Social Event Detection using Multimodal Clustering and Integrating Supervisory Signals

Georgios Petkos
Informatics and Telematics Institute
6^{th} Km. Charilaou-Thermis
Thessaloniki, Greece
gpetkos@iti.gr

Symeon Papadopoulos
Informatics and Telematics Institute
6^{th} Km. Charilaou-Thermis
Thessaloniki, Greece
papadop@iti.gr

Yiannis Kompatsiaris
Informatics and Telematics Institute
6^{th} Km. Charilaou-Thermis
Thessaloniki, Greece
ikom@iti.gr

ABSTRACT

A large variety of features can be extracted from raw multimedia items. Moreover, in many contexts, like in the case of multimedia uploaded by users of social media platforms, items may be linked to metadata that can be very useful for a variety of analysis tasks. Nevertheless, such features are typically heterogeneous and are difficult to combine in a unified representation that would be suitable for analysis. In this paper, we discuss the problem of clustering collections of multimedia items with the purpose of detecting *social events*. In order to achieve this, a novel *multimodal* clustering algorithm is proposed. The proposed method uses a known clustering in the currently examined domain, in order to supervise the multimodal fusion and clustering procedure. It is tested on the MediaEval social event detection challenge data and is compared to a multimodal spectral clustering approach that uses early fusion. By taking advantage of the explicit supervisory signal, it achieves superior clustering accuracy and additionally requires the specification of a much smaller number of parameters. Moreover, the proposed approach has wider scope; it is not only applicable to the task of social event detection, but to other multimodal clustering problems as well.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding; I.5.3 [**Pattern Recognition**]: Clustering; I.5.4 [**Pattern Recognition**]: Applications

General Terms

Theory, Experimentation, Algorithms

Keywords

Social media, Multimedia, Social event detection, Multimodal clustering

ICMR '12, June 5-8, Hong Kong, China

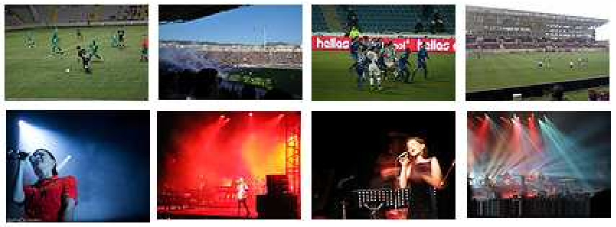

Figure 1: A set of sample pictures representing two classes of social events: soccer (top row) and concerts (bottom row).

1. INTRODUCTION

A lot of research effort has been put into processing unstructured and heterogeneous content that has been collected from the web. One of the most challenging aspects of this effort is probably the linking of web content to real world concepts. A very interesting task that has attracted a lot of interest recently and falls within the "matching to real world concepts" domain is the detection and processing of real world *social events* in collections of multimedia uploaded by users of social media platforms. Social events are events that are organized by people and attended mostly by people who are not directly involved in the organization of the events. Instances of such an event could be a soccer game, a concert, the screening of a movie, etc. A set of sample pictures that are collected from Flickr and depict social events can be seen in Fig. 1.

Similarly to many tasks that involve the analysis of multimedia content, the task of detecting and processing social events in multimedia collections is very challenging. This is due to the heterogeneity, multimodality and generally unstructured form of such content. In the case that these multimedia collections are retrieved from social medial platforms, the items in the collection are also typically linked to a rich set of metadata. Therefore, processing of such collections will need to deal with heterogeneous features and metadata such as: the time and geolocation that an item was captured, textual features such as tags and titles or even visual descriptors extracted from the raw content (e.g. SIFT). This paper treats the problem of social event detection as a clustering problem and attempts to deal with the heterogeneity and multimodality in multimedia collections by proposing a novel *multimodal* clustering approach. The essence of the method lies in predicting the "same cluster" relationship between pairs of items using the set of pairwise similarities for all modalities. An example clustering from the current domain (in our case, a clustering of items in which each cluster corresponds to a social event) is used to train the classifier that predicts the "same cluster" relationship. Finally, the set of all pairwise "same cluster" relationships is used to obtain the final multimodal clustering. This approach essentially achieves "supervised fusion" of the heterogeneous features that describe the data with the specific goal of retrieving clusters that are related to social events. The proposed approach is more general though and could also be used in other multimodal clustering tasks, in which different clusterings could correspond to separation in clusters that are relevant to different concepts, e.g. landmarks.

The rest of the paper is organized as follows. Section 2 reviews some relevant event detection and multimodal clustering approaches. Section 3 describes a typical multimodal spectral clustering algorithm that uses an early fusion strategy and subsequently describes the proposed method that utilizes a supervisory signal. Section 4 presents experimental evidence that supports the merits of the proposed method. Finally, Section 5 concludes the paper and discusses some future work.

2. RELATED WORK

2.1 Event detection

Event detection and processing in multimedia collections is a topic that has attracted a lot of interest in recent years. A straightforward approach for detecting event related items in multimedia collections is presented in [10]. Here, the description of events is retrieved from structured online sources such as last.fm or Upcoming. In addition, a set of multimedia items that are linked to these events by machine tags is retrieved from the same sources, providing an initial collection of multimedia items for each event. Subsequently, online sources of multimedia such as Flickr and YouTube are queried with specific properties of the identified events, e.g. location and title or location and time. Finally, since not all the retrieved results are related to the event to which the query corresponds, a visual similarity process is used to prune the retrieved items.

Another approach that utilizes additional online information sources is presented in [11]. The authors use geotagging information retrieved from online sources to determine the bounding box for a set of venues. Subsequently, they retrieve a set of photos that have location information that match the determined bounding box. They finally analyze the time distribution of retrieved photos to determine the set of events, which they compare to the actual set of events that occurred at the examined venues.

A cluster based approach is presented in [14]. Here, a set of photos is used to produce two image similarity graphs, one using visual features and the other using textual features. The two graphs are then combined in a single hybrid similarity graph and a community detection algorithm is used to cluster the nodes of the graph, i.e. the photos of the collection. The clusters are subsequently classified as representing either events or landmarks.

Moreover, in the 2011 MediaEval workshop, there was a social event detection challenge [15] on a collection of Flickr images. Many interesting approaches were used to tackle the challenge. For instance, [3] presents a classifier-based method, where items that are geotagged are used to build a set of initial clusters that correspond to events. The items of each cluster are then used to train a classifier that augments each cluster (event) with non-geotagged items. An approach that builds a classifier using explicit event descriptions from online event catalogues (such as last.fm, FBLeague, etc.) and performs some post-processing on the visual features to clean the classified data can be found in [9]. Another solution to the task is presented in [16], where a sequence of specific filtering, grouping and expansion rules are applied on the collection. A quite different view on the problem is presented in [17], where data are organized in a search engine and online sources are used to build groups of queries that are relevant to each location of interest. The search results for each group of queries correspond to location clusters and these are finally clustered according to time. Finally, an approach that treated the challenge by applying a sequence of clustering and filtering operations is presented in [21].

It is also useful to mention that, although this paper focuses on social event detection in collections of *multimedia* items, interesting work in the related field of event detection in *text streams* from the social media appeared recently. Some promising work can be found in [18], [22] and [19].

In general, many of the approaches that have appeared so far to tackle the problem of event detection in multimedia collections have used some form of online source to retrieve structured information that is related either to the events or the locations of interest. This is acceptable and could lead to enhancement of results. Nevertheless, this may not be possible in all cases as not all social events have a formal description in some online source. Moreover, in some cases human supervision will be necessary in order to direct the query, e.g. to select the most appropriate sources of information. Therefore, it would be important to also have a method that can handle plain data without much use of external sources. In addition, many of the approaches use some form of heuristics to improve results, such as "put all items that have been uploaded by the same user in the same day to the same cluster". Such heuristics make perfect sense. However, their use brings in some amount of uncertainty and in addition it may be difficult to cover all cases with such heuristics. Clearly, it would be much more handy to exploit any available data that can cover such cases, rather than to deploy a possibly incomplete set of heuristics.

2.2 Multimodal clustering

To cope with the aforementioned issues, this paper proposes the use of direct clustering on the features of multimedia items. This however presents some difficulties. Heterogeneous features can be extracted from multimedia items and also multimedia items may also be associated with a set of heterogeneous metadata which may offer important information for clustering. Here, some existing approaches for clustering data that are characterized by multiple heterogeneous features are presented.

In general, there are two classes of methods for dealing with multiple and heterogeneous features: early and late fusion [20]. In early fusion methods, features are combined before the main processing is executed. In late fusion methods, each modality is treated separately and only the results of processing each modalities data are combined. There are multimodal clustering methods though, such as the ones that are presented next, that do not clearly fall in the early or late fusion categories.

An interesting spectral clustering approach for multimodal clustering can be found in [4]. Spectral clustering comes in many flavors. It can be commonly seen as a method for finding a partitioning of the nodes of a graph, in which the sum of cross-group weights is minimized. For a common instance of spectral clustering (adopted to the multimodal scenario) please see the next Section. With regard to the aforementioned work, it is sufficient to mention that similarities according to the different modalities are combined by summing the inverses of regularized individual Laplacian matrices in an aggregate Laplacian matrix. This aggregate Laplacian is subsequently used to perform clustering exactly like in the common spectral clustering scenario. In a final step, the algorithm switches the final assignment of items to clusters, so that a cost function that measures the disagreement between the aggregate clustering and the individual clusterings for each modality is minimized.

A probabilistic approach to multimodal clustering is presented in [6]. Here, a hidden variable that represents a physical entity that generates data in different physical modalities is utilized. For instance, a person in a conversation recorded by a camera generates visual and auditory data. The goal is, given a set of multimodal observations, to infer the hidden variables that represent the cluster that each item belongs to. The generative model for each modality is assumed to be known and the parameters of the probabilistic model are learned by an Expectation Maximization algorithm. Once the model is learned, it can be used to perform inference of the hidden variables, i.e. to perform clustering.

The use of combinatorial Markov random fields (Comrafs) for multimodal clustering is proposed in [2]. Combinatorial Markov random fields are Markov random fields (undirected graphical models), in which at least one of the nodes represents a combinatorial random variable. In this work, each node represents a combinatorial random variable that is essentially a partitioning of the data, a clustering according to some modality. Multimodal clustering is formulated as an efficient inference procedure in this Comraf, specifically as a problem of most likely state estimation for a hidden variable that represents the aggregate multimodal clustering.

3. MULTIMODAL CLUSTERING USING A SUPERVISORY SIGNAL

As discussed, in order to deal with the shortcomings of existing approaches to the problem of event detection in collections of multimedia, this work proposes a direct clustering approach. Moreover, since the data that needs to be clustered are characterized by multiple heterogeneous features, the use of specialized multimodal clustering algorithms is necessary. This Section presents a novel multimodal clustering algorithm that utilizes an easy to obtain supervisory signal to guide the clustering process. First though, we present a multimodal spectral clustering algorithm that can be characterized as ta typical early fusion algorithm and that will be used as a baseline to evaluate the proposed method.

3.1 A baseline multimodal spectral clustering algorithm

A multimodal variant of the spectral clustering algorithm is outlined in Algorithm 1 and graphically depicted in Fig. 2. It uses a sort of early fusion approach: a weighted sum of the affinity matrices of the different modalities is computed and then - similarly to a common spectral clustering algorithm - the corresponding graph is pruned, the Laplacian matrix is computed and the minimum cut of the graph is found.

Algorithm 1 Multimodal spectral clustering with affinity matrix fusion

```
1: For each modality $m$, compute the affinity matrix $W_m$, with $W_m(i,j) = \exp(-d_m(i,j)^2/\sigma_m^2)$, where $d_m$ is the dissimilarity measure for modality $m$ and $\sigma_m$ is a scaling factor, chosen specifically for modality $m$.
2: Compute the aggregate affinity matrix $W_{tot}$ as $W_{tot} = \sum_{m=1}^{M} w_m W_m$.
3: Compute the pruned aggregate affinity matrix $\bar{W}_{tot}$: Initialize all the entries of $\bar{W}_{tot}$ to zero. For each line $i$ of $W_{tot}$ find the $k^{th}$ largest affinity. For each item $j$ in the current line that has affinity equal or larger to that threshold, set $\bar{W}_{tot}(i,j)$ and $\bar{W}_{tot}(j,i)$ to the value of $W_{tot}(i,j)$.
4: Compute the normalized Laplacian of $\bar{W}_{tot}$ as $\bar{L}_{tot} = \bar{D}_{tot}^{-1/2}\bar{W}_{tot}\bar{D}_{tot}^{-1/2}$, where $\bar{D}_{tot}$ is a diagonal matrix whose $i^{th}$ diagonal entry is the sum of the elements of the $i^{th}$ row of $\bar{W}_{tot}$.
5: Compute the $c$ largest eigenvalues of $\bar{L}_{tot}$ and the corresponding eigenvectors $v_1, v_2 ... v_c$.
6: Form the matrix $V$ by using the top $c$ eigenvectors as the columns of the matrix $V$.
7: Normalize the rows of the matrix $V$ to have unit length.
8: Use the k-means clustering algorithm to cluster the rows of $V$. Assign item $i$ of the original dataset to the $k^{th}$ cluster, if the $i^{th}$ row of $V$ has been assigned to the $k^{th}$ cluster.
```

Given that there are M modalities, the algorithm has the following parameters: M scaling factors σ_m, M weight factors w_m, the value of k for pruning weak edges from the equivalent graph and the number of clusters c. That makes for a total of $2M + 2$ parameters.

The described clustering algorithm has the following shortcomings. First, the final clustering result will be sensitive on both the scaling parameters and the weights that are assigned to the affinity / distances for each modality. Scaling and weighting factors are common in a variety of early fusion approaches. They are used so that the final solution is not affected by the fact that some modalities may have a different scaling than the rest and therefore they may dominate or be insignificant for the aggregate similarity measure. Since the final clustering result will depend on the scaling and weighting parameters, they will need to be set appropriately for different clustering tasks. The desired outcome of the clustering task may correspond to separation according to semantically different concepts and emphasis on the different features may need to be put. For instance, assume that the task is to discover important landmarks from the collection of images, then intuitively (but of course this would have to be investigated) better results may be expected if emphasis was put on location information (if available). If on the other hand, the goal of the multimodal clustering procedure is to determine groups of similar objects, e.g. buildings or tools, then better results may be expected if emphasis was put on textual metadata and visual features. Therefore, there may be different clustering goals when working on multimodal data and the parameters of the clustering procedure may need to be tuned separately for each clustering goal. It should also be noted that the dependence of the final result on weighting factors is an important issue in late fusion algorithms as well (e.g. in the case that predictions are combined using weighted sums).

Secondly, distances need to be specified even in cases when one of the features is missing for one of the datapoints. For instance, in the multimedia collection that is examined in this work, only around one fifth of the datapoints have location information. Clearly, when available, this is valuable information that can be used for the task of social event detection and cannot be ignored. In the clustering formalization that was described, the distance / affinity needs to be specified even if the value is missing for one of the two items. There are various options for the value to be used in that case. One could be to set the affinity to zero (or equivalently the distance to some very large constant). This however would bias the solution (how much depends on the weights of the modalities) towards grouping together items that may not be very close according to that modality but do have the corresponding values specified. Another option would be to use the minimum affinity between items for which the modality is not missing. Compared to the previous option, this reduces the bias introduced but is still a suboptimal solution. It would be better for this information to not play any role for the clustering of the data when not available.

3.2 Multimodal clustering integrating supervisory signals

As discussed, depending on the weights / scalings of the different modalities, it is possible to retrieve different clustering results. Instead of searching in the space of parameters, we propose to use a known clustering from the same domain, in order to determine how important the similarity according to the different modalities will be for the current clustering task. For most scenarios, it will be relatively easy to obtain such an example clustering. For the task at hand, it is possible to retrieve a set of pictures, e.g. from Flickr or last.fm, that correspond to some set of events.

We take advantage of this "supervisory signal" in the following way. Pairwise distances according to each modality that describe the example dataset are first computed. Then, for each pair of items, the distances according to each modality are compiled in a single vector. Additionally, the known clustering labels are used to form the target value for a binary classifier. A target value of $+1$ indicates that the two items to which the distance vector corresponds do belong in the same cluster, whereas a target value of -1 indicates that they do not belong in the same cluster. This data is used to learn the typical "same cluster" relationship for multimodal items according to the task at hand. Having learned such a classifier, one can form an indicator vector for each of the new items to be clustered. This indicator vector summarizes the "same cluster" relationship between each item and all other items to be clustered. Items that have similar indicator vectors should belong to the same cluster. Therefore, a final clustering result can be obtained by clustering the indicator vectors. The algorithm is graphically depicted in Fig. 3 and is described in Algorithm 2.

Algorithm 2 Multimodal clustering with supervisory signal

```
Retrieve a dataset $X$ of items that correspond to a known grouping. E.g. retrieve a set of pictures for a set of events from Flickr.
Extract the set of features $X_i^m$ for each of the items $i$, for each of the modalities $m$.
Compute the pairwise dissimilarities $DX_{ij}^m$ between all pairs of items in $X$ for all modalities.
Compile the dissimilarities $DX_{ij}^m$ for all modalities in a single vector $DX_{ij}$. Label each of these instances with $-1$ if the items $i$ and $j$ do not belong in the same clusters and with $+1$ if they do belong in the same cluster. Let $LX_{ij}$ denote the label.
Train a classifier $f(DX_{ij})$ that predicts the "same cluster" label $LX_{ij}$ using the vector of multimodal distances.
Given a new dataset $Y$ that needs to be clustered, extract the features and retrieve the metadata $Y_i^m$ for each modality.
Compute the pairwise distances $DY_{ij}^m$ between all pairs of items in $Y$ for all modalities.
Compile the distances $DY_{ij}^m$ in a single vector $DY_{ij}$.
Predict the "same cluster" indicators values $YL_{ij} = f(DY_{ij})$ for each pair of items in the dataset $Y$. Optionally, if the classifier $f$ can output probabilities or decision values post-process them using e.g. a threshold $\theta$ to obtain the final value of $YL_{ij}$.
Compile the $YL_{ij}$ in a single matrix $YL$.
Cluster the lines of $YL$ using some clustering algorithm, e.g. spectral clustering or k-means.
Each item $i$ in the dataset $Y$ is assigned to the cluster to which the $i^{th}$ line of $YL$ is clustered.
```

Essentially, the above algorithm introduces a supervisory signal to the multimodal clustering procedure. It uses an explicit instance of a relevant clustering to describe what it means, in terms of multimodal distances, for a pair of items to belong in the same cluster. Nevertheless, as our experiments will demonstrate later, we cannot expect that the classifier can achieve perfect accuracy as the two classes (distances that correspond to items that do and do not belong in the same cluster) have some overlap (in the experiments, the classifier achieves around 85% classification accuracy). The reason that we expect that this method will work, despite the fact that we cannot achieve perfect classification accuracy is that we compute the classification result for each pair of items, forming eventually an indicator vector that predicts quite accurately which items belong in the same cluster as the current item. Another item that belongs in the same cluster will have a - not exactly the same - but quite similar indicator vector as the first. On the other hand, an item that belongs in a different cluster will have a significantly different indicator vector. Considering that the number of items is large enough so that the indicator vector has sufficient dimensionality, it can be expected that the distances between indicator vectors that actually belong in the same cluster will be much smaller than the distances between indicator vectors that do not belong in the same cluster. An instance of two matching and an instance of two non-matching indicator vectors, taken from the experiments that will be described in the next section are shown in Fig. 4.

4. EXPERIMENTS

4.1 Dataset and tasks

To evaluate the multimodal clustering approaches that

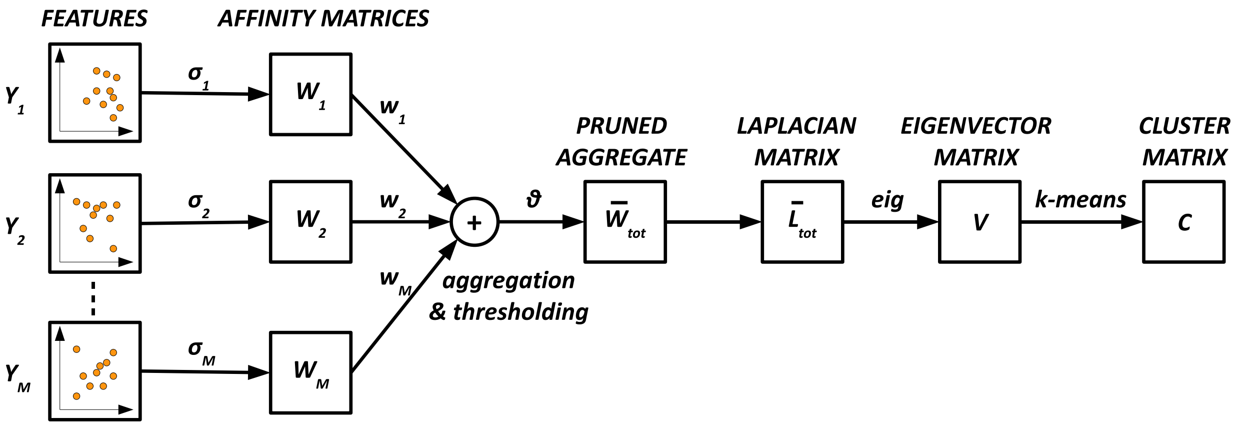

Figure 2: A spectral clustering approach that utilizes a sort of "early fusion" strategy. Its difference to standard unimodal spectral clustering is that it fuses affinity matrices by means of a weighted summation.

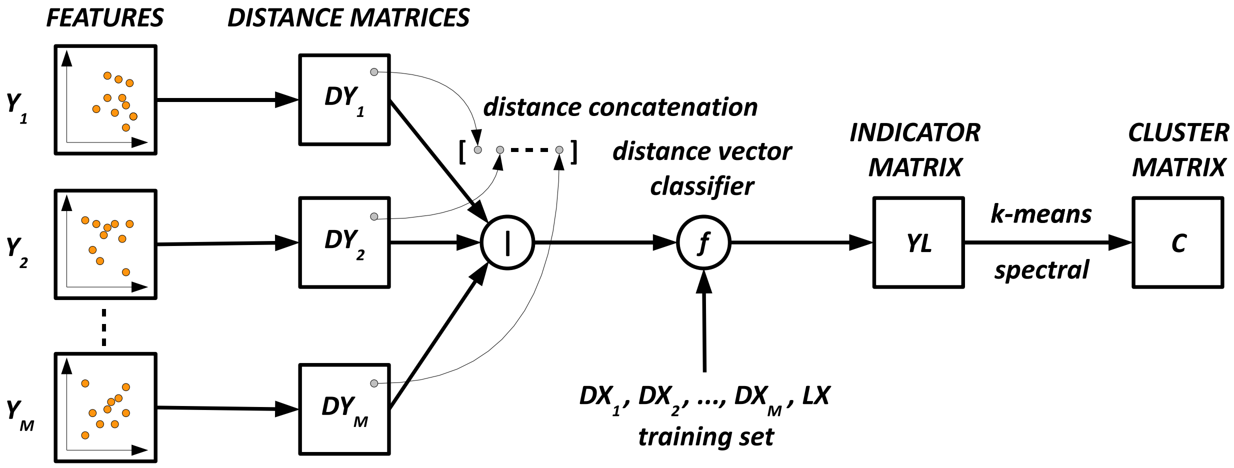

Figure 3: The proposed multimodal clustering approach that is used in this work for social event detection. An already clustered dataset X is used to obtain training examples consisting of distance vectors DX and corresponding labels LX. These training examples are used to learn the classifier f. The classifier f is used on the distance vectors DY_{ij} of the new dataset Y that needs to be clustered, in order to learn the indicator vectors that are compiled on the matrix YL. The indicator vectors are finally clustered to obtain the final clustering result.

were described in the previous Section, data from the MediaEval social event detection challenge [15] was used. The MediaEval challenge consisted of 2 tasks, in which photos that correspond to social events of a specific kind and occurred at specific locations needed to be determined from a set of 73645 photos collected from Flickr. The ground truth for the first task consisted of 434 items that belonged in 11 events and the ground truth for the second task consisted of 1640 items that belonged in 25 events. This makes a total of 36 tasks and 2074 items. It is important to note that only one fifth of the provided photos have explicit location information in the form of geotags.

A set of 10 different clustering tasks were generated from the set of 36 events that were available. That is, the set of 36 events was randomly split 10 times to two groups and each task consisted of clustering the items that belong in one of the two groups of events. The items that belonged in the other group of events was used to train the classifier in the proposed approach. It should also be stressed that for the proposed approach, not all available data was used to train the classifier, only a set of 40000 randomly selected positive and negative examples. In an example clustering with one thousand items, there are almost one million neighbourhood relationships, therefore only a small sample of the available training data is used.

4.2 Features and similarity measures

The following features and corresponding similarity measures are used:

1. Time, which is represented as a Unix timestamp. The absolute difference in hours between the timestamps of two items is used as the dissimilarity measure. It should be noted that this can also be a fraction.

2. Location, which is represented as a pair of latitude and longitude values. The geodesic distance in meters

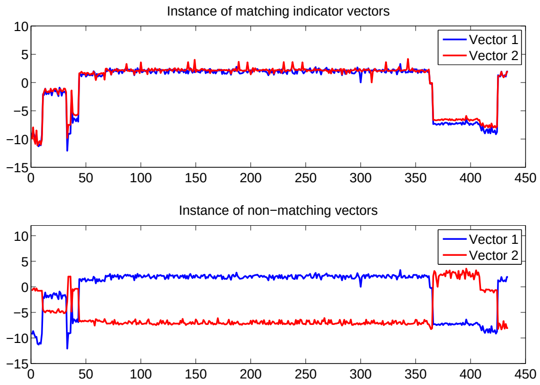

Figure 4: Instances of non-thresholded indicators vectors from the first clustering task in the experiments. (Top) Two matching indicator vectors that were placed in the same cluster. (Bottom) Two non-matching indicator vectors that were placed in different clusters.

is used as the dissimilarity metric for the items that are geotagged.

3. The Scale Invariant Feature Transform (SIFT) [12] was computed for each image and the cosine similarity is used. The quantity $1-$ cosine similarity is used as the dissimilarity measure.

4. Term frequency - inverse document frequency weights were computed for the tags that appear in each collection of documents and the cosine similarity between the vectors that represent each item was used as the similarity metric. Similarly to visual similarity, the quantity $1-$ cosine similarity is used as the dissimilarity measure.

4.3 Parameters and other options

As mentioned, given M modalities, the naive spectral clustering approach has $2M + 2 = 10$ parameters. Therefore, some search in the space of parameters needs to be performed in order to come up with a model that can achieve a good level of clustering accuracy.

The first parameters that need to be determined for our approach are related to the thresholding of the YL matrix. One option is to perform binary thresholding. The decision value for binary classification as computed by the SVM is 0. However, we may want to increase or decrease the threshold, so that we label as positive or negative only pairs of items for which the decision value is farther from the decision boundary. Moreover, we may want to use the difference between the decision value and the threshold as an indication of the confidence of the classifier prediction. Another option is to directly use the decision values to form the indicator vector without applying any thresholding. In addition, in the case that we do apply thresholding, we may decide that we want to flip the indications of the prediction values and use the negative of YL for thresholding. This would still make sense, as the negative would give an “inverted” indicator vector for each pair of items, which would still be more similar for items that belong in the same cluster than for items that do not belong in the same cluster. Please note that the implementation of the classifier does not ensure that positive values will be associated to positive decision values. This needs to be examined with respect to the training data that was used and is basically a feature of the implementation of LIBSVM. Finally, different clustering algorithms can be considered for the last step of the procedure, i.e. for the clustering of the indicator vectors. In the experiments, k-means clustering initialized with k-means++ [1] and plain spectral clustering will be considered. In both of these cases, the number of clusters is an open parameter. However, in some of the spectral clustering experiments, the number of clusters is estimated automatically from the eigenvalues of the Laplacian matrix [13]. Moreover, in the case of spectral clustering, there are two parameters. The first is the scaling factor, we trivially use 1. The other is the number of k neighbours to keep for each item, we use 20. These two parameters are important for the final clustering result when using spectral clustering. Moreover, appropriate values for these two parameters would vary as the length of the indicator vectors changes, i.e. as the number of items to be clustered changes. Nevertheless, the focus of the evaluation test is on the general concept of the use of the supervisory signal in multimodal clustering and therefore we ignore the search with respect to these parameters, which are also relevant only to the case that spectral clustering is used to perform the final step of the procedure. To conclude, the set of parameters / different versions of the algorithm to be tested for the proposed method are:

1. Hard, soft or no thresholding on the elements of the matrix YL.

2. The threshold value θ.

3. Whether to take the negative of the matrix YL or not.

4. The clustering algorithm to be used on the rows of the matrix YL and the number of clusters.

This set of parameters is far smaller than the set of parameters for the plain multimodal spectral clustering approach.

In addition, the parameters of the classifier need to be determined. In our experiments, a Support Vector Machine classifier is used and in particular the matlab implementation of LIBSVM [5]. The features that were input to the classifier (i.e. the similarities / dissimilarities according to the 4 modalities) were rescaled to the interval $[0, 1]$ apart from the location dissimilarity for the pairs that it was available, where it was rescaled to the interval $[0, 0.5]$ with the distance value 1 being used for the pairs for which at least one item does not have location information. This essentially reserves a part of the input space for the cases that the variable is missing. This is placed sufficiently far from the other data, so that it does not interfere with learning in the other areas of the space. As already mentioned, it is important to have training data in all areas of the space where the classifier will need to predict during testing. The alternative would be to learn 2 separate models, one for the case that the variable is missing and one for the case that it doesn't. In the case that more variables may be missing though, this would require learning a large number of models. The radial basis function kernel was used and appropriate values for the parameters of the classifier were determined using cross-validation (C=10000, $\gamma = 0.2$).

Table 1: Best NMI achieved by the two tested methods for each of the 10 runs and in all runs.

Run	Baseline	With supervisory signal	Difference (abs. and %)
1	0,8586	0,9469	0,0883 (10,28%)
2	0,7843	0,9197	0,1354 (17,26%)
3	0,7867	0,9164	0,1297 (16,48%)
4	0,8021	0,9062	0,1041 (12,97%)
5	0,7559	0,8939	0,138 (18,25%)
6	0,8410	0,8994	0,0584 (6,94%)
7	0,8153	0,9538	0,1385 (16,98%)
8	0,7097	0,8798	0,1701 (23,96%)
9	0,8058	0,9406	0,1348 (16,72%)
10	0,8223	0,8935	0,0712 (8,65%)
All	0,858	0,9538	0,0958 (11,16%)

A simple grid search approach was used to search over the parameter space for both methods. To avoid searching in the full 10-dimensional parameter space of the first method, it is recognized that the weighting factors w_m and the scaling factors σ_m have a similar (but definitely not equivalent) effect, i.e. they can be tuned to control the contribution of the similarity of each modality to the aggregate distance / affinity matrix. Therefore, we omit optimization with respect to the w_ms and set them all to the value 1. Eventually, we try to optimize with respect to only the σ_ms for the 4 modalities, the pruning parameter k and the number of clusters. For each of these parameters, 3 or 4 values are tested. For the proposed method, all combinations of the options mentioned earlier were checked. Again, 3 or 4 values were used for the thresholds, whereas the rest of the parameters are actually discrete.

4.4 Results

As suggested in [15], the performance measure used to evaluate the clustering accuracy is the Normalized Mutual Information (NMI) computed between the output clustering and the available ground truth.

The results can be seen in Table. 1 and Table. 2. Table 1 shows the best NMI achieved by each of the two methods on the 10 tasks and in overall. Table 2 shows the average NMI achieved by each of the two methods on the two tasks. The average is taken with respect to the tested set of parameters. The standard deviation between runs is also shown. It can be seen that for both tasks, the proposed approach consistently and significantly outperforms the plain multimodal spectral clustering approach. This result of course depends on the set of parameters that was chosen to be tested. Nevertheless, it still indicates, together with the previous Table, that without a lot of search in the parameter space, the proposed approach can achieve competitive results by taking advantage of the information provided by the clustering that was used to train the classifier.

For the proposed method, among the different options for thresholding, hard thresholding achieved the highest accuracy. The average NMI over all other parameters was 0,7694 ± 0,1130, whereas for soft thresholding it was 0,6095 ± 0,1263 and for no thresholding at all it was 0,6812 ± 0,1686.

It is also worth noting that the best NMI achieved by the contestants in task 1 of the MediaEval challenge was 0.63 and 0.67 in the second. Far higher scores are achieved using the proposed method. Of course the results are not directly comparable because our algorithm worked on a set of items that we already knew that belong in the target clustering, whereas the contestants worked on a larger dataset with many irrelevant items. Nevertheless, the clustering accuracy is very high and the score achieved is remarkable.

Table 2: Average and standard deviation of the NMI achieved by the two tested methods for the 10 runs and in all runs.

Run	Baseline	With supervisory signal
1	0,2996 ± 0,1855	0,6737 ± 0,1892
2	0,2701 ± 0,1560	0,7181 ± 0,1504
3	0,2869 ± 0,1657	0,7051 ± 0,1784
4	0,3051 ± 0,1787	0,7248 ± 0,1231
5	0,2859 ± 0,1534	0,7005 ± 0,1220
6	0,2863 ± 0,1688	0,6843 ± 0,1486
7	0,2992 ± 0,1901	0,6956 ± 0,1576
8	0,2389 ± 0,1264	0,6258 ± 0,1453
9	0,2468 ± 0,1396	0,7067 ± 0,1354
10	0,2500 ± 0,1533	0,6323 ± 0,2137
All	0,2769 ± 0,1643	0,6867 ± 0,1619

5. CONCLUSIONS AND FUTURE WORK

A novel multimodal clustering approach was presented and applied to the problem of social event detection in collections of multimedia items. The merit of the proposed clustering approach is that it learns using an explicit example of correct clustering in the current domain, eliminating the need to search for the appropriate fusion strategy. For the problem of social event detection, it is possible to obtain such an example clustering from online sources. For most other problems, similar example clusterings should not be hard to obtain. The use of the example clustering also guides the final clustering solution towards matching the semantics of the example clustering. For instance, if the example clustering corresponded to sets of landmarks, then it is likely that the final clustering would also correspond to groups of multimedia items that correspond to landmarks, rather than events. This will be investigated in future work. It is likely though that for fine semantical groupings, separate treatment of the groups of textual terms will most likely be required. The reason for this is that a lot of semantic and domain dependent information that is related to grouping of items is lost when computing term frequency - inverse document frequency weights.

A drawback of the algorithm is the fact that for a dataset with N items, the dimensionality of the indicator vector will also be N. This means that N^2 predictions from the classifier are required and this may be quite expensive for relatively large N. Moreover, in the last step, clustering on vectors of dimensionality N will have to be performed. Clustering of high-dimensional vectors may be a difficult task, as the notions of locality, neighbourhood and compactness become fuzzy in high dimensional spaces. Special algorithms have been designed for this problem [7]. Another possible solution to this problem could be to perform dimensionality reduction before applying the clustering algorithm, with the hope that the information lost during the process of rerepresenting the data is not crucial for the clustering result.

Moreover, an extension that we would like to examine in the future is the development of a pseudo-online version of the algorithm that would consume each time only a subset of the data of some maximum size. Another extension is the possibility of detecting outliers. This would also be useful for the pseudo-online procedure. A first thought may be that items that do are outliers would have indicator vectors that do not contain "same cluster" indicators. In a preliminary investigation, involving the addition of some 3000 random items to the tasks examined in the experiments, it was found that as the number of items grows, most items will have an increasing number of "same cluster" indicators. Therefore, a more versatile approach is required to this problem.

In future work, we also intend to apply the proposed multimodal clustering approach in other domains, e.g. semantic clustering of images using multiple extracted features. Moreover, we intend to compare the sensitivity of the resulting clustering to the selection of the example clustering and perform more extensive evaluations with richer example clusterings.

6. REFERENCES

[1] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[2] Ron Bekkerman and Jiwoon Jeon. Multi-modal clustering for multimedia collections. In *CVPR*. IEEE Computer Society, 2007.

[3] Markus Brenner and Ebroul Izquierdo. Mediaeval benchmark: Social event detection in collaborative photo collections. In Larson et al. [8].

[4] Xiao Cai, Feiping Nie, Heng Huang, and F. Kamangar. Heterogeneous image feature integration via multi-modal spectral clustering. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1977 –1984, June 2011.

[5] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[6] Vasil Khalidov, Florence Forbes, and Radu P. Horaud. Conjugate mixture models for clustering multimodal data. *Neural Computation*, 23(2):517–557, February 2011.

[7] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowl. Discov. Data*, 3:1:1–1:58, March 2009.

[8] Martha Larson, Adam Rae, Claire-Hélène Demarty, Christoph Kofler, Florian Metze, Raphaël Troncy, Vasileios Mezaris, and Gareth J. F. Jones, editors. *Working Notes Proceedings of the MediaEval 2011 Workshop, Santa Croce in Fossabanda, Pisa, Italy, September 1-2, 2011*, volume 807 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011.

[9] Xueliang Liu, Benoit Huet, and Raphaël Troncy. Eurecom @ mediaeval 2011 social event detection task. In Larson et al. [8].

[10] Xueliang Liu, Raphael Troncy, and Benoit Huet. Finding media illustrating events. In *ICMR'11, 1st ACM International Conference on Multimedia Retrieval, April 17-20, 2011, Trento, Italy*, 04 2011.

[11] Xueliang Liu, Raphael Troncy, and Benoit Huet. Using social media to identify events. In *WSM'11, ACM Multimedia 3rd Workshop on Social Media, November 18-December 1st, 2011, Scottsdale, Arizona, USA*, 11 2011.

[12] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, November 2004.

[13] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, pages 849–856. MIT Press, 2001.

[14] S. Papadopoulos, C. Zigkolis, Y. Kompatsiaris, and A. Vakali. Cluster-based landmark and event detection for tagged photo collections. *Multimedia, IEEE*, 18(1):52 –63, jan. 2011.

[15] Symeon Papadopoulos, Raphaël Troncy, Vasileios Mezaris, Benoit Huet, and Ioannis Kompatsiaris. Social event detection at mediaeval 2011: Challenges, dataset and evaluation. In Larson et al. [8].

[16] Symeon Papadopoulos, Christos Zigkolis, Yiannis Kompatsiaris, and Athena Vakali. Certh @ mediaeval 2011 social event detection task. In Larson et al. [8].

[17] Massimiliano Ruocco and Heri Ramampiaro. Ntnu@mediaeval 2011 social event detection task. In Larson et al. [8].

[18] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[19] Hassan Sayyadi, Matthew Hurst, and Alexey Maykov. Event detection and tracking in social streams. In Eytan Adar, Matthew Hurst, Tim Finin, Natalie S. Glance, Nicolas Nicolov, and Belle L. Tseng, editors, *ICWSM*. The AAAI Press, 2009.

[20] Cees G. M. Snoek, Marcel Worring, and Arnold W. M. Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, MULTIMEDIA '05, pages 399–402, New York, NY, USA, 2005. ACM.

[21] Yanxiang Wang, Lexing Xie, and Hari Sundaram. Social event detection with clustering and filtering. In Larson et al. [8].

[22] Jianshu Weng and Bu-Sung Lee. Event detection in twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain*. The AAAI Press, 2011.

Acknowledgments

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