IEICE TRANS. INF. & SYST., VOL.E97–D, NO.10 OCTOBER 2014

LETTER

Exploring Time Aware Features in Microblog to Measure TV Ratings

Joon Yeon CHOEH[†], ***Member***, **Hong Joo LEE**[††a)], ***and*** **Eugene J. S. WON**[†††], ***Nonmembers***

SUMMARY In measuring TV ratings, some features can be significant at a certain time, whereas they can be meaningless in other time periods. Because the importance of features can change, a model capturing the time changing relevance is required in order to estimate TV ratings more accurately. Therefore, we focus on the time-awareness of features, particularly the time when the words of tweets are used. We develop a correlation-based, time-aware feature selection algorithm which finds the optimal time period of each feature, and the estimation method using e-SVR based on top-n-features that are ordered by correlation. We identify that the correlation values between features and TV ratings vary according to the time of postings – before and after the broadcast time. This implies that the relevance of features can change according to the time of the tweets. Experimental results indicate that the proposed method has better performance compared with the method based on count-based features. This result implies that understanding the time-dependency of features can be helpful in improving the accuracy of measuring TV ratings.

key words: *time awareness, microblog, TV ratings*

1. Introduction

Social media has been a platform in which users can express their opinions, emotions, and experiences, as well as communicate with each other. In recent years, the use of microblog, such as Twitter, has been gaining popularity due to its ease of use, speed, and reach. It has the power to deliver and diffuse specific information to other people [1]. There has been much research on capturing social trends with microblogs; however, measuring television ratings has been given little attention so far [2].

Microblogging services allow users to publish brief messages and tag them with keywords. Allowing its users to conveniently share daily updates, microblogging helps people keep in touch [3]. Because users can connect to microblogs at any time and place via mobile device, they send tweets instantly as soon as they intend to express their emotions or opinions. Java, Song, Finin and Tseng [4] found that people use microblogging in order to talk about their daily activities as well as to seek or share information. In a similar way, microblog users are interacting with each other before and after watching TV or movies. As an active node with communication and distribution capabilities, the television viewer might want to communicate with others while watching in order to leave notes and comments for friends at specific moments of a television show [5]–[7].

TV ratings are important for content providers since the price of advertising is calculated based on it. Thus, content providers or TV stations want to monitor current trends of TV programs. In addition, many TV programs and stations use microblogs as their marketing channels. By leveraging social traces of microblogs, we can estimate ratings of TV programs. Based on these estimations, TV stations can plan a campaign for the shows and update their content to attract more viewers.

For measuring the TV ratings, we focus on the time when a feature is found. Some features can be significant in certain hours of the day or days of the week, whereas they are meaningless in other time periods. Because the importance of features can change during the day, the model capturing the time-changing relevance is required in order to estimate TV ratings more accurately. From this perspective, we develop an algorithm for finding the optimal time period when features are used; moreover, we adopt the Support Vector Regression for estimating TV ratings. We demonstrate that capturing the time awareness of features is vitally necessary for improving the estimation accuracy.

2. Methodology

2.1 Time-Aware Feature Selection

One of the most important parts of our method is the time-awareness of features, particularly the time when the words of tweets are used. The use frequency of each word in a tweet may differ according to the hour of day. For example, "wish to watch" is used before broadcasting, whereas "is fun" is used after broadcasting in most cases. In this study, we aim to find features with the optimal time period where the relevance to TV rating is maximized.

In order to prepare the input data for the algorithm, we extract morphemes from the chatter of tweets and use them as candidate features. The microblogs that we use as the data source are in Korean; thus, our feature extraction is based on a Korean corpus. All the features, f_i, extracted from the tweet data are added to set S_{init}.

To uncover the relationship between a specific term and TV ratings, we divide a week into 168 (7×24) time slots, and each time slot corresponds to one hour in a day of the week. However, calculating the relevance of every feature

Manuscript received March 3, 2014.
Manuscript revised June 3, 2014.
[†]The author is with Sejong University, Seoul, South Korea.
[††]The author is with The Catholic University of Korea, Bucheon, South Korea.
[†††]The author is with Dongduk Women's University, Seoul, South Korea.
a) E-mail: hongjoo@catholic.ac.kr (Corresponding author)
DOI: 10.1587/transinf.2014EDL8036

Input:
S_{init} = {f_i // all the extracted terms from the chatter of twitter}
n // total number of terms
h_{day} // the most-tweeted hour for each day of the week

Construct S_{day}:
for i=1 to n **do**
for j=1 to 7 **do**
calculate the frequency of f_i in the j-th day of the week
calculate $Corr(\boldsymbol{f}_{i,j})$
if Calculate $Corr(\boldsymbol{f}_{i,j}) > \delta_{corr}$
Add $f_{i,j}$ to S_{day}
$S_{day} = \{ f_{i,j} \mid f_i \in S_{init}, Corr(\boldsymbol{f}_{i,j}) > \delta_{corr} \}$

Construct S_{hour}:
for s=h_{day}-5 to h_{day}+5 **do**
calculate $Corr(\boldsymbol{f}_{i,j,s})$
add $f_{i,j,s}$ with the highest $Corr(\boldsymbol{f}_{i,j,s})$ to S_{hour}
$S_{hour} = \{f_{i,j,s} | f_{i,j} \in S_{day}, Corr(\boldsymbol{f}_{i,j,k})$ is highest for each $f_{i,j}\}$

Construct S_{final}:
for all $f_{i,j,s}$ in S_{hour} **do**
a,b ← s
$C_{old} \leftarrow 0$
$C_{new} \leftarrow Corr(\boldsymbol{f}_{i,j,s})$

for ($C_{old} < C_{new}$) **do**
$C_{old} \leftarrow C_{new}$
if $(Corr(\boldsymbol{f}_{i,j,s-1,e}) > Corr(\boldsymbol{f}_{i,j,s,e+1})$ **then**
$C_{new} \leftarrow Corr(\boldsymbol{f}_{i,j,s-1,e})$
if($C_{new} < C_{old}$) **then**
Add $f_{i,j,s,e}$ to S_{final}
else
s ← s-1
else
$C_{new} \leftarrow Corr(\boldsymbol{f}_{i,j,s,e+1})$
if($C_{new} < C_{old}$) **then**
Add $f_{i,j,s,e}$ to S_{final}
else
e ← e+1
Ouput:
$S_{final} = \{f_{i,j,s,e} | f_{i,j,s} \in S_{hour}\}$

Fig. 1 Detailed algorithm for time-aware feature selection.

for all the 168 slots takes too much time; this may cause serious problems with respect to the performance of the learning algorithm. This requires an efficient feature selection algorithm in order to find a relevant subset.

Thus, we develop a correlation-based, time-aware feature selection algorithm, as shown in Fig. 1. It consists of three major parts. In the first part, it discovers the relevant day of the week of each feature and constructs a feature set S_{day}. In order to select good features, we use Pearson correlation as the goodness measure. The role of this stage is to remove any irrelevant features and find the relevant day of the week for the features, which fulfill the requirement of the minimum correlation threshold. We denote by $\boldsymbol{f}_{i,j}$ the m dimensional vector containing the frequency of feature f_i in j-th day of k-th week for all the training data; $\boldsymbol{t}$ indicates the m dimensional vector containing TV ratings at k-th week. The Pearson correlation between $\boldsymbol{f}_{i,j}$ and $\boldsymbol{t}$ is given by the formula:

$$Corr(\boldsymbol{f}_{i,j}) = \frac{\sum_{k=1}^{m}(f_{i,j,k} - \overline{f})(t_k - \overline{t})}{\sqrt{\sum_{k=1}^{m}(f_{i,j,k} - \overline{f})^2}\sqrt{\sum_{k=1}^{m}(t_k - \overline{t})^2}} \quad (1)$$

where $\overline{f}$ is the mean of $f_{i,j,k}$ and $\overline{t}$ is the mean of t_k. Since j-th day of the week means the day of the week, j can range from 1 to 7. Subsequently, we exclude the feature whose correlation is lower than δ_{corr} (and p-value is above than 0.05), and construct a candidate feature set S_{day}.

$$S_{day} = \{f_{i,j} \mid f_i \in S_{init},\ Corr(\boldsymbol{f}_{i,j}) > \delta_{corr}\}$$

The second part searches for the most relevant time slot during the day. For each day of the week, we find the hour when the most tweets are posted. If we denote it as h_{day}, we choose a candidate period as from $h_{day} - 5$ to $h_{day} + 5$. As a result, a total of 11 time slots are selected as a candidate for each features in S_{day}. The time slot having the highest correlation is denoted as s; then, the selected features $\boldsymbol{f}_{i,j,s}$ are added to the feature set S_{hour}.

$$S_{hour} = \{f_{i,j,s} \mid f_{i,j} \in S_{day},\ Corr(\boldsymbol{f}_{i,j,s}) \text{ is highest for } f_{i,j}\}$$

The third part looks for the time period when the relevance of feature is maximized. In order to represent the period, we need notations s for the start time and e for the end time of the period. We design an algorithm that searches for the optimal time period using an incrementally growing window starting from the time slot in S_{hour}. The overall algorithm is described in Fig. 1.

After finding the feature set S_{final}, the features are ordered in descending order according to the correlation in order to decide the best-n-features. Then, the selected features are used as input data for training SVM (Support vector machine). The SVM, which recently has been widely utilized in the field of machine learning, is used for estimating TV ratings. As the SVM provides a powerful kernel function that can identify the structural features of data, it is known to provide a more reliable estimation.

In this study, ε-SVR (epsilon-Support Vector Regression) is used to identify a non-linear relationship between a target value and input feature, and the Radial Basis Function (RBF) was used as a kernel function.

If we sort features in S_{final} in descending order of $Corr(f_{i,j,s,e})$, each feature can be mapped to $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots \mathbf{x}_m\}$. Using this rearranged features, we prepare the training data set $\{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \ldots, (\mathbf{x}_m, t_m)\}$ with $x_i \in \boldsymbol{R}^n$ and $t_i \in \boldsymbol{R}^1$. Given a training set, eSVR require the solution of the following optimization problem:

$$\begin{aligned} \min_{\omega,b,\xi,\xi^*} \quad & \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n}\xi_i + C\sum_{i=1}^{n}\xi_i^* \\ \text{subject to} \quad & \omega^T\varphi(x_i) + b - t_i \le \varepsilon + \xi_i \\ & t_i - \omega^T\varphi(x_i) - b \le \varepsilon + \xi_i^* \end{aligned} \quad (2)$$

$$\xi_i,\ \xi_i^* \geq 0,\ i = 1, \ldots, n.$$

Usually, the above problem is converted into the following dual problem due to the high dimensionality of the variable w.

$$\begin{aligned} \min_{\alpha,\alpha^*} \quad & \frac{1}{2}(\alpha - \alpha^*)^T Q(\alpha - \alpha^*) + C\sum_{i=1}^{n}(\alpha_i - \alpha_i^*) \\ & + C\sum_{i=1}^{n} z_i(\alpha_i - \alpha_i^*) \qquad (3) \\ \text{subject to} \quad & e^T(\alpha - \alpha^*) = 0 \\ & 0 \leq \alpha_i,\ \alpha_i^* \leq C,\ i = 1, \ldots, n. \end{aligned}$$

where $Q_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) \equiv \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$.

After solving the above problem, the approximate function is

$$\sum_{i=1}^{n}(-\alpha_i + \alpha_i^*)K(x_i, x) + b \qquad (4)$$

Using the above function, we estimate the TV ratings.

3. Experimental Results

We selected two variety shows and one TV drama which were very popular nationwide for experimental evaluation. We collected data of the three TV programs using Twitter search API v1.1 and evaluated our method on the data set. We used the name of each program as the search keyword once per 10 minutes and gathered 776,544 tweets from January to October 2013.

3.1 Feature Analysis

Using our proposed algorithm, we extracted about 300 time-aware features for each program and added them to S_{final}. Each feature has a start hour s and an end hour e of its own time period. Figure 2 illustrates the distribution of the time period, in which we can see that the number of features is increasing around the broadcast time.

3.2 Evaluation

The performance of the proposed methodology was evaluated using the MSE (Mean Squared Error). Because MSE measures the average difference between true TV ratings and the estimated TV ratings implied by our method, the lower the calculated value is, the better is the estimation accuracy. In this study, a 10-fold cross-validation was used to evaluate the estimation accuracy.

The best-n-features are selected in descending order of the correlation and used as input data for ε-SVR. We conduct two experiments to verify the value of time-aware features for each TV program. The frequency of feature during the optimal time period is used as input data in experiment 1, whereas the frequency of feature during all day is used in experiment 2. Figure 3 shows the average MSE of the two experiments for each TV program.

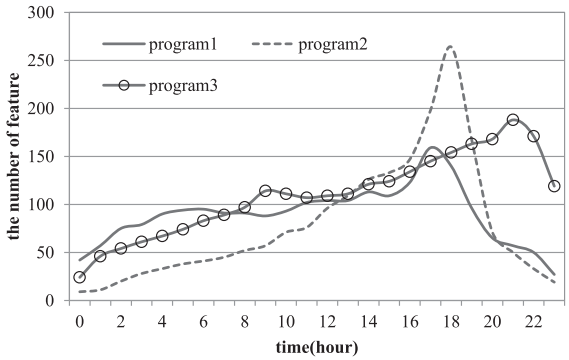

Fig. 2 Time period distribution of features in S_{final}.

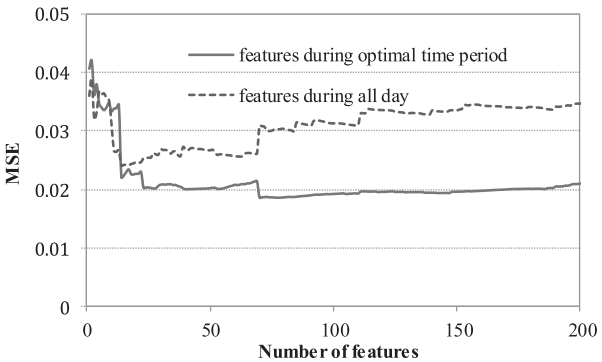

Fig. 3 Performance evaluation according to the number of features.

Table 1 Performance comparison of proposed method.

Category	Input	Program1	Program2	Program3
Count-based features	The number of Tweets	0.0135	0.0845	0.0148
	The number of Retweets	0.0212	0.0914	0.0153
	The number of Users	0.0222	0.0601	0.0154
	The number of Tweets+Retweets+Users	0.0186	0.0741	0.0152
Content-based features	TF-IDF	0.0132	0.0583	0.0165
	Features without optimal time period	0.0107	0.0451	0.0148
	Features with optimal time period	0.0087	0.0305	0.0129

The experimental result revealed that our proposed method of time-aware features provides better performance than the method of not using the optimal time period. The MSE of the time-aware method decreases as the number of features increases; however, they rise and converge on a certain value. Having too many features engenders a great deal of noise for estimating TV ratings.

We compared the performance of our method with the method using count-based features, where we adopt the frequency rate of the tweets, retweets containing the keyword and the number of users. In addition, a comparison with

TF-IDF method which is used for extracting important words was performed. Table 1 illustrates the experimental result in the comparison for the three TV programs. The experimental results indicate that time-aware features have better performance than count-based features and the extracted feature using TF-IDF. Among the methods based on content-based features, optimal time period method shows the best performance.

4. Conclusion

In this paper, we propose a novel approach for estimating TV ratings by using time-aware features extracted from microblogs. We design a new feature selection algorithm that finds the optimal time period of each feature as well as the estimation method using e-SVR, based on top-n-features that are selected by correlation. Our method proves its effectiveness in using the time-dependent features for measuring the response of the TV audience. The experimental results show that the proposed method has better performance and lower MSE compared with the method based on count-based features. Our results provide a suggestion on how to use the timing information of a microblog, such as the hour of day and day of the week, in predicting or estimating a social phenomenon. Future research will extend the method to work on other domain data.

References

[1] Y. Nam, I. Son, and D. Lee, "The impact of message characteristics on online viral diffusion in online social media services: The case of twitter," J. Intelligence and Information Systems, vol.17, no.4, pp.75–94, 2011.

[2] S. Wakamiya, R. Lee, and K. Sumiya, "Towards better TV viewing rates: Exploiting crowd's media life logs over twitter for TV rating," Proc. 5th International Conference on Ubiquitous Information Management and Communication, Paper 39, 2011.

[3] O. Günther, H. Krasnova, D. Riehle, and V. Schöndienst, "Modeling microblogging adoption in the enterprise," Proc. 15th AMCIS, Paper 544, 2009.

[4] A. Java, X. Song, T. Finin, and B. Tseng, "Why we twitter: Understanding microblogging usage and communities," Proc. 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis, pp.56–65, San Jose, California, 2007.

[5] D. Williams, M.F. Ursu, P. Cesar, K. Bergström, I. Kegel, and J. Meenowa, "An emergent role for TV in social communication," Proc. Seventh European Conference on European Interactive Television Conference, pp.19–28, 2009.

[6] N. Ducheneaut, R.J. Moore, L. Oehlberg, J.D. Thornton, and E. Nickell, "Social TV: Designing for distributed, sociable television viewing," Int. J. Human–Computer Interaction, vol.24, no.2, pp.136–154, 2008.

[7] M. Nathan, C. Harrison, S. Yarosh, L. Terveen, L. Stead, and B. Amento, "Collaboratv: Making television viewing social again," Proc. 1st International Conference on Designing Interactive User Experiences for TV and Video, pp.85–94, 2008.