There are significant challenges in using social streams to perform cross-domain popularity predictions. The main concern is to transfer knowledge across domains and align features that are common to both domains (e.g., video tags and social stream topic words as shown in Fig. 3). With respect to videos, we deem meta-data (like video tags) as representative features of the video. Thus, when we mention video features, it does not exclusively refer to 'visual' features. Specifically, some distinct challenges in making cross domain socialized predictions are:

- A unified framework to combine the social and multimedia feature information which has different domain-specific properties.
- An algorithm that can seamlessly propagate the knowledge (i.e., social topics) mined from the crowd-sourced social streams to the video domain.
- The scaling up and adaptation of the learning algorithm to the ever bursty real-time nature of the social streams.
- Capability of dealing with the noisy, incomplete, ambiguous, and short form nature of social stream data. For example, each tweet is limited to 140 characters and often improperly structured in grammar/syntax.

In this paper, we propose an approach to measure the *social prominence of a video* by using trends learned from Twitter streams as social sensors of video popularity. We claim that the social prominence of a video is substantially responsible for its bursty popularity in the video domain, depending on the topic that a video belongs to. The task of finding this topic requires construction of an algorithm capable of scalable real-time transfer learning between the domains of social streams and traditional media (like video). We call this transfer learning scheme *SocialTransfer*, which utilizes topics extracted from social streams to build an intermediate topic space in between the social and video domains.

SocialTransfer employs the Online Stream LDA model (OSLDA) to learn topics from social streams [13]. It is modeled as a graph based framework to resolve the transfer learning problem (what feature information is transferable and how) between the social and the video domains. Spectral analysis of this graph fetches the eigenvectors, using which we can represent both the social and the video feature information as a combined feature representation [9]. Since the stream is temporal nature, *SocialTransfer* also allows progressively updating the topic space and seamlessly incorporating newer trends into the transfer learning framework for socially aware popularity prediction. The real-time topic updating allows us to calculate the social prominence of a video in real-time, which makes prediction of bursty video popularity possible.

The framework we develop can be reused for several multimedia applications where social influence is capable of improving performance. Our results show that *SocialTransfer* considerably outperforms traditional learners without transfer learning in detecting online video content which will gain sudden/bursty popularity. Here, we summarize the main contributions of this work which have not been uncovered in previous research:

- *Modeling the social prominence of media content* in various social media domains on the Internet based on the visibility of the media topic in online social networks. This problem of modeling bursty popularity in videos is difficult to solve by video domain features alone and has not been previously analyzed from the perspective of social media.
- Intuitive realization and empirical evidence that media in some social media domains gains bursty/sudden popularity due to the increased popularity of the media topic in the another online social network domain. Thus, the *popularity signal is carried across domains of social media existence*. We show how this signal can be transferred across domains using the *SocialTransfer* algorithm.
- Large scale experiments on social and video data especially catered to test the effectiveness of the cross domain popularity penetration hypothesis, i.e., *how social network topic popularity affects the popularity of media with same topic across disparate social media domains*.

The remaining of this paper is organized as follows: Section II discusses related work. Section III provides an overview of the social popularity prediction problem in the light of *SocialTransfer*. Section IV introduces the *SocialTransfer* approach for scalable transfer learning from social stream data. In Section V, we formulate the social prominence of a video for socialized video popularity prediction. Section VI describes the experimental data (videos and tweets) and reports performance results. We conclude the paper in Section VII.

II. Related Work

We discuss related work in the field of transfer learning, mining social streams and online video popularity prediction.

A. Transfer Learning

Common machine learning techniques traditionally address isolated tasks. In contrast, transfer learning aims to transfer knowledge learned in one source domain and use it to improve learning in a related target domain. The source domain data Z_{src} contains the auxiliary data, while target domain Z_{tar} contains the training and test data. A comprehensive survey of transfer learning techniques is provided in [7]. A unified framework for transfer learning in scenarios ranging from cross-domain, cross-category and self-taught learning is described in [8]. Transfer learning has been previously used in various cases including classification, image clustering, collaborative filtering, and sensor based location prediction [7], [8].

In contrast to previous work [8], *SocialTransfer* can scale transfer learning to specifically incorporate social stream data as source domain and seamlessly combine topic learning with transfer learning in real-time. To *the best of our knowledge, a framework that can handle social stream topics distinctively as source domain for cross-domain social video popularity prediction has not been proposed before*. This is challenging due to the unique characteristics of social stream data [16].

B. Mining Social Streams

Social data from Twitter streams have been used for various innovative predictions [16], [23], [32] in recent times. Tweets can also be mined to build a relevant topic space using topic modeling [12], [15]. A topic space can act as bridge between the social and the traditional media domain, supporting multimedia applications like social video recommendation and social query suggestion [13]. Topic modeling aims to extract topics from large corpus of unlabeled documents by using generative models like Latent Dirichlet Allocation (LDA) [15]. There have been previous efforts to incorporate social data for recommendation [14], [22], but they do not use social streams specifically [12]. Social streams are more challenging to extract topics from; due to their dynamic, noisy, short and real-time nature [16]. Thus, large scale matrix decomposition is infeasible for social streams [14].

In previous research on mining social stream data, it has been assumed almost without exception that the entire tweet stream is available to the algorithm at the beginning of the run. This assumption is only applicable in ideal case; it does not hold in real life situations. In our paper, we simulate the tweet stream in pseudo real-time, where the *SocialTransfer* algorithm has not seen the entire tweet stream in advance. Instead, the complete timeline is divided into time slots, and a certain number of tweets occupy each time slot as they are generated in real life. Tweet chunks are fed to the *SocialTransfer* algorithm in time-sequential batches based on the time slots in which they are generated (pseudo real-time).

C. Video Popularity Prediction

Popularity prediction of online content has been a topic of consistent interest in the research community [1], [20]. Szabo *et al.* introduced a regression model to predict content popularity using the social news site Digg and YouTube videos [28]. The authors concluded that online videos demonstrate logarithmic

TABLE I
NOTATIONS OF AUXILIARY, TRAINING AND TEST DATA FOR SOCIAL TRANSFER

Dataset	Notation	Domain	Type	Instances	Labels
Training	χ_{train}	Z_{tar}	Online Video	$\chi_{train} = \{x_{tr}^{(m)}\}_{m=1}^{M}$	$\mathbb{C} = \{c^{(i)}\}_{i=1}^{l}$
Test	χ_{test}	Z_{tar}	Online Video	$\chi_{test} = \{x_{ts}^{(n)}\}_{n=1}^{N}$	N/A
Auxiliary	χ_{aux}	Z_{src}	Social Streams (e.g. tweet)	$\chi_{aux} = \{x_{ax}^{(k)}\}_{k=1}^{D}$	$\mathbb{C} = \{c^{(i)}\}_{i=1}^{l}$

view count growth. Other works have also suggested a gradual rise in popularity of online videos [20].

Recently however, Wattenhoffer *et al.* [26] and others [2], [25] have found that video portals like YouTube have significantly different popularity growth characteristics compared to traditional online streaming media. These findings are coherent with studies conducted by Ratkiewicz *et al.*, which show that contrary to the established logarithmic model, changes in popularity in fact occurs in bursts, whose magnitude and time-separation are broadly distributed [29].

Such bursts in content popularity might be due to social visibility (trending nature) of some related topic in context of the video [24]. The trend is indicative of the collective attention of users and boosts sudden popularity [31]. Categorization of videos has also been a hot topic for research. There have been efforts to categorize videos based on related tags [4] and comments [5]. Various patterns of video popularity in YouTube are discussed in [1], [2], [6].

III. Approach Overview

Before we can model the social popularity of a video, let us briefly discuss *social prominence* in light of *SocialTransfer* and provide a broader picture as to how the different components of the system interact. Then, in Sections IV and V, we will consequently elaborate on each of these components.

The bursty or sudden rise in popularity of a video observed in the video domain can be largely attributed to the social importance of the video topic in the Twitter sphere. This means before modeling popularity, we should first be able to classify a video as having a certain membership score to each topic in the intermediate topic space. The topic space is an abstract space containing several clusters of words belonging to various topics that reflect social trends in real time (Fig. 3). A topic modeling algorithm called OSLDA is used for learning topics from the social stream.

The main tasks in socialized video popularity prediction are twofold:

(1) **Detecting topic of a video:** This is achieved using the *SocialTransfer* algorithm, which can classify a video with its topic membership to different topics in the topic space by learning from social stream data. The concept of SocialTransfer was first introduced in our preliminary work [33], here we improve on that for video popularity prediction. *SocialTransfer* is explained in detail in Section VI-A.

(2) **Measuring popularity of the video topic** : This task involves modeling the *social prominence*. It requires calculating a popularity score for the video based on the video's traditional popularity (based on video view counts) and its 'transferred' social importance (based on video topic). The fusion of the traditional and social popularity metrics is called Trend-Aware Popularity *(TAP)* for a video and is described in Section V.

Therefore, given a test video, the first task is to find the topic they belong to. Once we know the topic, we can model the *TAP* for the video. We assume that the traditional popularity metric for a video is based only on its view counts. Therefore, if *TAP* is significantly different from the traditional popularity metric, we predict the video will gain bursty popularity and classify the video as bursty. Notice in Fig. 1 that for bursty videos, the curve has a slope greater than 45 degrees at burst points. We set a similar threshold for the TAP curve i.e., if any point in the curve has a slope greater than 45 degree then we consider it as significantly different from traditional view count growth. The reasoning behind this classification is that sudden popularity of a video is indicative that its topic must be getting social prominence, which has caused its recognition to shift from its traditional popularity score.

As shown in Fig. 4, the different pieces of our system are connected in the following way: (1) the OSLDA algorithm extracts topics from Twitter stream in real-time and populates the topic space, (2) the *SocialTransfer* algorithm allows for classifying videos with social media topics by using this topic space and the transfer graph. This labels a video with a topic learned from social domain. The algorithm also allows for continuous updating of the transfer graph and seamless integration of fresh topics as newer tweet data is encountered. (3) Finally, we calculate the social prominence of each topic and make an informed prediction that videos with social prominence will demonstrate bursty behavior/sudden rise in popularity. This would be empirical evidence that popularity signal of social media traverses across domains to affect video popularity. These components are subsequently explained in more detail in the next two sections.

IV. SocialTransfer

In this section, we will (A) first introduce the transfer learning problem that requires to be solved, and (B) provide brief overview of the *SocialTransfer*. Then, we will present (C) online topic modeling, (D) online spectral graph learning, and (E) algorithm for *SocialTransfer*.

A. The Transfer Learning Problem

In *SocialTransfer*, we have two datasets in the target domain; the target training data $\chi_{train} = \{x_{tr}^{m}\}_{m=1}^{M}$ with labels and the target test data $\chi_{test} = \{x_{ts}^{n}\}_{n=1}^{N}$ without labels. The training

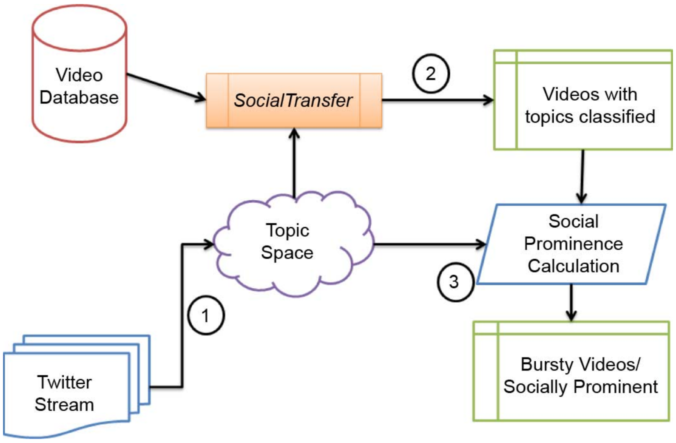

Fig. 4. Interaction among different components of the system: (1) Detecting topics from social streams, (2) Classifying video topics and (3) Modeling social prominence to detect bursty videos.

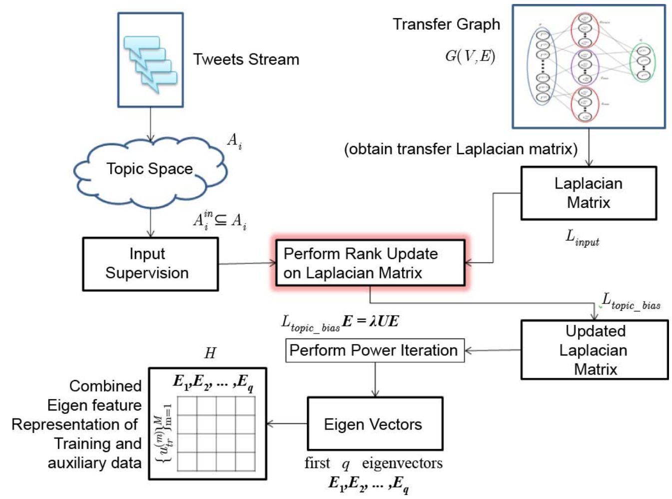

Fig. 5. The flow diagram addresses the overall approach in solving the two key problems of SocialTransfer: (1) learning the shared feature representation across domains in terms of eigenvectors using Spectral Learning (Power Iteration), and (2) reflecting the progressive inclusion of topics by updating the transfer Laplacian matrix.

data contains M instances whereas the test data contains N instances. Labeling the data is discussed in detail in Section V-A. Unlike traditional machine learning, we also have an auxiliary data set $\chi_{aux} = \{x_{ax}^k\}_{k=1}^D$, consisting of D tweets instances. We assume that the target data and the auxiliary data share the same categories (e.g., both a tweet and a video can be regarding music), but exist in different domains (e.g., tweets are social text-based micro-blogs while YouTube videos are part of a video publishing site). Consider a set of B videos in the target domain. For a video v_i, $1 \le i \le B$, we can represent the set of tags of v_i as $\{tags\,(v_i)\}$. Each tag in the set $\{tags\,(v_i)\}$. is a word, represented as w_j^i, $1 \le j \le |tags\,(v_i)|$.

Now consider a stream of D tweets picked from the source domain to be used for modeling the social topic space. For a tweet t_k, $1 \le k \le D$, let $tpw\,(t_k)$ represent the topical words in the topic of t_k (we consider only the principal topic, i.e., topic for which the conditional probability of topic given tweet is maximum). Then each instance/label of the twitter stream data can be represented as $t_k \to tpw(t_k)$. These instances can be combined into the auxiliary data set $\chi_{aux} = \{x_{ax}^k\}_{k=1}^D$.

As shown in Table I, all the instances $x \in \chi_{train} \cup^{\chi_{test}} \cup^{\chi_{aux}}$ are represented by the features in the feature space $\mathcal{F} = \{f_s^{(i)}\}_{i=1}^s$. Our goal is to learn an accurate classifier $f'(.)$ from χ_{train} and χ_{aux} that can predict the testing data with minimum classification error. We call this classifier $f'(\chi_{test})$. Thus, the goal of transfer learning is to minimize the prediction error on χ_{test} by leveraging the auxiliary data from χ_{aux}.

B. Brief Overview

The *SocialTransfer* framework tackles two distinct problems: (1) learning the interconnected pattern of shared features between the source and the target data, (2) *progressive inclusion of topics in pseudo real-time* since topics modeled from social stream (auxiliary data) changes with the real world trends.

Let us first focus on the first problem. The single transfer framework we use for this purpose is represented as a graph called the transfer graph G (see Fig. 6), which contains the videos, tweets, feature words and category information. To learn

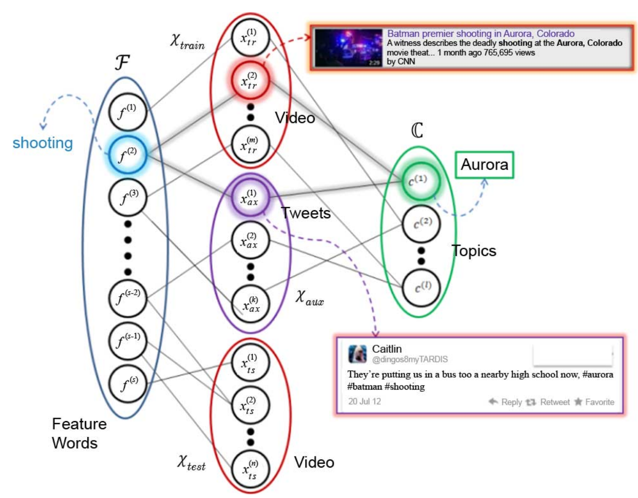

Fig. 6. Transfer graph for SocialTransfer with connections among auxiliary and target data including features and class labels.

the interconnected pattern of shared features between the source and the target data, we perform spectral analysis [11] of the transfer graph. Spectral graph analysis is the study and exploration of graphs through the eigenvalues and eigenvectors of matrices naturally associated with those graphs. The spectrum of a graph is an invariant and captures the structural properties of the graph irrespective of isomorphic changes. As shown in Fig. 5, spectral learning uses a technique called Power Iteration [17] to extract the eigenvectors from the Laplacian representation of the transfer graph (i.e., its Laplacian matrix). Although this will be described in detail in later sections, let us briefly explain these concepts. For the transfer graph G, its adjacency matrix G_{adj} and degree matrix G_{deg}, the Laplacian matrix of the transfer graph can be defined as: $G_{Lap} = G_{deg} - G_{adj}$. The Laplacian matrix is positive semi-definite, implying every eigenvalue of this matrix will be non-negative. The importance of such Laplacian eigenvalues is that several properties of the graph including graph invariants like maximum cut and connectivity are encoded in it. Power Iteration is a method of retrieving the Laplacian eigenvalue without relying on matrix decomposition. For a symmetric matrix, the rate of convergence of Power Iteration is $|\lambda_2|/|\lambda_1|$, where λ_1 is the largest eigenvalue, meaning it converges twice as rapidly compared to some other eigenvalue algorithms such as the Jacobi Method. The algorithm, when applied to specific situations, can be an extremely scalable way to calculate eigenvalues, due to its linear convergence and cost per step in order of $O(n^2)$. For example, Google uses Power Iteration to calculate the PageRank of documents in their search engine.

Spectral analysis of the transfer graph gives us the combined feature representation of the auxiliary and the training data using eigenvectors. This eigen feature representation reflects the intrinsic structure in terms of the principal components of the combined source and training data. Traditional learners (like Support Vector Machines/SVM [19]) can then use the combined features for prediction rather than using only the training features.

Next, let us address the second problem of how to progressively include social topics. Since the tweet stream is incrementally witnessed by the algorithm, the transfer graph needs to be updated in order to progressively include the twitter topics in pseudo-real time. This is achieved by treating the topics as input supervision before spectral learning as explained in the following sections. The role of input supervision is shown in Fig. 5, where selected topics act as input supervision for the Laplacian matrix which allows for smooth incorporation of social topics into the transfer learning framework.

C. Learning Topics From Social Streams via OSLDA

We use the Online Streaming LDA (OSLDA) model for real-time topic learning from Twitter stream [13], which generates a tweet-topic distribution and a topic-word distribution. Each topic is comprised of a group of related words called *topical* words (Table II). Topic learning treats each tweet as a document and builds a generative model to connect the tweet to one or more topics. Thus, the topic of a tweet contains words (topical words) that are related to the tweet words but might not be explicitly present in the tweet itself. Similar to the original LDA introduced by Blei *et al.* [15], we use certain category-indicative words in the vocabulary when detecting topics, such as the word 'actor' might indicate 'Entertainment' category. Similarly, the word 'goal' might indicate 'Sports'. This is a standard technique in topic modeling.

Since tweets contain less words than standard documents, it is advantageous to incorporate supervision where available, as is discussed in [12]. The supervision effectively allows to expand the number of words per document, by using word co-occurrence. In addition to this, since the model is online and streaming, we need to initialize the model with topics from a group of tweets (say conversations or tweets of an user) at the beginning, followed by adding additional documents as tweets with each batch of tweets appearing in each slot [13].

Note that each assigned topic consists of a cluster of topical words. During the process of topic modeling, these words have some statistical semantic similarity, which is why they were grouped in a cluster called the topic. We can limit ourselves to incorporating only selected topics from the topic space as input supervision (an additional set of labeled instances) for the transfer learning task. Thus for K topics in the global topic space, we can choose a particular set of topical words $A_i^{in} \subseteq A_i$, for $i = 1, 2, \ldots, K$ to act as the bias or input supervision to update the transfer graph before spectral learning. This sort of input topic supervision is fed into the transfer graph progressively, as is depicted in Fig. 5, where topics modeled in real-time from the social stream using OSLDA is used to update the transfer graph by means of a ranked update (5) on the transfer Laplacian matrix representation of the transfer graph. This allows progressive and seamless inclusion of topics into the transfer graph facilitating the social influence in transfer learning.

TABLE II
EXAMPLE TOPICAL WORDS AND RELATED TOPICS

Topical Words	**Assigned Topic**	**YouTube Category**
dance, adventure, photography, visit	events	Travel & Events
anime, hero, online, celebrity, diva	films	Films & Animation
iphone, games, showcase	electronics	Sci. & Tech
war, economy, army, revolution, blog, egypt	politics	News & Politics
trailer, show, live, watch	entertainment	Entertainment
wow, rap, jam, gaga	music	Music

D. Transfer Graph

The transfer graph's main purpose is to capture the cross-domain attributes of social streams and videos for using in the transfer learning task and model the relation between the auxiliary data from Twitter and the target video data. This 'transfer graph' (Fig. 6) contains the instances, features and class labels of the target data and the observed auxiliary data as vertices. The edges are set up based on the relations between the auxiliary and the target data nodes. The transfer graph presents a unified graph structure to represent the task of transfer learning from social domain to video domain.

As shown in Fig. 5, the transfer graph $G(V, E)$ consists of vertices representing instances, features or class labels, and edges E denoting co-occurrences between end nodes in the target and the auxiliary data i.e.,:

$$V = \chi_{train} \cup \chi_{test} \cup \chi_{aux} \cup \mathcal{F} \cup \mathbb{C}. \tag{1}$$

The weight of each edge where one of the end nodes belongs to $\mathbb{C}$ indicates the number of such co-occurrences. Let $\omega_{x,f}$ represent the importance of the feature $f \in \mathcal{F}$ that appears in instance $x \in \chi_{train} \cup \chi_{test} \cup \chi_{aux}$. Then, the weight of an edge where one of the end nodes belongs to $\mathcal{F}$ is indicated by $\omega_{x,f}$. The importance of a feature word $\omega_{x,f}$ can be calculated using the topic-word probability distribution matrix obtained from OSLDA. The total number of features and class label nodes remains fixed in the transfer graph. Let $T(x)$ represent the true label of the instance. If e_{ij} denotes the weight of an edge between two nodes ϑ_i and ϑ_j in the transfer graph, then edge weights can be assigned as:

$$e_{ij} = \begin{cases} \omega_{\vartheta_i,\vartheta_j} & \vartheta_i \in \chi_{train} \cup \chi_{test} \cup \chi_{aux} \bigwedge \vartheta_j \in \mathcal{F} \\ \omega_{\vartheta_j,\vartheta_i} & \vartheta_i \in \mathcal{F} \bigwedge \vartheta_j \in \chi_{train} \cup \chi_{test} \cup \chi_{aux} \\ 1 & \vartheta_i \in \chi_{train} \bigwedge \vartheta_j \in \mathbb{C} \bigwedge T(\vartheta_i) = \vartheta_j \\ 1 & \vartheta_i \in \chi_{aux} \bigwedge \vartheta_j \in \mathbb{C} \bigwedge T(\vartheta_i) = \vartheta_j \\ 1 & \vartheta_i \in \mathbb{C} \bigwedge \vartheta_j \in \chi_{train} \bigwedge T(\vartheta_j) = \vartheta_i \\ 1 & \vartheta_i \in \mathbb{C} \bigwedge \vartheta_j \in \chi_{aux} \bigwedge T(\vartheta_j) = \vartheta_i. \end{cases} \quad (2)$$

For all other cases except the ones mentioned in (2), we set $e_{ij} = 0$. The edge weights thus represent the occurrence/importance of a category or feature present in the auxiliary/target data, which will be eventually utilized as a distance metric during spectral clustering. Some nodes in the graph may be isolated with no edge connections. The matrix updating process (Section IV-E) adds new edges to the isolated nodes. The transfer graph G is usually sparse, symmetric, real and positive semi-definite, which allows the possibility of calculating its spectra efficiently [11]. The graph spectrum in terms of eigenvectors is the impression of the structure of relations among the source and target data. This structural relation between the cross domain data is the essence of transfer learning [8]. Thus, it is necessary to represent the source and target data as a transfer graph and then analyze their structural relation by learning the graph spectrum.

E. Learning Graph Spectra

The highlight of *SocialTransfer* is how it learns transfer graph spectra and incorporates new social topics into the transfer graph in real-time. This task is non-trivial, since if not properly done, it may incur substantial costs in terms of scalability (e.g., in eigen-feature extraction) and interoperability (in integration of topics) between topic modeling and transfer learning. In this section, we demonstrate how we achieve both these goals efficiently.

Once the transfer graph $G = (V, E)$ is built, we can use graph spectra analysis to form an eigen feature representation, which combines the principal component features from the training and the auxiliary data. In order to extract the top $-q$ eigenvectors of the transfer graph $G = (V, E)$, we first need to convert the graph into a Laplacian matrix. Let $\deg(\vartheta_i)$ denote the degree of the i-th vertex in G. Then the transfer graph Laplacian $L_{input} \Delta(l'_{i,j})_{|V| \times |V|}$, can be obtained as:

$$l_{i,j}' \Delta \begin{cases} \deg \vartheta_i & \text{if } i = j \\ -1 & \text{if } i \neq j \bigwedge e_{ij} = 1 \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

If the Laplacian eigen values are represented as:

$$\lambda_0 = 1 \geq \lambda_1 \geq \cdots \geq \lambda_p$$

then the eigen gap can be defined as: $eigengap = \lambda_q/\lambda_q - 1$. Since the Twitter stream is extremely dynamic, topics and trends change over time. This requires a feature extraction scheme that can reflect and scale with the social stream. Previous approaches for spectral feature representation in transfer learning have suggested the use of the normalized cut (Ncut) technique for eigenvector extraction [8]. However, our experiments (Fig. 9) showed that the normalized cut technique is incapable of scaling with the twitter stream.

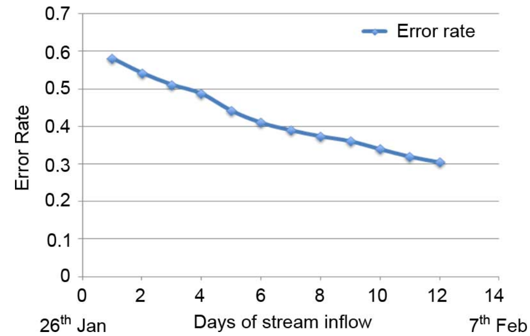

Fig. 7. Drop in prediction error rate with daily stream inflow from Twitter.

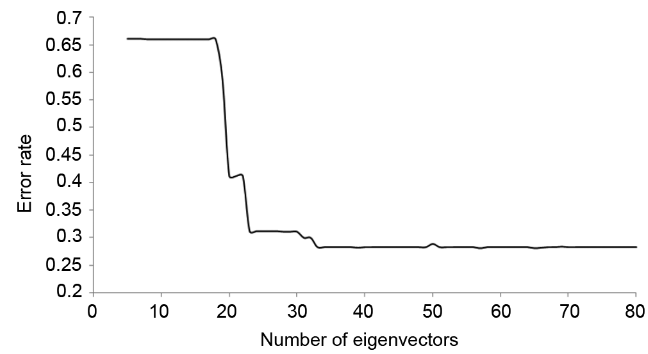

Fig. 8. The influence of the number of eigenvectors extracted on the error rate.

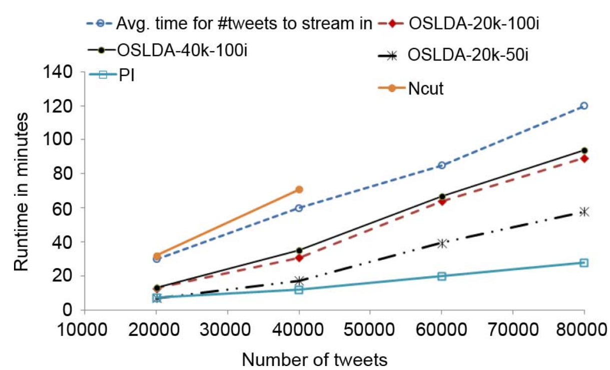

Fig. 9. Runtime comparison for topic modeling and eigen decomposition with incoming tweet stream in SocialTransfer.

Therefore, we use a Power Iteration technique for computing the q largest eigenvectors of L_{input} [17]. The method begins with a random $|V| \times q$ eigenvector matrix and iteratively performs matrix multiplication and ortho-normalization until convergence [11]. The speed of convergence of this method depends on the eigen gap, i.e., the difference between successive eigen values. In fact, Bach *et al.* mention that the number of steps required for the orthogonal convergence in the Power Iteration method is $O(1/1 - eigengap)$[11].

Since topics are updated in the topic space with time, we need to devise a way to progressively incorporate these new topics into the transfer graph. These topics could be incorporated by picturing them to be a time-dependent labeled bias (like a semi-supervised bias) which is an additional set of labeled instances acting as input supervision. One option for incorporating the semi-supervised topic bias as input supervision into the Laplacian representation of the transfer graph (L_{input}) is by producing a ranked update on L_{input} (see (5)). The update in effect recalculates the weights of edge/path between the features and the corresponding labels within the transfer graph, thus updating the characteristic of the Laplacian ((2), (3)). Essentially, the ranked update on the Laplacian using the topic bias adds positive weights between feature words that share the same topic and adds negative weights between feature words that belong to different topics. Thus, the target and the auxiliary data instances act as sort of virtual nodes enabling this re-weighing of the feature edges.

An additional reason for using the ranked update technique is that previous work [18] has also rigorously demonstrated that when Laplacians such as L_{input} is positive semi-definite, a ranked update can improve eigenvector extraction speed by spreading the eigen gap. The next subsection elaborates on how we use ranked updates to incorporate semi-supervised topic bias and update the transfer Laplacian.

1) Incorporating Social Topics: We know from topic modeling that the words in tweets can be clustered into topics. Let us consider there are K such topic clusters. The semi-supervised topic bias is implemented by assuming we know the correct topic labels for a subset of the feature words. This input is learned by topic modeling using OSLDA, which was described in Section IV-C.

The semi supervised bias consists of a set of topical words for each topic $A_i^{in} \subseteq A_i$, for $i = 1, 2, \ldots K$ that act as input supervision. Let us consider the simple case of two topic clusters A_1^{in} and A_2^{in}, such that $A^{in} = A_1^{in} \cup^{A_2^{in}}$ denotes the set of labeled bias instances. Also, consider $d_i = \sum_j e_{ij}$ and $vol\left(A_k\right) = \sum_{i \in A_k} d_i$. We can then define a regularization vector δ_1 as:

$$\delta_1\left(i\right) = \begin{cases} \frac{\sqrt{d_i}}{vol(A^{in})f(i)} & , i \in A^i n \\ 0 & , i \notin A^i n \end{cases} \tag{4}$$

where $f\left(i\right) = \sqrt{vol(A_2^{in})/vol(A_1^{in})}$ if $i \in A_1^{in}$ and $f\left(i\right) = -\sqrt{vol(A_1^{in})/vol(A_2^{in})}$ if $i \in A_2^{in}$.

The effect of the above (4) is to introduce a quadratic penalty if there is a violation in the topic bias label constraints. Said otherwise, this will cause vertices of features that belong to the same topic to cluster together while vertices of different topics will be assigned to separate clusters (due to the penalty). A rank-1 update on the original Laplacian can be made as:

$$L_{topic\_bias} = L_{input} + \gamma.\delta_1\delta_1^T. \tag{5}$$

Similarly, if there are K topics, we can modify the original matrix L_{input} with a rank $-k$ update [18] instead of a rank-1 update. This supervised ranked update firstly allows us to seamlessly incorporate streaming data progressively. Secondly, it aims at tuning certain algebraic properties of the input Laplacian matrix which are related to the convergence rate of the Power Iteration method, eventually speeding the eigen decomposition [18].

In summary, the input supervision using topics learned from the social stream allows us to implement rank $-k$ updates on the transfer-Laplacian matrix as a similarity learning mechanism, where vertex similarities are adjusted on the basis of the topic bias. Note that the number of nodes in the graph is not changed during updating (dimension $|V|$ is fixed); instead the updates only introduce new edges or re-weights existing edges in the graph as it iteratively reuses the eigenvectors from previous update. Due to lack of space, we refrain from describing in detail how the rank $-k$ update improves the speed of eigenvector extraction. In fact, the ranked update increases the eigen gap, which accelerates the convergence of the Power Iteration method. For a detailed explanation of how a supervised bias using rank $-k$ update accelerates the eigenvector extraction process, please refer to [18].

Algorithm for SocialTransfer

Once the first q eigenvectors $E_1, E_2, \ldots, E_q$ have been found by iteratively using the Power Iteration method with the topic-based input supervision, we can form a combined feature representation that depends on both the training and the auxiliary data. Traditional learners like SVMs can use the combined features that include the transfer task to train a classifier $f'\left(\chi_{test}\right)$. Described below, is the algorithm for *SocialTransfer* for classification in the target domain based on auxiliary social stream data.

Algorithm 1: *SocialTransfer*—Transfer Learning from Social

Input: A target classification task which includes the target training data set χ_{train}, the source auxiliary data set χ_{aux} and the target test data set χ_{test}.

Output: Classification result on χ_{test}

1. Construct the initial transfer graph $G(V, E)$ based on the social transfer clustering task (cf. **Section IV-D**).

2. Calculate transfer Laplacian matrix: L_{input} from G using **(3)**.

3. **for** each chunk of tweets entering the system **do**

  4. Calculate the regularization vector δ_1 using the input supervision of social topics A^{in} as shown in **(4)**.

  5. Perform semi-supervised topic bias update on transfer Laplacian: $L_{topic\_bias} = L_{input} + \gamma.\delta_1\delta_1^T$ as shown in **(5)**.

  6. Use Power Iteration to calculate the first q eigenvectors of L_{topic_bias}: $\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3, \ldots, \mathbf{E}_q$ which satisfy the generalized eigenproblem: $L_{topic\_bias}\mathbf{E} = \lambda UE$. The resulting eigenvectors will be used as initial eigenvectors for the next updated Laplacian matrix.

7. **end for**

8. Construct matrix H with $\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3, \ldots, \mathbf{E}_q$ as columns.

9. **for** each $x_{ts}^{(m)}$ in χ_{train} **do**

  10. Let $u_{tr}^{(m)}$ be the corresponding row in H w.r.t $x_{tr}^{(m)}$.

11. **end for**

12. Use a traditional classification algorithm (we use SVM) to train the classifier $f'(\chi_{test})$ based on $\mho_{tr} = \{u_{tr}^{(m)}\}_{m=1}^{M}$ instead of the original training set $\chi_{train} = \{x_{tr}^{(m)}\}_{m=1}^{M}$ and then classify $\chi_{test} = \{x_{ts}^{(n)}\}_{n=1}^{N}$ in the eigen feature space.

V. Social Popularity

In this section, we discuss how to utilize the *SocialTransfer* in calculating the *social prominence* of a video and estimate its social popularity. The steps include: (A) calculate the trending score for each topic (called *Tscore*) and use *SocialTransfer* classification to find the principal topic of a video. The trending score of the principal topic of a video is its *social prominence*; and (B) fusing *social prominence of a video* with its traditional popularity (based on view count) to estimate the final trend aware popularity score (*TAP*). (C) The final goal of this work, predicting which videos will demonstrate bursty nature based on their *TAP*.

A. Social Prominence

Trends are temporal dynamic entities, meaning they grow for a certain period of time, after which they suffer inevitable decay. In other words, trends remain socially prominent for some time and their attractiveness fades away. It is therefore necessary to include a time decay factor when modeling the trending score.

More formally, consider *SocialTransfer* receives a set of D tweets in one time slot; t_{cur} being the current time slot and t_{onset} is the time slot when the trend was first observed. We can then define the trending score of a topic z as:

$$Tscore_z = \frac{\sum_{k=1}^{|D|} P(z|d_k,\ t_{cur})}{|D|\ .\delta_z} \tag{6}$$

where $\delta_z = \phi(t_{cur}, t_{onset})$ is the time dependent decay factor which is a function of the current time slot and the time slot when the trend was first seen. The decay factor must actively respond to trend reoccurrences (i.e., when the trend rises after an initial fall). The decay can be formulated as:

$$tr = \begin{cases} 1,\ P(z|D, t_cur) \geq P(z|D, t_cur - 1) \\ 0,\ P(z|D, t_cur) < P(z|D,\ t_{cur} - 1) \end{cases}.$$

$$\delta_z = \begin{cases} 1, t_cur = t_onset \\ \delta_z, t_{cur} > t_{onset}\ and\ tr = 1 \\ \delta_z + \eta, t_cur > t_onset\ and\ tr = 0 \end{cases} \tag{7}$$

where $0 < \eta \leq 1$ depends on the category of the topic z (meme, music etc.). In addition to the usual trends, active decay can capture extremely dynamic trends like memes or sports related topics, which have short life spans compared to music or entertainment related trends.

For some video v, let z_v^* be the topic to which the video has maximum membership. This membership measure can be easily retrieved using SocialTransfer classification, since the output of the classification is the topic of the video. Then the *social prominence* of video v is $Tscore_{z_v^*}$.

B. Trend Aware Popularity (TAP)

In a traditional video ranking system (like in YouTube) videos with higher view counts are boosted in the rank list [6]. Thus, these videos get clicked more often, resulting in subsequent higher view counts for them [3]. Therefore, it is necessary to engineer a reasonable fusion of the traditional approach and our proposed social prominence approach. This fusion of the traditional popularity factors (like view counts) and the social prominence of the video is called the Trend Aware Popularity (TAP).

In formulating the final popularity score, we also need to take into account the time when the video was uploaded $((t_{upl}))$ since we need to discount the fact that older videos already have higher view counts. Thus, the net temporal Trend Aware Popularity score that we assign to a video v is:

$$TAP_v = \gamma.TScore_{z_v^*} + (1-\gamma)\,.\frac{t_{onset} - t_{upl}}{t_{cur} - t_{upl}}.\#(vc)_{t_{onset}}$$

where $\#(vc)_t$ represents the view count at time t and γ is a weighting factor that balances social vs. traditional popularity control. The above equation measures the social trend aware popularity of a video. The traditional popularity is reflected by the adjusted view count measure, which fractions the view count of a video based on when the video was uploaded in video domain, when the video topic trend was onset in social domain and when the prediction was performed.

C. Predicting Bursty Videos

The *TAP* score reflects the social popularity as well as the traditional (video domain) popularity for a certain video. Our hypothesis is that social popularity signal penetrates across media domains on the Internet. In other words, if a topic is substantially popular (trending) in the social domain, then media belonging to the same topic will gain popularity in other domains (in this case, video domain). Therefore, a ratio of *TAP* to a scaled $Tscore_{z_v^*}$. value will provide us with the quantitative estimation of the impact of the social signal in boosting the overall video popularity for some video v. The lower the value of this ratio, the higher the impact of the social prominence of the video in comparison to the adjusted view count score. Given the same social prominence, the ratio seems to favor videos with lower adjusted view count measure. However, this is not an issue, since the adjusted view count measure is lower when the trend has been seen for longer time period $(t_{cur} - t_{onset})$, which practically means that we are more sure of the prediction if we are exposed to more of past trend data. Thus, for a certain video, if this ratio is significantly lower than for others (lower 10th percentile), we predict the video will gain bursty popularity.

VI. Experiments

A. Data Description

Our study is based on a 3.5 million videos crawled from YouTube and 10.2 million tweets obtained from the NIST Twitter dataset [10]. The source domain is Twitter and the target domain is YouTube. The notations for data from each domain are included in Table I.

1) Twitter Data—Source Domain: The Twitter dataset consists of 10.2 million tweets generated in the US and collected between Jan 26th, 2011 and Feb 11th, 2011. We simulate the twitter data as a stream, with each batch of tweets representing approximately 5 minutes. The resulting rate at which tweets stream over the last week of Jan, 2011, where the 5 min batch time slots account for a total of 288 slots spanning 24 hours in the horizontal axis.

2) YouTube Videos—Target Domain: We collected YouTube data for 3.5 million videos using the YouTube API v2.0. The meta-data for each video includes video id, title, tags, view count, age (in days since uploading), category. We collect the view count for each of these videos from Jan 26th, 2011 and Feb 11th, 2011, to synchronize with the time period of our Twitter data. Then we label all the videos for which the slope of view count growth changed drastically over a period of 1 day (see Fig. 1) as bursty. Therefore, each video in dataset is converted to a instance/label combination: $\{\mathrm{video}_{id} : [\mathrm{tags}, \mathrm{bursty} = 1/0]\}$. For experimentation, a set of B videos are picked randomly from the dataset and divided into two sets for training and testing, called χ_{train} and χ_{test} respectively; where $|\chi_{train}| = M$ and $|\chi_{train}| = N$ represented as $\chi_{train} = \{x_{tr}^m\}_{m=1}^M$, $\chi_{test} = \{x_{ts}^n\}_{n=1}^N$, $M + N = B$ and $M/N \sim 1.5$. The class distribution of our video dataset mimics YouTube's inherent category skewness [2], [34] to a decent extent, which means most occurring category of video in our data set is *Entertainment* (34.5%), followed by *Comedy* (17.1%), *People & Blogs* (15.9%), *Film & Animation* (12.4%), *Sports* (10.8%), *News & Politics* (5.6%) etc.

B. Evaluation Settings

We test our social transfer learning model against traditional learners like SVM [19] which do not use any auxiliary social data in prediction. We used LibSVM with the Radial Basis Function kernel for SVM implementation. Here, the classification task is: given a test video, classify whether it is bursty or not $(\mathrm{bursty} = 1/0)$. For the experiments, we set $\gamma = 1.25$, limit the power method to extracting top-34 eigenvectors and include 60% of the topic space for input supervision. The reasoning of these choices is explained over the following sections. We have three datasets for transfer learning—the target training data, the target test data and the source auxiliary data. The target dataset consists of 3.5 million videos. Our training data consists of 60% videos randomly picked from these 3.5 million YouTube videos. The rest 40% videos (~ 1.4 million) are used for testing. As auxiliary data, we use the 10.2 million tweets from the Twitter stream. We ensure to extract topics from tweets based on approximately 90 categories (16 main+75 other) so that the source and target domains share same categories. Additionally, we also evaluate category-specific predictions based on six popular categories (Comedy, Film & Entertainment, Sports, People & Blogs, Music).

To measure the performance, we use error rate as a metric. Error rate is calculated as (1—*accuracy*) where,

$$accuracy = \frac{\#truePositives + \#trueNegatives}{\#truepositive + \#trueNegatives + \#falsePositives + \#falsenegatives}.$$

C. Results

In Table III, we report the average error in prediction for the Non-Transfer cases (SVM on training only) vs. *SocialTransfer*. Non-Transfer refers to application of the traditional SVM learner to the original target dataset with no social influence (only training features are used); *SocialTransfer* means to apply SVM on the combined feature representation learned using transfer learning from social data (training+auxiliary). The performance in Table III is measured in error rate by averaging 10 random repeats on each dataset by the two evaluation methods. For each repeat, we randomly select 5000 instances per category as target training data. We report the prediction error rate in each of the main categories, along with the overall error for the entire data set. The results are provided category specific to show that the algorithm does better in certain video categories, potentially due to the fact that more information about those categories can be extracted from the social media in the first place. We also report the standard deviation of the repeats in Table III. The two methods are well-tuned using 10-fold cross validation. The **overall gain using *SocialTransfer* is $\sim 39.9\%$ compared to non-transfer cases**. Please note that the overall error rate is averaged over all the main categories and not just the six categories shown in Table III. Performance improvement using transfer learning is most in category 'Music'. In all the major categories, *SocialTransfer* performs better than a traditional non-transfer learner. The F1-score of positive bursty videos the proposed *SocialTransfer* algorithm is 0.68 whereas for the non-transfer SVM it was 0.32.

Additionally, we ran a baseline Naive Bayes classifier, which produces an F1 score of 0.21 without any transfer of auxiliary data. If we replace the SVM in *SocialTransfer* with the Naive Bayes, the F1 score achieved is 0.49. The drop in performance of Naive Bayes in both transfer and non-transfer cases compared to SVM (-0.19 and -0.11 respectively) is expected. Naive Bayes is easy to implement, but it suffers from strong feature independence assumptions. Notice that this feature independence assumption is more costly in the transfer scenario, where the drop in performance is larger than in non-transfer scenario, potentially due to the heavy reliance of *SocialTransfer* on cross-domain feature alignment.

We also provide results of using a majority-class baseline classifier (in place of SVM in Algorithm 1). The F1 score of the final bursty video prediction in this case is 0.111. The distribution of bursty and non-bursty video in our dataset in 17% and 83% respectively. Thus, a majority-class baseline classifier, when directly applied to bursty video prediction, will classify every test video as non-bursty.

We further test the prediction error in the *SocialTransfer* framework based on several factors including scalability and learning capability per day of stream inflow:

1) Accuracy Variation With Stream Inflow: We test the rate at which the prediction error decreases with incoming stream of tweets every day across 12 days of the social data (Jan 26th–Feb 7th). Fig. 7 shows that there is a gradual decrease in error rate as more of the stream is seen by *SocialTransfer*. Lack of any sharp drops hints at the fact the social popularity is significantly trend category specific. On course of the 12 days, we see a 49.4% net reduction of error.

The classification is done continuously at various time points. This is why the decrease in error can be tracked each day as shown in Fig. 7. However, the results shown in Table III are calculated at the end of the entire period of time for which the dataset is available (Jan 26th–Feb 7th).

2) EigenVectors: Previously we mentioned that for the experiments, we fix the number of eigenvectors to be extracted from the transfer Laplacian to 34. The reason for this choice is due to results of Fig. 8, which shows the variation of the error rate with the number of eigenvectors extracted. We see

TABLE III
EXPERIMENTAL RESULTS OF ERROR RATE IN PREDICTING BURSTY VIDEOS FOR SOCIAL VIDEO POPULARITY. THE RESULTS ARE THE AVERAGES OF 10 RANDOM REPEATS ALONG WITH THEIR STANDARD DEVIATIONS. BOTH METHODS ARE TUNED WITH 10-FOLD CROSS VALIDATION

Category → *Approach* ↓	Overall	Comedy	Film & Animation	Entertainment	Sports	People & Blogs	Music
Non-Transfer	0.524 ± 0.031	0.623 ± 0.039	0.412 ± 0.033	0.386 ± 0.028	0.451 ± 0.062	0.324 ± 0.056	0.576 ± 0.028
SocialTransfer	0.311 ± 0.026	0.328 ± 0.043	0.389 ± 0.031	0.289 ± 0.022	0.225 ± 0.074	0.197 ± 0.029	0.236 ± 0.017

that when the number of eigenvectors extracted is greater than 34, the error rate is almost constant.

However, there is a trade-off between the time duration required for extraction vs. error rate of prediction for a certain number of eigenvectors that can be extracted. Thus, since the variation of reduction in error rate is not significant beyond 33–35 eigenvectors, we can safely assume that the extraction of more than 34 eigenvectors is not necessary.

3) Scalability: The speed at which the incoming stream of tweets is explored for topics by OSLDA together with the time required for eigen feature extraction from the transfer graph using spectral learning is important for maintaining scalability with the real-time social stream. In our system, the topic modeling is done in parallel with the eigenvector extraction to save time. Thus, our main aim should be to limit the time required to complete either of these tasks within the incoming tweet flow time.

Fig. 9 shows the comparison of runtimes for various settings of OSLDA, eigenvector extraction using power iteration (PI) and eigenvector extraction using Normalized cut (Ncut) with the time taken on average for an incoming chunk of tweets to stream in. For OSLDA, '20k' (in legend) refers to 20 topics extracted and '50i' refers to 50 iterations of the generative process. Experiments were run on a IBM server with 2.67 GHz processor and 8 GB RAM.

From Fig. 9, we can safely conclude that the model scales to incoming bursts of tweets, since the matrix decomposition with Power Iteration and the topic modeling using OSLDA require less time than the speed of incoming tweets. Note that the Normalized cut method (Ncut) does not scale as it takes longer time to extract eigenvectors than the speed of the incoming burst of tweets as shown in Fig. 9. Moreover, for more than 40,000 tweets, Ncut causes our system to run out of memory.

The experimental results suggest that the proposed approach is an improvement over non-transfer, but it does not ascertain that the cause of every bursty video is its prior topic popularity in Twitter. We show that we can improve the prediction of bursty videos by learning from Twitter trends.

VII. CONCLUSION

In this paper, we shed light upon popularity signatures of online videos. We explain why previous work has detected bursty nature of online popularity, and confirm that it is mostly due to the *social prominence* of such videos. To test our theory, we develop a novel cross-domain real time transfer learning approach based on social streams called *SocialTransfer*. Our proposed scheme can be applied to various multimedia applications which can be boosted by knowledge acquired from cross-domain social data. Here, we demonstrate the use of the *SocialTransfer* in realizing socialized video popularity prediction. Our study could provide indispensable insight into applications like popularity based caching and advertisement for bursty traffic.

Visual analytics and traditional object/semantic detection from video signals will be helpful in enriching the tags in the video, which would allow our model to utilize an increased number of features. We assume the worst case in our paper, that the video only possesses tags extracted from the title and/or entered by the user. Tag enrichment by comment extraction or visual object understanding will improve the prediction power of the model, since it alleviates the meta-data problem effectively reducing the semantic gap.

Experimental results show that *SocialTransfer* can outperform traditional learners by almost $\sim 60.1\%$ increase in accuracy of predicting videos that will gain *social prominence*, identified by their sudden/bursty popularity in the video domain. The main contribution of this work is the scalable model for cross-domain real time transfer learning from social streams that allows the social network trend signal to affect media popularity across disparate social media domains on the Internet, the formulation of *social prominence* of a video and the use of social topics in modeling novel multimedia phenomena that can hardly be realized by traditional multimedia techniques alone.

REFERENCES

[1] F. Figueiredo, F. Benevenuto, and J. M. Almeida, "The tube over time: characterizing popularity growth of youtube videos," *Proc. Web Search and Data Mining*, 2011.
[2] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of youtube videos," in *Proc. Int. Workshop on Quality of Service*, 2008.
[3] R. Zhou, S. Khemmarat, and L. Gao, "The impact of youtube recommendation system on video views," in *Proc. Int. Conf. Internet Measurement*, 2010.
[4] G. Toderici, H. Aradhye, M. Pasca, L. Sbaiz, and J. Yagnik, "Finding meaning on youtube: Tag recommendation and category discovery," in *Proc. IEEE Comput. Vision Pattern Recognit.*, 2010.
[5] K. Filippova and K. B. Hall, "Improved video categorization from text metadata and user comments," in *Proc. ACM SIGIR Conf. Res. and Develop. in Inf. Retrieval*, 2011.
[6] G. Chatzopoulou, C. Sheng, and M. Faloutos, "A first step towards understanding popularity in youtube," in *Proc. IEEE INFOCOM*, 2010.
[7] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.
[8] W. Dai, O. Jin, G. Xue, Q. Yang, and Y. Yu, "Eigentransfer: a unified framework for transfer learning," in *Proc. Int. Conf. Mach. Learning*, 2009.
[9] X. Ling, W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Spectral domain-transfer learning," in *Proc. ACM Conf. Knowledge Discovery and Data Mining*, 2008.
[10] [Online]. Available: http://trec.nist.gov/data/tweets/.

[11] F. R. Bach and M. I. Jordan, "Learning spectral clustering, with application to speech separation," *J. Mach. Learn. Res.*, vol. 7, pp. 1963–2001, 2006.

[12] D. Ramage, S. T. Dumais, and D. J. Liebling, "Characterizing microblogs with topic models," in *Proc AAAI Conf. Weblogs and Soc. Media*, 2010.

[13] S. D. Roy, T. Mei, W. Zeng, and S. Li, "Empowering cross-domain Internet media with real-time topic learning from social streams," in *Proc. IEEE Int. Conf. Multimedia and Expo*, 2012.

[14] H. Ma, T. C. Zhou, M. R. Lyu, and I. King, "Improving recommender systems by incorporating social contextual information," *ACM Trans. Inf. Syst.*, vol. 29, no. 2, 2011, article no. 9.

[15] M. Hoffman, D. Blei, and F. Bach, "Online learning for latent Dirichlet allocation," *Nature*, vol. 23, pp. 1–9, 2010.

[16] M. Naaman, "Social multimedia: highlighting opportunities for search and mining of multimedia data in social media applications," *Multimedia Tools Applicat.*, vol. 49, pp. 1–26, 2012.

[17] F. Lin and W. W. Cohen, "Power iteration clustering," in *Proc. Int. Conf. Mach. Learning*, 2010.

[18] D. Mavroeidis, "Minding the eigen-gap, or how to accelerate semi-supervised spectral learning algorithms," in *Proc. 22nd Int. Joint Conf. Artificial Intell.*, 2011.

[19] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. Annual Workshop Computational Learning Theory*, 1992.

[20] X. Cheng, C. Dale, and J. Liu, "Understanding the characteristics of Internet short video sharing: Youtube as a case study," *Comput. Res. Repository*, vol. 0707.3, 2007.

[21] G.-J. Qi, C. C. Aggarwal, Y. Rui, Q. Tian, S. Chang, and T. S. Huang, "Towards cross-category knowledge propagation for learning visual concepts," in *Proc. CVPR*, 2011, pp. 897–904.

[22] Q. Xu, W. Xiang, and Q. Yang, "Social-behavior transfer learning for recommendation systems," in *Proc. Int. Workshop Social Web Mining*, 2011.

[23] X. Jin, A. C. Gallagher, L. Cao, J. Luo, and J. Han, "The wisdom of social multimedia: using flick for prediction and forecast," in *Proc. ACM Multimedia*, 2010.

[24] T. Broxton, Y. Interian, J. Vaver, and M. Wattenhofer, "Caching a viral video," in *Proc. IEEE Social Interactions Analysis and Service Providers, Int. Conf. Data Mining*, 2010.

[25] G. Gursun, M. Crovella, and I. Matta, "Describing and forecasting video access patterns," in *Proc. IEEE INFOCOM*, 2011.

[26] M. Wattenhofer, R. Wattenhofer, and Z. Zhu, "The youtube social network," in *Proc. ICWSM*, 2012.

[27] D. A. Shamma, J. Yew, L. Kennedy, and E. F. Churchill, "Viral actions: Predicting video view counts using synchronous sharing behaviors," in *Proc. ICWSM*, 2011.

[28] G. Szabó and B. A. Huberman, "Predicting the popularity of online content," *Commun. ACM*, vol. 53, no. 8, pp. 80–88, 2008.

[29] J. Ratkiewicz, F. Menczer, S. Fortunato, A. Flammini, and A. Vespignani, "Traffic in social media II: Modeling bursty popularity," in *Proc. IEEE Int. Conf. Social Comput.*, 2010, pp. 393–400.

[30] [Online]. Available: http://blog.socialflow.com/post/7120245507/big-data-breaking-news-aurora-colorado.

[31] A.-L. Barabási, "The origin of bursts and heavy tails in human dynamics," *Nature*, vol. 435, pp. 207–211, 2005.

[32] S. Asur and B. A. Huberman, "Predicting the future with social media," in *Proc. ACM Int. Conf. Web Intell. and Intell. Agent Technol.*, 2010.

[33] S. R. Deb, T. Mei, W. Zeng, and S. Li, "Socialtransfer: Cross-domain transfer learning from social streams for media applications," in *Proc. ACM Multimedia*, 2012.

[34] J. E. Burgess, J. B. Green, Y. O. video, and P. culture*, YouTube: Online Video and Participatory Culture*. Cambridge, U.K.: Polity Press, 2009.

Suman Deb Roy is a Ph.D. Candidate in the Department of Computer Science at the University of Missouri and a Fellow at the Reynolds Journalism Institute (RJI) at the Missouri School of Journalism. He received his Bachelors degree in Computer Science from the National Institute of Technology (NIT), Durgapur in India, and is expected to receive his PhD degree in 2013.

His research interests include social multimedia, targeted social advertising, semantic analysis of natural languages and micro-blogs, transfer learning and applications of game theory in multi-agent systems. Suman leads the Social Computing Group at the University of Missouri. Suman has also enjoyed summer internships with Huawei Multimedia Networking Lab, NJ and Microsoft Research Asia where he developed algorithms for multimedia encryption and real-time learning from social streams respectively. Most recently, he held the position of Data Scientist at SocialFlow Inc., NY where he worked on several interesting projects on real time social network analysis. He is the recipient of the Gilliom Fellowship of Cyber Security Research and Mizzou Advantage Fund for his work on Media of the Future. Suman is a member of the IEEE Special Technical Community on Social Networking and is a recipient of the 2013 Missouri Honor Award for Outstanding Ph.D. Student.

Tao Mei (M'07–SM'11) is a Researcher with Microsoft Research Asia, Beijing, China. He received the B.E. degree in automation and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2001 and 2006, respectively. His current research interests include multimedia information retrieval and computer vision. He has authored or co-authored over 140 papers in journals and conferences, eight book chapters, and edited two books. He holds seven U.S. granted patents and more than 30 are pending.

Dr. Mei was the recipient of several paper awards from prestigious multimedia conferences, including the Best Paper Award and the Best Demonstration Award at ACM Multimedia in 2007, the Best Poster Paper Award at the IEEE MMSP in 2008, the Best Paper Award at ACM Multimedia in 2009, the Top 10% Paper Award at the IEEE MMSP in 2012, the Best Paper Award at ACM ICIMCS in 2012, the Best Student Paper Award at the IEEE VCIP in 2012, the Best Paper Finalist at ACM Multimedia in 2012, and the IEEE Transactions on Multimedia Best Paper Award 2013. He was the principle designer of the automatic video search system that achieved the best performance in the worldwide TRECVID evaluation in 2007. He received Microsoft Gold Star Award in 2010, and Microsoft Technology Transfer Awards in 2010 and 2012. He is an Associate Editor of *Neurocomputing* and the *Journal of Multimedia*, a Guest Editor of the IEEE TRANSACTIONS ON MULTIMEDIA, the *IEEE Multimedia Magazine*, the ACM/Springer *Multimedia Systems*, and the *Journal of Visual Communication* and *Image Representation*. He is the Program Co-Chair of MMM 2013, and the General Co-Chair of ICIMCS 2013. He is a Senior Member of ACM.

Wenjun Zeng (M'97–SM'03–F'12) is a Professor with the Computer Science Department of University of Missouri. He received his B.E., M.S., and Ph.D. degrees from Tsinghua University, the University of Notre Dame, and Princeton University, respectively, all in electrical engineering. His current research interest includes mobile computing, social media analysis, semantic search, distributed source/video coding, 3-D analysis and coding, multimedia networking, and content and network security.

Prior to joining University of Missouri in 2003, he had worked for PacketVideo Corp., Sharp Labs of America, Bell Labs, and Panasonic Technology. From 1998 to 2002, he was an active contributor to the JPEG 2000 and MPEG4 IPMP standard, where four of his proposals were adopted. He is/was an AE of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *IEEE TMM*, IEEE TRANSACTIONS ON INFORMATION FORENSICS & SECURITY, and IEEE TRANSACTIONS ON MULTIMEDIA (TMM), and was on the Steering Committee of *IEEE TMM* from 2009-2012. He is serving as the TPC co-Chair of the 2013 *IEEE International Workshop on Information Forensics & Security,* and a Guest Editor (GE) of *ACM TOMCCAP Special Issue on ACM MM 2012 Best Papers*. He served as the Steering Committee Chair of ICME in 2010 and 2011, and has served as the TPC Vice Chair of ICME 2009, the TPC Chair of the *IEEE CCNC 2007,* the TPC Co-Chair of the *Multimedia Communication and Home Networking Symposium* of IEEE ICC 2005. He was a GE of the *Proceedings of the IEEE's Special Issue on Recent Advances in Distributed Multimedia Communications* (January 2008) and the Lead GE of *IEEE TMM's Special Issue on Streaming Media* (April 2004).

Shipeng Li (F'10) joined Microsoft Research Asia (MSRA) in May 1999. He is now a Principal Researcher and Research Manager of the Media Computing group. He also serves as the Research Area Manager coordinating the multimedia research activities at MSRA. His research interests include multimedia processing, analysis, coding, streaming, networking and communications; digital right management; advertisement; user intent mining; eHealth; etc. From October 1996 to May 1999, Dr. Li was with Multimedia Technology Laboratory at Sarnoff Corporation as a Member of Technical Staff. Dr. Li has been actively involved in research and development in broad multimedia areas. He has authored and co-authored 6 books/book chapters and 250+ referred journal and conference papers. He holds 120+ granted US patents.

Dr. Li serves as adjunct professor at the Chinese University of Hong Kong and at University of Science and Technology of China. He is the past chair of Visual Signal Processing and Communications technical committee, a member of Multimedia Systems and Applications technical committee, of IEEE Circuits and Systems Society. He is a member of Multimedia Communication technical committee of IEEE Communication Society and a past member of Multimedia Signal Processing technical committee of IEEE Signal Processing Society. He served as an associate editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and on the editorial board of *Journal of Visual Communications and Image Representation*. He is the general co-chair for International Conference on Multimedia and Expo in 2014.

Dr. Li received his B.S. and M.S. in Electrical Engineering from the University of Science and Technology of China, Hefei, China in 1988 and 1991, respectively. He received his Ph.D. in Electrical Engineering from Lehigh University, Bethlehem, PA, USA in 1996. He was a faculty member in Department of Electronic Engineering and Information Science at University of Science and Technology of China in 1991-1992.

Dr. Li received the Best Paper Award at VCIP 2007, the Best Poster Paper Award at MMSP 2008 and the Best Paper Award in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2009). He also mentored his student to receive the Best Student Paper Award in VCIP 2005. In his 13 years with MSRA, Dr. Li has also mentored four MIT TR 35/100 Award Winners.