Face Clustering Experiments on News Video Images

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

***Abstract*—With the advent of the information age and an explosion in the availability of multimedia data, successful clustering of faces is absolutely essential for many applications such as video indexing and summarization, major cast detection, and even unsupervised face-database generation. This paper compares with each other, the combinations of existing approaches of dimensionality reduction and different methods of clustering on face images detected from a TV news broadcast clip. The real value of the paper is in providing future researchers an introductory overview of possible methods and recommend a method as a good starting point for face clustering from videos.**

***Index Terms*—face clustering, dimensionality reduction, kernel PCA, GPLVM, tSNE, hierarchical clustering.**

I. Introduction

The face recognition problem has come a long way since it's early solutions [1] and [2], and has been reviewed well in [3]. It has been applied to surveillance and biometrics with increasing efficiency. The human face is of interest in not just the aforementioned areas, but forms some of the most important subjects of multimedia information, such as videos, and are considered as high-level semantic features. An explosion in the generation and availability of such multimedia data has given rise to face clustering as a parallel problem with recognition. Clustering, where one is merely interested in discovering the disjoint classes of faces in a video or set of images, has many applications. A brief (introductory and in no means comprehensive) literature review is given in the paragraph to come.

Ref. [4] explains basic face clustering using SIFT features and agglomerative clustering, and provides insight into the efficiency of clustering with low-level descriptor features. A spectral clustering approach is suggested in [5]. It uses 2DPCA features from the face and constructs a pairwise distance matrix, comparing different spectral clustering methods. Clustering of faces for video indexing is shown in [6] using skin-colour based face detection, and the PCA-eigenspace for comparison between images for clustering. Another use of clustering could be extraction of the principal characters from the video [7]. This paper, describes an approach for automatically generating the list of major casts in a video sequence based on multiple modalities, specifically, both speaker and face information. Face clustering can be used to "clean up" an inaccurately labelled large database of faces [8] or building a new face dataset [9]. It can also be used to facilitate semi-supervised learning, in cases where it might be difficult to get exact labels [10]. All these practical applications of facial clustering, alongwith with ever increasing multimedia information, has generated a lot of research interest in recent years.

In this paper, clustering experiments are performed on the images obtained from a Finnish news broadcast clip. Various methods of dimensionality reduction, linear and otherwise, to project the data into a more representative subspace, are combined with various unsupervised clustering algorithms. The clustering algorithms chosen are simple and are taken from different cluster models, to obtain results with varying degrees of accuracy. The resulting clusters are analyzed in a two-fold way:

- The number of clusters the algorithm and the subspace projection method produces, before mixing up the members of the cluster, are matched against the true number of clusters
- The algorithm is run, with the parameter which determines the number of clusters set to the true number, and the degree of mismatch is checked

This paper is mainly aimed at providing a good starting point for future researchers trying to tackle the problem of face clustering and this is the major contribution.

II. Process Overview

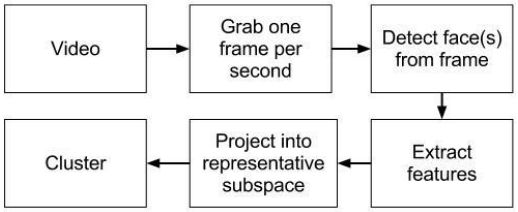

Figure 1. General process overview

The overall process overview for the experiments is shown in the figure above. The frames of the video clip are extracted at the rate of one frame per second and faces are detected from the frame using a Viola-Jones framework [11]. The face images so detected are further subjected to a simple PCA-eigenspace based method to remove non-face areas. Simple features representative of

Manuscript received December 18, 2012; revised February 4, 2013.

doi: 10.12720/joace.1.3.213-216

each face are then extracted from the faces detected and projected into a subspace which is more suitable for clustering. Then, the features in this more representative subspace are used for the actual clustering process. Various algorithms for the subspace projection and clustering are tried at the last two stages of the general process and the results are then analyzed.

III. Face Detection and Feature Extraction

A. Removal of Non-Faces

The detected faces from the Viola-Jones detector [11] may have false positives, i.e., non-face regions detected as faces. In order to tackle the problem, a simple PCA-based method was implemented [2]. A small subset of all the extracted images containing only face regions was chosen and projected to the 'face-space' using PCA. Then a new image is a facial image only if the distance of it's projection from the training 'face-space' is less than a threshold.

B. Feature Extraction

The faces are subjected to a feature extraction step to extract some simple representative features for further use. Simple texture based features have been found to describe facial images well and one particular type, the Local Binary Pattern (LBP), has been found to be very efficient [12], [13].

LBP is a simple and effective texture feature which labels each pixel of an image by thresholding its neighbourhood. The neighbourhood to be operated upon can be varied. An important advantage of using the LBP feature is the computational simplicity with which it can be calculated. Also, these features are quite ideal in our case as they are invariant to rotation and robust against illumination and contrast changes.

IV. Dimensionality Reduction and Clustering

The dimensionality of the feature space generated is in the order of thousands. Particularly, for the LBP feature used in this paper, the dimensionality is 7080. Hence, for handling data effectively and countering the "curse of dimensionality" without losing representative power, it is important to choose a proper dimensionality reduction mechanism to project the data into a lower dimensional subspace. Ideally, the reduced representation should have a dimension which equals the intrinsic dimension of the data, which is the number of parameters needed to account for the properties of the data [14]. In this paper, both linear and non-linear subspace projection methods are tested for clustering accuracy.

A. PCA

Principles components analysis [15] has been a traditional initiation choice when it comes to dimensionality reduction. It is well-documented and simple to understand and apply. Without going into the mathematical detail, it is simply the orthogonal decomposition of the data into its basis vectors, the so-called “principle components”. It is a linear technique in the sense that it embeds the data into a linear subspace of reduced dimensionality. A good way to find the number of components is to test the percentage variance explained by them. In our case, a mere 212 components can explain all the variance.

B. Kernel PCA

Traditional PCA is linear and cannot handle data manifolds that are otherwise. The Kernel PCA (KPCA) is the reformulation of PCA in high-dimensional space using the kernel function [16], [17]. The kernel function encodes the datapoints $\boldsymbol{x_i}$ into kernel matrix, where k_{ij} gives the datapoint at location i,j, calculated by:

$$k_{ij} = \mathsf{K}(\boldsymbol{x_i}, \boldsymbol{x_j}) \tag{1}$$

where, K is the kernel function [17].

The kernel matrix is then double-centered to correspond to the mean subtraction operation on the dataset in traditional PCA.

$$k_{ij} = -(1/2)(k_{ij} - (1/n)\sum_l k_{il} - (1/n)\sum_l k_{jl} + (1/n^2)\sum_{lm} k_{lm}) \tag{2}$$

Now, the first p principal eigenvectors, $\boldsymbol{a_i}$, of the kernel matrix are computed and the data is projected onto them as:

$$\boldsymbol{y_i} = [\sum_{j=1}^{n} a_1^{(j)} \kappa(\boldsymbol{x_j}, \boldsymbol{x_i}), \sum_{j=1}^{n} a_2^{(j)} \kappa(\boldsymbol{x_j}, \boldsymbol{x_i}) \ldots \sum_{j=1}^{n} a_d^{(j)} \kappa(\boldsymbol{x_j}, \boldsymbol{x_i})] \tag{3}$$

where, $a_1^{(j)}$ is the jth value in the vector $\boldsymbol{a_1}$.

In this problem, 210 components were extracted to explain the variance in the data. The choice of kernel function was not experimented with and was fixed at 'gaussian'.

C. *GPLVM* [18]

The Gaussian Process Latent Variable Model (GPLVM) places a Gaussian Process prior on the mapping from the latent space to the observed-data space. Ref. [18] shows that if depending on the covariance function of the GP, the mapping can be linear (equivalent to PCA) or non-linear. The difference of GPLVM with PPCA is that PPCA optimizes the weights and marginalizes over the latent variables, whereas GPLVM optimizes the latent variables and marginalizes over the weights.

The GPLVM uses scaled gradients to optimize both the latent parameters and the kernel. The subset of features selected for representing the dataset is called the *active set* and the rest is called the *inactive set* [18]. In our problem, an active set of 300 features is chosen by performing cross-validation on a labelled subset of the dataset. The kernel chosen is 'gaussian'.

D. t-Distributed Stochastic Neighbour Embedding

The t-Distributed Stochastic Neighbour Embedding (t-SNE) given in [19] is a variant of the Stochastic Neighbour Embedding (SNE), given in [20], and is much more efficient in learning low-dimensional representations from high-dimensional space and much easier to optimize.

t-SNE [19] starts by converting the distance matrix for the high-dimensional datapoints into joint probabilities that shows the similarities,

$$p_{ji} = \frac{\exp\left(-\left(\left(x_i - x_j\right(\right(^2 / 2\sigma^2\right)}{\sum_{k \neq l} \exp\left(-\left(\left(x_k - x_l\right(\right(^2 / 2\sigma^2\right)} \quad (4)$$

where, x_i is a datapoint and σ is the variance of the gaussian centered at the point. The value of p_{ii} is set to 0.

Similar conditional probabilities are also calculated for mapped points y_i, but this time from a t-distribution instead of a gaussian. This is done because the t-distribution has a much heavier tail than the gaussian, and can tackle the problem of "crowding of points" [19].

$$q_{ji} = \frac{\left(1+\left(\left(y_i - y_j\right(\right(^2\right)^{-1}}{\sum_{k \neq l}\left(1+\left(\left(y_k - y_l\right(\right(^2\right)^{-1}} \quad (5)$$

The cost function to be minimized is the KL divergence between p_{ij} and q_{ij}.

In practical cases, the cost function is minimized by a gradient descent and the value of the variances is determined by a user defined perplexity parameter, to which the algorithm is robust. In this paper, the value of the perplexity is fixed at 30 (as suggested in [19]). The number of dimensions of the mapped space is only 100, as determined by cross-validation on a smaller subset of labelled points.

E. Clustering Algorithms

The algorithms tested for the clustering of these dimensionality-reduced features were picked from different types of cluster models:

- Connectivity based model: Clustering based on distances of objects from other objects. Assumes that nearby objects are similar
- Centroid based model: Clusters are based on a central vector, which may or may not be a part of the dataset
- Distribution based model: Clusters are objects which follow the same distribution

Agglomerative hierarchical clustering with average linkage distance and cosine similarity was chosen for the connectivity based clustering algorithm [21]. It starts with placing each point in its own cluster (agglomerative) and combines clusters according to the average of all the distance pairs between the cluster objects. These distances in our case are cosine similarities. It is a representative of the type 1 cluster model as stated above. (Hierarchical clustering with other metrics, namely, Ward's distance, euclidean distance, complete linkage, single linkage, was tried but the average linkage with cosine similarity outperforms them)

For the other types, the k-means algorithm (for type 2) and the gaussian mixture model trained with the EM algorithm (for type 3) are chosen and tested.

V. Dataset

The dataset consists of 217 images detected by the Viola-Jones framework from the frames grabbed at the rate of one per second from a Finnish TV news broadcast clip. Out of the 217 images, 211 are face images and 6 images are wrongly detected non-face images, which get filtered out (section III, A). The 211 images form 8 disjoint clusters describing 8 different people. The dataset is difficult to tackle since it shows the problem of uneven distribution of samples. The newsreader has too many samples and some of the persons interviewed have too few. (Newsreader faces: row 1, column 1; sparse cluster: row 3, column 2)

The representative images of the people in the dataset are given as follows:

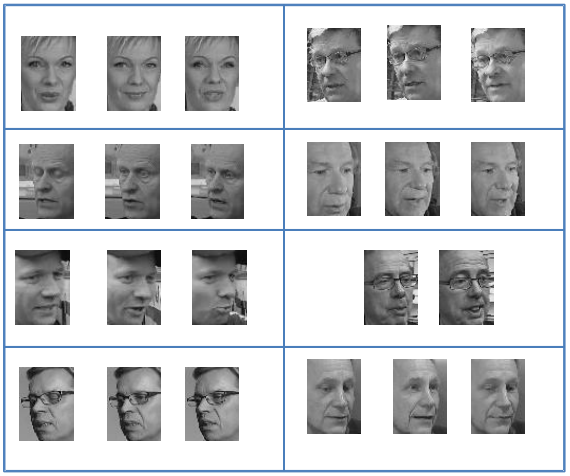

Figure 2. Dataset description

VI. Results and Conclusion

The results of the clustering algorithms were analyzed in a twofold way. For the first method of analysis, the algorithms were allowed to run to output from 30 to 5 clusters (the true number of clusters being 8), and the smallest size of the clustering produced for which the clusters were "pure" (a cluster is not "pure" if two or more different objects occupy that cluster). The results are shown below:

TABLE I. "Pure" Clusters

	PCA	KPCA	GPLVM	tSNE
Hierarchical	11	13	N.A.	9
Kmeans	24	28	N.A.	20
EM-GMM	30	N.A.	N.A.	26

Thus, as is clear from the above table, the tSNE with average-linkage cosine-similarity based agglomerative hierarchical clustering is superior to the other combinations. In particular, this clustering method outperforms the others. One probable reason for this could be that the assumption of the data coming from a mixture of gaussians is invalid. Also, as can be observed, PCA performs surprisingly well on the dataset. The state of the art tSNE performs the best and could be highly recommended as a dimensionality reduction tool. (Here, N.A. means no "pure" cluster for the specified type of experiment, where the number of clusters begins from a maximum of 30)

The second method of analysis is to run the algorithms to output 8 clusters, i.e., the true number of clusters, and visually inspect the clusters to check for the degree of "mixing" (the number of objects which get wrongly clustered together).

TABLE II. ANALYSIS OF DEGREE OF MIXING

	Degree of Mixing for cluster number:							
Method	#1	#2	#3	#4	#5	#6	#7	#8
Hierarchical								
PCA	P	P	P	2	PM	2	P	P
KPCA	P	3	2	P	1P	1P	1P	PM
GPLVM	6	5	5	6	4	6	6	5
tSNE	P	2	P	P	P	P	PM	PM
KMeans								
PCA	P	2	P	P	3	P	PM	2
KPCA	PM	P	P	P	PM	4	PM	PM
GPLVM	1P	4	1P	4	8	4	1P	1P
tSNE	P	PM	PM	PM	PM	PM	7	PM
EM-GMM								
PCA	PM	PM	3	2	P	PM	6	P
KPCA	3	P	2	PM	2	PM	P	PM
GPLVM	5	4	5	6	5	3	6	5
tSNE	PM	PM	P	PM	2	3	PM	P

Here, the symbols are defined as: P for "pure" cluster (without any "mixing"), numeric value for degree of "mixing" (number of different objects in the same cluster), 1P for "pure" cluster with only one object, PM for "pure" cluster of "major" face object (the major face object being the face which occurs the maximum times in the video; in this case, being the newsreader. See row 1, column 1, of Fig. 2)

To analyze, hierarchical clustering with tSNE features outperform the rest. Using this combination, only one cluster is "mixed" with degree 2, and the "major" face cluster (i.e., the newsreader face image) gets split into two "pure" clusters (see Table II). Also, as observed, the PCA features are able to cluster the "major" face object as one disjoint class, which might make them suitable if one wants to only identify the "major" face object from a video.

In all, the tSNE features perform the best, in this test, with hierarchical clustering and can be a good starting point for use for clustering faces.

ACKNOWLEDGMENT

I wish to thank my supervisors Jorma Laaksonen and Markus Koskela for helping me out with the dataset.

REFERENCES

[1] L. Sirovitch and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of Optical Society of America*, 1987, vol. 2, pp. 586-591.
[2] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, USA, 1991, pp. 586-591.
[3] R. Gross, J. Shi, and J. Cohn, "Quo vadis face recognition?" in *Proc. Third Workshop on Empirical Evaluation Methods in Computer Vision*, USA, 2001.
[4] P. Antonopoulos, N. Nikolaidis, and I. Pitas, "Hierarchical face clustering using SIFT image features," in *Proc. IEEE Symposium on Computational Intelligence in Image and Signal Processing*, USA, 2007, pp. 325-329.
[5] S. Foucher and L. Gagnon, "Automatic detection and clustering of actor faces based on spectral clustering techniques," in *Proc. Fourth Canadian Conference on Computer and Robot Vision*, Canada, 2007, pp. 113-122.
[6] C. Czirjek, N. E. O'Connor, S. Marlow, and N. Murphy, "Face detection and clustering for video indexing applications," in *Proc. Advanced Concepts for Intelligent Vision Systems*, Belgium, 2003.
[7] Z. Liu and Y. Wang, "Major cast detection in video using both audio and visual information," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, USA, 2001, vol. 3, pp. 1413-1416.
[8] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y-W. Teh, E. Learned-Miller, and D. A. Forsyth, "Names and faces in the news," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, USA, 2004, vol. 2, pp. 848-854.
[9] D. Ramanan, S. Baker and S. Kakade, "Leveraging archival video for building face datasets," in *Proc. IEEE International Conference on Computer Vision*, Brazil, 2007.
[10] T. Hertz, N. Shental, A. Bar-Hillel, and D. Weinshall, "Enhancing image and video retrieval: learning via equivalence constraints," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, USA, 2003, vol. 2, pp. 668-674.
[11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, USA, 2001, vol. 1, pp. 511-518.
[12] T. Ojala, M. Pietikäinen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," in *Proc. International Conference on Pattern Recognition*, Israel, 1994, vol. 1, pp. 582–585.
[13] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," in *Pattern Recognition*, 1996, vol. 29, pp. 51-59.
[14] L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik, "Dimensionality reduction: A comparative review," in *Tilburg University Technical Report*, 2009.
[15] K. Pearson, "On lines and planes of closest fit to systems of points in space," in *Philosophical Magazine 2*, 1901, pp. 559–572.
[16] B. Schölkopf, A. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299-1319, 1998.
[17] J. Shawe-Taylor and N. Christianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, UK, 2004.
[18] N. D. Lawrence, "Gaussian Process latent variable models for visualization of high dimensional data," in *Advances in Neural Information Processing Systems*, 2004, pp. 329-336.
[19] L. J. P. van der Maaten and G. E. Hinton, "Visualizing high-dimensional data using t-SNE," *Journal of Machine Learning Research* 9, pp. 2579-2605. 2008.
[20] G. Hinton and S. Roweis, "Stochastic neighbour embedding," in *Advances in Neural Information Processing Systems 15*, 2002, pp. 833-840.
[21] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall Inc., USA, 1988.