Twitter Hash Tag Prediction Algorithm

Tianxi Li and Yu Wu
Department of Computer Science
Stanford University
Stanford, CA 94305, USA
(tianxili, ywu2}@stanford.edu

Yu Zhang
Department of Computer Science
Trinity University
San Antonio, TX 77812, USA
yzhang@trinity.edu

ABSTRACT

Social media has demonstrated quick growth, in both directions of becoming the most popular activities in internet and of attracting scientific researchers to get better insights into the understanding into the underlying sociology. Real time micro-blogging sites such as Twitter, Flickr and Delicious use tags as an alternative to traditional forms of navigation and hypertext browsing. The tag system of those micro-blogging sites has unique features in that they change so frequently that it is hard to identify the number of clusters and so effectively carry out classification when new tags can come out at any time. In this paper, we propose to use Euclidean distance between points as the measurement of their similarity. Our method has advantages in easy data storage and easy accommodation to personal settings. In experiment, we compare our model with other classification functions and show that our model maintains a false positive rate lower than 15%. Our work is relevant for researchers interested in navigating of emergent hypertext structures, and for engineers seeking to improve the navigability of social tagging systems.

KEYWORDS

Social Networks, Twitter, Hash Tags, Social Tag Prediction, Ontology Based Distance

1 INTRODUCTION

Social media has demonstrated exponential growth, making it the most popular activity on the internet [1]. Real-time micro-blogging services such as Twitter, Flickr and Delicious are widely recognized for their social dynamics – how they both encapsulate a social setting propagate information across it [2, 3, 13, 14, 15, 21].

Social tagging [26, 29] is a method for Internet users to organize, store, manage and search for resources online [9, 12]. Trant [28] categorizes the existing works on social tagging into three broad topics: (a) on the *folksonomy* that results from the collective wisdom of users of the social tagging system; (b) on the *tagging behavior* of users, such as the incentives and motivation for tagging; (c) on the software aspects of the *social tagging systems*, for improving system performance and enhancing user satisfaction.

Social tag prediction belongs to the third category. In particular, it aims at enriching tags for Web resources that are untagged or inadequately tagged [12]. The Internet users are benefited by this technique because it make search in webs become easy [17]. The current research in this direction can be classified into three categories: (1) determining topics from hypertext content [23], (2) predict new trend on topics based on existing tags [5, 10, 17, 18], and (3) enriching tags from other similar or linked resources [4, 22, 27.

Twitter is one of popular web applications nowadays [19, 20, 25]. Twitter allows users to use “Hash tags” to classify their tweets. In this research project, we propose an algorithm to predict tags, by utilizing machine learning and network relatedness methods.

Hash tag prediction is different from normal texts classification mentioned in the above. In a real time micro-blogging site, we don’t know how many clusters needed to be found. In addition, the tag set changes so frequently that it is almost impossible to effectively carry out classification or clustering, since a new tag would force us to establish a new class and a new classification rule. Our intuition is: if we can measure the correlation between various tweets as the mathematical metric we can treat the collected tweets as points in a high dimensional space, and construct a network by the latent space model. We show that simple techniques are sufficient to extract key semantic content from tags and also filter out extraneous noise. We demonstrate the efficacy of this approach by comparing it with other classification functions and show that our model maintains a false positive rate lower than 15%.

The paper is structured as follows: In Section 2 we briefly introduce Twitter and its hash tag system. Section 3 presents our theoretic approach to assessing distance of tagging systems. We propose to use the ontology based distance between points as the measurement of their similarity. Our method has advantages in easy data storage and easy accommodation to personal settings. Section 4 presents and discusses the analysis results. Section 5 concludes the paper.

2 TWITTER

Twitter is one of the fastest growing Web 2.0 services. It is called a micro-blog because people can post short, quasi-public messages up to 140 characters in length. People create lists of others and are shown a list of all of the posts of those people. The substantive nature of the social tie on Twitter is attention-based [7, 8]. In addition to paying attention to one another by "following," Twitter users can address tweets to other users and can mention others obliquely in their tweets [11]. Another common practice is "retweeting," or rebroadcasting someone else's message (with attribution) so as to direct attention toward that person's tweets [1].

Twitter differs from other online social networking services in that ties are asymmetric [7, 8]. Consider friendship ties in LinkedIn, Facebook, or MySpace; in these services, when two people share a friendship tie, the tie is symmetrical; A being friends with B implies B is friends with A. This is not the case in Twitter; A can "follow" B, but B needs not follow A. People who are popular, such as basketball players or actors, can be followed by millions of people, but can barely pay attention to all of those who follow them.

The hash tag (the # sign followed by a phrase to a tweet, for example #superbowl) is probably the most important function of Twitter search, and the most used. The hash tag enables Twitter users to create searchable subject groups and so to be able to navigate the hypertext structures of the whole site. The power of the hash tag is that it creates very specific sets of content. If you want to know what other people think of the superbowl that just came on you can find it easier by searching for the hash tag than by searching for something similar in a normal search engine. Every day, many new hash tags are formed and this process can happen right before your eyes-heck. The frequent creation of new tags makes the prediction of tags challenging. This motivates us to develop the following method.

3 METHOD

3.1 Theory

An intuitive way to solve this problem is to use Euclidean distance between points as the measurement of their similarity. We developed our theory based on this distance. Since in a Euclidean Space, the distance is equivalent to the norm of a vector, we will focus our discussion on norms.

Let $\mathbf{u}_1, \mathbf{u}_2 \cdots \mathbf{u}_p$ be the standard bases (with unit norm) of a *p*-dimensional Euclidean Space. Then for any vector $\mathbf{v}$ with coordinates (x_1, x_2, …, x_{p-1}, x_p), we have $\mathbf{v} = \sum_{i=1}^{p} x_i \mathbf{u}_i$. Then the Euclidean norm of vector $\mathbf{v}$ is given by:

$$\|\mathbf{v}\|^2 = \mathbf{v}\cdot\mathbf{v} = \sum_{i=1}^{p} x_i \mathbf{u}_i \cdot \sum_{i=1}^{p} x_i \mathbf{u}_i = \sum_{i,j=1}^{p} x_i x_j \mathbf{u}_i \cdot \mathbf{u}_j \quad (1)$$

where · represents the inner product operation defined in the Euclidean Space. Clearly, if we assume $\mathbf{u}_i \cdot \mathbf{u}_j = 0$, that is, $\mathbf{u}_i$ and $\mathbf{u}_j$ are orthogonal, whenever *i≠j*, the Euclidean norm equals to $\|\mathbf{v}\|^2 = \sum x_i^2$. In our problem, the bases are the words in the dictionary. The preliminary assumption for Euclidean distance is that the bases are orthogonal to each other, that is, the words in dictionary are uncorrelated, which is against common sense. Therefore, we need to perform some transformation to capture this correlation.

In Equation (1), as $\mathbf{u}_i$ and $\mathbf{u}_j$ are unit vectors, their inner product is actually the cosine of the angle between them. Thus we can rewrite (1) in a matrix form as:

$$\|\mathbf{v}\|^2 = \begin{pmatrix} x_1 & \cdots & x_p \end{pmatrix} \begin{pmatrix} \cos\theta_{11} & \dots & \cos\theta_{1p} \\ \vdots & \ddots & \vdots \\ \cos\theta_{p1} & \cdots & \cos\theta_{pp} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix} = XMX^T \quad (2)$$

where $\cos\theta_{ii} = 1$, i=1, …, p, and x = (x_1, x_2, …, x_{p-1}, x_p).

Now we try to find the angle between each pair of terms in the dictionary and then calculate the matrix M . Notice that M is clearly a symmetric and non-negative definite matrix. If we decompose M in the way

$$M = CC^T \quad (3)$$

then (2) becomes

$$\|\mathbf{v}\|^2 = XCC^T X^T = \tilde{X}\tilde{X}^T \quad (4)$$

where $\tilde{X} = XC$. So the norm can be seen as the Euclidean norm of the transformed coordinates. Here we take (3) as the Eigen value decomposition of *M*, so $\tilde{X}$ could be the coordinates of vector $\mathbf{v}$ in a new coordinate system where axes are orthogonal to each other. Please note that we can use any other decomposition in the form of (3) to get the same norm in computation, even when *C* is not a square matrix. With this property, the computation becomes applicable.

3.2 Estimate the Cosine Matrix

First, we construct the preliminary weighted matrix, say *H*, by using the WordNet to initialize the semantic correlation among words from the dictionary. If two words t_i, t_j are similar to each other, and they both appear in one Tweet, we add positive weights for both words. This process can be expressed as

$$\hat{x}_i = x_i + \sum_{j \neq i}^{p} \rho_{ij} x_j \qquad (5)$$

where $\rho_{ij} \in (0,1)$, equals to one when t_i, t_j are similar words and zero otherwise. Here we take the same positive number ρ for all $\rho_{ij} \in (0,1)$, and if $\rho_{ij} > 0$, so is ρ_{ji}. Then we can construct the symmetric matrix *H* as:

$$\mathrm{H} = \begin{pmatrix} 1 & \cdots & \rho_{1p} \\ \vdots & \ddots & \vdots \\ \rho_{p1} & \cdots & 1 \end{pmatrix} \qquad \hat{X} = X\mathrm{H} \quad (6)$$

In the second step, we get m tweets, say $X_1 \cdots X_m$, and transform them by (4) to get $\hat{X}_1 \cdots \hat{X}_m$. Then by these data, we use cosine similarity in variable analysis to construct matrix M. Set the text matrix as the $m \times p$ matrix :

$$\Omega = (\hat{X}_1^T \quad \cdots \quad \hat{X}_m^T)^T = \begin{pmatrix} \hat{x}_{11} & \cdots & \hat{x}_{1p} \\ \vdots & \ddots & \vdots \\ \hat{x}_{m1} & \cdots & \hat{x}_{mp} \end{pmatrix}. \qquad (7)$$

We would estimate the cosine between the i^th and j^th terms as

$$\cos\theta_{ij} = \hat{X}_{\bullet i}^T \hat{X}_{\bullet j} \Big/ \left\|\hat{X}_{\bullet i}\right\| \left\|\hat{X}_{\bullet j}\right\| = \frac{\sum_{k=1}^{m} \hat{x}_{ki}\hat{x}_{kj}}{\sqrt{\sum_{k=1}^{m} \hat{x}_{ki}^2 \sum_{k=1}^{m} \hat{x}_{kj}^2}} \qquad (8)$$

The distance estimate obtained from formula (5) is equivalent to what proposed by [16], but with a better mathematical explanation. Note that since our data is represented as frequency, all the elements of the matrix Ω would be non-negative. So the cosine estimated in this way can only be non-negative. Therefore, all angles between words are cute or right angles. In this way, all words tend to be similar to each other in some degree. This may well incorporate the similarity elements, but might also be vulnerable to noise. In the following, we give a modified estimate which also includes the possibility of obtuse angle and takes dissimilarity into consideration, which is also the sample correlation in statistics,

$$\cos\theta_{ij} = \frac{\sum_{k=1}^{m} (\hat{x}_{ki} - \hat{x}_{\bullet i})(\hat{x}_{kj} - \hat{x}_{\bullet j})}{\sqrt{\sum_{k=1}^{m} (\hat{x}_{ki} - \hat{x}_{\bullet i})^2 \sum_{k=1}^{m} (\hat{x}_{kj} - \hat{x}_{\bullet j})^2}} \qquad (9)$$

Since the distance from formula (8) was named as Ontology Based Distance (OBD) in [16], here we call the distance in formula (9) centralized Ontology Based Distance (COBD). We will discuss the pros and cons of the two methods in experiments. In the following sub section, we will make another adjustment to the method.

3.3 Normalization

Note that the various scales of vectors may still cause us some problem. Consider a special case where $\tilde{X}_1 = (1,0,0)$, $\tilde{X}_2 = (10,0,0)$, $\tilde{X}_3 = (0,0,1)$. Obviously, $\tilde{X}_1$ and $\tilde{X}_2$ should have high similarity value between them. But in this case, the distance between $\tilde{X}_1$ and $\tilde{X}_3$ is much smaller.

To make our method more reasonable, before we compute the distance between transformed points, we need to rescale their distances to the original point as 1. And then we measure the Euclidean distance between normalized points.

3.4 Prediction of Tags

Finally, we predict tags based on the distance. One intuitive way is to simply select the tag of the closest tweet. In this case, it may be unwise to simply pick the closest tweet's tag, since that is not resistant to noise. To increase the accuracy, we collect a few closest tweets, and make the prediction based on tag ratios. Specifically, we will collect *n* initial closest tweets at first (*n* usually ranges from 4 to 6). Then from this point, we will keep adding tweets while check a certain tag has become dominate. If there is a tag with a ratio higher than 50%, we will choose this tag as our primary predicted tag. Since in some cases tags have very similar meanings (such as #government vs. #election), sometimes we will also pick a secondary tag to predict.

4 EVALUATIONS

To compare the performances of various distances discussed above, we use a test dataset consist of 400 tweets that are not included in the sample set we used to estimate matrix M. There are 4 different tags. We first process the OBD on a dataset with 665 tweets that are not in our test set, choose the best performance ρ (=0.2) and use it for both OBD and COBD. The table below shows the test result for Euclidean Distance (EucD), OBD and COBD.

	Test Error Rate	Type II Error
EucD	16.25%	5.1%
COBD	13.5%	4.6%
OBD	12.75%	4.2%

Table1: The test error rate and type II error for three distances. Type II error is the rate we assign a wrong tag to a particular tweet.

Both OBD and COBD outperform EucD, and OBD is the best one. If we see the data for different tags (not provided here for concise), we would find COBD is the most stable one, while EucD is far more unstable. But the disadvantage of COBD lies in computation. We need to estimate the cosine matrix M to construct the distance, which involves computation for matrices with tens of thousands rows and columns. It won't be a big problem for OBD since the matrices are sparse. But in COBD, the matrix becomes non-sparse, so we need many decompositions and transformations of matrices to make the computation applicable. Given their close performances, OBD is more practical in application, while the COBD is a better model theoretically.

The top picture in Figure 1 shows the COBD from other tweets to a random selected tweet. Different colors represent tweets with different tags. It can be seen that most of the tweets are very close to the 1.4142 distance boundary, and the majority of points falling in the circle are from the correct tag group. This indicates that tweets with different topics are projected onto orthogonal axes. The right plot illustrates the distance distribution. The lighter the color is, the shorter the corresponding distance is. Since the tweets are sorted by tags, we see that the distance within each group appears to be shorter, as shown by the light rectangles along the diagonal.

In Figure2, different colors represent what tag cluster the tweets belong to. A link will be added between a pair of nodes when they are near enough. In addition, the deeper color the line is, the higher the similarity value is. As we can see, the lines appear to be very dense among each tag cluster, and sparse between tweets with different tags. It indicates that tweets with the same tag cluster are near on average.

Due to the vagueness of many tweets, the correct rate of more than 86% is actually very high. Apart from the accuracy, our method has other advantages:

(1) The whole system is easy to store (we only need to store the *C* matrix in Equation (3)).

(2) It is easy to update when dictionary changes (only needs to compute an extra column and add it back to original matrix).

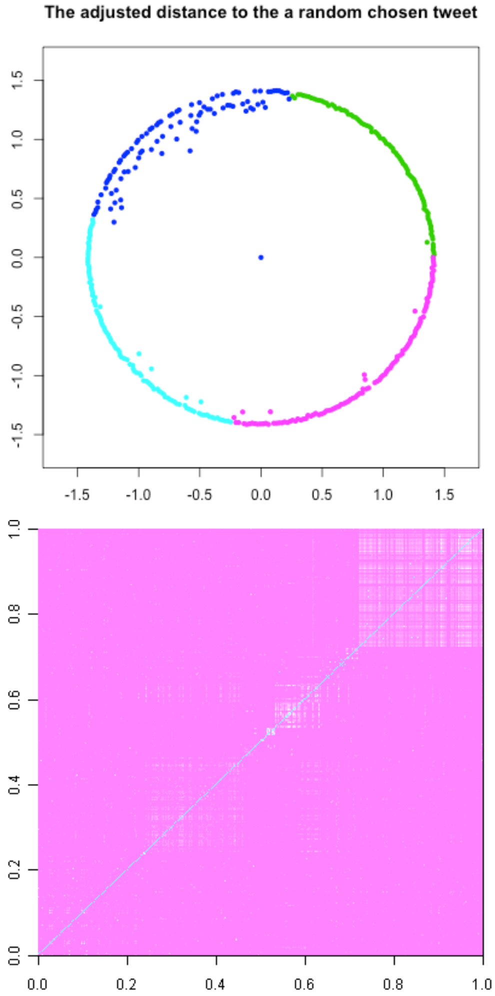

Figure 1 Distance to one point and the distribution of sample distance matrix

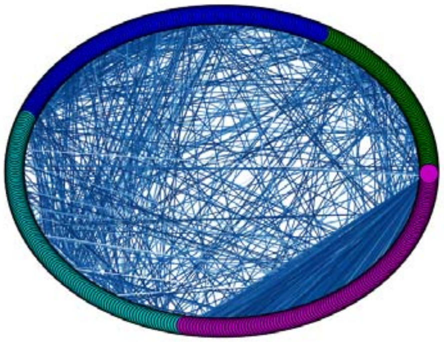

Figure 2 Prediction Visualization

(3) It won't lose power when the topics trend changes with time, and it can work with personal elements and settings, which makes it more flexible (since we can set the algorithm to only consider the distance of the objective tweet to certain subset of

other tweets, so elements like location, time, etc can be incorporated.)

(4) In addition, the distance provides us with the possibility to transform the twitter system and even other text systems into social networks by latent space approach. So we can use traditional social network methods to discuss the properties of such systems.

5 CONCLUSION

In this paper, we have presented a distance function to classify hash tags in Twitter. A major challenge to the social tag prediction problem with a micro-blog like Twitter is that the underlying dataset is updated frequently by millions of the Twitter online users. We propose a distance function that utilizes machine learning technology and latent space models. We map the collected tweets to a high dimensional space and construct a latent network to predict the similarity of these tags. Our model is general in terms of that it allows the flexibility of adapting users' personal settings. We show that simple techniques are sufficient to extract key semantic content from tags and also filter out extraneous noise.

REFERENCES

[1] Adar E, Adamic LA, Tracking Information Epidemics in Blogspace, 2005 IEEE/WIC/ACM International Conference on Web Intelligence, pp.207-214, 2005.

[2] Banerjee N, Chakraborty D, Dasgupta K, Mittal S, Joshi A, Nagar S, Rai A and Madan A, User Interests in Social Media Sites: An Exploration With Micro-blogs, the 18th ACM Conference on Information and Knowledge Management, 2009.

[3] Boyd D, Golder SA, and Lotan G, Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter. Proc. HICSS-43., 2010.

[4] [1] Budura A, Michel S, Cudr´e-Mauroux P, and Aberer K, To Tag Or Not to Tag -: Harvesting a Djacent Meta Data in Large-Scale Tagging Systems, SIGIR, pp. 733–734, 2008.

[5] Bundschus M, Yu S, Tresp V, Rettinger A, Dejori D, and Kriegel HP, Hierarchical Bayesian Models for Collaborative Tagging Systems, ICDM, pp. 728–733, 2009.

[6] Cha M, Mislove A, Adams B, Gummadi KP, Characterizing Social Cascades in Flickr, the First Workshop on Online Social Networks, 2008.

[7] Golder SA and Yardi S, Structural Predictors of Tie Formation in Twitter: Transitivity and Mutuality, IEEE International Conference on Social Computing, pp. 88 - 95, 2010.

[8] Golder SA, and Lotan G, Tweet, Tweet, Tetweet: Conversational Aspects of Retweeting on Twitter, HICSS-43, 2010.

[9] Helic D, Trattner C. Strohmaier M, and Andrews K, On the Navigability of Social Tagging Systems, IEEE International Conference on Social Computing, pp. 161 – 168, 2010.

[10] Heymann P, Ramage D, and Garcia-Molina H, Social Tag Prediction, SIGIR, pp. 531–538, 2008.

[11] Honeycutt C and Herring SC, Beyond Microblogging: Conversation and Collaboration Via Twitter. Proc. HICSS-42, 2009.

[12] Hu M, Lim EP and Jiang J, A Probabilistic Approach to Personalized Tag Recommendation, IEEE International Conference on Social Computing, pp, 33 – 40, 2010.

[13] Huberman B., Romero D, and Wu F, Social Networks that Matter: Twitter Under the Microscope, First Monday 14, 1-5, 2009.

[14] Jackson M and Yariv L, Diffusion on Social Networks, Economie Publique, 16: 3-16, 2005.

[15] Java M, Song X, Finin T and Tseng B, Why We Twitter: Understanding Microblogging Usage and Communities, the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis, pp.56-65, 2007.

[16] Jing L, Ng MK, Yang X, and Huang J, A Text Clustering System based on *k*-means Type Subspace Clustering and Ontology, International Journal of Intelligent Technology, 1(2), 2006.

[17] Krestal R, Fankhauser P, and Nejdl W, Latent Dirichlet Allocation for Tag Recommendation, RecSys '09, pp. 61–68, 2009.

[18] Krestel R and Fankhauser P, Tag Recommendation Using Probabilistic Topic Models, ECML PKDD Discovery Challenge, (497): 131–141, 2009.

[19] Krishnamurthy B, Gill P, Arlitt M, A Few Chirps About Twitter, the First Workshop on Online Social Networks, 2008.

[20] Kwak H, Lee C, Park K and Moon S, What Is Twitter, A Social Network or A News Media?, the 19th International Conference on World Wide Web, 2010.

[21] Lerman K and Galstyan A, Analysis of Social Voting Patterns on Digg, the First Workshop on Online Social Networks, 2008.

[22] Lu YT, Yu SI, Chang TC, and Hsu JY, A Content-Based Method to Enhance Tag Recommendation, IJCAI, pp. 2064–2069, 2009.

[23] Murfi H and Obermayer K, A Two-Level Learning Hierarchy of Concept Based Keyword Extraction for Tag Recommendation, ECML PKDD Discovery Challenge, (497): pp. 201–214, 2009.

[24] Plickert G, Cote RR and Wellman B, It's Not Who You Know, It's How You Know Them: Who Exchanges What With Whom? Social Networks, 29:405–429, 2007.

[25] Romero DM and Kleinberg J, The Directed Closure Process in Information Networks with An Analysis of Link Formation on Twitter, the International Conference on Weblogs and Social Media, 2010.

[26] Szabo G. and Huberman B, Predicting the Popularity of Online Content, Communications of the ACM, 2008.

[27] Subramanya SB and Liu H, Social Tagger - Collaborative Tagging for Blogs in the Long Tail, SSM 2009.

[28] Trant J, Studying Social Tagging and Folksonomy: A Review and Framework, Journal of Digital Information, 10(1): 1 - 44, 2009.

[29] Wasserman S and Faust K, Social Network Analysis: Methods and Applications. Cambridge University Press, 1994.