Analyzing Microblogs with Affinity Propagation

Jeon Hyung Kang
USC Information Sciences Institute
Marina del Rey, CA 90292, USA
jeonhyuk@usc.edu

Kristina Lerman
USC Information Sciences Institute
Marina del Rey, CA 90292, USA
lerman@isi.edu

Anon Plangprasopchok
USC Information Sciences Institute
Marina del Rey, CA 90292, USA
plangpra@isi.edu

ABSTRACT

Recently, there has been a great deal of interest in analyzing inherent structures in posts on microblogs such as Twitter. While many works utilize a well-known topic modeling technique, we instead propose to apply Affinity Propagation [4] (AP) to analyze such a corpus, and we hypothesize that AP may provide different perspective to the traditional approach. Our preliminary analysis raises some interesting facts and issues, which suggest future research directions.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval —*Clustering*; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Measurement, Human Factors

Keywords

Microblogging, Social Media, Twitter

1. INTRODUCTION

The microblogging service Twitter allows users to broadcast short messages, *tweets*, to their followers. Millions of users have enthusiastically embraced Twitter, using the 140 character limit to express opinion, describe experiences, and spread ideas and information. The resulting flood of data can potentially be mined to discover the "buzz" about products, people, and events, discover emerging trends, and facilitate real-time information search. One of the key challenges that need to be solved is to identify related tweets that are about the same topic.

Traditional topic modeling techniques used in document analysis exploit statistics of term co-occurrences to identify groups of related terms in a topic. Twitter's 140 character limit on tweets presents a challenge to these techniques. To address this data sparseness problem, some researchers have proposed to aggregate over all of user's tweets before using a topic modeling method [9], or to utilize manual-labels to guide the topic modeling inference to group related terms in such a sparse environment [8].

In this work, we instead propose to apply Affinity Propagation (AP) [4] to cluster a corpus of tweets. AP allows objects, in our case tweets, to choose exemplars that best represent them. A group of objects that have chosen the same exemplar can be considered to be in the same cluster, and therefore, on the same topic. The exemplar, then, is a tweet that best represents the topic. AP uses a similarity metric to evaluate how well the exemplar describes the object.

Affinity Propagation has several advantages over alternative clustering and topic modeling approaches. Traditional clustering algorithms greedily assign each object to the best cluster. AP, on the other hand, is a distributed clustering algorithm that finds the best assignment of all objects to clusters at the same time. Moreover, AP produces an exemplar that can best "summarize" the cluster. In Twitter data, exemplars can effectively compress the stream of data, reducing user's cognitive load in processing tweets. Once we find the grouping of tweets, we can also leverage the hashtags users created to label the tweets to suggest tags for the remaining tweets in the cluster.

Specifically, we describe an experiment that uses AP to cluster tweets that contain URLs to news stories. Users frequently tweet URLs to share interesting stories with their followers. These tweets are often *retweeted*, or re-broadcast by users who received the tweet to their own followers, often with a further comment or embellishment. Retweeting is used to spread a story to a wider audience than original poster had [3]. We extracted a small corpus of tweets from Tweetmeme (http://tweetmeme.com), a service that aggregates all Twitter posts to determine frequently retweeted URLs. Often there is more than one URL about the same story that is being retweeted by users. We use AP to cluster together related tweets and retweets.

2. TWEETS' CHARACTERISTICS

Twitter users are constrained to exhange messages using a flat text of 140 characters or less. With such limitation, many users typically use some structure conventions such as a user-to-message relation (i.e. initial tweet author, Replay, via, cc, by), type of message (i.e. Broadcast, conversation, or retweet messages), type of resources (i.e. URLs, hashtags, keywords). For example, the tweet "RT @xxx: Social Media to build Newspapers www.eee.bbb #newspaper" con-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
1st Workshop on Social Media Analytics (SOMA '10), July 25, 2010, Washington, DC, USA.
Copyright 2010 ACM 978-1-4503-0217-3 ...$10.00.

tains a type of message (retweet), a user-to-message relation (@xxx), a hashtag (#newspaper), an URL (www.eee.bbb), and words (Social Media to build Newspapers). In many cases, users retweet or reply to a tweet by altering the original message. In this paper, we only utilize message level contents, which include words, hashtag and URL.

3. AFFINITY PROPAGATION

Affinity Propagation [4] is a clustering algorithm that identifies a set of exemplar points that are representative of all the points in the data set. The exemplars emerge as messages are passed between data points, with each point assigned to an exemplar. AP attempts to find the exemplar set which maximizes the net similarity, or the overall sum of similarities between all exemplars and their data points.

In this paper, we describe AP in terms of a factor graph [7] on binary variables, as recently introduced by Givoni and Frey [5]. The model is comprised of a square matrix of binary variables, along with a set of factor nodes imposed on each row and column in the matrix. Following the notations defined in the original paper [5], let c_{ij} be a binary variable. $c_{ij} = 1$ indicates that node i belongs to node j (or, j is an exemplar of i); otherwise, $c_{ij} = 0$. Let N be the number of data points; consequently, the size of the matrix is $N \times N$.

There are two types of constraints that enforce cluster consistency. The first type, I_i, which is imposed on the row i, indicates that a data point can belong to only one exemplar ($\sum_j c_{ij} = 1$). The second type, E_j, which is imposed on the column j, indicates that if a point other than j chooses j as its exemplar, then j must be its own exemplar ($c_{jj} = 1$). AP avoids forming exemplars and assigning cluster memberships which violate these constraints. Particularly, if the configuration at row i violates I constraint, I_i will become $-\infty$ (and similarly for E_j).

In addition to the constraints, there is a similarity function $S(.)$, which indicates how similar a certain node is, to its exemplar. If $c_{ij} = 1$, then $S(c_{ij})$ is the similarity between nodes i and j; otherwise, $S(c_{ij}) = 0$. $S(c_{jj})$ evaluates "self-similarity," also called "preference", which should be less than the maximum similarity value in order to avoid all singleton points becoming exemplars. This is because that configuration yields the highest net similarity. In general, the higher the value of the preference for a particular point, the more likely that point will become an exemplar. In addition, we can assign the same self-similarity value to all data points, which indicates that all points are equally likely to be formed as exemplars.

A graphical model for affinity propagation is depicted in Figure 1, described in terms of a factor graph. In a log-form, the global objective function, which measures how good the present configuration (a set of exemplars and cluster assignments) is, can be written as a summation of all local factors as follows:

$$\begin{aligned}\mathbf{S}(c_{11},\cdots,c_{NN}) &= \sum_{i,j} S_{ij}(c_{ij}) + \sum_i I_i(c_{i1},\cdots,c_{iN}) \\ &+ \sum_j E_j(c_{1j},\cdots,c_{1N}). \qquad (1)\end{aligned}$$

That is, optimizing this objective function finds the configuration that maximizes the net similarity S, while not violating I and E constraints.

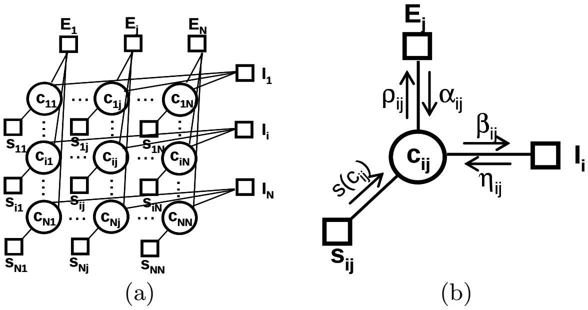


Figure 1: Binary variable model for Affinity Propagation proposed by Givoni and Frey[5]: (a) a matrix of binary hidden variables (circles) and their factors(boxes); (b)incoming and outgoing messages of a hidden variable node from/to its associated factor nodes.

The original work uses max-sum algorithm to optimize this global objective function, and it requires updating and passing five messages as shown in Figure 1(b). Since each hidden node c_{ij} is a binary variable (two possible values), one can pass a scalar message — the difference between the messages when $c_{ij} = 1$ and $c_{ij} = 0$, instead of carrying two messages at a time. The equations to update these messages are described in greater detail in the Section 2 of the original work [5].

Once the inference process terminates, the MAP configuration (exemplars and their members) can be recovered as follows. First, identify an exemplar set by considering the sum of all incoming messages of each c_{jj} (each node in the diagonal of the variable matrix). If the sum is greater than 0 (there is a higher probability that node j is an exemplar), j is an exemplar. Once a set of exemplars K is recovered, each non-exemplar point i is assigned to the exemplar k if the sum of all incoming messages of c_{ik} is the highest compared to the other exemplars.

3.1 Affinity Propagation for Analyzing Microblogs

In our setting, we treat each microblog post or tweet as a data point, and we wish to identify clusters of similar tweets. The similarity between tweets i and j, or $S(c_{ij})$, is determined from the textual similarity. In particularly, we simply use word frequencies of the tweets (weighted using TF-IDF scheme) to compute cosine similarities between them. Tweets' words are normalized as follows: words are stemmed and lowercased, and all non-word characters are discarded. We then straightforwardly run Affinity Propagation on these data points.

4. PRELIMINARY STUDIES

Tweetmeme (http://tweetmeme.com), aggregates all Twitter posts to determine frequently retweeted URLs, categorizes the stories these URLs point to, and presents them as news stories in a fashion similar to Digg's front page. We collected data from Tweetmeme using specialized page scrapers developed using Fetch Technologies's AgentBuilder tool. For each story, we retrieved the name of the user who posted the link to it, the time it was posted, the number of times the link was retweeted, and details of up to 1000 of the most recent retweets. For each retweet, we extracted the name of the user, the text and time stamp of the retweet. We were limited to 1000 most recent retweets by the structure of Tweetmeme. We extracted 398 stories from Tweetmeme that were originally posted between June 11, 2009 and July 3, 2009. Of these, 329 stories had fewer than 1000 retweets. Next, we used Twitter API to download profile information for each user in the data set. The profile included the complete list of user's friends and followers.

4.1 Collasping Identical Tweets

In our data set, there are many "identical" tweets, which contain an exact similar set of terms. These identical tweets usually cause the inference process to become unstable. Illustratively, suppose we have 5 tweets, where tweet #1 contains terms:"A X Y B," and the rest identically contains the terms: "X Y Z." With any similarity metric, tweet #2 thru #5 must belong to the same cluster. In addition, suppose that the tweet #1 is the most similar to "A X Y B." Hence, tweet #1 can choose any of tweets, #2 to #5, as its exemplar, which makes the exemplar assignment of this cluster keep changing through out the inference. To alleviate this problem, we could use randomly generated preferences but we collapsed all identical tweets into the same data point, which reduces the size of the data set from 1447 to 795 tweets.

4.2 Empirical Validations

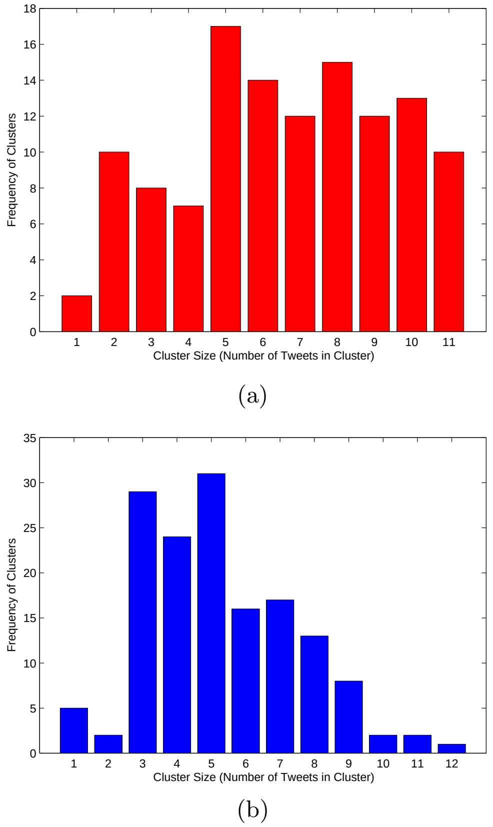
4.2.1 Evaluation Metrics

To evaluate the utility of AP in our setting, we first use tweets's URL as the actual cluster labels (ground truth). Specifically, we assume that all tweets that refer to the same URL belong to the same cluster. We then compute normalized mutual information (MI) [1]. Particularly, consider AP predicts a cluster division X from total n tweets, while the actual cluster division is Y. Consequently, the probability that a tweet is assigned to division X while it actually belongs to cluster division Y is $P(X = x, Y = y)$, which can be computed by N_{xy}/n, where N_{xy} is the number of nodes that were assigned to cluster x that actually belong to cluster y. Using this probability, we can calculate the normalized mutual information. The larger the value, the better the predicted clusters consistent with the actual ones.

4.2.2 Results

In Affinity Propagation, we cannot select the number of clusters but we can control them using different preference values. The preferences are placed on the diagonal of the similarity matrix and because we do not have any prior knowledge about appropriate preferences, we set them all equal to mean, median, min*2 and min value. Our best MI score was 0.8756 with similarity mean value as a uniform preference.

Figure 2 shows the distribution of actual cluster size and predicted cluster size. X-axis represents the number of tweets in cluster and y-axis represents frequency of occurrence of cluster size. While preserving the original cluster division based on URL, our model can combine tweets which originally belong to different URL cluster. Table 2 contains one example cluster from our data set about Beastie Boys tour cancellation due to Adam Yauch' cancer diagnosis. Note that none of these tweets contain hashtag and we only used texts to find this cluster. The first tweet was picked as an exemplar because it contains most important tf-idf weighted words than any other tweets in this cluster. Many individual user-generated tweets picked this tweet as an exemplar. Our result shows that it is a promising approach to find relevant clusters from different URLs while preserving initial URL based clusters.



(a)

(b)

Figure 2: The chart presents a distribution of the size (a number of cluster members) of (a) actual and (b) predicted clusters

5. CONCLUSION AND FUTURE WORK

In this paper, we applied AP to cluster related tweets and retweets that contain URLs to news stories and evaluated them using tweets's URL. We demonstrated that our approach is promising since it combines related tweets and retweets that are originally coming from different URLs while preserving original URL cluster division. In the future, we plan to develop different similarity measures. Since similarity measure play a crucial role in identifying the right exemplar, we will integrate other useful information, such as time factor or user-to-message relation. Second, we are interested in building efficient distributed learning algorithms to scale to real large data sets under the Map-Reduce framework. We will argue the effectiveness and efficiency of AP on real large data sets. Last, we also would like to use AP and its variant to answer the following questions:

- Is a tweet exemplar always the first tweet of its link?
- Are tweet exemplars usually generated from the same person?

ID	Tweet Text(URL)
356*	Adam Yauch has cancer, Beastie Boys cancel all dates :((http://www.brooklynvegan.com/archives/2009/07/adam_yauch_has.html)
34	Adam Yauch has cancer, Beastie Boys cancel all dates - BrooklynVegan (http://www.brooklynvegan.com/archives/2009/07/adam_yauch_has.html)
350	Yauch apologies for his cancer :((http://www.youtube.com/watch?v=u7CH3M7cECI)
351	RT: Adam Yauch (MCA) has cancer. Beastie Boys cancel all dates Oh man =/ (http://www.brooklynvegan.com/archives/2009/07/adam_yauch_has.html)
353	RT get well adam <3 @brooklynvega sad news alert: Adam Yauch (MCA) has cancer. Beastie Boys cancel all dates (http://www.brooklynvegan.com/archives/2009/07/adam_yauch_has.html)
354	If u hadn't heard.. RT @brooklynvegan sad news alert: Adam Yauch (MCA) has cancer. Beastie Boys cancel all dates (http://www.brooklynvegan.com/archives/2009/07/adam_yauch_has.html)
357	bloody hell, Adam Yauch has cancer (http://www.brooklynvegan.com/archives/2009/07/adam_yauch_has.html)
358	Send love. RT @brooklynvegan: sad news alert: Adam Yauch (MCA) has cancer. Beastie Boys cancel all dates (http://www.brooklynvegan.com/archives/2009/07/adam_yauch_has.html)
364	Beastie Boys cancel their tour as Yauch announces he has cancer and is undergoing surgery next week. Get well soon Adam. (http://beastieboys.com/)
365	Yikes. Beastie Boy Adam Yauch aka MCA has cancer of the saliva gland. Get well soon. We're all getting old (http://beastieboys.com/)

Table 1: The cluster of "beastieboy" tweets, having the tweet #356 as its exemplar.

ID	Tweet Text(URL)
16*	APOLLO LANDING SITES IMAGED BY LRO! | Bad Astronomy | Discover Magazine (http://blogs.discovermagazine.com/badastronomy/2009/07/17/apollo-landing-sites-imaged-by-lro/)
247	RT @badastronomer: HOLY FREAKING HALEAKALA! Apollo landing sites imaged by LRO!!! Incredible images! (http://blogs.discovermagazine.com/badastronomy/2009/07/17/apollo-landing-sites-imaged-by-lro/)
248	APOLLO LANDING SITES IMAGED BY LRO! - [@BadAstronomer] #GlobalAtheist (http://blogs.discovermagazine.com/badastronomy/2009/07/17/apollo-landing-sites-imaged-by-lro/)
250	Freaking Cool! Pix of the Apollo landings from the LRO... (http://blogs.discovermagazine.com/badastronomy/2009/07/17/apollo-landing-sites-imaged-by-lro/)
251	Apollo landing sites imaged by LRO!!! Incredible images! (via@TeresaKopec: @BadAstronomer) Brilliant. (http://blogs.discovermagazine.com/badastronomy/2009/07/17/apollo-landing-sites-imaged-by-lro/)
252	New Pictures of Apollo Landing Sites. Take THAT Conspiracy Theorists! (http://blogs.discovermagazine.com/badastronomy/2009/07/17/apollo-landing-sites-imaged-by-lro/)
253	Reading APOLLO LANDING SITES IMAGED BY LRO! | Bad Astronomy | Discover Magazine (http://blogs.discovermagazine.com/badastronomy/2009/07/17/apollo-landing-sites-imaged-by-lro/)
254	RT @meatleg: ROCKIN'!!!!! APOLLO LANDING SITES IMAGED BY LRO! | Bad Astronomy | Discover Magazine (http://blogs.discovermagazine.com/badastronomy/2009/07/17/apollo-landing-sites-imaged-by-lro/)
329	Sven & I are watching the simulation of the Apollo lunar landing on (http://kottke.org/apollo-11/)

Table 2: The cluster of "apollo landing photo" tweets, having the tweet #16 as its exemplar.

- Do the top users who are followed by many others always generate tweet exemplars?

From total $164,902$ tweets, there are $5,461$ tweets having unique texts and 346 tweets with unique hashtags. This is about 6% of tweets containing a hashtag, which coincides with the early finding of only about 5% of tweets having hashtags [2]. With manual inspection, we found that tweets with different content can have the same hashtag, as the tweets with the same URL can also have different hashtags. Different from the recent work that utilize hashtags to find topic distribution of tweets [8], or users' influence flow of a given topic [6], we also suspect whether or not a hashtag can be regarded as a representative term for a given tweet.

6. REFERENCES

[1] M. J. Barber. Modularity and community detection in bipartite networks. *Physical Review E*, 76, 2007.

[2] D. Boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *In Proceedings of the Hawaii International Conference on System Sciences*, 2010.

[3] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi. Measuring user influence in twitter: the million follower fallacy. In *In Proceedings of the 4th International Conference on Weblogs and Social Media*, 2010.

[4] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 312:972Ű–976, 2007.

[5] I. E. Givoni and B. J. Frey. A binary variable model for affinity propagation. *Neural Comput*, 21(6):1589–1600, 2009.

[6] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of KDD Workshop on Web Mining and Social Network Analysis*, 2007.

[7] F. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 2001.

[8] D. Ramage, S. Dumais, and D. Liebling. Characterizing microblogs with topic models. In *In Proceedings of the 4th International Conference on Weblogs and Social Media*, 2010.

[9] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *In Proceedings of the 3rd International Conference on Web Search and Data Mining*, 2010.