Enhancing Image and Video Retrieval: Learning via Equivalence Constraints

Tomer Hertz, Noam Shental, Aharon Bar-Hillel and Daphna Weinshall
{email: tomboy,fenoam,aharonbh,daphna@cs.huji.ac.il}
School of Computer Science and Engineering and the Center for Neural Computation
The Hebrew University of Jerusalem, Jerusalem
Israel 91904

Abstract

This paper is about learning using partial information in the form of equivalence constraints. Equivalence constraints provide relational information about the labels of data points, rather than the labels themselves. Our work is motivated by the observation that in many real life applications partial information about the data can be obtained with very little cost. For example, in video indexing we may want to use the fact that a sequence of faces obtained from successive frames in roughly the same location is likely to contain the same unknown individual.

Learning using equivalence constraints is different from learning using labels and poses new technical challenges. In this paper we present three novel methods for clustering and classification which use equivalence constraints. We provide results of our methods on a distributed image querying system that works on a large facial image database, and on the clustering and retrieval of surveillance data. Our results show that we can significantly improve the performance of image retrieval by taking advantage of such assumptions as temporal continuity in the data. Significant improvement is also obtained by making the users of the system take the role of distributed teachers, which reduces the need for expensive labeling by paid human labor.

Keywords: Learning from partial knowledge, semi-supervised learning, image retrieval, clustering

1 Introduction

Supervised learning techniques are designed to use training data where explicit labels are attached to a sample of data points. Obtaining labels in real life applications, such as image and video indexing, is a difficult task that requires human intervention. We argue that in many real life situations training data of different sorts can be extracted automatically or very cheaply from the data. Specifically, we focus on two sources of information:

1. There is often inherent temporal continuity in the data that can be exploited. For example, in video indexing objects extracted from successive frames in roughly the same location can be assumed to come from the same 3D object. Similarly, in surveillance applications we automatically obtain small sequences of images that are known to contain the same intruder. Alternatively, when two people simultaneously walk in front of two *distinct* cameras, the corresponding images are known to contain different people.

2. Anonymous users of a retrieval system can be asked to help annotate the data by providing information about small portions of the data that they see. For example, in what may be viewed as generalized relevance feedback, we may ask the users of a retrieval engine to annotate the set of images retrieved as an answer to their query. Thus, cooperative users will provide a collection of small sets of images which belong to the same category. Moreover, different sets provided by the same user are known to belong to different categories. We cannot use the explicit labels provided by the different users because we cannot assume that the subjective labels are consistent.

The discussion above presents two seemingly unrelated applications that share the same essential property: the presence of intrinsic relations between data points that can be naturally (or cheaply) discovered. This paper addresses the problem of using such information to enhance clustering and classification performance.

1.1 Our approach

As described above, our analysis assumes that apart from having access to a large amount of unlabeled data, we are also provided with additional information in relational form. We focus on *equivalence constraints*, which determine whether two data points come from the same

class or not. We denote the former as "is-equivalent" constraints, and the latter as "not-equivalent" constraints. We note that in both cases, the labels themselves are unknown. We studied three alternative techniques to use equivalence constraints (see details in Section 2):

1. Constrained Gaussian mixture models (GMM): we show how to incorporate equivalence constraints in a GMM formulation using the EM algorithm. As it turns out, "is-equivalent" constraints can be easily incorporated into EM, while "not-equivalent" constraints require heavy duty inference machinery such as Markov networks.

2. Relevant Component Analysis (RCA) combined with nearest neighbor classification: In this algorithm we use "is-equivalent" constraints to determine the dimensions relevant for classification and to implement a semi-supervised learning process. The algorithm was first introduced in [9].

3. Clustering with graph-based algorithms: equivalence constraints are directly incorporated into the Typical-cut algorithm [4] as weights on edges.

1.2 Related work

There has been numerous work in the field of semi-supervised learning. Most of these papers consider the case of partial labeling in which a large unlabeled data set is augmented by a much smaller labeled data set [7, 10]. There are a few papers which exploit equivalence constraints as well. In [12] equivalence constraints are introduced into the K-means clustering algorithm; but since the algorithm computes hard partitioning of the data, the constraints are introduced in a heuristic manner. Another example of introducing equivalence constraints into a graph-based clustering algorithm is given in [13], showing nice improvement in image segmentation.

Different forms of partial information have also been explored. The notion of "preference learning" was studied in [6], where a utility function which corresponds to the teacher's preferences is learned. It is interesting to note that in [6] a set of ordinal labels is assumed to exist, and relations are extracted from labels rather than obtained directly.

2 Using equivalence constraints

In this section we describe three alternative (but related) techniques to use equivalence constraints: Section 2.1 presents a constrained Expectation Maximization (EM) formulation of a Gaussian mixture model (GMM). Section 2.2 describes the RCA algorithm, and Section 2.3 outlines a constrained graph based clustering algorithm.

2.1 Constrained EM

A Gaussian mixture model (GMM) is a parametric statistical model which assumes that the data originates from a weighted sum of several Gaussian sources. More formally a GMM is given by: $p(x|\Theta) = \Sigma_{l=1}^{M} \alpha_l p(x|\theta_l)$, where α_l denotes the weight of each Gaussian, θ_l its respective parameters and M denotes the number of Gaussian sources in the GMM. EM is a widely used method for estimating the parameter set of the model (Θ) using unlabeled data [3]. The algorithm iterates between two steps:

- 'E' step: calculate the expectation of the log-likelihood over all possible assignments of data points to sources.
- 'M' step: differentiate the expectation w.r.t current parameters.

Equivalence constraints modify the 'E' step in the following way: instead of summing over *all* possible assignments of data points to sources, we sum only over assignments which comply with the given constraints. For example, if points x_i and x_j form an "is-equivalent" constraint, we only consider assignments in which both points are assigned to the *same* Gaussian source. If on the other hand, these points form a "not-equivalent" constraint, we only consider assignments in which each of the points is assigned to a *different* Gaussian source.

There is a basic difference between "is-equivalent" (positive) and "not-equivalent" (negative) constraints: While positive constraints are transitive (i.e. a group of pairwise "is-equivalent" constraints can be merged using a transitive closure), negative constraints are not transitive. The outcome of this difference is expressed in the complexity of incorporating each type of constraint into the EM formulation. Therefore we begin by presenting a formulation for positive constraints (Section 2.1.1), and then present a different formulation for negative constraints (Section 2.1.2). We conclude by presenting a unified formulation for both types of constraints (Section 2.1.3).

2.1.1 Incorporating "is-equivalent" constraints

We begin by defining a *chunklet*: a small subset of data points that are known to belong to a single unknown class. Chunklets can also be obtained by the transitive closure of the group of "is-equivalent" constraints.

In this settings we are given a set of unlabeled data points, and a set of chunklets. In order to write down the likelihood of a given assignment of points to sources, a probabilistic model of how chunklets are obtained must be specified. We consider two such models:

1. Data points are sampled i.i.d, without any knowledge about their class membership, and only afterwards chunklets are selected from these points.

2. Chunklets are sampled i.i.d, with respect to the weight of their corresponding source (points within each chunklet are also sampled i.i.d).

Although the first model above seems more sound statistically, it suffers from two major drawbacks: (i) All our current suggestions for how to automatically obtain chunklets do not fit this statistical model. (2) Unlike the standard EM formulation for a GMM, it does not have a closed form iterative solution for the sources' weights. In this case we must apply generalized EM (GEM) [3] using gradient ascent. We therefore defer the discussion of this model to Section 2.1.3.

The second model suggested above often complies with the way chunklets are "naturally" obtained. For example, in surveillance applications data is obtained in chunklets. Therefore the probability of obtaining a chunklet of a specific person is proportional to the number of events that the person was tracked by the surveillance cameras.

Fortunately, the EM update rules for the second model have a closed form solution. More specifically, let $\{x_i\}_{i=1}^N$ denote the data points. Let $\{X_j\}_{j=1}^L,\ L < N$ denote the distinct chunklets, where each X_j is a set of points x_i (one or more) such that $\bigcup_{j=1}^L X_j = \bigcup_{i=1}^N x_i$. Let y_i denote the label assignment of point i, and $Y_j = \{y_j^1 \dots y_j^{|X_j|}\}$ denote the label assignment of the chunklet X_j. The iterative EM equations for the parameters of the l'th model are:

$$\begin{aligned}
\alpha_l^{new} &= \frac{1}{L}\sum_{j=1}^{L} p(Y_j = l|\{X_j\},\Theta^{old}) \\
\mu_l^{new} &= \frac{\sum_{j=1}^{L}\bar{X}_j p(Y_j = l|\{X_j\},\Theta^{old})|X_j|}{\sum_{j=1}^{L} p(Y_j = l|\{X_j\},\Theta^{old})|X_j|} \\
\Sigma_l^{new} &= \frac{\sum_{j=1}^{L}\Sigma_{jl}^{new} p(Y_j = l|\{X_j\},\Theta^{old})|X_j|}{\sum_{j=1}^{L} p(Y_j = l|\{X_j\},\Theta^{old})|X_j|} \\
\Sigma_{jl}^{new} &= \frac{\sum_{x_i\in X_j}(x_i-\mu_l^{new})(x_i-\mu_l^{new})^T}{|X_j|}
\end{aligned}$$

where $\bar{X}_j$ denotes the mean of the points in chunklet j, $|X_j|$ the number of points in chunklet j, and Σ_{jl}^{new} the mean covariance matrix of the jth chunklet. The chunklet probability is:

$$\begin{aligned}
& p(Y_j = l|\{X_j\},\Theta^{old}) \\
&= p(\{y_j^1 = l \dots y_j^{|X_j|} = l\}|\{X_j\},\Theta^{old}) \\
&= \frac{\alpha_l^{old}\prod_{x_i\in X_j} p(x_i|y_i^j = l,\Theta^{old})}{\sum_{m=1}^{M}\alpha_m^{old}\prod_{x_i\in X_j} p(x_i|y_i^j = m,\Theta^{old})}
\end{aligned}$$

As can be readily seen, the update rules above effectively treat each chunklet as a single data point weighed according to the number of elements in it. Note that in our formulation unconstrained data points appear as chunklets of size 1.

2.1.2 Incorporating "not-equivalent" constraints

The probabilistic description of a data set using a GMM attaches to each data point two random variables: an observable and a hidden. The hidden variable of a point describes its source label, while the data point itself is an observed example from the source. Each pair of observable and hidden variables is assumed to be independent of the other pairs. Note, however, that negative equivalence constraints violate this assumption, as dependencies between the hidden variables are introduced.

Specifically, assume we have a group $A = \{(a_i^1, a_i^2)\}_{i=1}^M$ of pairs of indices corresponding to M pairs of points that are negatively connected, and define the event $E_A = \{Y \text{ complies with the constraints}\}$.

$$\begin{aligned}
p(X,Y|\Theta,E_A) &= p(X|Y,\Theta,E_A)p(Y|\Theta,E_A) \\
&= \frac{p(X|Y,\Theta)p(E_A|Y)p(Y|\Theta)}{p(E_A|\Theta)}
\end{aligned} \tag{1}$$

We denote the constant $p(E_A|\Theta)$ as Z. Using the independence of samples we get

$$p(X,Y|\Theta,E_A) = \frac{1}{Z}1_{Y\in E_A}\prod_{i=1}^{N} p(y_i|\Theta)p(x_i|y_i,\Theta)$$

Expanding $1_{Y\in E_A}$ gives the following expression

$$\begin{aligned}
& p(X,Y|\Theta,E_A) = \\
& \frac{1}{Z}\prod_{(a_i^1,a_i^2)}(1-\delta_{y_{a_i^1},y_{a_i^2}})\prod_{i=1}^{N} p(y_i|\Theta)p(x_i|y_i,\Theta)
\end{aligned} \tag{2}$$

As a by-product of its local components, the distribution in (2) can be readily described using a Markov network. Each observable data point depends, in a Gaussian manner, on a hidden variable corresponding to the label of its source. Negative constraints are expressed by edges between hidden variables which prevent them from having the same value (see Fig. 1).

We derived an EM procedure which maximizes $log(p(X|\Theta,E_A))$ entailed by this distribution. The update rules for μ_i and Σ_i are still

$$\begin{aligned}
\mu_l^{new} &= \frac{\sum_{i=1}^{N} x_i p(y_i = l|X,\Theta^{old})}{\sum_{i=1}^{N} p(y_i = l|X,\Theta^{old})} \\
\Sigma_l^{new} &= \frac{\sum_{i=1}^{N}\widehat{\Sigma_i} p(y_i = l|X,\Theta^{old})}{\sum_{i=1}^{N} p(y_i = l|X,\Theta^{old})}
\end{aligned}$$

where $\widehat{\Sigma_i} = (x_i - \mu_l^{new})(x_i - \mu_l^{new})^T$. Note that now $p(y_i = l|X, \Theta^{old})$ are inferred using the net. The update rule of α_l is more intricate, since this parameter appears in the normalization factor $Z = p(E_A|\Theta)$

$$\begin{aligned} Z &= \sum_Y p(E_A|Y)p(Y|\Theta) \\ &= \sum_{y_1} \cdots \sum_{y_N} \prod_{i=1}^{N} \alpha_{y_i} \prod_{(a_i^1, a_i^2)} (1 - \delta_{y_{a_i^1}, y_{a_i^2}}) \end{aligned}$$

This factor can be calculated using a net which is similar to the one discussed above but lacks the observable nodes. We use such a net to calculate Z and differentiate it w.r.t α_l, after which we perform gradient ascent.

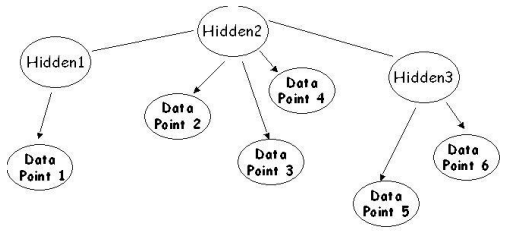

Figure 1. Illustration of the Markov network structure required for incorporating "is-equivalent" and "not-equivalent" constraints. Data points $2-4$ are positively constrained, and so are points $5-6$. Data points $2-4$ have a negative constraint with point 1 and with point $5-6$.

2.1.3 Combining "is-equivalent" and "not-equivalent" constraints

Both kinds of constraints can be combined in a single Markov network by a rather simple extension of the network described in the previous section. The data distribution can be expressed by a Markov network similar to the network from the previous section, where every pair of data points related by a positive constraint share a hidden father node (see Fig 1).

The complexity of an EM round is dominated by the inference complexity, which is $O(M^{induced\ width(G)})$ by using the Jtree algorithm [8]. Hence the algorithm is limited to $O(N)$ "not equivalent" constraints. The general case of $O(N^2)$ is NP-hard, as it may be reduced to the graph coloring problem. To achieve scalability to large sets of constraints two approximations are used: the graph is replaced by its minimal spanning tree, and the normalization factor Z is approximated (for details see [1]).

2.2 Relevant Component Analysis

Relevant Component Analysis (RCA) is a method that seeks to identify and down-scale global unwanted variability within the data. The method changes the feature space used for data representation (or equivalently the distance between data points in feature space), by a linear transformation which assigns large weights to "relevant dimensions" and low weights to "irrelevant dimensions" (cf. [11]). Hopefully, in the new feature space the inherent structure of the data could be more easily unraveled. The method can be used as a preprocessing step both for unsupervised clustering of data, or for using nearest neighbor classification. Specifically, we do the following:

1. Assume that chunklets are provided as input (Fig. 2a-c).[1] For each chunklet: subtract the chunklet's mean from all of the points belonging to it (Fig. 2d).

2. Merge the recentered chunklets into a single data set and compute the covariance matrix of the merged set (Fig. 2d).

3. Compute the whitening transformation W associated with this covariance matrix (Fig. 2e) and apply it to the data. Apply the transformation W to the original data points: $x_{new} = Wx$ (Fig. 2f).

What the whitening transformation W effectively does, when applied to the original feature space, is give lower weight to directions in feature space in which the variability is mainly due to within class variability, and is therefore "irrelevant" for the task of classification.

In section 3 we provide results using RCA on both unsupervised clustering and nearest neighbor classification of real image data. In [2] we provide the theoretical justification of RCA. RCA is analyzed from an an information-theoretic view and is shown to be the optimal procedure under several criteria.

2.3 Constrained graph-based clustering

Graph based clustering methods represent the data points as a graph in which the nodes correspond to data points, and the edge values correspond to the similarity between pairs of data points. In this representation there are several ways of introducing equivalence constraints, which depend on the specific clustering algorithm used.

In the examples below we describe results of incorporating equivalence constraints in the Typical Cut clustering algorithm [4]. The incorporation of *is-equivalent* constraints is done by assigning 0 weight to all edges connecting chunklet points. However, in order to incorporate *not-equivalent*

[1] "not-equivalent" constraints are ignored in the current version of RCA.

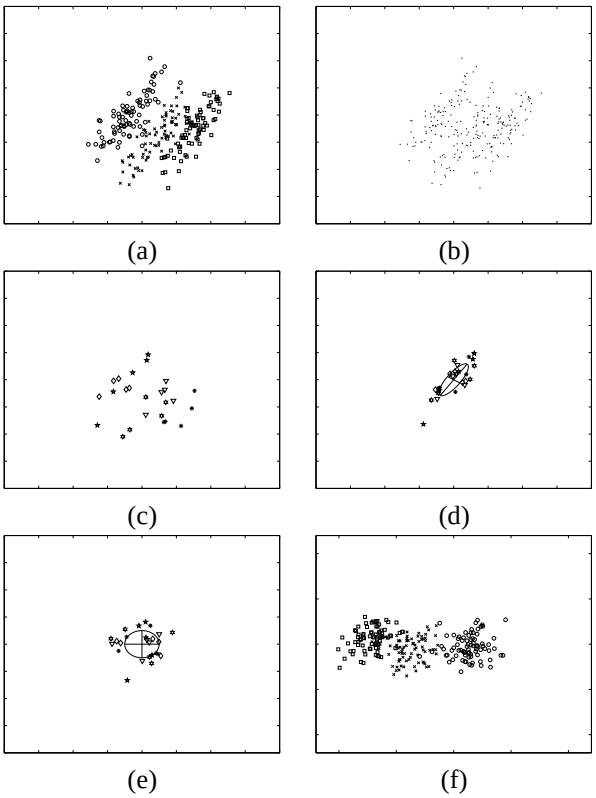

Figure 2. An illustrative example of the RCA algorithm applied to synthetic Guassian data. (a) The fully labeled data set with 3 labels. (b) Same data unlabeled; clearly the classes' structure is less evident. (c) The set of chunklets that are provided to the RCA algorithm. (d) The centered chunklets, and their empirical covariance. (e) The whitening transformation applied to the chunklets. (f) The original data after applying the RCA transformation to it.

constraints edges should be assigned the value $-\infty$. Such assignment poses a technical difficulty when using graph cut methods, and as a result has not been implemented yet.

3 Experimental results

3.1 Image retrieval and clustering of facial images

To evaluate the performance of our algorithms we used a subset of the Yale face database B [5] which contains a total of 1920 images, including 64 frontal pose images from 30 different subjects (see examples in the top row of Fig. 3). In this database the variability between the images of the same person is due mainly to different lighting conditions. We automatically centered all the images using optical flow. Images were then converted to vectors, and each image was represented using the first 60-100 PCA coefficients.

Our experimental setup We constructed an experimental design using the Yale B face database and a simple distributed retrieval engine. The retrieval engine used a naive

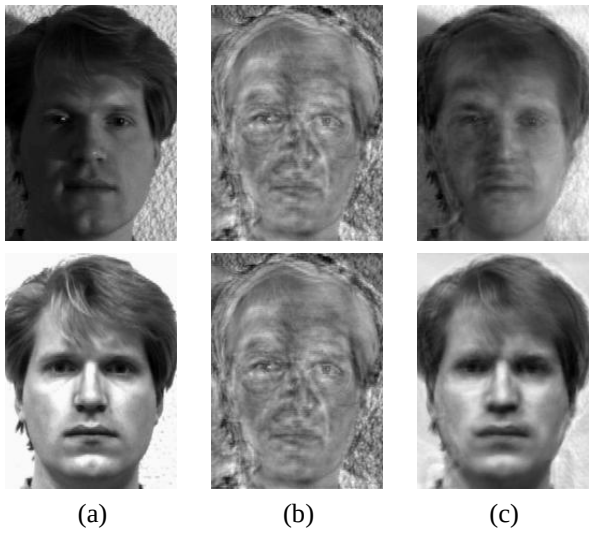

Figure 3. (a) Original images taken from the yaleA image database. (b) Reconstructed images using RCA. (c) Reconstructed images using PCA. Notice that although the original images are taken under very different lighting conditions, their RCA reconstructions cancel out this effect.

metric for locating the K-nearest neighbors of a query image. The naive metric was a Mahalanobis distance which used the total covariance matrix of the dataset as weights.[2]

The experiment was conducted as follows: each user selected a facial image (data point) from the database, and presented it as a query to the system. The retrieval engine computed its K-nearest neighbors and presented them to the user. The user was then asked to partition the set of retrieved images into equivalence classes (i.e., into groups of pictures containing the same individual). Each user thus supplied the system with both positive and negative constraints, in what may be viewed as generalized relevance feedback. These equivalence constraints were used to test the performance of our algorithms.

Using the RCA algorithm and the procedure described above for obtaining equivalence constraints, we studied the effect of the number of data queries while K (the number of neighbors) was set to 10. The beneficial effect of RCA on nearest neighbor classification is shown in Fig. 4.

It is also interesting to observe the visual effect of applying RCA to facial images with varying lighting conditions. We computed the RCA transformation using a set of chunklets, applied it to the data, and then reconstructed the transformed images. Fig. 3 illustrates the "effect" of RCA when applied to facial images of a single subject.

We used similarity constraints obtained in the distributed learning scenario to test our constrained EM algorithms.

[2] This naive metric proved to be empirically superior to the simple Euclidean metric.

	Regular EM		EM with "is equivalent"		EM with "is+not equivalent"	
# queries	Purity	Accuracy	Purity	Accuracy	Purity	Accuracy
10	$59 \pm 6\%$	$64 \pm 5\%$	$80 \pm 5\%$	$82 \pm 5\%$	$85 \pm 7\%$	$88 \pm 5\%$
15	$59 \pm 6\%$	$64 \pm 5\%$	$82 \pm 6\%$	$83 \pm 5\%$	$87 \pm 6\%$	$88 \pm 5\%$

Figure 5. Purity (precision) and accuracy (recall) scores of EM face clustering. The results were averaged over 100 tests.

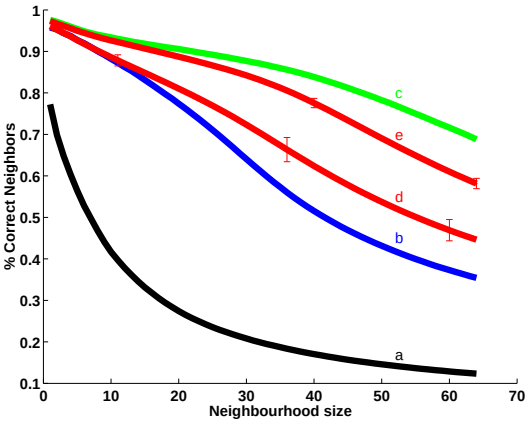

Figure 4. Mean cumulative neighbor purity (percentage of correct neighbors) before and after applying RCA to the Yale B facial image database. The graph compares different Mahalanobis distance matrices: (a) Euclidean distance (b) Euclidean distance after a global whitening of the data. (c) inner class covariance matrix. (d,e) RCA estimators of the inner class covariance matrix using 20 and 60 distributed queries respectively. Results were averaged over 50 chunklet realizations, while the error bars depict standard deviations. The results are shown for 20, and 60 queries. As can be seen, performance improves as the number of queries increases, although a large part of the performance gain is obtained with 20 queries.

The task was the clustering of 640 facial images belonging to 10 subjects. We compared the performance of the following: (1) Regular EM, (2) Positively constrained EM and (3) Fully constrained EM (both positive and negative constraints). The constraints were obtained using distributed queries of 15 faces each, randomly selected. Fig. 5 shows the results. As may be seen, incorporating the constraints improves both purity (precision) and accuracy (recall) scores by 20-25%, depending on the number of constraints used. Most of the improvement can be achieved using the positive constraints alone, in which case the algorithm has a closed form fast implementation.

3.2 Visual surveillance

In the surveillance application data was collected by a stationary indoor surveillance camera, from which short video clips were extracted. The beginning and end of each clip were automatically detected by the appearance or disappearance of a moving target. The database included many clips, each displaying only one person; however, the identity of the intruder was unknown. The task was to retrieve images from this database, based on individual query images.

The task and our approach In this experiment the data was taken in entirely natural conditions, and was therefore rather challenging. Specifically, in a certain video clip an intruder may have walked all over the scene, coming closer to the camera or walking away from it. Thus the size and resolution of each image varied dramatically. In addition, since the environment was not controlled, images included variability due to occlusions, reflections and (most importantly from our perspective) highly variable illumination. In fact, the illumination changed dramatically across the scene both in intensity (from brighter to darker regions), and in spectrum (from neon light to natural lighting). Fig. 6 displays several images from one input clip.

Our goal was to devise a representation that would enable effective retrieval of images. We found that the only low-level attribute that could be reliably used in this application was color. Therefore our task was to accomplish some sort of color constancy, i.e., to overcome the irrelevant variability created by the changing illumination.

Figure 6. Several images from a video clip of one intruder.

Image representation and the application of our methods Each image in a clip was represented using its color

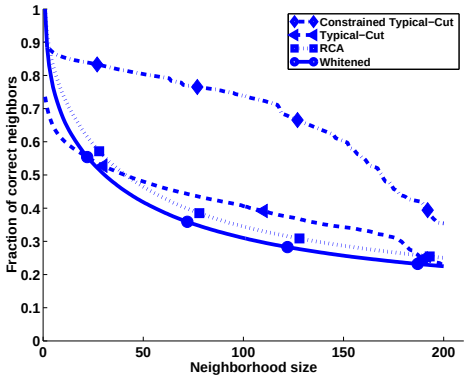

Figure 7. Fraction of correct retrievals as a function of the number of retrieved images. Remark about retrieval in the clustering cases: Both clustering algorithms produce a hierarchy of clusters, i.e. a dendrogram, which is used in the retrieval procedure. In order to retrieve the neighbors of a certain point i, one locates the smallest cluster which includes i, and retrieves its neighbors in ascending distance from i (ignoring neighbors that belong to the same chunklet as i). If needed one climbs up the dendrogram, and repeats this process disregarding points that were part of lower levels in the dendrogram.

histogram in $L^*a^*b^*$ space (we used 5 bins for each dimension). Since a clip forms a chunklet by definition, we can naturally apply our methods without further preprocessing of the data. We tested the effect of chunklets in two of our methods, i.e, graph based clustering and RCA. We used 4600 images from 130 clips (chunklets) of 20 different people. We tested the percent of correct retrievals as a function of the number of retrieved images, over four methods of k-nearest neighbor classification: (1) Whitened feature space: First whiten the data, and then sort the neighbors using the $L1$ distance between images in the transformed feature space. (2) RCA: Perform RCA and then sort the neighbors using the $L1$ distance in the new feature space. (3) Unconstrained Typical-cut algorithm. (4) Constrained Typical-cut: Cluster the data in the whitened feature space using a constrained version of the Typical-cut algorithm. Results are shown in Fig. 7 As may be seen using constrained Typical-cut significantly outperforms all other methods. A smaller but still noticeable difference exists between RCA and the whitened space.

4 Concluding remarks

In this paper, our key observation was that equivalence constraints between data points can be automatically obtained in some applications. It is also possible to obtain such partial information in a distributed manner. We presented a number of novel algorithms which make use of equivalence constraints and significantly enhance classification and clustering performance in image retrieval and in surveillance applications.

References

[1] A. Bar-Hillel, T. Hertz, N. Shental and D. Weinshall. Learning via Equivalence Constraints, with applications to the Enhancement of Image and Video Retrieval. Technical Report no. 2002-51. see *http://leibniz.cs.huji.ac.il/tr/acc/2002/HUJI-CSE-LTR-2002-51_techRep.pdf*

[2] A. Bar-Hillel, N. Shental, T. Hertz and D. Weinshall. Learning Distance Functions using Equivalence Relations. Submitted to ICML, 2003.

[3] A. P. Dempster, N. M. Laird and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. In *Journal of the Royal Statistical Society*, B, 39, 1-38, 1977.

[4] Y. Gdalyahu, D. Weinshall, and M. Werman. Self organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization. *IEEE Tran. PAMI*, 23(10):1053–1074, 2001.

[5] A.S. Georghiades, P.N. Belhumeur and D.J. Kriegman From Few To Many: Generative Models For Recognition Under Variable Pose and Illumination", IEEE Int. Conf. on Automatic Face and Gesture Recognition, page 277-284, 2000.

[6] R. Herbrich, T. Graepel, P. Hollmann-Sdorra, and K. Obermayer. Supervised learning of preference relations. In *Fachgruppentreffen Maschinelles Lernen*, pages 43–47, 1998.

[7] D.J. Miller and H.S. Uyar. A mixture of experts classifier with learning based on both labeled and unlabeled data. In *NIPS*, volume 9, pages 571–578, 1997.

[8] J. Pearl. Probabilistic reasoning in intelligent systems. Morgan Kaufmann, 1998.

[9] N. Shental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis. In *Proc. of ECCV*, Copenhagen, DK, June 2002.

[10] M. Szummer and T. Jaakkola Partially labeled classification with markov random walks. In *NIPS*, volume 14, 2001.

[11] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.

[12] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proc. of ICML*, pages 577–584, 2001.

[13] S. X. Yu and J. Shi. Grouping with bias. In *NIPS*, 2001