Multidimensional Community Detection in Twitter

Nasser Zalmout*
Department of Computing
Imperial College London
nasser.zalmout12@imperial.ac.uk

Moustafa Ghanem
School of Science and Technology
Middlesex University, London
m.ghanem@mdx.ac.uk

***Abstract*—We present and apply a generic methodology for multidimensional community detection from Twitter data. The approach builds on constructing multiple network structures based on the similarity and interaction patterns that exist between different users. It then applies traditional network centric community detection techniques to identify clusters of users. The paper also approaches the issues of dynamicity and evolution in Social Media by developing a Bayesian classifier that maps new users to the detected communities. Using a data set of UK political Tweets, we evaluate the factors affecting the quality of the detected communities. We also investigate how the accuracy of the classifier is affected by the dynamicity of the network evolution and the time elapsed between community detection and classifier application.**

I. Introduction

A. Motivation

Humans are social beings by nature, always trying to group themselves instinctively within communities and societies. We tend to group ourselves with people with similar ideologies and backgrounds, as this would facilitate easier understanding and communication. Communication patterns also tend to intensify for members in the same group, in comparison to members from the outside. These facts are the essence of the sociological principles of *Influence and Homophily* [1]; where people eventually tend to develop similar views and opinions over different issues, based on the characteristics of these groups and patterns of communication. The Social Media, or the virtual social world in general, is a reflection of the real world. People in the virtual world still instinctively group themselves with people sharing similar backgrounds, forming a kind of virtual communities.

Twitter is a social networking tool with more than 500 million users with over 340 million tweets sent daily[1]. In Twitter, users publish a stream of short messages called 'tweets' and the social network itself is structured so that users can 'follow' each other, thus adding the followed user's tweets to the follower's news feed. The author of a tweet may add the '#' symbol as prefix to arbitrary words in its content which become known as a 'hashtag'. These hashtags can then be used for identifying messages that discuss the same topics. Authors also have the ability to rebroadcast, or 'retweet', another tweet and the ability to reply to specific users as well as mention them in tweets.

Within Twitter, as any other social network, a community can be regarded as a cluster of users that are characterised by having dense connections amongst themselves and sparse connections with other users or clusters. Traditionally, one can use the explicit connections that exist in Twitter (as formed via the 'follow' relationships between users) to study its community structures. However, other implicit communities do also exist. These are the communities formed when considering the relationships between the contents of users' messages; e.g. when considering the hashtags and topics discussed and/or the retweets and mentions of other users. Such relationships form multiple dimensions that construct different network structures that represent communities with similar views and similar communication patterns. Automatically detecting these hidden communities is of importance for many applications that require identification of users with similar opinions and tastes, including political analysis, advertising and recommendation engines.

B. Related Work and Contribution

Community level analysis in social networks [2] provides generic tools to examine the different models of influence and knowledge propagation. The principles of community detection have found their applications in a wide spectrum of fields; ranging from biological and social sciences[3], to the web [4]. Tang and Liu [2] provide a comprehensive review of the theory behind community detection and mining of the Social Media content. Moreover, Tang et al. [5] investigate the general multidimensional heterogeneity of communication in Social Media, utilising the communication of the users at different Social Media platforms to enhance the accuracy of the community detection procedure.

This paper presents a generic methodology for applying multidimensional community detection to Twitter. It studies the factors affecting the quality of the detected community structures and investigates the challenges for learning classifiers that map users to existing community structures. It builds on the theory and practices outlined in the related work, investigating the multidimensional nature of the communication in Twitter for community detection. Unlike Tang et al. [5] who focused on detecting communities across multiple social networks, this paper examines the dimensions of different within Twitter, presenting a practical procedure to calculate the distance scores for pairs of users. In addition to leveraging the interaction and intensities patterns, as used by Giatsoglou et al. [6], we also take into account the content similarity dimensions in Twitter. We also build up on Tang and Liu's [2] proposed snapshot approach to tackle dynamicity in Social Media, by learning a Bayesian classifier to map new users to existing communities. Finally, we also present a case study based on a data set of UK political Tweets, to evaluate the practicality of the approach and to investigate the factors affecting the quality of the detected communities and classifier.

*Currently at Birzeit University, Ramallah, Palestine
[1] statisticbrain.com/twitter-statistics

978-1-908320-20/9/$25.00©2013 IEEE

II. Background

A. Homophily and Social Network Analysis (SNA)

Homophily, or "love of the same"[7], can be defined as the tendency of people to associate and bond with other people who share similarities with them. *Homophily* forms a fundamental principle in social networks in general. Miller McPherson et al. [1] have explained the presence of homophily within social networks, and explained how homophily has powerful effects and implications for the information people receive and attitudes they show.

Communities formed within Social Media content are the direct effects of the homophily principle. Structure-based social network analysis represents relationships between people as a graph and uses graph-based algorithms to extract subgroups or communities. Such graph structures can be of great importance for understanding how information propagates and spreads in a system of people. Modeling communities of people (nodes) and interactions (connections) is natural as the structure builds up on two basic facts: (1). People tend to group into clusters as a result of communication opportunities for which people tend to meet, both physically and virtually (2). Communication is more influential and frequent within these clusters, such that people within the same cluster tend to develop similar views. [8].

In this context, the community structure can be defined essentially as a form of grouping or clustering of the nodes. The resulting clusters are characterised by having dense connections amongst the nodes within the community, and sparse connections for the inter-clusters nodes[2]. These connections might have different semantics, as discussed later in the paper, but they all contribute to the communities' structure.

B. Network-Centric Community Detection Algorithms

Various types of community detection algorithms exist in the literature. These include network-centric, hierarchy-centric, node-centric, and group-centric algorithms[7]. In this paper we focus on network-centric algorithms. Such algorithms study the overall topology of the network, aiming at obtaining possible partitions from within the network. The algorithms usually include some criterion or metric defined upon all network partitions, rather than a certain group or individual nodes. A good example of such algorithms is the Modularity Maximisation algorithms.

Modularity in a network can be defined as a quality measure for network partitions. For any given network, with m different edges, and assuming that the network is partitioned into k different communities, modularity, Q, can be defined as [2]:

$$Q = \frac{1}{2m} \sum_{l=1}^{k} \sum_{i \in C_l, j \in C_l} Aij - \frac{d_i d_j}{2m}$$

where, d_i is the degree of the vertex i, or the number of edges incidents of vertex i, and Aij is the number of edges between vertices i and j. In the formula, the expected edges count between any given pair of nodes is represented by $\frac{d_i d_j}{2m}$, the strength of a community is represented by $\sum_{i \in C, j \in C} Aij - \frac{d_i d_j}{2m}$, and the term $\frac{1}{2m}$ is used to normalise the modularity value to between -1 and 1.

One simple approach for finding communities in a network structure is based on modularity maximization. A possible approach for achieving this is by using a greedy algorithm that begins off with all nodes belonging to their own separate communities. The algorithm then merge those two communities that make a better overall modularity score. The process continues until a local modularity maximum is found. This is approach used in this paper and the main community detection algorithm used is the *Fastgreedy* algorithm provided by the *igraph* tool[2]; graph theory and network analysis software packages. The algorithm produces a *Dendrogram* based clustering that facilitates easy control over the number of returned clusters, or communities and relies on a greedy optimization method, applied to a hierarchical agglomerative approach. Each node starts off at separate communities then consequent merges occur based on the modularity score, until no further modularity enhancement can be observed.

III. Twitter Community Detection Methodology

A. Multidimensional Heterogeneity

Communication patterns within in Social Media are generally of a multidimensional heterogeneous form[2]. Different users, or nodes, tend to use multiple forms of communication and knowledge sharing. For Twitter, these patterns can be classified into two broad sets of dimensions:

- Interaction dimensions, where nodes engage in a direct interaction form that has two distinct users. These include include replies, retweets, or mentions (when mentioning actual users within the data set).
- Similarity dimensions, where the behaviour of different nodes can be compared for a given a certain topic or field of operation. Examples include hashtags, web links and mentions similarity.

Each of these dimensions contributes to the community structure with different proportions. However, handling the heterogeneity, and assessing how the different proportions at which these dimensions contribute to the community detection could be challenging.

B. Data Flow

The general dataflow for the community detection process for Twitter content is presented in Figure 1. The process begins by the input data set of Tweets, along with the affiliated data items of individual Tweets, including the user information, mentions, hashtags, retweeters, reply_to_tweet if the Tweet was a reply, links... etc. The data set contents are used as the input for the process of User Profiling. User profiling here is the grouping of all the data items (entities), in all the Tweets in the data set, that belong to the same user together in the form of (user: entities). These groupings would include all the entities (hashtags, mentions, retweets...etc) that each of the users has used or was linked to.

The user profiles are used to generate the different dimensions (mentions of the community detection process, or the Interaction and Similarity matrices. These matrices are considered as distance matrices between any given pairs of users. The intersection

[2]igraph.sourceforge.net/doc/python/igraph.pdf

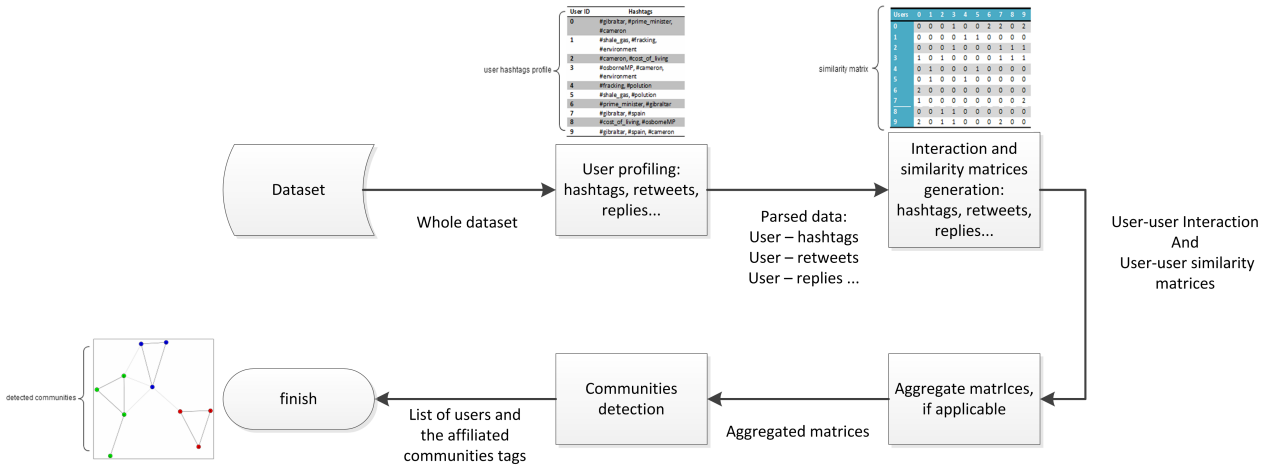

Figure 1. Data flow of the community detection process for Twitter

of the rows and columns represents the interaction, or similarity, between the different pairs of users. These matrices are of size $n \times n$, where n is the number of users in the data set, and they are initialised to zero. The distance matrices are usually sparse, so a better approach is to use sparse matrices to store the data, enhancing the memory efficiency. Some of the dimensions (distance matrices) can then be aggregated, which is used to reduce the number of dimensions for the consequent analysis procedures.

The pseudo code below represents the algorithm used to build the similarity matrices. Another, quite similar, algorithm is used to build the interaction matrices.

Algorithm 1 Building the similarity matrix

Input: SQL table entries in $(user\_id, entity)$ form, entity like hashtags for example
Output: similarity matrix S

1: Map the user_id into sequential range of $[0, n)$
2: Initialise the sparse matrix S with the shape (n, n)
3: **for all** $entity$ in $entities$ **do**
4: compute (user, count(entity)) for users that used this entity
5: **for all** pair of users i and j that used the entity **do**
6: $S(i, j)$ = sim(count(entity) of i, count(entity) of j)
7: **end for**
8: **end for**
9: return S

The similarity function *sim* uses intersection to find the similarity between the pairs of users. Jaccard's index (as an intersection based index) can also be used here as a similarity metric. The similarity metric within this scope has to take into account the different entities, and the frequency in which the entity was used by the pair of users.

The execution time of the algorithm is normally of order $O(h \times u^2)$, where h is the number of distinct entities in the data set, and u is the number of users that used each distinct entity. However, we note that the complexity here depends on the approach used to iterate the pairs of users, by using appropriate data structures and optimization the execution time can be reduced to nearly order $O(h \times u)$, as in Python's Itertools.

C. Community Detection Algorithm and Parameters

The output matrices for each entity can then be aggregated before being passed to the community detection algorithms (*Fastgreedy* algorithm in this paper). The aggregation in this context is conducted by normalising the matrices and adding the results to form aggregate matrices. The corresponding communities for each matrix are then assigned to the related users. Each community corresponds to one of the interaction or similarity matrices. We also note that the behaviour of the community detection algorithm is affected by two key parameters, the 'Edge weight threshold' and 'Number of clusters'.

- Edge weight threshold: The resulting similarity and interaction matrices represent the weight of similarity, or interaction, any given pairs of users share, as an edge in the overall network graph. Many of these edges represent noisy or occasional communication that don't contribute to the community separation. Setting a threshold for the edge weight would help reducing the effect of these edges on the overall analysis.
- Number of clusters: Having a high number of resulting communities decreases the accuracy of how which the detected communities reflect reality. An upper limit for the number of clusters would be to assign each individual user to separate community but this would produce no communities at all. The trade off is to assign a number of clusters low enough to capture natural communities that exist in the data, but not too small to lose information.

D. Constructing a Community Classifier

The communities defined in a Twitter data set are based on their graph properties that exist between users based on the interaction and similarity dimensions. It is not difficult to build an automatic community classifier that assigns new unseen users as they join the social network to one of the identified communities. One approach is to use supervised machine learning, e.g. a Bayesian classifier, with the characteristics and dimensions of

the available communities as a feature-set for the classifier, as the one developed and used within our system. However, it is important to note that such community classifier is built based on a snapshot of the data of the interactions between users that define the community structure at, or up to, a particular point in time. As discussed later in the paper, the community structure typically evolves as time passes and as new interactions are recorded.

IV. Evaluation Approaches

A. Approach and Metrics for Community Detection Evaluation

One approach for the evaluation of community detection process is to compare the output clusters to some existing known communities that form ground truth. In this case, two main evaluation metrics can be used. The first is based on Set Matching, where the resulting community clusters are compared against the clusters of the ground truth information. The second is based on Counting Pairs; where the evaluation is operated over any pair of items in the distribution, and then the results of all the nodes within each community are averaged [9].

For the Set Matching evaluation approach, the *Purity* metric can be used:

$$Purity = \sum_i \frac{|C_i|}{N} max_j Precision(C_i, L_j)$$

where C_i is the cluster i from the detected communities.L_j is a given ground truth category j. The precision of a cluster C_i for a given category L_j is denoted as:

$$Precision(C_i, L_j) = \frac{|C_i \cap L_j|}{|C_i|}$$

Purity is a good measure to detect and scale down with noisy elements within the clusters. However, there are several issues regarding Purity as a sole evaluation metric that renders it inefficient: (1) Purity tends to be biased to small clusters, that is, small clusters are more likely to attain better Purity measures. (2) Purity is associated with individual clusters alone, so changes with other clusters would not have an effect on the individual score. That is, Purity does not reward the grouping of items of different clusters from the same category together. [9]

For the Counting pairs approach, the *B-Cubed* metric can address some of shortcomings of the purity metric. The approach is based on estimating the B-Cubed precision and B-Cubed recall affiliated to each item, rather than clusters, in the distribution. The item B-Cubed precision reflects the amount of items in the same cluster that belong to the item's category. Similarly, The B-Cubed recall affiliated with an item reflects the amount of items from its category that appear in its cluster [9].

The recall of a cluster C_i for a given category L_j is the $Precision(L_i, C_j)$.

A correctness measure can be defined to represent the correctness between two edges e and $e\prime$:

$$Correctness = \begin{cases} 1 & iff L(e) = L(e\prime) \leftrightarrow C(e) = C(e\prime) \\ 0 & otherwise \end{cases}$$

The B-Cubed precision of an item can be defined as the proportion of items in the item's cluster, that share the item's category. The overall B-Cubed precision can be calculated by average the precision of all items in the distribution. Since the average is calculated over items, there's no need to use weighted mean of the clusters or categories. The same approach is used for calculating the B-Cubed recall, by replacing "cluster" with "recall" in the reasoning above [9].

$$Prec\,BCubed = Avg_e[Avg_{e\prime.C(e)=C(e\prime)}[Correctness(e, e\prime)]]$$

$$Recall\,BCubed = Avg_e[Avg_{e\prime.L(e)=L(e\prime)}[Correctness(e, e\prime)]]$$

The recall and precision can then be combined using the F-measure as follows:

$$F(Recall, Prec) = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

B. Evaluating Community Classifier and Community Evolution

In general, a snapshot Bayesian community classifier can be developped and evaluated once communities have been detected and labelled in a data set. The general approach can proceed as follows:

1) Divide the original tweets data set, that has the detected communities' information, into training and testing data sets.
2) To avoid overfitting, the procedures assures that no users affiliated with the training tweets is included with the testing tweets.
3) Train a Bayesian classifier using the training data set. The hashtags, mentions and links are considered as features for the classification process, and the corresponding community information as labels.
4) Apply the generated Bayesian Classifier over the testing data set with the same features.
5) Evaluate the resulting labels. The newly generated labels as the predicted_labels, and the already available community labels as the true_labels and then calculate the F-score.

However, we note that the key challenge for this simple approach is Twitter content is dynamic with constant evolution, large amounts of data is propagated on daily basis and that the detected communities themselves may evolve in time. In this case, it is important to study the effect of increasing the time span between the training and testing data sets, to examine how accurate the system will remain with the transition from a snapshot to a different one. A more realistic evaluation approach would be dividing the data by choosing the training data set only from the first half of the time-span. We then use second half of the time-span as testing data set. Moreover, the testing data itself can be divided based on different time periods to investigate how the accuracy of the classifier varies with the time elapsed from model training to model application.

V. Case Study

A. Data Set

In order to evaluate our community detection framework and tools we needed to collect a Twitter data set with known communities that can act as ground truths. The approach we used

in was to collect Tweets from UK MPs (Members of Parliament). A list of the Twitter accounts of 423 UK MPs, classified by party affiliation, was retrieved from news website Tweetminster[3]. We then collected the Tweets of the MPs via the Twitter API. The statistics of the collected data is summarized in Table 1. In our experiments, we used the political affiliation of the MPs as ground truths for the natural community structures that exist in the data. Our assumption here is that these communication and interactions patterns that exist in the data should, somehow, correlate to the political affiliation of the MPs. Although we do not expect that the community detection approach would cluster the MPs cleanly into the three main UK political parties, we argue that it is reasonable to expect that MPs from the same party would cluster into a small number of communities around party political messages or discussions.

Item	Count
dates	29/7 - 06/8 2013
users	300
Tweets	9266
unique hashtags	1380
total hashtags	4836
unique mentions	5380
total mentions	23630
unique links	1302
total links	1589
replies	119
retweets	244

Table I
STATISTICS RELATED TO THE MPS DATA SET

B. Evaluating Community Detection

For our first experiment, we evaluated how changing the parameters of the community detection algorithm affects the quality of the detected communities. The two key parameters to investigate are the number of clusters and the edge_weight threshold value.

The graphs in Figure 2 show the variation of both Purity and BCubed measures as the numbers of generated clusters is increased. As can be seen, Purity tends to increase with increasing the number of clusters since the number of users per cluster drops and only the more "pure" members remain. Also, in general, the BCubed value tends to drop as we increase the number of clusters. The reason being the big drop of users count per cluster in both cases, even if the accuracy, or precision, of each cluster increases.

Similar behaviour can be seen when studying the variation of the edge_weight threshold parameter, as shown in Figure 3. Purity also tends to increase with increasing the edge_weight threshold, as the noisy edges get eliminated with increasing the threshold of the edge's weight to be considered in the analysis. The BCubed value tends to decrease with increasing the threshold, as the number of users per community tends to drop as well after eliminating more of the considered noisy edges.

We also investigated how using different communication dimensions (hashtags, mentions, retweets... etc) affects the performance. For example, we found that eliminating the replies dimension from the analysis enhances the overall performance, as shown in the dashed lines the graphs in Figure 2. We also found that removal of the frequent, or common, entities from the data set improved the quality of the communities. The dashed graphs in Figure 3 show the effect of removing the most commonly used hashtags (e.g. #UK) and mentions (e.g. youtube). These common entities might be of general relevance to all users in all communities, so including them in the analysis process might produce erroneous results.

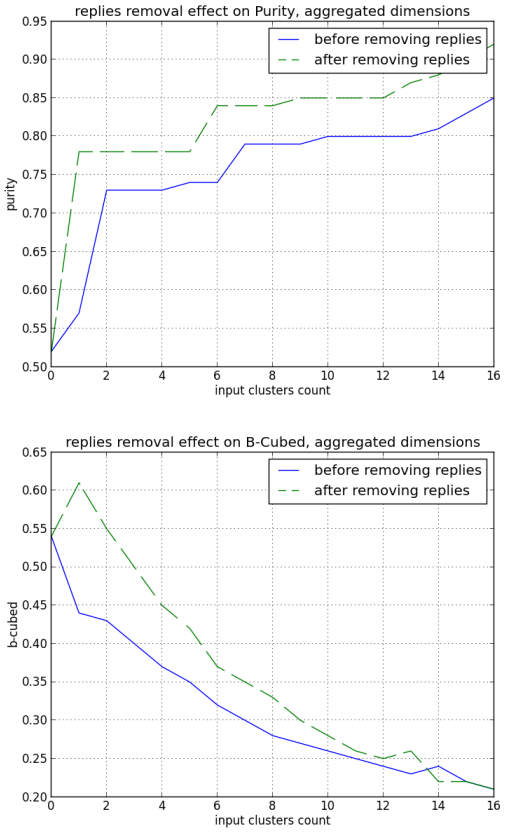

Figure 2. Purity and BCubed for varying numbers of clusters

We conducted various other experiments that are not reported here for space restrictions. We summarize some of the main findings here with more details provided in These are described in more detail in [10]. For the similarity dimensions, the mentions dimension performed better individually, as the number of mentions in the system was big in comparison to the others. The hashtags and links dimensions performed better when aggregated. Also, for the interaction dimensions, the replies dimension performed very poor in general, as replies were found to drive isolated specific discussions. Moreover, the aggregation of all interaction dimensions performed better than the individual dimensions. This is probably because the number of edges per dimension was low, so aggregating the efforts gave better results.

C. Evaluating Community Classifier

For evaluating the community classifier we do not use the political affiliation as labels, but rather use the community tags generated by the community detection algorithm outlined above. In our experiments this was based on 224 users automatically

[3]tweetminster.co.uk

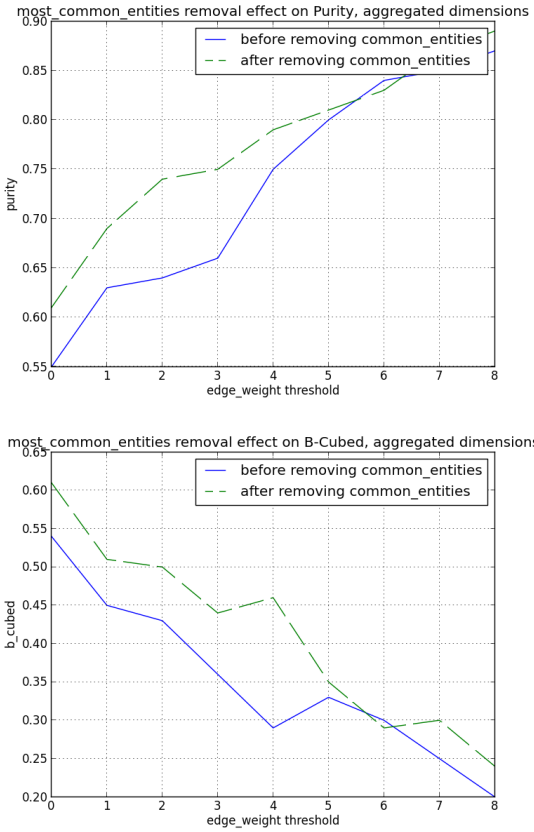

Figure 3. Purity and BCubed for different threshold values

clustered into 5 different communities with sizes 73, 50, 51, 21 and 29 users based on an edge_weight threshold value of 4. The data set was divided into training and test sets across different time spans for the week-long data, the total number of tweets, 9266, has been divided into a training data set of 5040 tweets covering the first half of the time-span, and with time span segments for the second half with sizes 823, 945, 1013, and 820 tweets. 625 tweets were eliminated as they were from users used in the training phase.

As can be seen at Figure 4, we notice that increasing the time span period difference between the testing and training data sets decreases the accuracy of the analysis. Clearly, the detected communities best describe the characteristics of the data space at the given time period.

VI. Summary and Future Work

This paper presented and evaluated a generic approach for multidimensional community detection from Twitter content. We evaluated the quality of the generated communities using a number of measures on a data set of UK political tweets. Our results indicate that the approach is practical. They showed that certain similarity dimensions, e.g. topic hashtags and links tend to perform better when aggregated. In addition, certain interaction dimensions, e.g. replies between users, tend to perform worse and result in isolated communities. Moreover, they showed that removing the most_common_entities, like the hashtags or mentions of very high frequency in the data set, enhances the accuracy. Our future work in this direction includes utilising

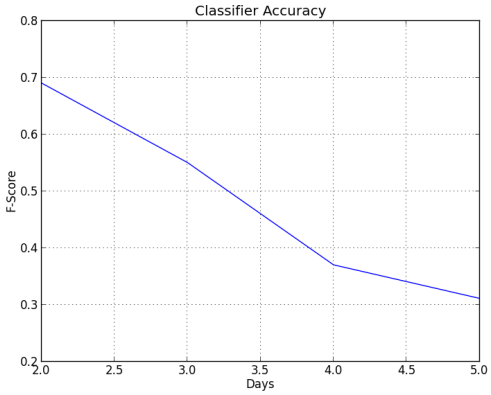

Figure 4. The relationship between the accuracy and time span difference between testing and training data sets.

overlapping community detection algorithms, developing more advanced dimension aggregation techniques, and also evaluating the approach on other data sets and conditions.

We also developed and evaluated a Bayesian community classifier that maps new users to the detected communities. The classifier can be used to address the issues of dynamicity and evolution in the context of Social Media. This approach builds up on Tang and Liu's [2] proposed snapshot approach, by providing a practical classifier based solution. Our results indicate that although the classifier starts with good accuracy, its accuracy deteriorates as the time elapsed from detecting the communities and classifier application increases. This indicates that further investigations are needed into the practicality of such developing and using such classifiers. Our future work in this area includes modeling the dynamic evolution of the detected communities by further exploiting the snapshot approach, and the development of the appropriate dimension representation and aggregation techniques for community mapping.

References

[1] L. S.-L. Miller McPherson and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, pp. 415–444, 2001.

[2] L. Tang and H. Liu, *Community Detection and Mining in Social Media*. Morgan and Claypool Publishers, 2010.

[3] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, pp. 7821–7826, 2002.

[4] S. R. Ravi Kumar, Prabhakar Raghavan and A. Tomkins, "Extracting large-scale knowledge bases from the web," *Proceedings of the 25th VLDB Conference*, pp. 639–650, 1999.

[5] X. W. Jiliang Tang and H. Liu, "Integrating social media data for community detection," *Modeling and Mining Ubiquitous Social Media*, pp. 1–20, 2011.

[6] D. C. Maria Giatsoglou and A. Vakali, "Community detection in social media leveraging interactions and intensities," *Web Information Systems Engineering - WISE 2013 Proceedings, Part 2*, pp. 57–72, 2013.

[7] S. Papadopoulos, "Community detection in social media," *Center for Research and Technology - HELLAS, Information Technology Institute*, 2011.

[8] M. K. Ronald Burst and S. Tasselli, "Social network analysis: Foundations and frontiers on advantage," *The Annual Review of Psychology*, 2013.

[9] J. A. Enrique Amigo, Julio Gonzalo and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Departamento de Lenguajes Sistemas Informaticos UNED, Madrid, Spain*, 2009.

[10] N. Zalmout, "Mining the Social Web: Community Detection in Twitter, and its Application in Sentiment Analysis," Master's thesis, Department of Computing, Imperial College London, London, SW7 2AZ, UK, 2013.