Improved Hierarchical Clustering for Face Images in Videos

Integrating positional and temporal information with HAC

Subhradeep Kayal
Aalto University School of Science, Finland
subhradeep.kayal@aalto.fi

ABSTRACT

Efficient techniques for face clustering, along with speech and text recognition, can provide the means for rapid browsing and accurate retrieval of videos from large video databases. Observing that video data contains information about not only the face features, but also the temporal ordering of the faces and positional coordinates of the face-regions within the video frame, an attempt is made to consolidate all of these details into the framework of the flexible and intuitive hierarchical clustering algorithm. This paper outlines a novel initialization mechanism for the hierarchical clustering to follow, based on the temporal and positional information of the face-samples extracted from a video. Experiments with news broadcast videos show that the novel algorithm is considerably more efficient than it's parent, and is promising for future exploration.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*Video Analysis*; I.4.9 [**Image Processing and Computer Vision**]: Applications; I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*

General Terms

Theory, Experimentation, Algorithms

Keywords

Face Clustering, Hierarchical Clustering

1. INTRODUCTION

With the increasing amount of multimedia data, a genuine need has risen to develop more efficient methods for browsing, content description and indexing. Such methods are essential for systems such as video libraries, which store large volumes of multimedia data with suitable tagging. A general survey of state-of-the-art methods for video indexing can be found in [10] and [1]. Face clustering, in particular, is an important mechanism to affect video indexing and summarization, by describing the content of the video in terms of *who* is present in it [2][3].

This paper is aimed at enhancing the highly flexible Hierarchical Agglomerative Clustering (HAC) algorithm, by including temporal and positional information, in order to efficiently use it for clustering of faces in videos. The paper modifies and adds to the work done in [5], and compares the results of clustering of faces detected from 3 Finnish news broadcast clips to the traditional HAC algorithm. The results show that the novel variant improves the parent algorithm considerably for all 3 videos.

2. DATASET DESCRIPTION

The dataset consists of 3 news videos provided by the Finnish channel YLE, each approximately 20 minutes long. Faces are detected from each video, using the procedure mentioned next, and are manually labelled to form clusters, which will be used as ground truth. Some more information about the videos: (1) Video 1 consists of 1778 detected faces, labelled to form 13 ground truth clusters, (2) Video 2, 2043 detected faces, 23 ground truth clusters and (3) Video 3, 2529 detected faces, 20 ground truth clusters

3. DETECTION OF FACES, FEATURE EXTRACTION AND DIMENSIONALITY REDUCTION

3.1 Face Detection

Faces are detected from video frames, which are extracted at a rate of 5 frames per second. Once a frame has been captured, face-like regions are detected by running the Viola-Jones detector [13] on it and keeping only those regions which are reasonably sized. In this paper, all regions which are of dimensions more than 100x100 pixels are regarded as reasonably sized, and all regions which are smaller are discarded. This is a good assumption for the present case, since the dataset consists of news videos wherein important people, whose faces should be effectively detected and clustered, will have close-up shots and, hence, reasonably large face regions.

The face-like regions are saved with a label which contains two valuable pieces of information: the time at which the face occurred in the video and the positional coordinates of the center of the bounding box which contains the face.

ICMR '14 April 01 - 04 2014, Glasgow, United Kingdom
Copyright 2014 ACM 978-1-4503-2782-4/14/04 ...$15.00.
http://dx.doi.org/10.1145/2578726.2578792

3.2 Filtering of non-faces

Amongst all the face-like regions extracted from the video, some are erroneously detected non-face regions. To reject these regions and perform a clean-up, two additional filtering steps are performed.

The first step is based on the observation that human skin colour can be used to distinguish between face and non-face regions. It has been found that the chrominance component of human skin has good clustering properties [4], especially in the HSV and YCbCr colour-spaces. Using this property to filter individual pixels as skin or non-skin pixels, the fraction of the total pixels which qualify as skin pixels is counted. If this fraction is not substantial, and less than a predetermined threshold, the region is discarded as a non-face region. A good threshold has been found to be 0.8, i.e., at least 80% of the region must comprise skin pixels for it to be a face region.

The second step uses a simple PCA-based filtering algorithm [12] to get rid of the remaining non-face regions which might pass through step 1. Here, a set of example face images is chosen and projected to the 'face-space' using PCA. Then, a region is accepted as a face if the distance between its projection on the 'face-space' and itself is less than a threshold.

These two steps together form an accurate filter to remove non-face regions. Feature extraction and clustering is performed only on those images which pass through both of these steps and, thus, get validated as a true face region.

3.3 Feature Extraction

It is important to extract features which can be rapidly calculated, but at the same time are descriptive enough to adequately capture the properties of one facial image and differentiate it from another. For this purpose, the Gabor features are chosen, which have been widely used in face recognition since the work in [6].

The Gabor feature is formed as by taking the absolute value of the result of convolution of each image with the Gabor filter. A 2D Gabor filter, in the form most commonly used in the context of face recognition, is given by:

$$\phi_{f,\theta,\gamma,\eta}(x,y) = \frac{f^2}{\pi\gamma\eta} e^{-(\alpha^2 x'^2 + \beta^2 y'^2)} e^{j2\pi f x'}$$

where,

$$x' = x\cos\theta + y\sin\theta \text{ and } y' = -x\sin\theta + y\cos\theta$$

Here, f and θ control the scale and orientation of the filter.

Usually, a bank of filters, at several scales and frequencies, is used, instead of a single filter, to most effectively capture all the available information contained in a facial image. The most suitable filter bank, as past studies suggest [11], uses 5 scales and 8 orientations.

The absolute values of each individual 2D output of convolution, i.e., of a particular face region and a Gabor filter at a particular scale and frequency, is taken and concatenated row-wise to form different row vectors. Finally, all such row vectors, for a particular face image, are concatenated to form one high dimensional feature vector per face image.

3.4 Dimensionality Reduction via Non-linear Embedding

The Gabor features are high-dimensional and complex. However, often, the underlying structure can be described by only a small number of features. To deal with this problem of high dimensionality and extract of a small number of highly descriptive features, a popular non-linear dimensional reduction technique, called Locally Linear Embedding (LLE) [8], is used.

The optimization problem in LLE is an eigenvalue problem and does not involve local minimas, making it an accurate dimensionality reduction method. Essentially, LLE tries to learn the intrinsic manifold, embedded in the high-dimensional space, in the dataset. A deeper study of LLE as a dimensionality reduction tool in face recognition can be found in [14].

In this paper, LLE is performed on the 2D matrix which is generated from each video, such that each row represents the Gabor feature vector for a face-sample. The optimal values for the number of reduced dimensions and the LLE parameter, the number of nearest neighbours [8], is selected by cross-validation.

4. RELEVANT BACKGROUND : THE HAC ALGORITHM

This section is aimed at introducing the concepts related to Hierarchical Agglomerative Clustering (HAC), for ease of understanding the latter sections in the paper, where the HAC algorithm is modified.

Hierarchical Clustering, simply put, tries to create a hierarchy of clusters. There are two approaches which can be used, a top-down approach (Divisive Clustering) and a bottom-up approach (Agglomerative Clustering). Agglomerative clustering works by placing each point in it's own cluster and combining clusters, according to some similarity metric, until there is one cluster left. Thus, it creates a cluster-tree, with different levels representing different clustering outcomes. The divisive approach is the opposite, meaning that it begins with all the points in one cluster, and proceeds by gradually dividing the cluster into smaller clusters. The agglomerative approach is more widely used because it is easier to understand and implement.

In HAC, clusters are combined based on some similarity measure. Three commonly used similarities are single-link (SL), complete-link (CL) and average-link (AL) similarities, described in the following equations:

$$SL(i,j) = \min\{d(a,b) \forall a \in C_i, b \in C_j\} \tag{1}$$

$$CL(i,j) = \max\{d(a,b) \forall a \in C_i, b \in C_j\} \tag{2}$$

$$AL(i,j) = \frac{1}{|C_i||C_j|} \sum_{a \in C_i} \sum_{b \in C_j} d(a,b) \tag{3}$$

where, $d(a,b)$ is the distance metric between points a and b, such that point a belongs to cluster C_i, and point b belongs to cluster C_j. $|C_i|$ refers to the size of cluster C_i.

The process of HAC can be outlined as:

1. Calculate the distance matrix D_s for the data points in the dataset, such that $D_s(i,j)$ is the distance between the i^{th} and j^{th} points.

2. Initialize the algorithm such that all the datapoints represent a singleton cluster.

3. Calculate similarity between clusters using any suitable similarity metric (see equations (1), (2) and (3)).

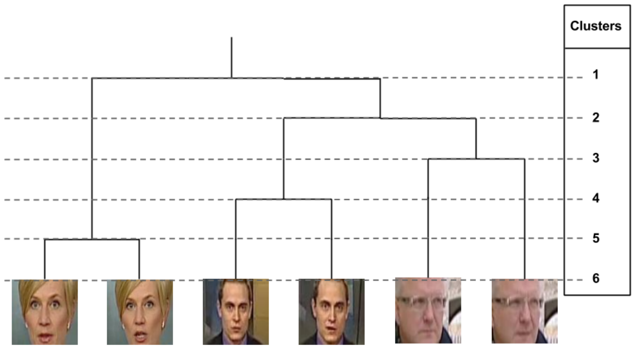

Figure 1: HAC algorithm performed on 6 faces

4. Combine the nearest pair of clusters.

5. Continue till one cluster remains.

As can be seen from Figure 1, HAC forms a cluster tree, whose different labels represent different clustering outcomes. The required number of clusters can be selected by choosing the appropriate level on the clustering tree.

5. PROPOSED ALGORITHM

The work done in this paper modifies and improves the work in [5] so as to improve the efficiency of the HAC algorithm. In order to keep this paper self-contained, parts of [5] will be restated in this section, with modifications added.

5.1 Temporal Clustering

The temporal clustering in [5] proceeds via selection of a window parameter (w), which is the maximum allowable temporal difference between two faces for them to be in the same cluster. This step is modified entirely to avoid the heuristic selection of the parameter w.

The algorithm for organizing the face regions into clusters, by their temporal similarity, automatically detects 'jumps' in the sequence of the time-stamps by simple peak detection. It is stated as follows:

1. Sort the unique time-stamps in ascending order.

2. Detect the 'jumps' in this sequence by detecting the peaks of it's first difference. This can be done by detecting the non-zero second differences of the sequence of temporal labels (see Figure 2 for a pictorial example). Let the corresponding temporal labels for the detected peaks be $T = \{t_1, t_2, \ldots, t_n\}$.

3. Partition the face-samples according to their time-stamps and according to T. This means that all the faces with time-stamps between 0 and t_1 are in cluster 1, from t_1 to t_2 are in cluster 2, and so on.

5.2 Reclustering using Positional Information

This step accounts for the fact that every frame in the video may have the presence of two or more faces, at different spatial coordinates in the frame. Hence, every temporal cluster will also contain face-samples of two or more individuals, whose spatial coordinates, or positional label, are different. This information is used in order to further enhance the initialization mechanism to the HAC algorithm.

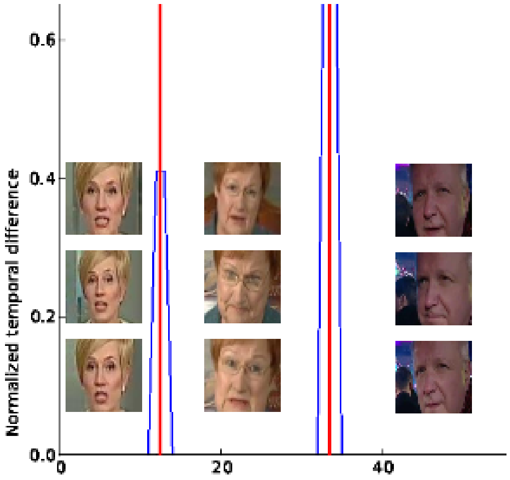

Figure 2: An example of how the initial temporal clustering works: The figure shows a plot (in blue) of the magnitudes of the second difference of the series of temporal labels, after normalization. For ease of viewing, red vertical lines have been added to show the peaks prominently. A few example faces have been added to each temporal cluster. Note also that the figure has been cropped for better viewing.

The algorithm for reclustering based on positional information uses Gaussian Mixture Models (GMMs), to model the clusters, whose parameters are inferred by the Expectation-Maximization algorithm. The problem of preselecting the number of clusters is solved by using the model with the lowest BIC (Bayesian Information Criterion) score [9].

The algorithm is stated as:

1. For a particular temporal cluster, read the positional labels (X-Y coordinates of the center of the face bounding box) for the face-samples in it

2. Fit GMMs, with components varying from 1 to max, to the position data. The value of max is the upper limit for the number of faces that maybe present in the frame. A good guess for max is:

$$max = \frac{\text{Area of frame}}{\text{Area of smallest face sample}}$$

3. Select the optimal components for the GMM to be that for which the BIC score is minimum, and recluster according to it.

4. Repeat for all temporal clusters

5.3 Clustering based on HAC

The clusters which are formed thus are the seed clusters for the HAC algorithm. Apart from traditional HAC, these seed clusters are also used with the modified HAC algorithm (THAC) described in [5]. A brief description of the THAC algorithm is as follows:

1. Calculate the feature distance matrix, D_s using the Gabor feature vectors of the face-samples and, addi-

Table 1: Results

	H			C			V-score		
Method	1	2	3	1	2	3	1	2	3
HAC	0.655	0.866	0.74	0.618	0.736	0.789	0.636	0.796	0.764
TP_HAC	0.784	0.775	0.87	0.802	**0.84**	0.831	0.793	0.806	0.85
TP_T-HAC	**0.817**	**0.93**	**0.881**	**0.819**	0.764	**0.862**	**0.818**	**0.836**	**0.871**

tionally, the temporal distance matrix, D_t, given by:

$$D_t(i,j) = |t_i - t_j| \tag{4}$$

where, t_i and t_j are time-stamps.

2. Weight D_t from (4) with an exponential decay term and construct the matrix D such that:

$$D(i,j,it) = D_s(i,j) + D_t(i,j) * \exp(-it/N) \tag{5}$$

where, it is the present iteration number and N is the total number of samples.

3. Use D as the distance matrix for the HAC algorithm, updating it with every iteration.

6. METHOD OF ANALYZING THE RESULTS

There exist many methods of cluster analysis in literature. In the light of this paper, the V-measure [7] is chosen as the criterion to quantify cluster purity, and uses the knowledge of ground truth class assignments of the samples. The V-measure is calculated by taking the harmonic mean of two parameters, namely: (1) Homogeneity (H), which takes a high value if each cluster contains members of only a single class and (2) Completeness (C), which takes a high value if all members of a particular class are assigned the same cluster. Thus, it provides some information about on how 'pure' and 'complete' the clusters are.

7. RESULTS

The results mentioned in this paper are based on the algorithms:

1. Hierarchical Agglomerative Clustering with a cosine distance metric and average-link similarity measure, or **HAC**.

2. HAC initialized with novel method, or **TP_HAC**.

3. Spatio-temporal HAC [5] with novel initialization, or **TP_T-HAC**.

The results in Table 1 lists the values for Homogeneity, Completeness and V-measure for the 3 videos (sub-columns 1-3). It can be seen that the novel initialization step improves the HAC algorithm, with the spatio-temporal HAC performing even better.

8. REFERENCES

[1] R. Brunelli, O. Mich, and C. Modena. A survey on the automatic indexing of video data,. *Journal of Visual Communication and Image Representation*, 10(2):78 – 112, 1999.

[2] C. Czirjek, S. Marlow, and N. Murphy. Face detection and clustering for video indexing applications. In *Proceedings of Advanced Concepts for Intelligent Vision Systems*, pages 2–5, 2003.

[3] S. Eickeler, F. Wallhoff, U. Lurgel, and G. Rigoll. Content based indexing of images and video using face detection and recognition methods. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 1505–1508, 2001.

[4] C. Garcia and G. Tziritas. Face detection using quantized skin color regions merging and wavelet packet analysis. *IEEE Transactions on Multimedia*, 1(3):264–277, 1999.

[5] S. Kayal. Face clustering in videos: GMM-based hierarchical clustering using spatio-temporal data. In *13th UK Workshop on Computational Intelligence (UKCI)*, pages 272–278, 2013.

[6] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. Von Der Malsburg, R. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311, 1993.

[7] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 410–420, 2007.

[8] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[9] G. Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, 1978.

[10] C. Snoek and M. Worring. Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools and Applications*, 25(1):5–35, 2005.

[11] V. Struc and N. Pavesic. The complete gabor-fisher classifier for robust face recognition. *EURASIP Journal on Advances in Signal Processing*, 2010(1):847680, 2010.

[12] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proceedings of the 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591, 1991.

[13] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 511–518, 2001.

[14] T. Zhang, S. Li, S. Wu, and L. Tao. Face recognition dimensionality reduction based on LLE and ISOMAP. In W. Du, editor, *Informatics and Management Science V*, volume 208 of *Lecture Notes in Electrical Engineering*, pages 775–780. Springer London, 2013.