Scalable Multi Stage Clustering of Tagged Micro-Messages

Oren Tsur
Department of Computer Science
The Hebrew University
Jerusalem 91904, Israel
oren@cs.huji.ac.il

Adi Littman
Department of Computer Science
The Hebrew University
Jerusalem 91904, Israel
adilittman@cs.huji.ac.il

Ari Rappoport
Department of Computer Science
The Hebrew University
Jerusalem 91904, Israel
arir@cs.huji.ac.il

ABSTRACT

The growing popularity of microblogging backed by services like Twitter, Facebook, Google+ and LinkedIn, raises the challenge of clustering short and extremely sparse documents. In this work we propose SMSC – a scalable, accurate and efficient multi stage clustering algorithm. Our algorithm leverages users practice of adding tags to some messages by bootstrapping over virtual non sparse documents. We experiment on a large corpus of tweets from Twitter, and evaluate results against a gold-standard classification validated by seven clustering evaluation measures (information theoretic, paired and greedy). Results show that the algorithm presented is both accurate and efficient, significantly outperforming other algorithms. Under reasonable practical assumptions, our algorithm scales up sublinearly in time.

Categories and Subject Descriptors

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

Keywords

scalability, clustering, short documents, microblogging, Twitter, hashtags

1. INTRODUCTION

Document clustering is one of the basic tasks required as a first step in many text processing efforts such as topic modeling, content-based recommendation and question answering, among others. As the popularity of microblogging is growing at an increasing rate, there is thus a need for accurate clustering of micro-messages on a large scale.

While document clustering is well studied, applying traditional clustering methods on micro-messages fails due to the inherent sparseness of micro-messages. Moreover, classic clustering techniques are incapable of handling the massive stream of data posted in microblogging services (e.g. over 200 million messages are posted on Twitter every day[1]). In this paper we develop an accurate and efficient clustering algorithm for tagged micro-messages such as Twitter tweets. Our algorithm exploits tagged data, allowing to convert the document clustering task to a tag clustering problem in order to overcome sparseness and improve running time. Given a fixed number of hashtags, our algorithm scales up in sublinear time as we employ the inefficient clustering algorithm only on the reduced space of the hashtags, regardless the number of tweets or the desired number of clusters. The sublinearity holds under the assumption that the number of hashtags is orders of magnitude smaller than the corpus size. This assumption is very reasonable and holds in our data.

[1] According to the official Twitter blog: http://blog.twitter.com/2011/06/200-million-tweets-per-day.html

Stage 1 Given a collection of micro-messages D and T, a set of tags appearing in D – we create a set of virtual documents D'. The number of virtual documents in D' is $\|T\|$ – the number of tags in T. Each $d^t \in D'$ is a concatenation of all micro-messages in D that contain a specific tag t. If some d contains more than one tag it will be concatenated to more than one virtual document in D'. Each d^t is now represented as a feature vector based on its words. The vectors based on the virtual documents in D' are not sparse since they are based on the concatenation of multiple micro-messages.
Stage 2 We now use a kMeans algorithm in order to cluster the virtual documents in D'. The kMeans algorithm operates on a reduced space of $\|T\|$, the number of tags, instead of $\|D\|$. C', the partition achieved for $d^t \in D'$ is actually a solution to a hashtag clustering task, since each $d^t \in D'$ corresponds to a tag $t \in T$. As it is assumed that $\|T\| << \|D\|$, the typically time consuming kMeans terminates much faster (see Scalability section).
Stage 3 In the third stage, each virtual document d^t is redivided to $\{d^t_i\}$ the set of micro-messages it was originally composed from. Each micro-message d^t_i is now being assigned to the same clusters the virtual document d^t was assigned to.

Table 1: stages of the Scalable Multi-Stage clustering algorithm

Related Work While many works focus on document clustering and topic models, clustering of micro-messages is addressed sparsely. The major challenge in clustering of micro-messages (or even short documents) is sparseness, thus traditional techniques in which documents are represented as TF-IDF feature vectors does not perform well. A comparative study, comparing three clustering algorithms: kMeans, singular value decomposition (SVD) and affinity propagation (AP) on a small set of only 600 tweets they conclude that affinity propagation is best suited for (a small number of) short texts [4]. Another challenge arising with the increasing popularity of microblogging is efficiency. Sculley [8] presented a modified kMeans algorithm designed for large scale sparse web page clustering. Our experiments show this modification is not well suited to microblogs. Keywords and tags are sometimes used to improve clustering of long documents [1, 2], although in these cases the full content is ignored.

2. CLUSTERING ALGORITHM

Problem definition Given a set of tags T, a large set of sparse micro-messages D, where each $d \in D$ contains at least one tag $t \in T$, and given k, a desired number of clusters: find an optimal partition with k sets. Our Scalable Multi-stage Clustering (SMSC) algorithm is portrayed in Table 1.

Scalability One of the drawbacks of the standard kMeans is its practical running time. Given N data points and k clusters and assuming m (m') iterations till convergence[2], the practical running

[2] We assume m is bounded and relatively small, in our experiments we allow a maximum of 100 iterations, which proved sufficient.

of the kMeans is $T(kMeans) = m \cdot k \cdot N^d + N = O(N^d)$ (d is the dimension of the vector space). On the other hand, $T(SMSC) = m' \cdot k \cdot g^d(N) + N$, where g is a sublinear function of N (in our case $g(N) = \sqrt{N}$), thus $T(SMSC) = O(g^d(N))$.

3. EXPERIMENTAL FRAMEWORK

Corpus statistics Our corpus consists of over 417,000,000 tweets. The hashtag frequency presents a long tail distribution where the 1000 most frequent hashtags cover 43% of hashtag occurrences. Following [7] we manually classified the 1000 most popular hashtags in our dataset, creating a gold standard. A thousand tweets were sampled for each of the hashtags in our list, creating a set of one million tweets to cluster.

Data Representation We took the standard bag of words approach, representing each $d^t \in D'$ by the *tf-idf* values of the n most frequent words in our data. We experimented with various values of n, results are reported for $n = 10000$.

3.1 Baseline Algorithms

Baseline 1 Dist.BL assigns tweets to clusters according to the class distribution of the gold standard, thus cluster sizes correspond to class sizes.

Baseline 2 We use the the classic kMeans [3], one of the most widely used algorithms for clustering.

Baseline 3 In order to address issues of scalability and sparseness, Sculley [8] proposed the Web-Scale Fast kMeans (WSFkM) algorithm which includes two modifications of the standard kMeans algorithm. WSFkM uses mini-batch optimization for kMeans which reduces computation costs; a projected gradient descent is used providing a projection into the L1 ball.

3.2 Evaluation Methods

A meaningful evaluation of clustering results is a challenging task. Even provided with a gold standard external measure, a number of competing measures can be used, each has its advantages and disadvantages. We used seven evaluation measures: the pairwise Rand index (RI); the entropy based V, VI and NVI measures; the mapping based measures: Greedy many-to-one and Greedy one-to-one; and the F measure (see [5] for a survey of the evaluation methods). Note that measures differ in range and interpretation.

All measures were computed on two different gold-standard sets:
Tweets Gold Standard (TGS) We sampled 1000 tweets from our dataset and had them manually classified by two human annotators. The annotators agreed on 97.4% of the sampled tweets, that constituted the TGS gold standard. In order to classify tweets, our SMSC algorithm reduces the clustering task to a hashtag clustering task. The TGS set allows us to evaluate the algorithm directly on tweets and to verify the conformity between the indirect hashtag-based multi-stage clustering and the tweet content.

Hashtags Gold Standard (HGS) The core assumption driving this work is that tags can be leveraged for bootstrapping in content based clustering of a sparse collection of documents. In order to classify tweets, our SMSC algorithm reduces the clustering task to a hashtag clustering task. We thus follow the classes definition and classification provided by [7], creating a gold-standard of manually classified *hashtags*. The HGS set allows us to test our algorithm on a larger scale as only the hashtags need to be manually classified.

4. RESULTS

Table 2 presents results of a baseline of the standard kMeans and results of our algorithm (SMSC) against the 1000 tweets in the tweet's content gold standard (TGS). Our algorithm's Rand index is 0.8, the Greedy many-to-one is 0.52 and the F score is 0.44. The greedy mapping and the F score values confirm that classification of such a sparse data is hard and that class definitions are fuzzy. However, our algorithm significantly outperforms the baseline in all seven measures. Results of the standard kMeans are similar, in most measures, to the distribution-based baseline, demonstrating the challenge posed by sparseness and the limitations of the standard approach and highlighting the contribution of the SMSC algorithm. Achieving these results on the TGS set establishes the validity of the reduction from tweets to hashtags that is performed in stage 1 of SMSC.

Alg.	RI ↑	V ↑	VI ↓	NVI ↓	GM1 ↑	G11 ↑	F ↑
Dist.BL	0.68	0.02	3.6	1.95	0.356	0.21	0.24
kMeans	0.72	0.07	3.63	1.96	0.38	0.2	0.22
SMSC	**0.8**	**0.29**	**2.82**	**1.51**	**0.52**	**0.42**	**0.44**

Table 2: Clustering results against the tweets gold standard (TGS). For ease of reading, we indicate (in all tables) whether a higher score is better (↑) or whether a lower score is better (↓).

Alg.	RI ↑	V ↑	VI ↓	NVI ↓	GM1 ↑	G11 ↑	F ↑
Dist.BL	0.72	0.001	3.8	2.01	0.31	0.18	0.21
kMeans	0.75	0.06	3.74	1.96	0.36	0.22	0.23
WSFkM	0.22	0.002	(1.91)	(1.004)	0.31	0.31	0.3
SMSC	**0.80**	**0.27**	2.94	1.54	**0.5**	**0.4**	**0.46**

Table 3: Clustering results against the hashtag gold standard (HGS)

Scalability Table 3 presents results of four algorithms, evaluated against the million tweets in the HGS set. Our SMSC algorithm outperforms the three baselines in all measures[3].
Surprisingly, the Web-Scale Fast-kMeans baseline ([8]), designed to address large and sparse collection of web pages, performed poorly on our data. Examining the actual clusters produced by the WSFkM baseline, we see that one big cluster was created containing all tweets but a handful. We attribute that to the batch sampling used in order to achieve scalability. This sampling cannot handle extremely sparse micro-messages, although proven suitable for clustering web pages.

Running times Experimenting on a single machine with 6 Intel(R) Core(TM) i5-2400 3.10GHz CPU units and 8GB memory, the standard kMeans converges after 18632 seconds while our SMSC converges after only 129 seconds, demonstrating a great improvement in execution time as well as in clustering quality.

5. REFERENCES

[1] G. Begelman, P. Keller, and F. Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland*, pages 15–33. Citeseer, 2006.

[2] C. Brooks and N. Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *Proceedings of the 15th international conference on World Wide Web*, pages 625–632. ACM, 2006.

[3] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA, 1967.

[4] A. Rangrej, S. Kulkarni, and A. Tendulkar. Comparative study of clustering techniques for short text documents. In *Proceedings of the 20th international conference companion on World wide web*, pages 111–112. ACM, 2011.

[5] R. Reichart, O. Abend, and A. Rappoport. Type level clustering evaluation: New measures and a pos induction case study. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 77–87. Association for Computational Linguistics, 2010.

[6] R. Reichart and A. Rappoport. The nvi clustering evaluation measure. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 165–173. Association for Computational Linguistics, 2009.

[7] D. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 695–704. ACM, 2011.

[8] D. Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178. ACM, 2010.

[3]The WSFkM baseline seemingly achieves the best results in the VI and NVI measures, however, these results are meaningless as the WSFkM creates one cluster containing almost all instances and $k - 1$ clusters each containing 1 or two instances. Since VI and NVI favor big clusters, the case of which most instances are assigned to the same cluster achieves low (good) score. This very peculiarity is discussed in the section *Extreme Cases for the Two Measures* in [6].