Social Event Detection via sparse multi-modal feature selection and incremental density based clustering

Sina Samangooei
ss@ecs.soton.ac.uk

Jonathon Hare
jsh2@ecs.soton.ac.uk

David Dupplaw
dpd@ecs.soton.ac.uk

Mahesan Niranjan
mn@ecs.soton.ac.uk

Nicholas Gibbins
nmg@ecs.soton.ac.uk

Paul Lewis
phl@ecs.soton.ac.uk

Electronics and Computer Science, University of Southampton, United Kingdom

ABSTRACT

Combining items from social media streams, such as Flickr photos and Twitter tweets, into meaningful groups can help users contextualise and effectively consume the torrents of information now made available on the social web. This task is made challenging due to the scale of the streams and the inherently multimodal nature of the information to be contextualised. We present a methodology which approaches social event detection as a multi-modal clustering task. We address the various challenges of this task: the selection of the features used to compare items to one another; the construction of a single sparse affinity matrix; combining the features; relative importance of features; and clustering techniques which produce meaningful item groups whilst scaling to cluster large numbers of items. In our best tested configuration we achieve an F1 score of 0.94, showing that a good compromise between precision and recall of clusters can be achieved using our technique.

1. INTRODUCTION

In their June 2013 WWDC keynote, Apple announced a new photo collection feature for their iOS mobile operating system. With the evocative tag-line "Life is full of special moments. So is your photo library", Apple noted the importance of clustering social streams by their belonging to some real life event. This, along with the plethora of mobile and desktop applications which offer some degree of event detection in user photo streams, demonstrates that detecting events in multimedia streams has real practical utility for users. In this work we present our approach to achieving clustering of social media artefacts towards addressing task 1 in the Social Event Detection (SED) challenge of the Mediaeval 2013 multimedia evaluation [3]. Task 1 asks that a collection of Flickr photos be organised into events such that events are defined as "events that are planned by people, attended by people and the media illustrating the events are captured by people". The Flickr items in this task contain metadata beyond the content of the image itself. Namely, the Flickr photos are guaranteed to include accurate: Flickr IDs, user IDs and time posted. The photos may also contain in varying degrees of accuracy location information, time taken according to the camera, and textual information including title, tags and a description.

2. METHODOLOGY

The overarching strategy of our technique is the construction of a square sparse affinity matrix whose elements represent the similarity of two items of social media. Once this object is created, two clustering algorithms are applied. However, the creation of such a matrix for any number of items beyond a trivial number is a time consuming $O(n^2)$ operation which scales poorly. Therefore, the first stage of our process was the efficient construction of such an affinity matrix.

Given the SED2013 training set we know for $\sim 300,000$ objects there exist $\sim 14,000$ clusters. The average number of items per cluster in the training set is ~ 20. From this information we know that the similarity between most objects must be 0 if similarity is a reasonable indication of cluster membership. However, inducing this sparsity after feature extraction and comparison of the social media objects is without merit. At this point the expensive operation has already been performed and is only forced to 0 after this fact. To address this issue, we construct a Lucene [1] index of the items to be clustered. The items are indexed using their metadata, each meta data component is given a field in the Lucene index. Then, for each item in the dataset we construct a custom Lucene query based on the item's metadata, receiving an artificially limited number of documents. We then extract features from, and compare distances using, only the documents returned by this query. Once the work is done to construct this Lucene index, this operation has a complexity of $O(n)$ which allows a much faster construction of the affinity matrix.

Once documents are filtered using Lucene, the affinity matrix is constructed. The items being clustered are inherently multi-modal. These modalities include time information (both posted and taken), geographic information, textual information (tags, descriptions and titles) as well as the visual information of the Flickr photos themselves. Any of these modalities might serve as a strong signal of cluster membership. Photos taken in the same place, or at the same time, or containing similar text might all serve as strong indication of these photos being of the same event. However on their own the features might also serve to confuse unrelated events, for example, two events happening on a Friday, but one in Nottingham and one in London. Therefore, the first stage in the construction of a unified affinity matrix is a separate affinity matrix for each of these features, while the second step is the correct combination of the affinity matrices. Inspired by Reuter and Cimiano [2] we use a logarithmic distance function for our two time features. We also use a logarithmic distance function for our geographic Haversine based distance function. Fundamentally, this forces distances and times beyond a certain distance to count as being infinitely far, or as having 0 similarity. For the textual features we use the TF-IDF score

MediaEval 2013 Workshop, October 18-19, 2013, Barcelona, Spain

[1] `http://Lucene.apache.org/core/`

Table 1: Results from our four runs.

	DBSCAN (best-weight)	Spectral (best-weight)	DBSCAN (average-weight)	Spectral (average-weight)
F1	0.9454	0.9114	0.9461	0.9024
NMI	0.9851	0.9765	0.9852	0.9737
F1 (Div)	0.9350	0.8819	0.9358	0.8663
Random Base F1	0.0589	0.0580	0.0597	0.0569
Div F1	0.8865	0.8534	0.8864	0.8455

with the IDF statistics calculated against the entire corpus of Flickr objects. We also experimented with SIFT based visual features for image feature affinity matrix construction. However, we found this feature only made F1 scores worse in the training set and the visual features were completely ignored in all submitted runs against the test set. If any given feature is missing or empty for either object represented by a particular cell in the affinity matrix, for the purpose of the sparse affinity matrix it is treated as being "not present" rather than 0 similarity. The distinction here is important for how these affinity matrices are combined.

While [2] constructed vectors of similarity, we choose to fuse the similarity features into a single similarity score to construct a fused affinity matrix. We experimented with various feature-fusion techniques to combine them. Combination schemes including: *product*, *max*, *min*, and *average* were tested, average was shown to work well. Different feature weightings were also investigated. To calculate the combined affinity w_{ij} of the i^{th} image with the j^{th} image, the feature affinities w_{fij} for all features $f \in F_{ij}$ were used where F_{ij} is all the features which had values for both images i and j, i.e. those features "present" in the affinity matrix:

$$w_{ij} = \sum_{f}^{F} p_f w_{fij}, \quad \sum_{f}^{F} p_f = 1 \tag{1}$$

where p_f is the weight of a given feature f. The final affinity matrix produced by this process was used by the clustering techniques below.

2.1 Event Clustering

The exact number of events in the corpus was unknown and hard to estimate. We explored clustering techniques which work without an explicit k. The first technique we experimented with was a modified version of the classic Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm, implementing a fast version of the neighbourhood selection stage which worked against sparse affinity matrices. Secondly, we explored more sophisticated spectral clustering techniques which interprete the affinity matrix as weights of edges on a graph and uses graph theoretic techniques to automatically estimate the number of clusters present in the graph. The data items are projected into a metric space which ensures better separation between the clusters. We used a classic cosine similarity nearest neighbour DBSCAN to cluster in the spectral space.

In response to challenges of scale that both DBSCAN and Spectral clustering methods face given enough data, we developed a general incremental clustering technique which takes advantage of the streaming nature of social media data. Namely, if we assume the data appears in an incremental fashion, we can cluster the data in small windows. If we then allow the windows to grow and perform the clustering again, we might notice that certain clusters and their members are stable across window growth. This stability could be defined as the cluster members not changing whatsoever, or a relaxed form could define stability as paired clusters with high overlap. Regardless, once a cluster is defined as stable those items could be removed and not involved in the next window size increase. In this way, the effective number of items to be clustered in any given round will increase as items arrive but will also decrease as clusters become stable. In this way we were able to successfully apply the spectral clustering algorithm to the large training set of 300,000 items unlike Petkos et al. [1] who also applied a spectral clustering technique to event detection, but on a comparatively small sample size.

3. EXPERIMENTS AND RESULTS

For the runs submitted we first had to explore and set the various parameters of our approach against the training set. Our first parameter was the weightings with which the feature affinity matrices were combined. We performed a search across the weighting simplex and found that for the training set the best F1 score was achieved when the features were weighted as: $p_{timetaken} = 3$, $p_{timeposted} = 0$, $p_{geolocation} = 1$, $p_{description} = 1$, $p_{tags} = 3$, $p_{title} = 0$. However, we noticed that the F1 achieved by the top feature weightings on the simplex were very similar. Therefore, to avoid over-fitting we selected the top 1000 F1 ranked selections on the weightings simplex and got the weightings average in the training set. This resulted in the features weighted as: $p_{timetaken} = 2.1$, $p_{timeposted} = 1.8$, $p_{geolocation} = 1.4$, $p_{description} = 0.7$, $p_{tags} = 1.7$, $p_{title} = 0.3$. We performed a similar line search to find the optimal values for DBSCAN's parameters ($eps = 0.45$, $minpts = 3$) and Spectral Clustering's parameters ($eigengap = 0.8$, $eps = -0.6$). The 4 runs we submitted for the MediaEval 2013 SED Task 1 were therefore: DBSCAN with the best weighting, Spectral clustering with the best, DBSCAN with the average weighting and spectral clustering with the average. All these runs were performed using our incremental clustering technique. An overall summary of the results from the runs can be seen in Table 1.

4. ACKNOWLEDGMENTS

The described work was funded by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreements 270239 (ARCOMEM), and 287863 (TrendMiner).

5. ADDITIONAL AUTHORS

Additional author: Jamie Davies (email: `jagd1g11@ecs.soton.ac.uk`) Neha Jain (email: `nj1g12@ecs.soton.ac.uk`), and John Preston (email: `jlp1g11@ecs.soton.ac.uk`)

6. REFERENCES

[1] G. Petkos, S. Papadopoulos, and Y. Kompatsiaris. Social event detection using multimodal clustering and integrating supervisory signals. In *Proc. ICMR*, 2012.

[2] T. Reuter and P. Cimiano. Event-based classification of social media streams. In *In Proc. ICMR*, 2012.

[3] T. Reuter, S. Papadopoulos, V. Mezaris, P. Cimiano, C. de Vries, and S. Geva. Social Event Detection at MediaEval 2013: Challenges, datasets, and evaluation. In *MediaEval 2013 Workshop*, Barcelona, Spain, October 18-19 2013.