Learning To Predict Trending Queries: Classification – Based

Chi-Hoon Lee
Yahoo Inc.
chihoon@yahoo-inc.com

HengShuai Yao
University of Alberta
hengshua@cs.ualberta.ca

Xu He
Yahoo Inc.
xuhe@yahoo-inc.com

Su Han Chan
Yahoo Inc.
suchan@yahoo-inc.com

JieYang Chang
University of Washington
jycjerry@uw.edu

Farzin Maghoul
Yahoo Inc.
fmaghoul@yahoo-inc.com

ABSTRACT

Among the many tasks driven by very large scaled web search queries, it is an interesting task to predict how likely queries about a topic become popular (a.k.a. trending or buzzing) as the news in the near future, which is known as "Detecting trending queries." This task is nontrivial since the realization of buzzing trends of queries often requires sufficient statistics through users' activities. To address this challenge, we propose a novel framework that predicts whether queries become trending in the future. In principle, our system is built on the two learners. The first is to learn dynamics of time series for queries. The second, our decision maker, is to learn a binary classifier that determines whether queries become trending. Our framework is extremely efficient to be built taking advantage of the grid architecture that allows to deal with the large volume of data. In addition, it is flexible to continuously adapt as trending patterns evolve. The experiments results show that our approach achieves high quality of accuracy (over **77.5%** true positive rate) and yet detects much earlier (on average **29** hours advanced) than that of the baseline system.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Query Log Analysis; Trending Prediction; Classification

1. INTRODUCTION

One of the interesting tasks dealing with large scaled web search logs is the prediction of future trending queries. The goal of an early trending detection of queries is to effectively predict how likely queries about a topic become popular (a.k.a. trending or buzzing) as the news in the near future. As an example, when a royal baby was born in U.K., people started looking for more details of news; what the gender of the baby is, how much the baby weighs, etc. If web services such as Yahoo are able to capture early popularity of related queries on the topic (e.g., "royal baby born") and lead users to right contents, users's experiences are apparently enhanced, which in turn drives better monetization opportunities.

There are two major challenges in early trending detection– the effectiveness and the earliness. The first is to effectively identify trending queries, which are distinguished from popular queries and accidentally spiked queries. The second is how early trending queries are detected. Particularly, the earliness is tightly related to the service paradigm associated with passively oriented search engine services. To address these challenges, we propose a novel system that predicts whether queries that refer to the news become trending in the future. Our framework is composed of two learners that are trained via historical patterns. The former provides intrinsic descriptive features based on the dynamics of query intensities. Thereafter, the latter learns a classifier to make a decision based on the features. The classifier is then used for large scaled streaming data to detect if a query becomes trending in the near future. Our system is able to continuously learn temporal variances of trending patterns and is designed on the Hadoop architecture [4] to efficiently deal with large scaled data.

2. EARLY DETECTION OF TRENDS (eDOT): LEARNING TO CLASSIFY

Our framework, "**eDOT**", is to predict if a query from streaming data becomes trending in the near future. Two machine learned estimators are required – a regression model that learns the intensity change of a query over time and a classifier to make a decision to tell if the intensity change of a query over time is significant to be trending shortly. It is important to note that the regression model in our system is learned to provide the dynamics of the intensity changes, not to produce estimates. Algorithm 1 illustrates the overview of our system that highlights the two learning procedures.

First, **eDOT** collets training data $\mathbf{D} = \{\mathbf{q}_i, y_i\}_{i=1}^n$ from historical observations, where $\mathbf{q}_i$ is the sequence of query q_i's intensities over time having length d_i. Specifically, $\mathbf{q}_i$ is a feature vector corresponding to the form of $(f_1, f_2, \cdots, f_{d_i})$,

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577315.

Algorithm 1: Early Detection of Trends: **eDOT**

```
  Input  : Training Data, $\mathbf{D}= \{\mathbf{q}_i, y_i\}_{i=1}^n$
  Output: Model Parameter, $\omega$
1 $\Theta$ = LearnAutoRegression($\{\mathbf{q}_i\}_{i=1}^n$, $p$);
  // $\Theta = \{\theta_i\}_{i=1}^n$, where $\theta_i = \{\alpha_1^i, \alpha_2^i, \cdots, \alpha_p^i\}$
2 $\omega$ = LearnAClassifier($\Theta$, $\{y_i\}_{i=1}^n$);
```

where f_i is defined as pair (t_i, c_i). t_i denotes a time stamp and c_i the count for t_i; both retrieved from query logs at Yahoo!. Label y_i for $\mathbf{q}_i$ is in either trending (+1) or non-trending (-1).

Given training data $\mathbf{D}$, we first learn the dynamics of intensities of a query by following principles in Auto Regressive (AR) models [6]. A key principle behind an AR model is to optimize weights of p previous timestamps of time series data under an assumption that a current observation at time k is the linearly weighted combination of previous p intensities. An AR model with order p, $AR(p)$, is to estimate $\hat{c}_{k+1} = \sum_{j=(k+1)-p}^{p} \alpha_j c_j + \alpha_0$, where α_0 is a bias term and $\theta_i = \{\alpha_0^i, \alpha_1^i, \cdots, \alpha_p^i\}$ is a set of parameters that linearly combine the previous intensities of query q_i. A parameter α_k that corresponds to a weight of intensities at k^{th} timestamp quantifies how fast the dynamics of intensities is changing from $(k-1)^{th}$. By reformulating $\hat{c}_{k+1}$ by $\hat{c}_{k+1} = \alpha_0 + \alpha_k \cdot (c_k - c_{k-1}) + (\alpha_k + \alpha_{k-1}) \cdot (c_{k-1} - c_{k-2}) + (\alpha_k + \alpha_{k-1} + \alpha_{k-2}) \cdot (c_{k-2} - c_{k-3}) + \ldots$, it is clear to see that α_k weighs the gradient at k^{th} timestamp and thus the magnitude of weights associated with gradients is interpreted as intrinsic indicators to explain the dynamics of changes. In order to optimize order p as features in learning a classifier, we formulate a score $s_{\mathbf{Q}}(p)$ (defined in Equation (1) using a data set $\mathbf{Q}$) to quantify the system performance at p with respect to the effectiveness and the earliness. In this paper, the earliness and the effectiveness are assumed to be equally weighted, while the mistake rate is penalized.

$$s_{\mathbf{Q}}(p) = \frac{ET_{\mathbf{Q}}(p) \times TPR_{\mathbf{Q}}(p)}{FPR_{\mathbf{Q}}(p)}, \tag{1}$$

where $ET_{\mathbf{Q}}(p) = \sum_{q_i \in \mathbf{Q}} ADT(q_i, p) \times Pop(q_i)$ quantifying how early a system for AR order p can detect over queries (i.e. $ADT(q_i, p)$) with weighted popularity of each query (i.e. $Pop(q_i)$). Note that $ET_{\mathbf{Q}}(p)$ is to adjust the earliness of the system at p by weighting higher to popular queries. $TPR_{\mathbf{Q}}(p)$ and $FPR_{\mathbf{Q}}(p)$ denote the True positive rate and False positive rate, respectively. Although higher orders might be a candidate for the optimal value, the score at $p = 10$ was found statistically significant compared to $p = 5$ at level 0.01, while no statistically significant evidence was found compared to higher orders such as 25 and 30. Note that learning growth dynamics is not sufficient for the trending decision task. For instance, when a manual or heuristic based decision step is additionally required to produce a final outcome, it makes finding an "optimal" threshold challenging and thus suffers from ineffective predictions [2]. To address the challenge, we train a classifier to learn what patterns of growth dynamics of time series data have been recognized as trending and/or non-trending with the "linearity" assumption in AR.

Our system is flexible to employ any type of classifiers from a generative approach (e.g., naïve bayes) to a discriminative one (e.g., logistic regression). In this paper, we learn a model minimizing the hinge loss (based on a support vector machine (SVM)) to take advantage of its robust performance [3, 1].

3. EXPERIMENTS

This section reports some of our experimental results to demonstrate the effectiveness and the efficiency of **eDOT**. We first quantify the effectiveness compared with Weighted Majority Voting (WMV) algorithm [5]. In addition, a baseline system (currently deployed in Yahoo! production) is used to compare the efficiency. To train models and test their performance, we generate data sets using users' queries from the query logs of several months in 2013.

Although WMV algorithm has been shown to perform well in predicting whether a topic will be a trend in an online social network [5], **eDOT** achieves higher effectiveness from the testing set having 2,243 positive and 5,000 negative samples. Note that training samples (i.e., 2607 positive and 2000 randomly selected negative samples) are built from June in 2013, while test data are from Sep. 2013. Refer to Table 1 for the summary.

Table 1: Performance Comparison: Effectiveness

Algorithms	Precision	TPR	FPR
WMV	61.33%	68.27%	19.39%
eDOT	**71.83**%	**72.31**%	**12.76**%

To deal with temporal variances in streaming data, we incrementally update **eDOT** by following online learning [3], which generates a new model **eDOT++**. That is, given a model and additionally labeled data, **eDOT** is updated to **eDOT**++, enhancing the performance of **eDOT**: 73.49% Precision, 77.90% TPR, and 10.44% FPR. With respect to the earliness, our **eDOT** detects much earlier (on average, **29** hours advanced) than that of the baseline system.

4. CONCLUSIONS

In this work, we propose a novel framework that predicts if a query becomes trending in the near future, achieving promising results with advancement of early detection over the baseline. One interesting remark from the experiments is that our system can detect "Texas-California pipeline" as trending query buzzing in June due to its issue related to fuel supply, while the baseline missed it. This highlights the unnecessary penalization towards our approach. We currently perform more elaborated tests by using real users' responses and further explore more robust on-line learning principles.

5. REFERENCES

[1] C. Cortes and V. Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[2] N. Golbandi et. al. Expediting search trend detection via prediction of query counts. In *WSDM 2013.*

[3] Vowpal Wabbit: A fast out-of-core learning system. `http://hunch.net/~vw/`.

[4] Apache Hadoop. `http://hadoop.apache.org/`.

[5] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108:212–261, 1994.

[6] S.M. Pandit and S.M. Wu. *Time series and system analysis, with applications.* Wiley, 1983.