A Discriminative Learning Framework with Pairwise Constraints for Video Object Classification

Paper 1134

Abstract

In video object classification, insufficient labeled data may at times be easily augmented with pairwise constraints on sample points, i.e, whether they are in the same class or not. In this paper, we proposed a regularized discriminative learning approach which incorporates pairwise constraints into a conventional margin-based learning framework. The proposed approach offers several advantages over existing approaches dealing with pairwise constraints. First, as opposed to learning distance metrics, the new approach derives its classification power by directly modeling the class boundary. Second, most previous work handles labeled data by converting it to pairwise constraints and thus leads to much more computation. The proposed approach can handle pairwise constraints together with labeled data simultaneously so that the computation is greatly reduced. The performance of the proposed approach is evaluated on a people classification task with two surveillance video datasets.

1. Introduction

Learn with insufficient training data in classifying or recognizing objects/people has recently become an interesting topic [1, 2]. One solution for this problem is to integrate new knowledge sources that are complementary to the training data. In this paper, we are particularly interested in how to incorporate additional pairwise constraints to improve classification performance in video. More specifically, a pairwise constraint between two examples describes whether they belong to the same class or not, which provides a relationship between the labels rather than labels themselves. The inherent characteristics, that is, the sequential continuity and multi-modalities of video streams allow us to pose different types of constraints to boost the learning performance. These constraints can at times be obtained automatically or only with little human effort.

Figure 1 illustrates several examples of pairwise constraints in a scenario of classifying people's identity from surveillance video. First, constraints can be obtained from knowledge of temporal relations. For instance, two spatially overlapping regions extracted from temporally adjacent frames can be assumed to share the same labels

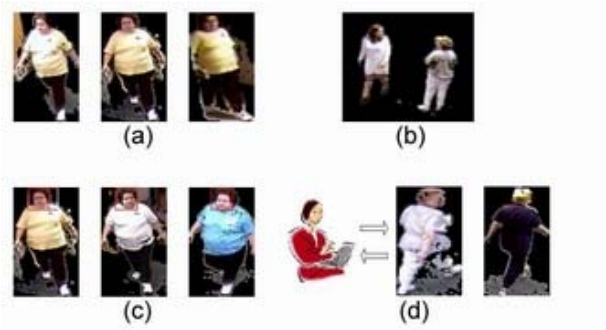


Figure 1: Examples of different kinds of pairwise constraints. (a) Temporal constraints from a single tracking sequence, (b) Temporal constraints of different regions extracted at the same time, (c) Constraints provided by comparing faces, (d) Constraints provided by user feedback

whereas two regions appeared simultaneously in a camera cannot be labeled as the same. Second, we can extract constraints from various modalities such as visual(face) and auditory(voice) cues. For example, conventionally, if we want to automatically identify a person's face from a video sequence, we need to train a model from many training samples of the same person with different head poses and under different lighting conditions. With the representation of pairwise constraints, we only need a face comparison algorithm to provide the pairwise relation between examples without building statistical models for every possible subject under every possible circumstance. This provides an alternative framework to aggregate different modalities, especially when the training examples of people of interest are limited or not available at all. Finally, constraints can also come from human feedback. In contrast to relevance feedback which asks users for label information, providing pairwise constraints does not necessarily require users to have prior knowledge or experience with the data set.

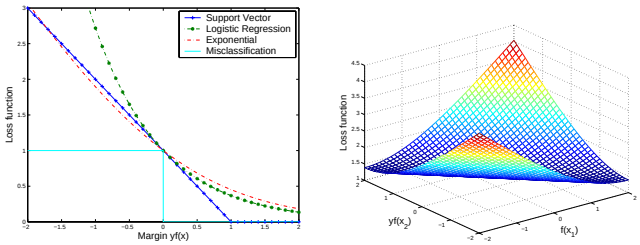
In previous research some efforts have been made to help both supervised and unsupervised learning with pairwise constraints [3, 4, 5, 6]. In the context of graph partitioning, S. Yu et al[3] has successfully integrated pairwise constraints into a constrained grouping framework, leading to improved segmentation results. In more closely related work proposed by Xing et al [6], a distance metric learn-

ing method is proposed to incorporate pairwise information and solved by convex optimization. However, it contains an iterative procedure with projection and eigen value decomposition which is computationally expensive and sensitive to parameter tuning. By comparison, relevance component analysis (RCA) [4] is a simple and efficient approach for learning a full Mahalanobis metric. In this work, a chunklet is defined as a subset of points that belong to the same class but the identity of this class is unknown. An inverse of the covariance matrix of all the center-points in the chunklets is computed as a Mahalannobis distance. However, only positive constraints could be utilized in this algorithm. In [4], Shental et al also propose a constrained Gaussian mixture model which incorporate the positive and negative pairwise constraints into GMM model using EM algorithm. More recently, pairwise constraints have been found useful in the context of kernel learning. Kwok et al[5] formulates the kernel adaptation problem as a distance metric learning problem searching for a suitable linear transform in the kernel-induced feature space, even if it is of infinite dimensionality. Most of the above techniques focus on learning (Mahalanobis) distance metrics or generative classifiers by estimating the joint probability $p(x, y)$. However, for the task of classification, discriminative classifiers which learn the posterior $p(y|x)$ directly have their own advantages because the decision boundary might be simple even when true underlying distance metric is complex. Moreover, for most of these algorithms the only way of dealing with labeled data is to convert the labels into the pairwise constraints between every data pair. This drawback makes the implementation rather inefficient and thus does limit usage in real applications.

In this work, we propose a new regularized discriminative learning approach which naturally incorporates pairwise constraints into a conventional margin-based learning framework. The proposed approach allows the classifiers using additional pairwise constraints with labeled data to model the decision boundary directly, instead of resorting to seek an underlying distance metric which could be much more complex. Analogous to kernel logistic regression [7], we also derive a kernelized representation of our proposed pairwise learning framework using a logistic regression loss function, which is called "pairwise kernel logistic regression" in this work. This algorithm is evaluated in the context of classifying people's identities from the surveillance video.

2. Discriminative Learning with Pairwise Constraints

Formally, the goal of classification is to produce a hypothesis $f : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X}$ denotes the domain of possible examples, $\mathcal{Y}$ denotes a finite set of classes. The learning algorithm typically takes a set of training examples $(x_1, y_1), ..., (x_m, y_m)$ as input, where $y_i \in \mathcal{Y}$ is the label assigned to example $x_i \in \mathcal{X}$. Moreover, in addition to the data with explicit labels, there is another set of pairwise constraints $(x_{11}, x_{12}, y'_1) \ldots (x_{n1}, x_{n2}, y'_n)$ available from entire data pool including both labeled and unlabeled data, where $y'_i \in \{-1, 1\}$ is the pairwise constraint assigned to two examples $x_{i1}, x_{i2} \in \mathcal{X}$. For the sake of simplicity, $(x_{i1}, x_{i2}, 1)$ will be called the positive constraints which means the example pair (x_{i1}, x_{i2}) belongs to the same class and $(x_{i1}, x_{i2}, -1)$ the negative constraints defined similarly.



(a) (b)

Figure 2: Comparison of loss functions (a) Comparison of four different loss functions against margin $yf(x)$. The losses are misclassification $I(sgn(f) \neq y)$, exponential $exp(-yf)$, support vector $(1 - yf)_+$, logistic regression $log(1 + exp(-yf(x))$. (b) The pairwise loss function in Eq(6) against $f(x_1)$ and $yf(x_2)$

2.1 Regularized loss function with pair-wise information

We begin our discussion with the case of binary classification. Many machine learning algorithms attempt to minimize the regularized empirical risk

$$\min_f R_{reg}(f) = \sum_{i=1}^{m} L(y_i, f(x_i)) + \lambda\Omega(\|f\|_{\mathcal{H}}), \qquad (1)$$

where L is the empirical loss function, $\Omega(\cdot)$ is some monotonically increasing regularization function on the domain $[0, +\infty]$ which controls the complexity of the hypothesis space, $\mathcal{H}$ denotes a reproducing kernel Hilbert space(RKHS) generated by some positive definite kernel K, $\|\cdot\|_{\mathcal{H}}$ the corresponding norm and λ is the regularization constant. The empirical loss function $L(y_i, f(x_i))$ is usually set to a function of "margin" $yf(x)$ [8], i.e. $L(y_i, f(x_i)) = \widetilde{L}(y_i f(x_i))$. With different choices of loss functions and regularization terms, we can derive a large family of well-studied algorithms from Eq(1). For example, the solution for parameters in a support vector machine (SVM) is the minimization of the following regularized loss

function,

$$\sum_{i=1}^{m} max(1 - y_i f(x_i), 0) + \lambda \|w\|_{\mathcal{H}}^2. \quad (2)$$

Therefore, the SVM can be viewed as a binary margin-based learning algorithm with loss function $\widetilde{L}(z) = max(1 - z, 0)$ and regularization factor $\|w\|_2^2$. More examples can be found in the work of [8]. To illustrate, figure 2(a) shows a comparison of four different loss functions against the margin $yf(x)$.

Within this learning framework, pairwise constraints can be introduced as another set of empirical loss functions, which penalize the violation of given constraints,

$$\sum_{i=1}^{m} L(y_i, f(x_i)) + \mu \sum_{i=1}^{n} L'(y'_i, f(x_{i1}), f(x_{i2})) + \lambda\Omega(\|f\|_{\mathcal{H}}), \quad (3)$$

where we call $L'(y'_i, f(x_{i1}), f(x_{i2}))$ pairwise loss functions. It is desirable for the pairwise loss function to give a high penalty when $f(x_{i1})$ and $y'_i f(x_{i2})$ have different signs but low penalty otherwise. Meanwhile, the loss functions should be robust to noisy data. Therefore, the problem can be translated into seeking a family of loss functions with the above properties. Analogous to misclassification loss, we can choose

$$L'(y'_i, f(x_{i1}), f(x_{i2})) = I(sgn[f(x_{i1})] \neq sgn[y'_i f(x_{i2})]),$$

which gives a unit penalty for violation of pairwise constraints, and no penalty at all otherwise. Although minimizing this exact loss may be worthwhile, in this form it is generally intractable to solve and even worse, it is not robust to noisy data without the ability to penalize large errors more heavily. Following the idea of "margin", we can also choose a pairwise loss function to be a monotonic decreasing function of $y'_i f(x_{i1}) f(x_{i2})$, i.e.

$$L'(y'_i, f(x_{i1}), f(x_{i2})) = \widetilde{L}'(y'_i f(x_{i1}) f(x_{i2})).$$

However, in most cases this function is not a convex function and thus finding global optimum is no longer guaranteed. Taking all these factors into account, we choose loss function L' a monotonic decreasing function of the difference between the predictions of two pairwise constraints $f(x_{i1}) - y'_i f(x_{i2})$, i.e.,

$$L'(y'_i, f(x_{i1}), f(x_{i2})) = \widetilde{L}'(f(x_{i1}) - y'_i f(x_{i2})),$$

which plays a similar role as the residues $y - f(x)$ in regression. The intuition is that the prediction difference can be a "soft" measure of how possible it is the pairwise constraints would be violated. When $\widetilde{L}'$ is convex, most of these pairwise loss functions have a nice property of convexity to $f(x_{i1})$ and $f(x_{i1})$, and thus allows us to apply standard convex optimization techniques.

Similar to the loss function in regression, the pairwise loss function should be symmetric for any example pair, i.e., $\widetilde{L}'(f(x_{i1}) - y'_i f(x_{i2})) = \widetilde{L}'(y'_i f(x_{i2}) - f(x_{i1}))$. Therefore, $\widetilde{L}'$ is an even function and could be represented as $\widetilde{L}'(x) = \widetilde{L''}(x) + \widetilde{L''}(-x)$, where $\widetilde{L''}$ now can be any monotonic decreasing function $f : \mathcal{X} \to \mathbf{R}$. To ensure the label loss function and pairwise loss function are comparable, we usually choose $\widetilde{L''}$ in the same form as $\widetilde{L}$. Putting all these together, our primal optimization problem has the following form,

$$\sum_{i=1}^{m} \widetilde{L}(y_i f(x_i)) + \mu \sum_{i=1}^{n} \widetilde{L}(f(x_{i1}) - y'_i f(x_{i2})) + \mu \sum_{i=1}^{n} \widetilde{L}(y'_i f(x_{i2}) - f(x_{i1})) + \lambda\Omega(\|f\|_{\mathcal{H}}). \quad (4)$$

Note that when the number of pairwise constraints n is zero, it trivially degraded to a margin-based learning problem with only labeled data.

A special case for Eq(4) is to fit a linear decision boundary on the input feature space, i.e., $f(x)$ can be expressed in form of $w^T x$ and $\|f\|_{\mathcal{H}} = \|w\|$ in the L_2 space. Substituting $f(x) = w^T x$ and $\|f\|_{\mathcal{H}} = \|w\|$ into Eq(4), we have

$$\sum_{i=1}^{m} \widetilde{L}(y_i w^T x_i) + \mu \sum_{i=1}^{n} \widetilde{L}(w^T x_{i1} - y'_i w^T x_{i2}) + \mu \sum_{i=1}^{n} \widetilde{L}(y'_i w^T x_{i2} - w^T x_{i1}) + \lambda\Omega(\|w\|). \quad (5)$$

It can be shown that the objective function of Eq(5) when $\mu = 1$ is equivalent to the objective function of Eq(1) with a expanded labeled data set, which includes $2n$ pseudo-labeled data $(x_{i1} - y'_i x_{i2}, 1)$ and $(x_{i1} - y'_i x_{i2}, -1)$ in addition to original labeled data. This property is intriguing because it allows a quicker implementation for linear kernel classifiers by means of adding $2n$ new training examples without modifying existing algorithms or software packages.

Note that in our experimental implementation, we adopt the logistic regression loss function as the empirical loss function $\widetilde{L}(x) = \log(1 + e^{-x})$, yielding

$$\sum_{i=1}^{m} \log(1 + e^{-y_i f(x_i)}) + \mu \sum_{i=1}^{n} \log(1 + e^{f(x_{i1}) - y'_i f(x_{i2})}) + \mu \sum_{i=1}^{n} \log(1 + e^{y'_i f(x_{i2}) - f(x_{i1})}) + \lambda\Omega(\|f\|_{\mathcal{H}}). \quad (6)$$

because it can be easily solved by unconstrained optimization techniques. However, our discussion can be extended to other loss functions as well. Figure 2(b) depicts the pairwise loss function used in Eq(6).

2.2 Kernelization

In this section, we present the kernelized representation of the primal problem Eq(6) using the representer theorem [9]. This representation allows simple learning algorithms to construct a complex decision boundary by projecting the original input space to a high dimensional feature space, even infinitely dimensional in some cases. This computationally intensive task is achieved through a positive definite reproducing kernel K and the well-known "kernel trick".

To begin, let $C(\cdot)$ represent the empirical loss and $\Omega(\|f\|_{\mathcal{H}}) = \|f\|^2_{\mathcal{H}}$. Therefore, the primal problem Eq(6) can be rewritten as,

$$\min_{f\in\mathcal{H}} C(\{y_i, f(x_i)\}, \{y'_i, f(x_{i1}), f(x_{i2})\}) + \lambda\|f\|^2_{\mathcal{H}}. \quad (7)$$

The loss function $C(\cdot)$ is pointwise, which only depends on the value of f at the data points $\{f(x_i), f(x_{i1}), f(x_{i2})\}$. Therefore by the representer theorem, the minimizer $f(x)$ admits a representation of the form

$$f(\cdot) = \sum_{i=1}^{m'} \alpha_i K(\cdot, \bar{x}_i)), \quad (8)$$

where $m' = m + 2n$, $\bar{x}_i = \{x_1, \ldots, x_m\} \vee \{x_{11}, \ldots, x_{m1}\} \vee \{x_{12}, \ldots, x_{m2}\}$ is a expended training set including labeled examples x_i and examples from every pairwise constraints $\{x_{i1}, x_{i2}\}$.

In the following, denote by $\mathbf{K}$ the $m' \times m'$ Gram matrix. Moreover, denote by $\mathbf{K}_l$ an $m \times m'$ matrix containing top m rows of $\mathbf{K}$ corresponding to x_i, i.e., $\mathbf{K}_l = [\mathbf{K}(x_i, \bar{x_j})]_{m\times m'}$. Similarly, denote by $\mathbf{K}_{l1}$ and $\mathbf{K}_{l2}$ the $n \times m'$ matrices containing n rows of $\mathbf{K}$ corresponding to x_{i1} and x_{i2} respectively. We derive the kernelized representation of logistic regression loss function by substituting Eq(8) into Eq(6),

$$R(\alpha) = \vec{1}^T \log(1 + e^{-\mathbf{K}_p\alpha}) + \mu\vec{1}^T \log(1 + e^{\mathbf{K}'_p\alpha}) + \mu\vec{1}^T \log(1 + e^{-\mathbf{K}'_p\alpha}) + \lambda\alpha\mathbf{K}\alpha, \quad (9)$$

where $\alpha = \{\alpha_1 \ldots \alpha_{m+2n}\}$, the regressor matrix $\mathbf{K}_p = diag(y_1 \ldots y_m)\mathbf{K}_l$ and the pairwise regressor matrix $\mathbf{K}'_p = \mathbf{K}_{l1} - diag(y'_1 \ldots y'_n)\mathbf{K}_{l2}$.

To find the minimizer α, we derive the parameter estimation method using the Newton-Raphson method to iteratively solve the equation. Since the optimization function is convex, Newton method can guarantee the finding of the global optimum. The gradient and Hessian are as follows,

$$\frac{\partial R(\alpha)}{\partial\alpha} = \mathbf{K}_p^T\mathbf{p} + \mu(\mathbf{K}_p'^T\mathbf{p} - \mathbf{K}_p'^T(1-\mathbf{p})) + \lambda\mathbf{K}^T\alpha, \quad (10)$$

$$\frac{\partial^2 R(\alpha)}{\partial\alpha^2} = \mathbf{K}_p^T\mathbf{W}\mathbf{K}_p + 2\mu\mathbf{K}_p'^T\mathbf{W}'\mathbf{K}_p + \lambda\mathbf{K}^T, \quad (11)$$

where $p(x), p'(x)$ denote the logistic model

$$\mathbf{p}(x) = \frac{e^{\mathbf{K}_p\alpha}}{1+e^{\mathbf{K}_p\alpha}}, \mathbf{p}'(x) = \frac{e^{\mathbf{K}'_p\alpha}}{1+e^{\mathbf{K}'_p\alpha}},$$

and $\mathbf{W}, \mathbf{W}'$ denote the corresponding weighted matrices $diag(\mathbf{p}(x_i)(1-\mathbf{p}(x_i)))$ and $diag(\mathbf{p}'(x_i)(1-\mathbf{p}'(x_i)))$.

It can be shown that the Newton updates are $\alpha \mapsto \alpha - (\frac{\partial^2 R(\alpha)}{\partial\alpha^2})^{-1}\frac{\partial R(\alpha)}{\partial\alpha}$. In practice, we solve this optimization problem with a subspace trust region method based on the interior-reflective Newton method described in [10]. In the rest of this paper, we will call this learning algorithm pairwise kernel logistic regression (PKLR).

2.3 An Illustrative Example

To show the advantages of incorporating pairwise constraints into discriminative learning, we prepared a synthetic spiral dataset shown in figure 3(a) which is nonlinearly separable. There are a total of 201 positive examples and 199 negative examples. 40 training examples are randomly sampled from each class. Additional 4 pairs of positive constraints are also provided on the dataset. With only the labeled data, the conventional kernel logistic regression(KLR) misclassifies the tails of two spirals due to insufficient labeled data (figure 3(b)). The additional positive constraints might be useful to correct the bias. However, applying the RCA algorithm [4] with these constraints only leads to slightly performance improvement shown in figure 3(c), since the true distance metric cannot be simply modeled by a Mahalanobis distance. In contrast, the PKLR algorithm learns a much better boundary shown in figure 3(d) by using pairwise constraints to model the decision boundary directly. In this example, we intentionally only generate the positive constraints to provide a relatively fair comparison with the RCA algorithm. In fact, negative constraints can naturally be applied in the PKLR framework.

3. Extension to Multi-class Classification

In the following discussion we extend our learning framework to multi-class classification. As a first step, it is worthwhile to consider how to present pairwise constraints [1] in the context of a one-against-all classifier, where it means that the negative class is less-defined anything else. Positive constraints still hold because if data pairs are considered the same object they must belong to the same class. However, negative constraints, which means two examples are not the same object, can no longer be interpreted as that two examples are in different classes because it might be the case

[1] Note that in multi-class object classification, a pairwise constraint indicates whether a pair of examples are the same object or not, instead of whether they belong to the same class in a one-against-all classifier

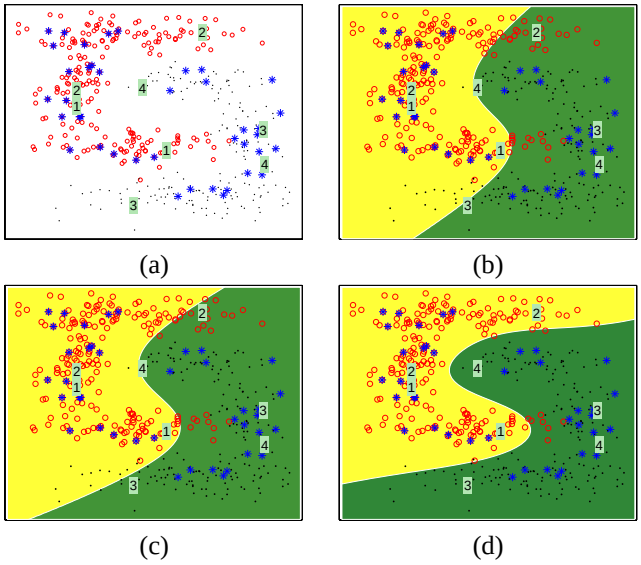


Figure 3: An illustration of pairwise kernel logistic regression applied to synthetic data (a) The synthetic dataset. "$\circ$" and "$\cdot$" denotes positive and negative examples, "$*$" denotes training data and each pair of framed numbers denote positive constraints. (b) Decision boundary of KLR. (c) Decision boundary of KLR in the metric space learned by RCA. (d) Decision boundary of PKLR

they both belong to the negative class. Therefore for negative constraints, we can only penalize the cases where they are both labeled as positive. Thus, the modified loss function can be defined as,

$$Reg'(f) = \sum_i \widetilde{L}(y_i f_i) + \mu \sum_{y'_i=-1} \widetilde{L}(-f_{i1} - f_{i2})$$
$$+\mu \sum_{y'_i=1} (\widetilde{L}(f_{i1} - f_{i2}) + \widetilde{L}(f_{i1} - f_{i2})) + \lambda\Omega(\|f\|_{\mathcal{H}}), \quad (12)$$

where f_i denotes $f(x_i)$. One-against-all classifiers allow the learning algorithm to handle new types of objects in the test set by classifying every unseen objects into the negative class. This is important especially when the number of the training examples is small.

With the aid of this one-against-all representation, we can simply extend our algorithm to multi-class classification with some output coding schemes. We choose a loss-based output coding schemes to construct a multi-class classifier using multiple binary classifiers [11],

$$\hat{y} = arg\min_r \sum_{s=1}^{S} L_M\left(m_{rs} f_s(x)\right).$$

where S is the number of binary classification problems, s is the their indices, r is the class index, m_{rs} is the elements of coding matrix and $f_s(x)$ are the prediction for x using classifier s. The loss function L_M we choose is the same as $\widetilde{L}(x)$, i.e. $L_M(x) = \log(1 + e^{-x})$. The one-against-all coding matrix is adopted in our experiments. Note that if only positive constraints is available, we can also use the other coding schemes as long as there are no zero entries in the coding matricies, such as ECOC coding schemes.

4. Experiments

In the experiments that follow, we applied the PKLR algorithm to the task of classifying people identities with two real-world surveillance video datasets.

4.1. Data Collections and Preprocessing

To test the performance of the PKLR algorithm, we collected two different datasets from a geriatric nursing home surveillance video. One of the datasets was extracted from a 6 hour long, single day and single view video. The other dataset was extracted from video across 6 consecutive days from the same camera view. Both collections were sampled at a resolution of 320×240 and a rate of 30 frames per second. The moving sequences of subjects were automatically extracted using a background subtraction tracker. The silhouette images, each of which corresponds to the extracted silhouette of a moving subject, are sampled from the tracking sequence every half second. In this experiment, we mainly experimented on images that did not have any foreground segments containing two or more people. Finally, we obtain the single day dataset with 63 tracking sequences or 363 silhouette images for 6 subjects, and the multiple day dataset with 156 tracking sequences or 1118 silhouette images for 5 subjects.

Because of the relative robustness of color histograms to variations of target appearance, we represent the images using a histogram of HSV color spaces in the following experiments, where each color channel has a fixed number of 32 bins. Thus we have a total of 96 one-dimensional features in the color histogram. Some examples of these two datasets are depicted in figure 4. From these examples, it can be seen that the silhouette images are collected from various lighting environments and the subjects walked in arbitrary directions. For each subject, the color representation is relatively stable in the single day dataset, but it is much more diverse in the multiple day dataset, which makes learning harder.

4.2. Selecting Informative Pairwise Constrains from Video

As mentioned in section 1, there are several types of pairwise constraints that can be extracted from a video stream. In this paper, we pay particular attention to two types of pairwise constraints:

Figure 4: Examples of images from the datasets collected from a geriatric nursing home. (a) Examples of 6 subjects in the single day dataset. Each column refers to a different subject, (b) Examples of 5 subjects in the multiple day dataset.

Temporal Constraints This type of constraints is obtained by knowing the temporal relation in video sequences. For example, a sequence of extracted regions generated from tracking a single moving object can be assumed to indicate a single person. On the other hand, two regions extracted simultaneously from a camera cannot be the same person.

Active Constraints In analogy to active learning paradigms, this type of constraints is obtained from users' feedback. Typically, the system gives users the most ambiguous pair of examples and users provide the constraint label as feedback.

However, for the video data there are always too many pairwise constraints to incorporate. To address this, we would like to select the most informative pairwise constraints before applying our learning algorithm. One important observation is that surveillance video data generally arrive in the form of image tracking sequences. If the constraint between every image pair of tracking sequences G_1 and G_2 has to be modeled, the pairwise loss function in Eq(4) will be expanded to a sum of $|G_1||G_2|$ terms,

$$L'(y', f(G_1), f(G_2)) = \sum_{i=1}^{|G_1|} \sum_{j=1}^{|G_2|} \widetilde{L}(f(x_i) - y' f(x_j))$$

for every $x_i \in G_1$ and $x_j \in G_2$[2]. In the case where either $|G_1|$ or $|G_2|$ is large the computational complexity will be very large. However, it is reasonable to assume that the images in a single sequence are similar to each other and thus the pairwise constraints $(x_i, x_j), x_i \in G_1, x_j \in G_2$ are likely to be redundant. Based on this assumption, we aggregate all of the sequence constraints using the centroids μ_i of every sequence images as an approximation. Therefore, we have the following pairwise loss function: when $G_1 = G_2 = G$, $L'(y', f(G_1), f(G_2)) = \sum_{i=1}^{|G|} \widetilde{L}(f(x_i) - f(\mu))$, or when $G_1 \neq G_2$, $L'(y', f(G_1), f(G_2)) = \widetilde{L}(f(\mu_2) - y' f(\mu_1))$.

Another observation can help to further reduce the number of pairwise constraints, i.e., it is not necessary to incorporate the pairwise constraints for which the KLR algorithm already provide correct predictions. But this criterion is not applicable in practice since true constraints are not known for unlabeled sequence pairs. As an alternative, we first choose the most ambiguous sequences based on the prediction ambiguity of KLR, and then construct the corresponding pairwise constraints. Since our following experiments are dealing with multi-class classification, we adopt a selection strategy called best-worst case model proposed in [2], of which the rationale is to choose the most ambiguous sequences by maximizing the expected loss for the predicted label,

$$arg \max_{x} \min_{r \in \mathcal{Y}} \sum_{s=1}^{S} L_M(m_{rs} f_s(x)). \tag{13}$$

Figure 5 summarizes the learning process with the selection strategies for pairwise constraints. A kernel logistic regression algorithm is first applied with only the labeled data. The top K ambiguous sequences $\{G_1, ..., G_K\}$ are selected based on Eq(13). For each sequence G_i, we add a temporal constraint $(G_i, G_i, 1)$ into constraint set. For any pairs of sequences that overlap, a negative constraint $(G_i, G_j, -1)$ will be constructed. Moreover, the nearest training sequence to g_j in terms of kernel distance is coupled with G_i to form a active constraint (G_i, G_j, y_{ij}), which pairwise labeling is requested from users. Finally, the PKLR algorithm is trained with both existing labeled data and additional pairwise constraints.

4.3. Performance Evaluation

Our experiments are carried out in the following way. Each dataset is first split into two disjoint sets based on temporal

[2]Note that G_1 and G_2 can be the same sequence G, which refers to modeling the self-similarity of sequence G.

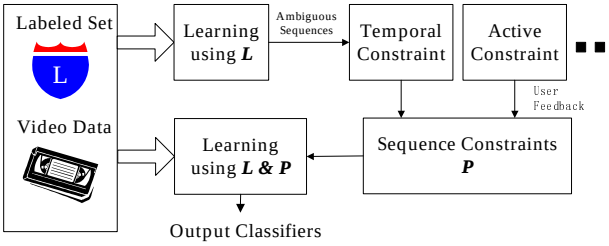


Figure 5: The flowchart of learning process. See text for details.

order. Training images are randomly drawn from the first set, which contains 50% of the first set's sequences. The rest images are used as test images. For every specific parameter setting, we increase the number of sequence constraints from 0 to N until the classification performance is relatively stable. N was chosen to be 20 in the single day dataset and 40 in the multiple day dataset. In terms of active constraints, we simulated the human labeling process using true pairwise constraints without actually asking a human to label in each iteration. For each experiment, the training set is repeatedly re-sampled 10 times to provide a more stable estimation of performance.

For evaluation, the prediction error rate is reported. The baseline performance uses the KLR classifiers with majority voting scheme, i.e. each image is predicted independently and for each sequence the majority label is predicted as true labels. We used the RBF kernel $K(x_i, x_j) = e^{-\rho \|x_i - x_j\|^2}$ with $\rho = 0.08$ in all of our experiments, which was chosen by maximizing the accuracy with cross-validation in the training set. Also, we empirically set the regularization parameters λ to be 0.001, and pairwise coefficient μ to be 1.

The first series of experiments compare the effectiveness of the PKLR algorithm using different types of pairwise constraints together with the baseline classifier shown in figure 6(a) and 6(c). Three different curves are plotted, indicating the performance of the PKLR algorithm using temporal constraints, using active constraints and using both of them. For both datasets we observed that the error rate can be considerably reduced even with a small number of constraints. Learning with temporal constraints is effective in the single day dataset but unable to get improvement in the multiple day dataset. This is partially due to the diverse color representation in the multi-day data. It degrades the effectiveness of temporal constraints which cannot capture long term relations between examples. However, active constraints, if available from users, can be more effective to reduce the error in both datasets. Moreover, the combination of both constraints produced a higher performance. For the first dataset, it reduces error rate from 20% down to 4% with 20 pairs of both type of constraints. For the second dataset, it again reduces error rate from 22% down to 8% with 40 pairs of both type of constraints.

In figure 6(b) and 6(d), we also compare the performance of the PKLR algorithm with the RCA algorithm using the same amounts of pairwise constraints. We use the RCA algorithm to learn a better distance metric before applying the KLR for prediction. An identity matrix ϵI is added to the inner chunklet covariance matrix to make it invertable. Because RCA can only take the positive constraints as input, another curve is depicted for PKLR algorithm with the presence of only positive constraints. A combination of temporal and active constraints is applied in all three experiments. The results show that our algorithm achieves superior performance to the RCA algorithm even without negative constraints. On the other hand, the experimental results also demonstrate the usefulness of incorporating negative constraints.

5. Conclusion

We have presented a discriminative classification framework which can learn the decision boundary with labeled data as well as additional pairwise constraints. The experiments with two surveillance video datasets demonstrated the proposed approach could achieve considerable improved performance with pairwise constraints, compared to the baseline classifier which uses labeled data alone and majority voting scheme. The proposed approach also outperforms a metric learning algorithm using pairwise constraints called RCA algorithm when using the same number of pairwise constraints.

Future work includes incorporating different types of noisy multi-modal pairwise constraints, such as face recognition and speaker identification. It would be interesting to study how these different types of pairwise constraints can improve the performance of a discriminative classifier. We would also like to point out that although our learning framework and previous work on learning distance metric exploit the pairwise constraints in a different way, they are somehow complementary. It may be possible to apply the proposed learning framework in a new distance metric learned from the other algorithms, which would be explored further.

References

[1] F. Li, R. Fergus, and P. Perona, "A bayesian approach to unsupervised one-shot learning of object categories," in *Proceedings of the Intl. Conf. on Computer Vision*, Oct 2003.

[2] R. Yan, J. Yang, and A. G. Hauptmann, "Automatically labeling data using multi-class active learning,"

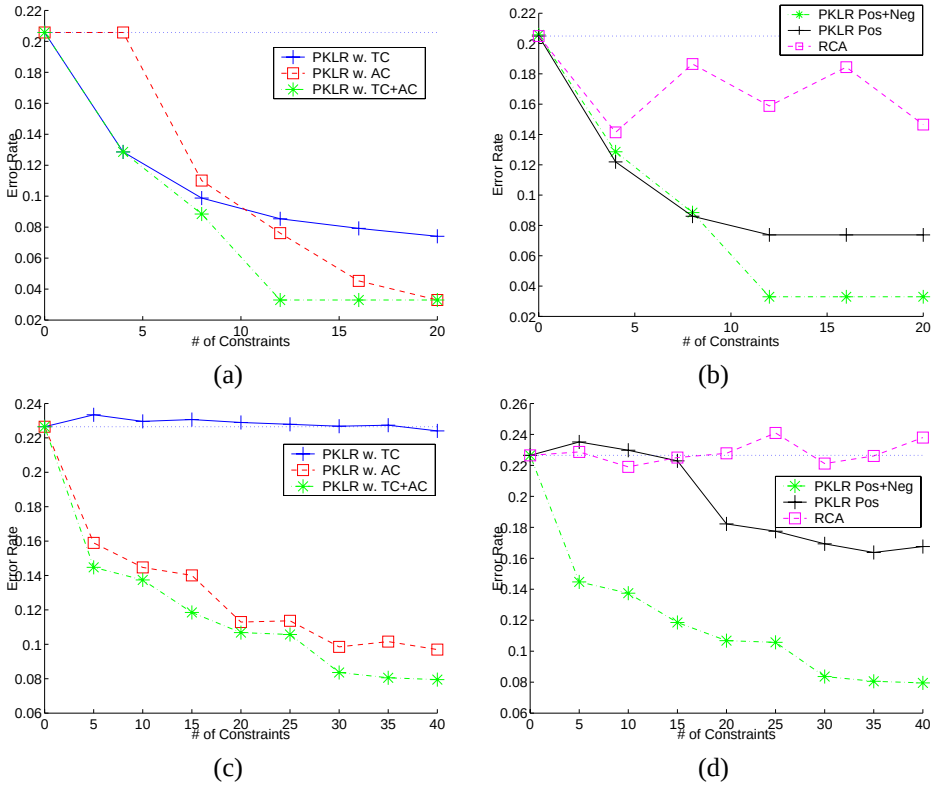


Figure 6: Summary of the experimental results (a) The error rate of PKLR against number of constraints with different constraint types in the single day dataset. The number of constraints is growing from 0 to 20 at a step 4. Each result is obtained by 10 repeated runs with different randomly drawn training and testing images. We compared three cases: using temporal constraints only, active constraints only and both types of constraints. (b) Comparison of PKLR with all constraints, PKLR with positive constraints and KLR with RCA algorithms in the single day dataset. (c), (d) are similar to (a), (b) except reported in the multiple day dataset. The number of constraints is growing from 0 to 40 at a step 5.

in *Proceedings of the Intl. Conf. on Computer Vision*, 2003, pp. 516–523.

[3] S. X. Yu and J. Shi, "Grouping with directed relationships," *Lecture Notes in Computer Science*, vol. 2134, pp. 283–291, 2001.

[4] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall, "Enhancing image and video retrieval: Learning via equivalence constraints," in *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*, Madison, WI, June 2003.

[5] J. T. Kwok and I. W. Tsang, "Learning with idealized kernel," in *Proceedings of the 20th Intl. Conf. on Machine Learning*, Washington, DC, Aug 2003.

[6] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russel, "Distance metric learning with applications to clustering with side information," in *Advances in Neural Information Processing Systems*, 2002.

[7] J. Zhu and T. Hastie, "Kernel logistic regression and the import vector machine," in *Advances in Neural Information Processing Systems*, 2001.

[8] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning. Springer Series in Statistics*, Springer Verlag, Basel, 2001.

[9] G. Kimeldorf and G. Wahba, "Some results on tchebycheffian spline functions," *J. Math. Anal. Applic.*, vol. 33, pp. 82–95, 1971.

[10] T.F. Coleman and Y. Li, "An interior, trust region approach for nonlinear minimization subject to bounds," *SIAM Journal on Optimization*, vol. 6, pp. 418–445, 1996.

[11] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," in *Proc. 17th Intl. Conf. on Machine Learning*, 2000, pp. 9–16.