User Community Detection, based on Temporal Topical Interest Similarity

Authors:
João Santos - 81083
João Simão - 81654
(IST - Universidade de Lisboa)

December 6, 2018

Abstract

In this Network Science project we present a procedure, primarily based on a paper presented by Fani Et. al [3], in a 2017 ACM Conference on Information and Knowledge Management. The main contribution of the paper was presenting a way to represent users with similar topical interests in their posts, having into account the temporal dimension, and then use this representation to do Community Detection.

1 Introduction

The social networks we live in have been increasing in terms of complexity over the course of human existence, and with the shift of these networks to the online world, this increase has been accelerated radically.

Also increasing radically in terms of complexity, is the structure of the information we consume nowadays. The number of information providers is vast and diverse in terms of perspective and motivation, and one could even affirm that not all of these providers are high-fidelity sources, a fact that many times may, or not, influence people to make decisions based on wrong information. This has impacts on all industry and economical systems, which are all increasingly dependent on the flow of information and opinions that occur in these networks, and the high complexity of this information flow creates the need of intelligent ways to reason about it, being for the safety or utility of the participants. This has sparked a big increase in researching such huge real-world networks.

One process that has been vastly investigated is user community detection, and it's very easy to understand why it's useful to know that certain users are similar and active with each other, being for market segmentation and prediction, or for studying social aspects of these users.

In the particular scope of Fani Et. al [3], there is a combination of being able to detect user communities, and being able to specify in which time window they were actually similar, which is presented to be very useful.

In our project, we present a procedure based on Fani Et. al, but with some simplifications.

In Section 2, we describe in some detail, the various steps we went through to get from a database containing tweets, to our Temporal User Communities. In Section 3, we present the results achieved for the graph created for Temporal User Communities Detection. In Section 4, we give some final remarks and conclusions of the work done, and in Section 5 we discuss what we could improve in our procedures.

2 Summary of Intermediary Procedures

2.1 Initial Dataset

We decided to use, as a starting point, a database provided by Abel Et. al, used in [1], which had information derived from 2,316,204 tweets, posted by 1619 users. We then queried the database to obtain, for a total period of two months, the tweets of the participant users, grouping these by userId and creationTime. This was done in order to follow the procedures of Fani Et. al, which proposed representing all users daily tweets as single documents.

The final result of this step was a text file with a total of 55856 lines, each line being the aggregation, by user and date, of all the corresponding tweets.

2.2 Annotation of Wikipedia Entities

Fani Et.al proposed annotating the documents with entities defined in Wikipedia, to improve the topic detection, being that using these entities has proven to give better results than using mere words, which in Twitter, are in many cases unusable.

To get these annotations, we used the TagMe Restful API combined with the coroutine based networking python library Gevent, which we used in order to improve performance, as it took several time to make all the requests one by one.

The result of this step was a text file with a total of 27836 lines, each line being the Wikipedia entities given by TagMe, for all that specific user's daily posts. The reduction of documents was due to TagMe not being able to annotate all documents, fact dealt with, by not using the empty documents, and working only with meaningful data, even if reduced.

2.3 Creation of User Documents LDA Model

Having aggregated a set of user daily documents, annotated with Wikipedia Entities, the next step was to create, for each document, a topic distribution, with the objective of knowing how much each user contributed to each topic, or by other words, how much those topics can be inferred from their posts.

Based in Fani Et. al experimental setup, which proposed using a Latent Dirichlet Allocation model to do topic modelling, we decided to use the semantic modelling tool Gensim, and more specifically, the gensim.ldamulticore class, which uses all CPU cores in parallel, improving the training time.

LDA models require two important things, an apriori number of topics, which in the case of our dataset, was already investigated to be 18, and for the input documents to be in the CBOW format, which represents documents as dictionaries with numerical keys, meaning unique tokens/words, and with the values meaning the frequency of each unique token in that specific document. To fulfill the latter requirement, we created a dictionary to give each unique word a unique numeric value, and then processed our entity annotated file to be a vector of dictionaries in CBOW format, with each dictionary representing one document. We used this vector as the corpus for creating the LDA model.

We computed various LDA models with different parameters, and picked one, based on the coherence of the top-k words contributing for each topic, and to do that we used the LDA visualization python package pyLDAvis.

The model we finally used, ran 50 iterations with a window size of 300 documents, for a total amount of time of around 120 minutes. We present a visualization sample, of the second topic's k-words given by pyLDAvis for the model we finally used in Figure 1.

2.4 User Topic-based Similarity Graph

After having a topic distribution for every document in our corpus, using the python package NetworkX, we built a graph in which each node represents a user in a certain day, and the edges represent the similarity between users, with their weights being the numerical value of this similarity.

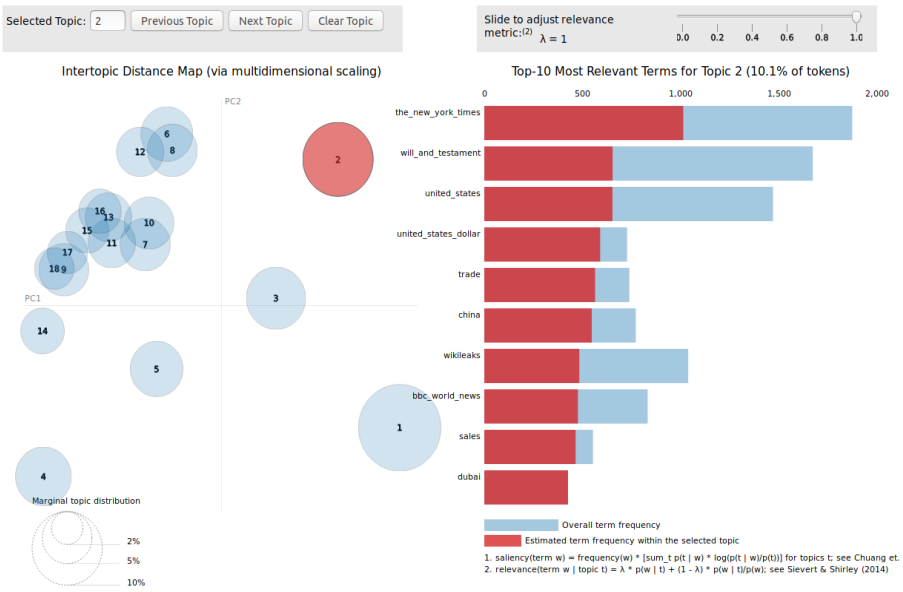

Figure 1: Snapshot of local server created with pyLDAvis, presenting, for each topic, the top-k words.

As a similarity measure, we used the Hellinger distance, normally used to quantify the similarity between two probability distributions (range between 0 and 1, with 1 meaning that two distributions are as distant as representable by the measure), which, in our case, has the same meaning as the similarity between two users topical interest.

When adding edges to our graph, we defined a cutoff of 0.2 (Hellinger distance), meaning that we only added edges if the distance between the nodes was smaller than 0.2. As a consequence of this, certain users were discarded, as they had no relevant similarity with others, and therefore presented no further interest, in our project scope.

2.5 Community detection

To do community detection, we decided to use Gephi, as we were already familiarized with its functioning, and being that its performance would still be acceptable for the graph we created. The algorithm we used [2], was already provided by Gephi in its default installation, and groups users in the modularity class.

3 Results

In Table 1, we present some metrics we computed for our graph, in Figure 2, the community size distribution and in Figure 3, the visualization of the main communities we were able to compute.

Graph Nodes	14209
Graph Edges	364751
Modularity	0,926
Average Degree	43,437
Average Clustering Coefficient	0,684

Table 1: Some computed metrics for the user graph.

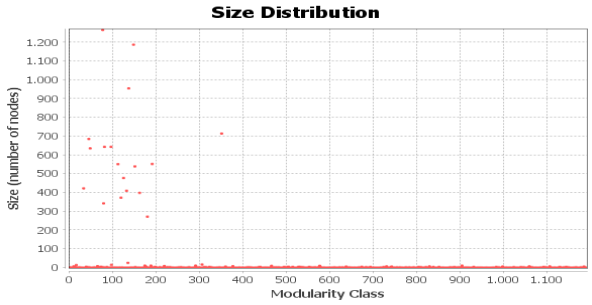

Figure 2: Community size distribution graph, calculated with Gephi.

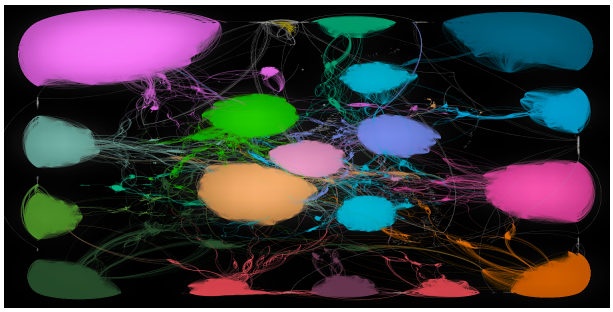

Figure 3: Graph with layout given by Force Atlas 2 algorithm, and color partitioned by modularity class. 18 main communities highlighted.

4 Final Remarks

The graph we built presents 18 main communities, being that each of those communities represent users similarity towards a topic, having also a temporal dimension. We consider that, with that said, the procedure we present is coherent and can be applied to detect user communities, where users have similar contribution towards a certain topic.

A drawback of our project is that we don't test these calculated communities at an application level, which could be done in a lot of ways, being one measuring the quality of content recommendation done using these communities. This kind of application level observation is said by Fani Et. al to be the best option to test these kinds of approaches, as normal measures such as Jaccard Index require gold standard communities, which are not easily retrievable for real-world social networks, due to their huge complexity, in terms of structure and content.

5 Possible Improvements

The LDA model package we used offers the possibility to customize the minimum probability for each topic, which we defined as 0, but if bigger, we could have had dense vectors instead of sparse ones, and that way we could have applied different similarity measures, and see the variation in the results. We could have also used various similarity measures and compute an intern one weighting in, all the various measures calculated.

Being that computing the intermediary procedures with these amounts of data takes a big amount of time, and being that the quality of the steps done, for example in the creation of the LDA model, depends a lot on parameters variation and experimentation, we could have introduced parallel computing in more steps, and use the time saved, to experiment with these parameters and possibly improve the results achieved.

References

[1] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Analyzing user modeling on twitter for personalized news recommendations. In Joseph A. Konstan, Ricardo Conejo, José L. Marzo, and Nuria Oliver, editors, *User Modeling, Adaption and Personalization*, pages 1–12, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[3] Hossein Fani, Ebrahim Bagheri, and Weichang Du. Temporally like-minded user community identification through neural embeddings. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 577–586, New York, NY, USA, 2017. ACM.

[4] Matthew D. Hoffman, David M. Blei, and Francis Bach. Online learning for latent dirichlet allocation. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, pages 856–864, USA, 2010. Curran Associates Inc.

[5] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.