Graph-Based Multimodal Clustering for Social Event Detection in Large Collections of Images

Conference Paper · January 2014

DOI: 10.1007/978-3-319-04114-8_13

Georgios Petkos
Information Technologies Institute (ITI)
24 PUBLICATIONS **724** CITATIONS
SEE PROFILE

Symeon Papadopoulos
The Centre for Research and Technology, Hellas
256 PUBLICATIONS **4,720** CITATIONS
SEE PROFILE

Ioannis (Yiannis) Kompatsiaris
The Centre for Research and Technology, Hellas
1,023 PUBLICATIONS **14,035** CITATIONS
SEE PROFILE

Graph-Based Multimodal Clustering for Social Event Detection in Large Collections of Images

Georgios Petkos, Symeon Papadopoulos, Emmanouil Schinas,
and Yiannis Kompatsiaris

Information Technologies Institute, Centre for Research and Technology Hellas

Abstract. A common approach to the problem of SED in collections of multimedia relies on the use of clustering methods. Due to the heterogeneity of features associated with multimedia items in such collections, such a clustering task is very challenging and special multimodal clustering approaches need to be deployed. In this paper, we present a *scalable graph-based multimodal clustering* approach for SED in large collections of multimedia. The proposed approach utilizes example relevant clusterings to learn a model of the "same event" relationship between two items in the multimodal domain and subsequently to organize the items in a graph. Two variants of the approach are presented: the first based on a batch and the second on an incremental community detection algorithm. Experimental results indicate that both variants provide excellent clustering performance.

Keywords: Social media, Social event detection, Multimodal clustering.

1 Introduction

The wide availability of low cost media capturing devices along with the advent and massive adoption of online social publishing platforms have significantly transformed the behaviour of casual online users, turning them to a large extent into media content producers. Considering the huge growth that social media have had in the recent years, it is clear that there is a growing amount of diverse content that covers a huge range of real-world activities and that can be used to "sense the world". For instance, web content obtained from social media has been used in applications such as detecting breaking news [24], landmarks [12] or more recently social events [20,19]. Nevertheless, such content is often noisy, presents large heterogeneity and is often not well-structured and therefore presents many challenges to analysts.

In this paper we discuss the problem of discovering social events in collections of web multimedia. By social events, we mean events that are attended by people and are represented by multimedia content shared online. Instances of such events could include concerts, sports events, public celebrations or even protests. Formally, given a photo collection denoted by $\mathbb{P} \triangleq \{p\}$, where p stands for a photo and its metadata (owner, time, tags, location if available, etc.), we consider methods that produce a set of photo clusters $\mathbb{C} \triangleq \{c\}$, each cluster c comprising only photos $\mathbb{P}_c$ associated with a single event. Various approaches

C. Gurrin et al. (Eds.): MMM 2014, Part I, LNCS 8325, pp. 146–158, 2014.

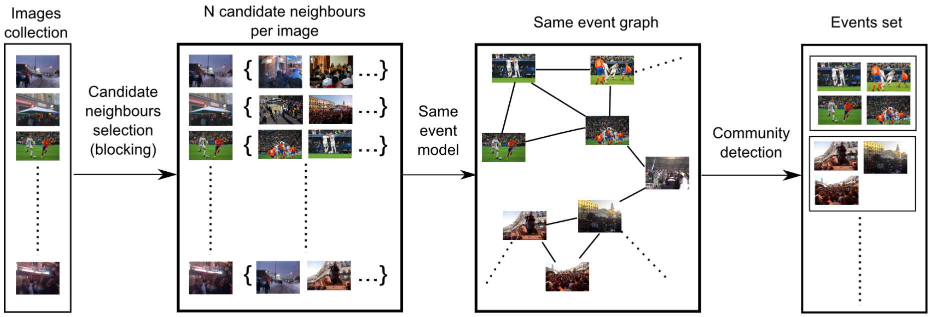

Fig. 1. Schematic representation of the proposed approach

have been proposed for tackling the problem of Social Event Detection (SED) in collections of multimedia. Some of them rely on auxiliary information obtained from online sources and directories, such as LastFM, EventFul and DBPedia [4,13]. Such approaches retrieve information about actual events or venues and attempt to match this information with items in the collection. A second class of methods do not utilize external information sources; instead, they attempt to cluster the items of the collection, so that the resulting clusters represent events [2,22,26,23]. This is clearly a more general approach than the first one, as it is does not rely on potentially unavailable external information.

In this paper we present a novel scalable approach for *multimodal* clustering that we apply to the problem of SED in collections of multimedia. The proposed approach utilizes a model that has been trained to predict whether a pair of items (images) belong to the same cluster (event), using as input the set of per modality distances between the pair of items (images). In the rest of the paper, we will refer to this as a "same event" (SE) model. We compute the predictions of this SE model for pairs of items in our collection and we organize the items in a graph. This graph has a node for each image of the collection and an edge between two nodes indicates that the prediction of the SE model for the corresponding pair of images was positive. Additionally, similarly to [26], we adopt an appropriate strategy to avoid having to predict the SE relationship between each possible pair of images, thereby making the approach applicable to large datasets. This involves computing the output of the SE model for each item only against its nearest neighbours according to each modality. Finally, we apply a community detection algorithm on the resulting graph. The proposed approach comes in two variants; in the first, batch community detection is applied on the graph, whereas in the second an incremental algorithm is applied. Figure 1 presents an overview of the proposed approach. Interestingly, the proposed approach is applicable to a variety of other multimodal clustering algorithms, as long as an example clustering that can be used for training the SE model is available. To the best of our knowledge, the presented *incremental* approach is one of the first to tackle the problem of incremental multimodal clustering.

The rest of this paper is structured as follows. Section 2 presents some related work. Subsequently, Section 3 describes in more detail the proposed approach. Section 4 presents some empirical results and finally Section 5 provides the conclusions and discusses some future work.

2 Related Work

2.1 Social Event Detection

The task of SED in collections of multimedia has attracted a lot of interest in the last years. Indicative of this is that a relevant task has been organized as part of the MediaEval Benchmark in 2011 [20] and 2012 [19], the data of which is used in the current work for evaluation. Most approaches commonly apply a sequence of clustering or filtering operations in order to obtain the required set of events, e.g. [21].

Of particular interest for this work are approaches that treat the problem as a clustering task. Since items in this problem are typically multimodal, this is a *multimodal clustering* task. Multimodal clustering is a challenging task, for which various approaches have been proposed (see Section 2.2). The essence of the problem is that appropriate similarity measures are required that take into account heterogeneous modalities. Such similarity measures in which the contribution of each individual modality is appropriately weighted may be determined either by prior knowledge or a search process. A solution to this problem that was utilized by previous approaches to event detection [2,26,23] uses an auxiliary example clustering, in which clusters are known to represent events, as a supervisory signal. Such a known clustering (collection of events) has been used in two different ways, in conjunction with different clustering procedures.

In the first, which will be refered to as the **item-cluster** approach, a part of the data is used to obtain an SE model that predicts whether an item belongs to some event, using as input the set of distances between the item and a prototype (aggregate) representation of the event. The auxiliary data can be used to obtain the training data for such a model by computing aggregate representations for part of the items that belong to the collection of events and then sampling from the rest of the items. This is the approach followed by [2] and [26]. Subsequently, this model can be used for clustering the incoming items in an incremental fashion, by computing the prediction of this SE model for the sets of similarities between the new item and the prototype representations of the already identified clusters. The set of predictions is either used to assign the item to the best matching cluster/event or to generate a new event. The prototype representation of each event is the average of the items assigned to it.

The second way that such an example clustering can be used will be refered to as the **item-item** approach. It involves learning a similar SE model that predicts whether two items, instead of an item and a cluster prototype, belong to the same cluster, again taking as input the corresponding set of similarities [23,2]. An item-item approach avoids the need to maintain a prototype representation, but requires a different clustering procedure. In the case of [2], the predictions of the model between an item and all other items that were assigned to some cluster are used to obtain an estimate of cluster membership by simple averaging. In [23], the SE model is applied on all pairs of items in the collection. For each item, a vector is maintained that contains the SE relationship of that item to all other items in the collection. Finally, it is assumed that items that belong to the same cluster will have similar sets of SE relationships to the set of items in the collection and therefore a final assignment to events is obtained by clustering the corresponding vectors. However, this approach is clearly not scalable as it has quadratic complexity to the collection size; it requires the evaluation of the SE model for all pairs of images. Moreover, it requires maintaining vectors of which the size increases with the dataset size. In this work, we propose a scalable item-item approach that deals with the problem of quadratic complexity by utilizing a candidate neighbour selection step and with the problem of large size neighbourhood vectors by utilizing a graph to store the SE relationships.

2.2 Multimodal Clustering

The task of clustering items which are expressed through heterogeneous modalities has commonly been treated using fusion approaches, of which two common classes exist: early and late fusion [29]. Early fusion approaches combine the features/modalities in some specific representation before the main analysis process - in this case clustering - is applied. Late fusion methods apply the main processing function to each modality separately and instead combine the results obtained for each modality. A recent spectral clustering approach that combines modalities at an intermediate level is presented in [5]. An important set of methods includes probabilistic approaches that utilize graphical models representations [11,3]. Finally, it is worth noting the connection between multimodal clustering and ensemble clustering [7], in which the goal is to combine a set of clusterings (produced e.g. from a set of different algorithms) in some optimal manner. A multimodal clustering problem can be cast into an ensemble clustering problem by performing a set of clusterings - e.g. one per modality - and then combining the individual clusterings. This is also one of the proposed approaches for event detection in [2].

2.3 Community Detection

Community detection is used to cluster the SE graph that is constructed as part of the proposed approach. There is a large body of work in this area, and a few comprehensive surveys on the topic [6,17]. The bulk of existing work examine the problem in the context of static graphs (i.e. batch mode). Recently, the problem of dynamic community detection (i.e. online mode) is increasingly gaining importance. The use of community detection for event detection is not new. For instance, [22] applies community detection on image similarity graphs for the extraction of landmarks and events in large image collections.

3 Approach Description

The proposed approach, has already been outlined in the introduction and in Figure 1. It consists of three components: The first retrieves for each image in the collection a set of images that are candidates for belonging to the same event. Subsequently, an SE model is employed to predict which of these candidate SE relationships are likely to hold. This leads to the generation of a graph that represents the SE relationships between the images of the collection. At the final step, a community detection algorithm is applied on this graph to extract clusters of images that constitute the detected events. We examine two variants, using either a batch or an incremental community detection algorithm. In the following, we provide more details on each of these components.

Candidate Selection. An important component of the proposed approach relies on the capability for fast retrieval of same-event candidates per incoming item for each of the different modalities, so that the SE model can be applied only on the candidate items. This is a technique, which in the context of related work, was first utilized by [26], but was used before in the field of record linkage and is commonly refered to as *blocking* [25]. Blocking greatly reduces the number of SE predictions required and can result in scalable implementations. To this end, all items in the collection are indexed in appropriate structures. For instance, textual metadata (title, tags) are inserted in a full-text index (e.g. Lucene) for rapidly retrieving documents with high textual similarity to a query document. Other metadata such as time and location are indexed by use of B-Trees, whereas visual similarity can be indexed by a variety of content-based retrieval approaches (in our implementation, we opted for the use of Product Quantization in tandem with Asymmetric Distance Computation [9]).

Multimodal "Same-Event" Model. The item-cluster comparison has the disadvantage that it relies on the prototype representation of the cluster being an accurate representation of the underlying event. Thus, if there are incorrectly assigned items, this may significantly affect the accuracy of the prototype representation. Due to the potential problem with averaging incorrectly assigned items into a prototype cluster representation, we adopt an item-item approach. Formally, given two images, p^i and p^j, which are expressed through a set of k features $p^i_1...p^i_k$ and $p^j_1...p^j_k$ respectively, we compute the vector that contains the per-modality distances $d(p^i, p^j) = [d(p^i_1, p^j_1), d(p^i_2, p^j_2)...d(p^i_k, p^i_k)]$. The SE model is then a function of $d(p^i, p^j)$:

$$SEM(p^i, p^j) = f(d(p^i, p^j))$$

and predicts if the images p^i and p^j belong to the same event, i.e. it is a classifier and for the following we assume that a predicted value of $+1$ denotes that the two images belong to the same event, whereas a value of -1 denotes the opposite. Having a separate example clustering, in which each cluster represents an event, it is straightforward to obtain training data for the SE model. In our scenario, where we use images from the social media, Flickr in particular, we use the following set of features and similarity measures:

1. *Uploader identity.* The identity of the Flickr user who uploaded the picture. We utilize a binary similarity measure where the value 0 is used for pairs of pictures uploaded by different users whereas the value 1 is used for pairs of pictures uploaded by the same user.
2. *Actual image content.* From each image we extract one global descriptor, GIST [16] and a set of local descriptors, SURF [1], which we aggregate using the VLAD scheme [10]. The distance between a pair of GIST descriptors or SURF-VLAD descriptors is computed using Euclidean distance.
3. *Textual information.* Images uploaded by users are typically accompanied by a title, a description and a set of tags. We utilize both the Term Frequency - Inverse Document Frequency and BM25 [14] weighting schemes, resulting in six textual-based distance elements per pair of items.
4. *Time of media creation.* Instead of using the difference of the time of creation between a pair of items, we use three binary distances. The first takes the value 0 if the difference in time of creation is larger than 6 hours and the value 1 if it is smaller. The other two are similar but set the threshold to 12 and 24 hours respectively. This is an approach that empirically led to better results when training the classifier compared to using the absolute time distance (various time scales were tried).
5. *Location.* Not all items in the collection come with location data. For pairs of items that do come with location information though, we compute two distance measures. The first is the geodesic distance in kilometers and the second is a boolean indicator which is 1 if the geodesic distance is smaller than 1 kilometer and 0 if it is larger.

In order to handle the case of missing location information we train two models. The first takes as input all the aforementioned distances and is used for pairs of items that both come with location information. The second takes as input the same set of distances except the location-based and is used when at least one of the two items for which SE model is evaluated does not have location information. In other scenarios, different types of data may be available or used, however, the presented approach is applicable without significant changes.

Graph-Based Clustering. The final part of the algorithm clusters the items in the collection. Our algorithms organize the items in a graph, in which the existence of an edge indicates that the SE model has predicted that the corresponding items belong to the same event. More formally, given a set of images to be clustered P, we generate a graph $G = (P, E)$, whose set of vertices is P, i.e. the set of photos to be clustered. The set of edges E, contains the pairs of images (p^i, p^j) for which $SEM(p^i, p^j) = +1$ and either p^i is a candidate neighbour of p^j or p^j is a candidate neighbour of p^i.

A community detection algorithm is then applied on the graph. The proposed approach comes in two variants, which utilize either a batch or an incremental community detection algorithm and are described below.

Batch community detection. In the first variant of our approach, a batch community detection algorithm is applied. The selected algorithm is the Structural Clustering Algorithm for Networks (SCAN) [30]. This choice was motivated by some desirable properties of the algorithm: (a) computational efficiency, (b) possibility to leave spuriously connected nodes out of the clustering, (c) it can identify not only communities, but also hubs and outliers. This last property, is particularly interesting for the following reason. The predictions of the SE model will inevitably be imperfect. If they were perfect, then it would suffice to find the connected components of the graph. Therefore, in the case of noisy edges, one has to take into account the case of nodes that are sparsely connected to some cluster(s) due to incorrect predictions of the SE model. Such nodes are likely to be classified as outliers and can therefore be assigned to a separate cluster/event rather than to their erroneously adjacent clusters. Outliers on the other hand are assigned to the adjacent community to which it has the largest number of connections. SCAN is controlled by two parameters, μ and ϵ, which determine the minimum number of nodes and the minimum "tightness" in a community.

Incremental community detection. In the second variant of our approach, an incremental algorithm is applied. There are few incremental community detection algorithms in the literature. We opted for Quick Community Adaptation (QCA) [15], due to its efficiency and simplicity of implementation. QCA is an expansion of a previous physics-inspired, non-incremental approach [31]. It maintains the detected community structures up-to-date by appropriate processing operations in the event of four different graph changes: a) new node creation, b) new edge addition, c) node removal and d) edge removal. In short, in all operations, forces that attempt to pull a node inside adjacent communities are computed and each node is pulled to the community that applies the strongest attracting force. In all operations, the forces are appropriately computed for all affected nodes. For more details please see [15]. In our scenario we do not consider node or edge removals: neither is it likely that an item is removed from the collection, nor would an SE relationship between a pair of items be evaluated a second time. Finally, it should be noted that QCA is parameter free.

4 Experimental Evaluation

The proposed graph-based multimodal clustering method was tested on data from the 2012 MediaEval SED task [19]. This consists of three challenges that call for the detection of social events of specific types (there are three target types of events: soccer events, technical events and events related to the Indignados movement) that took place in specific geographic locations in a collection of approximately 167,000 images collected from Flickr. Out of those, 7,779 images did indeed belong to one of the 149 target events of the ground truth (79 soccer, 18 technical and 52 Indignados events). For examples of images and events of these classes, please see [18].

The SE models are trained using a Support Vector Machine classifier (SVM), notably the Weka [8] implementation. Various other classifiers were tested but the SVM classifier produced the best results. Its average classification accuracy on a sequence of sets on test data was 98.58%. A close second in accuracy among the tested algorithms was a decision tree, which resulted in an average accuracy score of around 96.62%.

Table 1. NMI for the graph-based batch and incremental methods, as well as an item-cluster based method (only event images are used in these runs)

	Batch	**Incremental**	**Item-cluster [2]**
Avg.	0,924	0,934	0,898
Std.	0,019	0,021	0,027

Table 2. NMI using both event and non-event images. The set of images is randomly selected from the complete 2012 challenge dataset.

Labelling acc.	**# images**	**Batch**	**Incremental**	**Item-cluster [2]**
0,95	15352	0,4824	0,5164	0,3954
0,90	22876	0,3421	0,3683	0,2899

As mentioned, to avoid evaluating the SE relationship between all possible pairs of items, a candidate selection mechanism is utilized. For each item, we retrieve the items with which it has the largest similarity with respect to the textual features (50 items chosen), time (150 items chosen), location (50 items chosen, if location is available), GIST (50 items chosen) and VLAD/SURF (50 items chosen). Items are indexed according to the different modalities in appropriate structures, so that the nearest neighbours for each modality can be obtained very fast. For the batch procedure, all items are indexed before the main processing is carried out, whereas for the incremental procedure items are indexed as they arrive. The following index implementations were used: Lucene for the textual features, a MySQL database for time and location features and the approximate nearest neighbor search method of [9] for the visual features.

As mentioned above, the SCAN algorithm used by the batch method has two parameters that may affect the quality of the final result. For these experiments, the ϵ parameter was empirically set to 0.7 and the μ parameter was set to 3. This experimentation was carried out on a separate random sample of events.

In a first set of experiments, the images that belonged to a random subset of 30 events (out of the 149 events in the collection) were used to train the SE model, whereas the images in the remaining events were used for testing the proposed methods. We generated 10 random tasks in this manner. The utilized measure of clustering quality is Normalized Mutual information (NMI). The two variants of the proposed algorithm are tested against the item-cluster approach of [2], where a threshold of 0.5 was used on the output of the SE model. That is, an incoming item was assigned to the best matching cluster if the probability output by the model was above 0.5, otherwise it was used to generate a new event. Table 1 presents the obtained results. The results demonstrate that both variants of the graph-based approach achieve very high clustering accuracy with an NMI clearly above 0.9. An important thing to notice is that the performance of the incremental algorithm is by no means inferior to the performance of the batch algorithm. Contrary, on average, the performance of the incremental algorithm is slightly higher than that of the batch algorithm, even though the difference is not statistically significant. The performance of both graph-based approaches is

Table 3. NMI of proposed methods using limited sets of features

	Batch	**Incremental**
Visual	0,8020 ± 0,0193	0,8179 ± 0,0151
Textual	0,7925 ± 0,0255	0,7792 ± 0,0310
Visual + time	0,9244 ± 0,0195	0,9360 ± 0,0183
Textual + time	0,9016 ± 0,0173	0,9049 ± 0,0209

higher than the performace of the tested item-cluster approach and the difference is statistically significant at a 0.95 confidence level.

In the previous experiments, we clustered sets of images that do belong to the target events. In the following, we examine the scenario that non-event pictures are also included in the collection to be clustered. More particularly, we use the data from the 2011 challenge for training. The test set is obtained from the complete collection of 167,000 images of the 2012 challenge as follows. Each image is randomly labeled as event or non-event with the probability of being labeled correctly being p and use only the images that have been labeled as events. This experiment will provide further comparison of the methods and will also test the robustness of the method in the presence of many spurious images. We test two different values for p: 0,95 and 0,9 which almost double and triple respectively the number of images to be clustered (originally 7,779 images belong to some event). It should be noted also, that in some recent work [27] we attempt to classify images as either representing a social event or not, we obtain accuracy values similar to these values (in particular 0,8962). The results can be seen in Table 2. The NMI has dropped significantly, however, we still observe that the item-item approaches are superior to the item-centroid approaches and that the parameter free incremental method is superior to the batch method. Finally, it should be noted that although these experiments are closer to the initial MediaEval SED challenge scenario, the results cannot be compared to results reported for the challenge. In order to do this, one would have to also perform further processing, e.g. to filter events or images as representing an event of a particular type. For an application of this method to the challenge together with these steps please see [28].

We also investigate the importance of the set of features used for computing the predictions of the SE model. We conducted the same run of experiments on limited sets of features. The results can be seen in Table 3. These surprisingly good results, especially for the case that only visual or textual features are used, are due to the fact that blocking is still applied. In the same experiments executed without blocking, the average NMI obtained using only visual features was 0,030 and 0,7148 for the textual features. From Table 3 it can also be seen that time is a crucial feature for the performance of the SE model.

Moreover, we examine whether the type of events used for training the SE model is important for detecting events of a different type. In the previous run of experiments, the 30 randomly sampled events were randomly chosen from the three categories (soccer, technical and Indignados). We now use all available

Table 4. NMI achieved by training and testing on different types on events

<table>
<tr><th colspan="4">Batch</th></tr>
<tr><th></th><th>Soccer</th><th>Technical</th><th>Indignados</th></tr>
<tr><td><b>Soccer</b></td><td>-</td><td>0,8658</td><td>0,8494</td></tr>
<tr><td><b>Technical</b></td><td>0,7967</td><td>-</td><td>0,8977</td></tr>
<tr><td><b>Indignados</b></td><td>0,9645</td><td>0,8456</td><td>-</td></tr>
<tr><th colspan="4">Incremental</th></tr>
<tr><th></th><th>Soccer</th><th>Technical</th><th>Indignados</th></tr>
<tr><td><b>Soccer</b></td><td>-</td><td>0,8892</td><td>0,8667</td></tr>
<tr><td><b>Technical</b></td><td>0,7661</td><td>-</td><td>0,7735</td></tr>
<tr><td><b>Indignados</b></td><td>0,9845</td><td>0,8482</td><td>-</td></tr>
</table>

events from one type of events to learn the SE model, which we then use to cluster the items that belong to one of the other two types of events. The results can be seen in Table 4. Clearly, the type of event used for training does have an effect on the quality of the produced clustering for different types of events. Most notably, using technical events for training the SE model and then using it for producing clusters that represent other types of events produces lower quality results using either variants of the graph-based method. However, it is noteworthy that in some cases training with a completely different type of events can lead to very high performance, e.g. in the case that Indignados events are used for training and soccer events are detected with an NMI of over 0.96 for both variants of the approach.

5 Conclusions and Future Work

This paper proposed two variants of a novel multimodal clustering approach and presented an application on the problem of SED in collections of multimedia. The proposed method utilizes the so-called "same event" (SE) model, which predicts whether a pair of items belong to the same cluster or not. The SE model is used to organize the collection in a graph, on which a community detection algorithm is applied. The two flavors of our method utilize either a batch or an incremental community detection algorithm. Empirical results indicate that the proposed algorithms achieve high quality clusterings. Interestingly, the performance of the incremental algorithm is not inferior to that of the batch algorithm.

Compared to the related approaches in [2] and [26], our approach computes the predictions of the SE model on pairs of images rather than on pairs of an image and an event. In these approaches the event representation is the result of an averaging of the features of the items that have been assigned to the event. Nevertheless, in the case of incorrectly assigned items, it is possible that the representation of the event may be significantly erroneous in some features. This may subsequently result in more items being erroneously assigned to the event, further affecting the representation of the event and leading to a progressive deterioration of the quality of the clusters. On the other hand, an item-item approach does not suffer from this issue and it is likely that the empirical advantage of the examined approaches over the item-cluster approach is due to this reason. Also, compared to the aforementioned approaches, the proposed approach does not require learning an additional model for determining whether a new event needs to be added (as in [26]) or setting a relevant threshold for the best matching event (as in [2]). Instead, the incremental approach can automatically determine whether a new item should join an existing cluster or not.

Compared to the approach in [23], where a SE model was also used between pairs of items, the graph-based methods are much more scalable and require far less resources. For instance, the approach in [23] requires that all pairwise same event relationships are maintained. Thus, all N^2 same event relationships need to be computed and the results need to be stored. Moreover, the sets of SE relationship of nodes are compared using Euclidean distance, which may make items with very few irrelevant neighbours appearing close to each other. Contrary, the proposed approaches do not require to either compute or store all the SE relationships.

In the future we intend to test the framework with other community detection algorithms as well as to apply the framework to other multimodal clustering tasks. Moreover, we plan to extend our recent work on distinguishing between event and non-event images. Regarding evaluation, we plan to explore the effect that the number of candidate neighbours has on the results.

Acknowledgments. This work is supported by the SocialSensor FP7 project, partially funded by the EC under contract number 287975.

References

1. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-Up Robust Features (SURF). Comp. Vis. Image Underst. 110(3), 346–359 (2008)
2. Becker, H., Naaman, M., Gravano, L.: Learning similarity metrics for event identification in social media. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM 2010, pp. 291–300. ACM, New York (2010)
3. Bekkerman, R., Jeon, J.: Multi-modal clustering for multimedia collections. In: CVPR (2007)
4. Brenner, M., Izquierdo, E.: Mediaeval benchmark: Social Event Detection in collaborative photo collections. In: MediaEval. CEUR Workshop Proceedings (2011)
5. Cai, X., Nie, F., Huang, H., Kamangar, F.: Heterogeneous image feature integration via multi-modal spectral clustering. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 1977–1984 (June 2011)
6. Fortunato, S.: Community detection in graphs. Physics Reports 486(3-5), 75–174 (2010)
7. Goder, A., Filkov, V.: Consensus clustering algorithms: Comparison and refinement. In: Ian Munro, J. (ed.) Proceedings of the Workshop on Algorithm Engineering and Experiments, ALENEX 2008, San Francisco, California, USA, pp. 109–117. SIAM (January 19, 2008)
8. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explorations Newsletter 11(1), 10–18 (2009)

9. Jegou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(1), 117–128 (2011)
10. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: 23rd IEEE Conference on Computer Vision & Pattern Recognition, CVPR 2010, pp. 3304–3311. IEEE Computer Society, San Francisco (2010)
11. Khalidov, V., Forbes, F., Horaud, R.P.: Conjugate mixture models for clustering multimodal data. Neural Computation 23(2), 517–557 (2011)
12. Li, Y., Crandall, D.J., Huttenlocher, D.P.: Landmark classification in large-scale image collections. In: IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, pp. 1957–1964. IEEE (2009)
13. Liu, X., Troncy, R., Huet, B.: Using social media to identify events. In: ACM Multimedia 3rd Workshop on Social Media, WSM 2011, Scottsdale, Arizona, USA, November 18-December 1, p. 11 (2011)
14. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
15. Nguyen, N.P., Dinh, T.N., Xuan, Y., Thai, M.T.: Adaptive algorithms for detecting community structure in dynamic social networks. In: 30th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2011, Shanghai, China, April 10-15, pp. 2282–2290. IEEE (2011)
16. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. Int. J. Comput. Vision 42(3), 145–175 (2001)
17. Papadopoulos, S., Kompatsiaris, Y., Vakali, A., Spyridonos, P.: Community detection in social media. Data Mining and Knowledge Discovery 24(3), 515–554 (2012)
18. Papadopoulos, S., Schinas, E., Mezaris, V., Troncy, R., Kompatsiaris, I.: The 2012 Social Event Detection dataset. In: 4th ACM Multimedia Systems, Dataset Session, MMSys 2013, Oslo, Norway, February 27-March 1 (2013)
19. Papadopoulos, S., Schinas, E., Mezaris, V., Troncy, R., Kompatsiaris, Y.: Social Event Detection at MediaEval 2012: Challenges, Dataset and Evaluation. In: MediaEval 2012 Workshop, Pisa, Italy, October 4-5 (2012)
20. Papadopoulos, S., Troncy, R., Mezaris, V., Huet, B., Kompatsiaris, I.: Social Event Detection at Mediaeval 2011: Challenges, dataset and evaluation. In: MediaEval. CEUR Workshop Proceedings (2011)
21. Papadopoulos, S., Zigkolis, C., Kompatsiaris, Y., Vakali, A.: CERTH@Mediaeval 2011 social event detection task. In: MediaEval. CEUR Workshop Proceedings (2011)
22. Papadopoulos, S., Zigkolis, C., Kompatsiaris, Y., Vakali, A.: Cluster-based landmark and event detection for tagged photo collections. IEEE Multimedia 18(1), 52–63 (2011)
23. Petkos, G., Papadopoulos, S., Kompatsiaris, Y.: Social event detection using multimodal clustering and integrating supervisory signals. In: Proceedings of the 2nd ACM International Conference on Multimedia Retrieval, ICMR 2012, pp. 23:1–23:8. ACM, New York (2012)
24. Phuvipadawat, S., Murata, T.: Breaking news detection and tracking in twitter. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol. 3, pp. 120–123 (2010)
25. Rendle, S., Schmidt-Thieme, L.: Scaling record linkage to non-uniform distributed class sizes. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 308–319. Springer, Heidelberg (2008)

26. Reuter, T., Cimiano, P.: Event-based classification of social media streams. In: Proceedings of the 2nd ACM International Conference on Multimedia Retrieval, ICMR 2012, pp. 22:1–22:8. ACM, New York (2012)
27. Schinas, E., Mantziou, E., Papadopoulos, S., Petkos, G., Kompatsiaris, Y.: CERTH @ Mediaeval 2013 Social Event Detection Task. In: MediaEval. CEUR Workshop Proceedings (2013)
28. Schinas, E., Petkos, G., Papadopoulos, S., Kompatsiaris, Y.: CERTH @ Mediaeval 2012 Social Event Detection Task. In: MediaEval. CEUR Workshop Proceedings, vol. 927 (2012)
29. Snoek, C.G.M., Worring, M., Smeulders, A.W.M.: Early versus late fusion in semantic video analysis. In: Proceedings of the 13th Annual ACM International Conference on Multimedia, MULTIMEDIA 2005, pp. 399–402. ACM, New York (2005)
30. Xu, X., Yuruk, N., Feng, Z., Schweiger, T.A.J.: Scan: a structural clustering algorithm for networks. In: Proceedings of the 13th ACM SIGKDD, KDD 2007, pp. 824–833. ACM, NY (2007)
31. Ye, Z., Hu, S., Yu, J.: Adaptive clustering algorithm for community detection in complex networks. Physical Review E 78(4) (2008)