A Discriminative Learning Framework with Pairwise Constraints for Video Object Classification

Rong Yan
Carnegie Mellon University

Jian Zhang
Carnegie Mellon University

Jie Yang
Carnegie Mellon University

Alexander Hauptmann
Carnegie Mellon University

Follow this and additional works at: http://repository.cmu.edu/compsci
A Discriminative Learning Framework with Pairwise Constraints for Video Object Classification

Rong Yan, Student Member, IEEE, Jian Zhang, Jie Yang, and Alexander G. Hauptmann

Abstract—To deal with the problem of insufficient labeled data in video object classification, one solution is to utilize additional pairwise constraints that indicate the relationship between two examples, i.e., whether these examples belong to the same class or not. In this paper, we propose a discriminative learning approach which can incorporate pairwise constraints into a conventional margin-based learning framework. Different from previous work that usually attempts to learn better distance metrics or estimate the underlying data distribution, the proposed approach can directly model the decision boundary and, thus, require fewer model assumptions. Moreover, the proposed approach can handle both labeled data and pairwise constraints in a unified framework. In this work, we investigate two families of pairwise loss functions, namely, convex and nonconvex pairwise loss functions, and then derive three pairwise learning algorithms by plugging in the hinge loss and the logistic loss functions. The proposed learning algorithms were evaluated using a person identification task on two surveillance video data sets. The experiments demonstrated that the proposed pairwise learning algorithms considerably outperform the baseline classifiers using only labeled data and two other pairwise learning algorithms with the same amount of pairwise constraints.

Index Terms—Video object classification, pairwise constraints, discriminative learning, margin-based learning.

1 INTRODUCTION

Learning with insufficient training data in classifying or recognizing objects/people [1], [2], [3] has recently become an interesting topic [4], [5]. One solution for this problem is to integrate new knowledge sources that are complementary to the insufficient training data. In this paper, we are particularly interested in how to incorporate additional pairwise constraints to improve classification performance for video objects. To be more specific, a pairwise constraint between two examples describes whether they belong to the same class or not, which can provide information for the relationship between the labels rather than labels themselves.

The inherent characteristics of video streams, that is, the sequential continuity and multivalidities, allow us to pose different types of constraints to boost the learning performance. Moreover, these constraints can often be obtained automatically or only with little human effort. Fig. 1 illustrates several examples of pairwise constraints in a scenario of classifying people’s identities from surveillance video. First, pairwise 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 whereas two regions appearing simultaneously in a camera cannot be labeled as the same. Second, we can extract constraints from various modalities such as visual (face) [6] and auditory (voice) cues [7]. For example, if we want to automatically identify a person’s face from video sequences, conventionally we need to learn from many training examples of the same person with different head poses and under different lighting conditions. However, 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 way 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. Typically, the system can select and show some pairs of video sequences to a human annotator who can judge whether these examples depict the same subjects or not. Unlike the general relevance feedback process which forces users to annotate with the exact labels, asking feedback in the form of pairwise constraints does not require users to have prior knowledge or experience with the data set and, in some degree, it helps to protect the privacy of human subjects in the video.

In recent years, researchers have realized the usefulness of incorporating pairwise constraints into different kinds of learning algorithms. As discussed in the next section, a large amount of previous work managed to use pairwise constraints to achieve reasonable performance improvement in various tasks such as clustering [8], [9], [10], [11] and distance metrics learning (e.g., Mahalanobis, cosine distance, and Bregman divergence) [12], [13], [14], [15]. However, relatively less attention has been placed on using additional pairwise...
constraints to support the classification (supervised learning) task. In this case, the most general method to leverage pairwise constraints is to learn a better distance metric before applying the supervised learning algorithm. But, for classification, it is more natural to directly model the decision boundaries as has been done in discriminative classifiers, because the decision boundaries might be simpler to estimate even when true underlying distance metrics are either complex or against the model assumption. Moreover, since extant work usually focuses on the unsupervised learning problem, it does not need to provide a principled way to handle the labeled data. But, in our case, we have to consider incorporating the labeled data into the learning framework since they are the most useful information sources available. It is infeasible to convert every pair of labeled data into pairwise constraints because this usually leads to a prohibitive computation with an unreasonably large number of constraints.

In this work, we propose a regularized discriminative learning framework which naturally incorporates pairwise constraints into a conventional margin-based learning algorithm. The proposed framework is able to use additional pairwise constraints together with labeled data to model the decision boundary directly, instead of resorting to estimating an underlying distance metric which could be much more complex. With this approach, we investigate two families of pairwise loss functions, i.e., convex and nonconvex pairwise loss functions under the proposed learning framework. Analogous to kernel logistic regression (KLR) [16] and support vector machines (SVMs) [17], we derive three pairwise learning algorithms by plugging in the hinge loss and the logistic loss functions into this framework. These algorithms are evaluated in the context of classifying people's identities from surveillance video.

2 RELATED WORK

The classification of visual objects is a perceptual and cognitive task which is fundamental to human vision. Despite the fact that these objects may vary somewhat in shape, color, texture, etc., a human can detect and recognize a multitude of familiar and novel objects through vision without any effort. However, it is a very challenging problem for machines. Object classification has been an active research area in computer vision community for last two decades, though problems of interest have been changing over time, from automatic target recognition (ATR) [18], [19], Optical Character Recognition (OCR) [20], to face detection and recognition [21]. Visual object recognition has made great progress in recent years because of advances in learning theories, which is evident in several recent papers [22], [23], [24], [4], [1], [2].

Along another research direction, efforts have been made to help both supervised and unsupervised learning with pairwise constraints [8], [9], [12], [25], [10], [26], [13], [14], [27]. In the context of graph partitioning, Yu and Shi [8] has successfully integrated pairwise constraints into a constrained grouping framework, leading to improved segmentation results. Wagstaff et al. [9] introduce pairwise constraints into the k-means clustering algorithm for unsupervised learning problems. In more closely related work proposed by Xing et al. [14], a distance metric learning method is proposed to incorporate pairwise information and solved by convex optimization. However, the method contains an iterative procedure with projection and eigenvector decomposition which is computationally expensive and sensitive to parameter tuning. By comparison, relevant component analysis (RCA) [12] is a simpler and more efficient approach for learning a full Mahalanobis metric. A whitening transformation of the covariance matrix of all the centerpoints in the chunklets is computed as a Mahalanobis distance. However, only positive constraints can be utilized in this algorithm. In [25], Shental et al. propose a constrained Gaussian mixture model which incorporates both positive and negative pairwise constraints into a GMM model using the expectation-maximization (EM) algorithm. Basu et al. [10] studied a new approach for semisupervised clustering by adding additional penalty terms into the objective function. They also proposed an approach to actively select the most informative constraints rather than selecting them at random. In [15], they also used pairwise constraints to learn more

Fig. 1. Examples of different kinds of pairwise constraints. (a) Temporal constraints from a single tracked sequence. (b) Temporal constraints of different regions extracted at the same time. (c) Constraints provided by comparing faces. (d) Constraints provided by user feedback.
advanced metrics such as parameterized Bregman distances or directional distances. Kumar and Hebert [26] presented a discriminative learning framework for the classification of the image regions by incorporating interactions from neighborhood nodes together with the observed examples. Pairwise constraints have also been found useful in the context of kernel learning. Kwok and Tsang [13] formulated the kernel adaptation problem as a distance metric learning problem that searches for a suitable linear transform in the kernel-induced feature space, even if it is of infinite dimensionality.

Learning with pairwise constraints is also related to the semisupervised learning problem, which attempts to leverage a large number of unlabeled data to boost the classifier built from a small number of labeled data. Work by Nigam et al. [28] handled the unlabeled data by using a combination of the EM algorithm [29] and a naive Bayes classifier to augment text classifiers and demonstrated that unlabeled data can be used to improve the accuracy of text classification. Cotraining [30] is one of the most well-known multiview semisupervised learning algorithms. The idea of cotraining is to incrementally update the classifiers of multiple views so that a large number of unlabeled data to boost the classifier can utilize additional information about the relationship among the labeled and unlabeled data as the vertices in the weighted graph, where the edge weights encode the similarity between instances. For learning the part-based appearance models, Xie and Peréz [32] extended the GMM model to the semisupervised case where most of positive examples are corrupted with clutter but a small fraction are uncorrupted. Compared with these semisupervised learning algorithms, the algorithms leveraging pairwise constraints can utilize additional information about the relationship between pairs of examples other than the unlabeled data itself.

3 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. A learning algorithm typically takes a set of training examples \( \{(x_1, y_1), \ldots, (x_m, y_m)\} \) as input, where \( y_i \in \mathcal{Y} \) is the label assigned to the example \( x_i \in \mathcal{X} \). In addition to the data with explicit labels, there is another set of pairwise constraints \( \{(x_{i1}, x_{i2}, z_1), \ldots, (x_{i1}, x_{i2}, z_n)\} \) constructed from both labeled and unlabeled data, where \( z_i \in \{-1, 1\} \) 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 mean the example pair \( (x_{i1}, x_{i2}) \) belongs to the same class and \( (x_{i1}, x_{i2}, -1) \) the negative constraints which mean the pair \( (x_{i1}, x_{i2}) \) belongs to different classes.

3.1 Regularized Loss Function with Pairwise Information

Recently, researchers have found that many supervised learning algorithms can be generalized into a learning framework which aims at minimizing the regularized empirical risk [33],

\[
\min_f R_{\text{reg}}(f) = \frac{1}{m} \sum_{i=1}^{m} \tilde{L}(y_i, f(x_i)) + \lambda \Omega(\|f\|_{\mathcal{H}}), \tag{1}
\]

where \( \tilde{L} \) 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 \( K \), \( \| \cdot \|_{\mathcal{H}} \) is the corresponding norm, and \( \lambda \) is the regularization constant. The empirical loss function \( \tilde{L}(y_i, f(x_i)) \) is usually set to the loss of a function of “margin” \( y f(x) \) [33], i.e., \( \tilde{L}(y_i, f(x_i)) = L(y_i f(x_i)) \). With different choices of loss functions and regularization terms, we can derive a large family of well-studied algorithms from (1). For example, the support vector machines (SVMs) can be viewed as a binary margin-based learning algorithm with loss function \( L(z) = \max(1 - z, 0) \) and regularization factor \( \|w\|^2 \). To illustrate, Fig. 2a shows a comparison of four different loss functions against the margin \( y f(x) \), including misclassification loss \( L(\text{sgn}(f) \neq y) \), exponential loss \( \exp(-yf) \), hinge loss \((1 - yf)_+ \), and logistic loss \( \log(1 + \exp(-yf)) \).

Under this learning framework, pairwise constraints can be introduced as another set of empirical loss functions in an attempt to penalize the violation of the given constraints,

\[
\mathcal{O}(f) = \frac{1}{m} \sum_{i=1}^{m} L(y_i, f(x_i)) + \mu \sum_{i=1}^{m} L'(f(x_{i1}), z_i f(x_{i2})) + \lambda \Omega(\|f\|_{\mathcal{H}}), \tag{2}
\]

where we call \( \mu \) pairwise factors and \( L'(f(x_{i1}), z_i f(x_{i2})) \) pairwise loss functions. In the rest of this paper, we simplify the notation of \( f(x_i) \) to \( f_i \) and, thus, \( L'(f(x_{i1}), z_i f(x_{i2})) \) can be written as \( L'(f_{i1}, z f_{i2}) \). Equation (2) enjoys the nice property that when the number of pairwise constraints \( n \) is zero, it trivially degrades to a margin-based learning problem with only the labeled data. To complete the definition of the learning framework in (2), we still need to determine a family of appropriate pairwise loss functions for the pair of examples. Although there are lots of ways to design the pairwise loss functions, we want to seek for a family of loss functions that satisfies the following properties,

1. \( L' \) is commutable, i.e., \( L' \) have the same value when \( f_{i1} \) and \( f_{i2} \) exchange their positions, or equally \( L'(f_{i1}, z f_{i2}) = L'(f_{i2}, z f_{i1}) \), because the constraints would not change if the examples exchange their positions.

2. \( L' \) is even, i.e., \( L' \) have the same value when \( f_{i1} \) and \( f_{i2} \) reverse their signs, or equally \( L'(f_{i1}, z f_{i2}) = L'(-f_{i1}, -z f_{i2}) \), because the constraints would not change if the predictions reverse their signs.

3. \( L' \) has correct decision boundaries, i.e., \( L' \geq L'(0, 0) \) when \( f_{i1} \) and \( z f_{i2} \) have different signs but \( L' \leq L'(0, 0) \) when they have the same signs. This property ensures that the goal of minimizing the objective functions \( \mathcal{O}(f) \) could provide predictions consistent with most of the given pairwise constraints.

4. \( L' \) is a convex function for both \( f_{i1} \) and \( z f_{i2} \), i.e., the value \( L' \) at the midpoint of every interval in \([-\infty, \infty]\) does not exceed the average of its value at the end of its interval. This property indicates the existence of a unique global optimum and allows simpler parameter estimation methods.

1. Note that we only consider the nontrivial case that for each inequality there is at least one point where the inequality is strictly satisfied.
Unfortunately, it can be shown that the last two properties conflict with each other. That is to say, no matter how you design pairwise loss functions, it is impossible for them to have correct decision boundaries and be convex at the same time (see Appendix A). As a trade-off, we have to determine which of these two properties is supposed to be satisfied. In the rest of this section, we describe two possible families of pairwise loss functions and discuss their relationships: One is the nonconvex pairwise loss functions which have correct decision boundaries, the other is the convex pairwise loss functions with incorrect decision boundaries.

3.2 Nonconvex Pairwise Loss Functions

In order to provide a family of pairwise loss functions which are commutable, even and also have correct decision boundary, the simplest case is to choose the binary loss function analogous to the misclassification loss,

\[ L_{\text{binary}} = I(sgn(f_{i1}) \neq sgn(z_if_{i2})) \],

which gives a unit penalty for violation of pairwise constraints and no penalties otherwise. Although minimizing this exact misclassification loss may be worthwhile, it is generally intractable to optimize because of its discontinuity. Even worse, it is not able to penalize large errors more heavily.

To provide a continuous family of pairwise loss functions, we introduce the following nonconvex pairwise loss functions,

\[ L'_{\text{nonconv}} = LL(f_{i1}) + LL(z_if_{i2}) - LL(f_{i1} + z_if_{i2}), \]

where \( LL(x) = L'(x) + L'(-x) \) and \( L'(x) \) can be any convex loss function such as the logistic loss and the hinge loss function. To ensure the empirical loss function and pairwise loss function are comparable, we usually choose \( L_0 \) in the same form as \( L \), and, thus, \( LL(x) = L(x) + L(-x) \). Therefore, the primal optimization problem has the following form,

\[
O(f) = \frac{1}{m} \sum_{i=1}^{m} L(y_if_i) + \frac{\mu}{n} \sum_{i=1}^{n} \left( LL(f_{i1}) + LL(z_if_{i2}) - LL(f_{i1} + z_if_{i2}) \right) + \lambda \Omega(\|f\|_{\mathcal{H}}).
\]  

(3)

Appendix B proves that \( L'_{\text{nonconv}} \) is commutable even, and has correct decision boundaries under some general conditions. Two additional advantages make it a preferred choice compared to the binary loss function. First, \( L'_{\text{nonconv}} \) is able to place more penalties on larger errors. Second, its continuity allows efficient optimization approaches to be applied such as the EM algorithm and quadratic programming. However, the function of \( L'_{\text{nonconv}} \) is no longer convex and, thus, it is possible for the optimization algorithm to get trapped in a local optimum.

3.3 Convex Pairwise Loss Functions

This section considers the family of convex pairwise loss functions based on the intuition that the prediction difference of two examples, i.e., \( f_{i1} - z_if_{i2} \), can be a “soft” measure of how
possible the pairwise constraints would be violated. Therefore, we choose loss function \( L' \) to be a monotonic decreasing function of prediction difference \( f_{i1} - z_{i1}f_{i2} \), i.e., \( \hat{L}(f_{i1} - z_{i1}f_{i2}) \), which plays a similar role as the residues \( y - f(x) \) in regression. Meanwhile, the pairwise loss function should be symmetric for any example pair and, therefore, \( \hat{L} \) could be represented as \( \hat{L}(x) = \hat{L}^P(x) + \hat{L}^N(-x) \), where \( \hat{L}^P \) now can be any monotonic decreasing function \( f: \mathcal{X} \rightarrow \mathcal{R} \). By choosing \( \hat{L}^P \) to be empirical loss function \( L \), we obtain the convex pairwise loss function

\[
L_{\text{conv}} = L(f_{i1} - z_{i1}f_{i2}) + L(z_{i1}f_{i2} - f_{i1})
\]

and the corresponding primal optimization problem,

\[
\mathcal{O}(f) = \frac{1}{m} \sum_{i=1}^{m} L(y_i, f_i) + \sum_{i=1}^{n} \left( L(f_{i1} - z_{i1}f_{i2}) + L(z_{i1}f_{i2} - f_{i1}) \right) + \lambda \Omega(||f||_H).
\]

When \( L(x) \) is convex to \( x \) (true for most loss functions), it is not difficult to verify that the pairwise loss function \( L_{\text{conv}} \) is also convex to \( f_{i1} \) and \( f_{i2} \), which allows us to apply standard convex optimization techniques to solve the primal optimization problem. Generally speaking, minimizing a convex pairwise loss function is much more efficient than minimizing a nonconvex pairwise loss function, which comes at the price that \( L_{\text{conv}} \) cannot provide the correct decision boundaries. But, this disadvantage can be largely overcome by the fact that \( L_{\text{conv}} \) actually serves as a convex upper bound of \( L_{\text{nonconv}} \) (see Appendix C). The upper bound is usually tight because \( L_{\text{conv}} \) and \( L_{\text{nonconv}} \) are equal if and only if \( f_{i1} = -z_{i1}f_{i2} \). This property guarantees the global optimum of the corresponding convex objective functions in (4), although which have incorrect decision boundaries, can still provide a reasonable approximation for the optimum of the nonconvex objective functions in (3).

A special case for (4) is to fit a linear decision boundary on the input feature space, i.e., \( f(x) = w^Tx \) and \( ||f||_H = ||w|| \) in the \( L_2 \) space. Substituting \( f(x) = w^Tx \) and \( ||f||_H = ||w|| \) into (4), we have

\[
\mathcal{O}(f) = \frac{1}{m} \sum_{i=1}^{m} L(y_i, w^Tx_i) + \sum_{i=1}^{n} \left( L(w^Tx_{i1} - z_{i1}w^Tx_{i2}) + L(z_{i1}w^Tx_{i2} - w^Tx_{i1}) \right) + \lambda \Omega(||w||).
\]

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

2. This function has another interpretation as follows: For a pairwise constraint \( (x_{i1}, x_{i2}, z_{i}) \), we would like to penalize two cases based on the predictions: 1) \( f_{i1} > 0 \) and \( z_{i}f_{i2} < 0 \) and 2) \( f_{i2} < 0 \) and \( z_{i}f_{i1} > 0 \). If we use \( L(f_{i1} - z_{i}f_{i2}) \) to penalize the first case and \( L(z_{i}f_{i2} - f_{i1}) \) for the second case, we have exactly the pairwise loss function described above.

4 Algorithms

In this section, we substitute two widely applied loss functions, i.e., the logistic loss function and hinge loss function, into both the nonconvex pairwise objective functions in (3) and the convex pairwise objective functions in (4). We derive three different but closely related learning algorithms from the proposed discriminative learning framework, i.e., convex pairwise kernel logistic regression (CPKLR), convex pairwise support vector machines (CP SVM), and nonconvex pairwise kernel logistic regression (NP KLR). In the following, we describe these three algorithms in more details, present their optimization approaches, and conclude with an illustrative example.

4.1 Convex Pairwise Kernel Logistic Regression (CPKLR)

We begin our discussion by considering convex pairwise objective functions in (4), which can be easily solved by convex optimization techniques. In the first algorithm, we adopt the logistic regression loss function \( L(x) = \log(1 + e^{-x}) \) as the empirical loss function, yielding

\[
\mathcal{O}(f) = \frac{1}{m} \sum_{i=1}^{m} \log(1 + e^{y_i f(x_i)}) + \frac{\mu}{n} \sum_{i=1}^{n} \log(1 + e^{f(x_{i1}) - z_{i1}f(x_{i2})}) + \frac{\mu}{n} \sum_{i=1}^{n} \log(1 + e^{-f(x_{i2}) + z_{i1}f(x_{i2})}) + \lambda \Omega(||f||_H).
\]

Fig. 2b depicts the pairwise loss function used in (6). In the following discussions, we present the kernelized representation of the primal problem (6) using the representer theorem [34]. 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 seemingly computationally intensive task can be easily achieved through a positive definite reproducing kernel \( K \) and the well-known “kernel trick.” To begin, let \( C(\cdot) \) represent the empirical loss and \( \Omega(||f||_H) = ||f||^2_H \). Therefore, the primal problem (6) can be rewritten as,

\[
\min_{f \in \mathcal{H}} C((y_i, f(x_i)), (z_{i1}, f(x_{i1}), f(x_{i2}))) + \lambda ||f||^2_H.
\]

The loss function \( C(\cdot) \) is pointwise, which only depends on the value of \( f \) at the data points \( \{f(x_t), f(x_{t1}), f(x_{t2})\} \). Therefore, by the representer theorem, the minimizer \( f(x) \) admits a representation of the form

\[
f(\cdot) = \sum_{i=1}^{m'} a_i K(\cdot, \mathbf{x}_i),
\]

where \( m' = m + 2n \),

\[\mathbf{x}_i \in \{x_1, \ldots, x_m\} \cup \{x_{n1}, \ldots, x_{n1}\} \cup \{x_{12}, \ldots, x_{n2}\}\]

is an expanded training set including labeled examples \( x_t \) and examples from every pairwise constraints \( \{x_{t1}, x_{t2}\} \).

In the following, denote by \( K \) the \( m' \times m' \) Gram matrix. Moreover, denote by \( K_t \) an \( m' \times m' \) matrix containing top \( m \) rows of \( K \) corresponding to \( x_t \), i.e., \( K_t = [K(\mathbf{x}_t, \mathbf{x}_i)]_{i=1}^{m} \). Similarly, denote by \( K_{n1} \) and \( K_{n2} \) the \( n \times m' \) matrices containing \( n \) rows of \( K \) corresponding to \( x_{n1} \) and \( x_{n2} \), respectively. We
derive the kernelized representation of logistic regression loss function by substituting (8) into (6),
\[
R(\alpha) = \frac{1}{m} \sum_{i=1}^{m} \hat{y}_i \log(1 - e^{-K_p \alpha}) + \mu \frac{1}{n} \sum_{i=1}^{n} \log(1 + e^{-K_i \alpha}) + \lambda \alpha \Lambda \alpha,
\]
where \( \alpha = \{a_1, \ldots, a_{m+2n}\} \), the regressor matrix \(K_p = \text{diag}(y_1, \ldots, y_m)K_i\) and the pairwise regressor matrix \(K_p' = K_{11} - \text{diag}(z_1, \ldots, z_n)K_{22}\).

To find the minimizer \( \alpha \), we derive the parameter estimation method using the interior-reflective Newton method described in [35]. Finally, the shifts of \( z_i \) can be derived from the hinge loss function by substituting (8) into (6),

\[
L_{\text{conv}} = \max((1 + f_{i1} - z_if_{21}), (1 - f_{i1} + z_if_{21})) = \max(1 + f_{i1} - z_if_{21}, 1 - f_{i1} + z_if_{21}) = 1 + |f_{i1} - z_if_{21}|.
\]

Obviously, we can show that \( L_{\text{conv}} \) in (13) is still convex in terms of \( z_i \) but more sensitive to the constraint violation. Fig. 2c plots the pairwise loss function used in (13). By substituting this pairwise loss function into the optimization objective function and ignoring the constant terms, we have

\[
O(f) = \frac{1}{m} \sum_{i=1}^{m} (1 - y_if(x_i)) + \mu \frac{1}{n} \sum_{i=1}^{n} |f(x_i) - z_if(x_{i2})| + \lambda \Omega(||f||_{H}).
\]

Let us first consider the family of linear prediction functions where \( f(x) = w^T x \). In this case, by replacing the absolute and the hinge loss functions, we can get the primal optimization form as follows:

\[
\min \frac{1}{m} \sum_{i=1}^{m} \xi_i + \mu \frac{1}{n} \sum_{i=1}^{n} \eta_i + \lambda w^T w
\]

s.t. \( \xi_i \geq 1 - y_i w^T x_i, \xi_i \geq 0, i = 1..m \)
\( \eta_i \geq w^T x_i - z_i w^T x_{i2}, \eta_i \geq -w^T x_i + z_i w^T x_{i2}, i = 1..n. \)

We take the Lagrangian as usual to get the dual form,

\[
L(w, \xi, \eta, \alpha, \beta, \gamma) = \frac{1}{m} \sum_{i=1}^{m} \xi_i + \mu \frac{1}{n} \sum_{i=1}^{n} \eta_i + \lambda w^T w
\]

\[+ \sum_{i=1}^{m} \alpha_i (1 - y_i w^T x_i - \xi_i) - \sum_{i=1}^{m} \beta_i \xi_i
\]

\[+ \sum_{i=1}^{n} \gamma_i^+ (w^T x_{i1} - z_i w^T x_{i2} - \eta_i)
\]

\[+ \sum_{i=1}^{n} \gamma_i^- (-w^T x_{i1} + z_i w^T x_{i2} - \eta_i).
\]

Setting all the derivatives of the primal variables to be zero, we have

\[
\frac{1}{m} = \alpha_i + \beta_i,
\]

\[
\frac{\mu}{n} = \gamma_i^+ + \gamma_i^-,
\]

\[-2\lambda w = -\sum_{i=1}^{m} \alpha_i y_i x_i + \sum_{i=1}^{n} (\gamma_i^+ - \gamma_i^-) x_{i1}
\]

\[+ \sum_{i=1}^{n} (-z_i \gamma_i^+ + z_i \gamma_i^-) x_{i2}
\]

\[= -\sum_{i=1}^{m} \alpha_i y_i x_i + \sum_{i=1}^{n} (\gamma_i^+ - \gamma_i^-) (x_{i1} - x_{i2}).
\]

According to the Karush-Kuhn-Tucker (KKT) dual-complementarity condition, we have \( \alpha_i, \beta_i, \gamma_i, \gamma_i^- > 0 \). By plugging the above equations back into the Lagrange in (16) and denoting \( \gamma_i = \gamma_i^+ - \gamma_i^- \), the dual form can be rewritten as,
\[ \Theta(\alpha, \gamma) = \sum_{i=1}^{m} \alpha_i - \frac{1}{4\lambda} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j) + \frac{1}{4\lambda} \left\{ \sum_{i,j=1}^{n} \gamma_i \gamma_j [K(x_{i1}, x_{j1}) + z_i z_j K(x_{i2}, x_{j2}) - z_i K(x_{i1}, x_{j2})] \right\} \]

subject to the following conditions,

\[
0 \leq \alpha_i, \alpha_j \leq \frac{1}{m}, \quad i, j = 1..m,
\]

\[
-\frac{\mu}{n} \leq \gamma_i, \gamma_j \leq \frac{\mu}{n}, \quad i, j = 1..n,
\]

where the kernel function \( K(x_i, x_j) \) is the inner product of \( x_i \) and \( x_j \), i.e., \( \langle x_i, x_j \rangle \). We can find that the dual objective function only relies on the inner product of the input variables. Therefore, we can place any kind of positive definite kernel functions into (2) and allow it to produce nonlinear predictions even in infinite dimensions. This quadratic programming problem can be solved by modifying the sequential minimal optimization (SMO) algorithm [36]. After the dual objective is optimized, \( w \) can be computed with (19) and the prediction is made by,

\[ P_{\text{pred}}(x) = \frac{1}{2\lambda} \left\{ \sum_{i=1}^{m} \alpha_i y_i K(x_i, x) - \sum_{i=1}^{n} \gamma_i [K(x_{i1}, x) - z_i K(x_{i2}, x)] \right\}. \]

In the rest of this paper, we will call this learning algorithm convex pairwise support vector machines (CPSVM). This algorithm is efficient because its dual form only contains \( m + n \) number of variables.

### 4.3 Nonconvex Pairwise Kernel Logistic Regression (NPKLR)

Until now, we mainly studied the variants of the convex pairwise loss functions and their corresponding pairwise learning algorithms, i.e., the CPKLR and CPSVM algorithms. Both algorithms are computationally efficient and guaranteed to converge to the global optimum because of the convexity of their loss functions. However, their classification performances are likely to be degraded due to the drawback that they cannot provide correct decision boundaries.

In this section, we investigate the third type of the pairwise objective functions which is derived from the family of nonconvex pairwise loss functions \( L_{\text{nonconv}} \) with the logistic loss function \( L(x) = \log(1 + e^{-x}) \). By substituting the logistic loss into (3), our optimization goal becomes minimizing the following objective function

\[ \mathcal{O}(f) = \frac{1}{m} \sum_{i=1}^{m} \log \left( 1 + e^{-y_i f(x_i)} \right) + \lambda \Omega(\|f\|_{\mathcal{H}}) \]

\[ + \frac{\mu}{n} \sum_{i=1}^{n} \left\{ \log \left( 1 + e^{f(x_1)} \right) + \log \left( 1 + e^{f(x_2)} \right) - \log(1 + e^{f(x_1)+z f(x_2)}) \right\} \]

\[ - \log(1 + e^{f(x_1)+z f(x_2)}) \]

In contrast to convex pairwise loss functions, the nonconvex pairwise loss function above can provide correct decision boundaries and hopefully produce more accurate predictions than its convex counterpart. With some further manipulations for the objective function in (21), we can derive an equivalent form as follows:

\[ \mathcal{O}(f) = -\frac{1}{m} \sum_{i=1}^{m} \log \left( \frac{1}{1 + e^{-y_i f(x_i)}} \right) + \lambda \Omega(\|f\|_{\mathcal{H}}) \]

\[ + \mu \sum_{i=1}^{n} \log \left( \frac{1}{1 + e^{-y_i f(x_1)}} \right) \]

\[ + \frac{1}{1 + e^{-y_i f(x_2)}} \]

This formulation naturally provides a Bayesian interpretation for the proposed objective function. Let \( P(y|x) \) denote the conditional probability for an example pair \( (x, y) \), \( P(z|x_1, x_2) \) denotes the conditional probability for a constraint pair \( (x_1, x_2, z) \) and \( y_1, y_2 \) denotes the labels of \( (x_1, x_2) \). The posterior mode of the parameters \( \theta \) can be written as,

\[ \arg \max_{\theta} p(\theta|x, y, z) = \arg \max_{\theta} p(\theta)L(\theta) \]

\[ = \arg \max_{\theta} (\log p(\theta) + \log L(\theta)). \]

The first term \( \log p(\theta) \) is the logarithm of the prior probability for parameters \( \theta \). If we assume the prior probability \( p(\theta) \) to be proportional to the exponential function \( e^{-\mu \lambda \Omega(\|f\|_{\mathcal{H}})} \), we can recover the regularization term in (22). The second term above is the log-likelihood for all examples and constraints where,

\[ \log L(\theta) = \sum_{i=1}^{m} \log P(y_i|x_i; \theta) + \sum_{i=1}^{n} \log P(z_i|x_1, x_2; \theta). \]

The sample space which satisfies the constraint \((x_1, x_2, z)\) can be partitioned into two mutually exclusive events. One is the event of \( y_1 = 1 \) and \( y_2 = z \), and the other is the event of \( y_1 = -1 \) and \( y_2 = -z \). By assuming the prediction of \( x_1 \) and \( x_2 \) are independent to each other, we obtain

\[ P(z_i|x_1, x_2; \theta) = P(y_1 = 1|x_1; \theta)P(y_2 = z|x_2; \theta) + P(y_1 = -1|x_1; \theta)P(y_2 = -z|x_2; \theta). \]

Similar to logistic regression, the conditional probability \( P(y|x) \) can be represented as the Sigmoid function \( 1/(1 + \exp(-gf(x))) \). By substituting (25) and (24) into (23), we can exactly recover to the formulation of the nonconvex pairwise loss function in (22) except the additional weight \( \mu \) in (22) allows more flexibilities in the implementation.

The major difficulty for explicitly minimizing the optimization objectives lies in the log-sum form of the pairwise loss function. Therefore, we apply the expectation-maximization (EM) algorithm [29] to iteratively optimize the objective function. For each constraint pair \((x_1, x_2, z_i)\), we define \( z_1, z_2 \) as the hidden variable where only one of them is 1 and the other is 0. So, \( E(z_1) + E(z_2) = 1 \), where \( E(z) \) is the expectation of \( z \) and here \( E(z) = P(z = 1) \). Let us denote \( A = -\frac{1}{m} \sum_{i=1}^{m} \log P(y_i|x_i; \theta) + \lambda \Omega(\|f\|_{\mathcal{H}}) \) which is the component irrelevant to pairwise constraints. According to the Jensen’s inequality, we have to maximize

\[ \frac{1}{2}\|\theta - \theta^*\|^2 \mathcal{L}(\theta). \]
\[-\mathcal{O}(f) = -A + \frac{\mu}{n} \sum_{i=1}^{n} \log \left( \frac{1}{1 + e^{f(x_i)}} \right) \]
\[+ \frac{1}{1 + e^{f(x_i)}} \frac{1}{1 + e^{-z_i(f(x_i))}} \]
\[\geq -A + \frac{\mu}{n} \sum_{i=1}^{n} \left[ E(z_{i1}) \log \left( \frac{1}{1 + e^{f(x_i)}} \right) \right] \]
\[+ E(z_{i1}) \log E(z_{i1}) \]
\[+ \frac{\mu}{n} \sum_{i=1}^{n} \left[ E(z_{i2}) \log \left( \frac{1}{1 + e^{-z_i(f(x_i))}} \right) \right] \]
\[+ E(z_{i2}) \log E(z_{i2}) \].

Therefore, the EM algorithm can proceed as follows:

- **(E-step).** For each \( i \), set the hidden variables to be
\[E(z_{i1}) = \left( 1 + e^{f(x_i)} \right)^{-1},\]
\[E(z_{i2}) = \left( 1 + e^{-z_i(f(x_i))} \right)^{-1}.\]

- **(M-step).** Maximize the objective functions \(-\mathcal{O}(f)\) given the hidden variables
\[\theta = \arg \max_{\theta} \frac{1}{m} \sum_{i=1}^{m} \log \left( \frac{1}{1 + e^{-z_i(f(x_i))}} \right) - \lambda \Omega(\|f\|_\gamma)\]
\[\geq \frac{1}{m} \sum_{i=1}^{m} \left[ E(z_{i1}) \log \left( \frac{1}{1 + e^{f(x_i)}} \right) \right] \]
\[+ E(z_{i2}) \log \left( \frac{1}{1 + e^{-z_i(f(x_i))}} \right) \]
\[+ E(z_{i2}) \log E(z_{i2}) \].

The M-step can actually be solved as a weighted logistic regression problem. If \( f(x) \) belongs to the family of linear prediction functions the M-step can be solved by any gradient descent methods. The kernel version of the algorithm can be derived by modifying the M-step to be weighted kernel logistic regression using the same technique presented in Section 4.1.

In the rest of this paper, we will call the learning algorithm above nonconvex pairwise kernel logistic regression (NPKLR). The NPKLR algorithm is less efficient than its convex counterpart because it needs to run multiple iterations of kernel logistic regression containing \( m + 4n \) examples.

### 4.4 An Illustrative Example

To show the advantages of incorporating pairwise constraints into the framework of discriminative learning, we prepared a synthetic spiral data set shown in Fig. 3a which is non-linearly separable. There are a total of 201 positive examples and 199 negative examples. Forty training examples are randomly sampled from each class, an additional four pairs of positive constraints are also provided on the data set, as shown in Fig. 3b. We use kernel logistic regression (KLR) as the underlying learning algorithm.

As shown in Fig. 3c, with only the labeled data the conventional KLR algorithm misclassifies tails of two spirals due to insufficient labeled data. The additional positive constraints might be useful to correct the bias. However, applying the RCA algorithm [12] with these constraints only leads to slightly performance improvement shown in Fig. 3d since the true distance metric cannot be simply modeled by a Mahalanobis distance. In contrast, the CPKLR algorithm learns a much better boundary shown in Fig. 3e by using pairwise constraints to model the decision boundary directly. The nonconvex cousin, i.e., the NPKLR algorithm, further provides a slight improvement over the CPKLR algorithm with a better decision boundary, as shown in Fig. 3f.

### 5 Extension to Multiclass Classification

In the following discussions, we extend our learning framework to the multiclass classification. As a first step, it is worthwhile to consider how to present pairwise constraints in the context of a one-against-all classifier, where it means that the positive class is a certain object and the negative class is less-defined anything else. Positive constraints still hold in this case because if data pairs are considered the same object they must belong to the same class. However, negative constraints, which means two examples are different objects, can no longer be interpreted as that two examples are in different classes because it might be the case 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 convex pairwise loss function can be defined as,

\[\mathcal{O}(f) = \sum_{i} L(y_i, f_i) + \mu \sum_{z_{i1} = 1} L(-f_{i1} - f_{i2})\]
\[+ \mu \sum_{z_{i1} = 1} L(-f_{i1} - f_{i2}) + L(f_{i2} - f_{i1}) + \lambda \Omega(\|f\|_\gamma),\]

where \( f_i \) denotes \( f(x_i) \). Similarly the modified nonconvex pairwise loss function can be defined as

\[\mathcal{O}(f) = \sum_{i} L(y_i, f_i) + \mu \sum_{z_{i1} = 1} \left( L(-f_{i1} - f_{i2}) \right)\]
\[+ \mu \sum_{z_{i1} = 1} \left( L(-f_{i1} - f_{i2}) + L(f_{i2} - f_{i1}) \right) + \lambda \Omega(\|f\|_\gamma).\]

One-against-all classifiers allow the learning algorithm to handle new types of objects in the test set by classifying every unseen object into the negative class. This is important because in the testing phase there are always some unseen objects to predict especially when the number of the training examples is small.

Under this one-against-all representation, we can simply extend our algorithm to multiclass classification with some output coding schemes. We choose a loss-based output coding scheme to construct a multiclass classifier using multiple binary classifiers [37],

\[\hat{y} = \arg \min_f \sum_{r=1}^{g} L_M(m_{rf}(x)),\]

3. Note that in multiclass 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 positive/negative class in a one-against-all classifier.
where \( S \) is the number of binary classification problems, \( s \) is 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 \( L(x) \). \( M \) is the one-against-all coding matrix here. Note that, if only positive constraints are available, we can also use the other coding schemes as long as there are no zero entries in the coding matrices, such as ECOC coding schemes.

6 Experiments

In the experiments that follow, we applied the three proposed pairwise learning algorithms to the task of classifying people identities on two data sets from real-world surveillance video. First, we introduce how we collect and preprocess the data of people identities, followed by discussing the strategies to select a limited number of pairwise constraints from the video. Finally, we describe the experimental setting and evaluate the results using various pairwise learning algorithms.

6.1 Data Collections and Preprocessing

To examine the performance of the proposed algorithms, we collected two different data sets from a geriatric nursing home surveillance video. One data set was extracted from a six hour long, single day, and single view video. The other data set was extracted from video across six consecutive days from the same camera view. Both collections were sampled at a resolution of 320 \( \times \) 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
experienced on images that did not have any foreground segments containing two or more people. Finally, we obtain the single day data set with 63 tracking sequences or 363 silhouette images for six subjects, and the multiple day data set with 156 tracking sequences or 1,118 silhouette images for five subjects.

Because of the relative robustness of color histograms to appearance variations, we represent the silhouette images using a histogram of HSV color spaces in all of our experiments, where each color channel has a fixed number of 32 bins. Thus, we have a total of 96 one-dimensional features in the histogram. Sample images from both data sets are depicted in Fig. 4. Taking a closer look at these examples, it can be found 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 data set, but it is much more diverse in the multiple day data set which makes learning more difficult. Note that the data set, but it is much more diverse in the multiple day.

Fig. 4. Examples of images from the data sets collected from a geriatric nursing home. (a) Examples of six subjects in the single day data set. Each column refers to a different subject. (b) Examples of five subjects in the multiple day data set.

6.2 Selecting Informative Pairwise Constraints 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:

- 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 pairs of examples and users provide the label of positive/negative constraints as feedback. However, even only considering two types of constraints, there are always too many pairwise constraints available for the video data. For example, if there are \(10^3\) training images in the data set, all possible pairwise constraints between them is close to \(5 \times 10^9\) which is unaffordable for most of the learning algorithms. To address this, we would like to select the most informative pairwise constraints before applying the proposed learning algorithm. One useful observation to reduce the number of constraints is that surveillance video data generally arrive in the form of image tracking sequences. If we want to model all the constraints between every image pair of tracking sequences \(G_1\) and \(G_2\) for convex pairwise loss functions, (3) will be expanded to a sum of \(|G_1||G_2|\) terms,

\[
L'(f(G_1), zf(G_2)) = \sum_{i=1}^{|G_1|} \sum_{j=1}^{|G_2|} L'(f(x_i), zf(x_j))
\]

for every \(x_i \in G_1\) and \(x_j \in G_2\)\(^4\). In the case when either \(|G_1|\) or \(|G_2|\) is large, the computational effort will be very prohibitive. 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 probably redundant. Given this assumption, we approximate all of the sequence constraints with the centroids \(\mu_i\) which is the mean color histogram of every sequence images. Therefore, we have the following pairwise loss function: when \(G_1 = G_2 = G\),

\[
L'(f(G_1), zf(G_2)) = \sum_{i=1}^{|G|} L(f(x_i), f(\mu))
\]

or when \(G_1 \neq G_2\),

\[
L'(f(G_1), zf(G_2)) = L(f(\mu_1), zf(\mu_1)).
\]

Another observation can help to further reduce the number of pairwise constraints, i.e., it is not necessary to

4. Note that \(G_1\) and \(G_2\) can be the same sequence \(G\), which refers to modeling the self-similarity of sequence \(G\).
incorporate the pairwise constraints for which the learning algorithm without constraints already provide correct predictions. But, this criterion is not directly applicable since true constraints are not known for unlabeled sequence pairs. As an alternative, we decided to choose the most ambiguous sequences in analogy to the typical active learning algorithm and construct the corresponding pairwise constraints based on the predictions of learning algorithms with labeled data only. This is because pairwise constraints between ambiguous sequences can usually offer higher information gain and reduce the volume of the version space faster. Since our experiments are dealing with multiclass classification, we adopt a sample selection strategy called best-worst case model proposed in [5], of which the rationale is to choose the most ambiguous sequences by maximizing the expected loss of the sequence \( G_i \),

\[
G^* = \arg \max_G L(G),
\]  

(29)

where \( L(G) = \max_{x \in G} \min_{s \in S} L(f_s(x)) \) is the loss of classification prediction for the sequence \( G \) and \( f_s(x) \) is the \( s \)th binary classifier for the example \( x \). Fig. 5 summarizes the learning process with the selection strategy for pairwise constraints. The kernel logistic regression or support vector machines are first applied with no constraints. The top \( K \) ambiguous sequences are selected based on (29). For each sequence \( G_i \), we add the related temporal constraints and active constraints into the constraint set. Finally, the pairwise learning algorithm is trained with both existing labeled data and additional pairwise constraints.

This criterion intends to maximize the ambiguities of selected examples and meanwhile make them distinguishable from each other. For each pair of constraints provided by above strategies, we request the pairwise labels \( y_{ij} \) from users and use it to generate an active constraint \((G_i, G_j, y_{ij})\). After all these constraints are available, the learning algorithm can be easily relearned using both the existing labeled data and the additional pairwise constraints.

6.3 Experimental Setting

Our experiments are carried out in the following way. Each data set is first split into two disjoint sets based on temporal order. Training images are randomly drawn from the first set, which contains 50 percent of its video sequences. The rest images are used as test images. For each 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 data set and 40 in the multiple day data set. 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. The MIN sampling strategy is applied unless stated otherwise.

For evaluation, the prediction error on testing data is reported. The baseline performance of the CPKLR and NPKLR algorithms uses KLR with a majority voting scheme, i.e., each image is predicted independently and then the majority label for each sequence is predicted as true labels. Similarly, the baseline performance of the CP SVM algorithm uses SVMs with a majority voting scheme. 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 \( \lambda \) to be 0.01, and pairwise coefficient \( \mu \) to be 1.

We also compared the proposed approaches with the following two learning algorithms utilizing pairwise constraints. The first one is called relevant component analysis (RCA) [12], which is an efficient algorithm for learning a full Mahalanobis metric by linear transformation. In this work, the authors define a chunklet as a subset of points that belong to the same class but the identity of this class is unknown. Given the chunklets, the covariance matrix \( S_{ch} \) of all the center-points in the chunklets is computed. The Mahalanobis distance is generated from the whitening transformation of \( S_{ch} \). In the implementation, we added an identity matrix \( \epsilon I \) to the covariance matrix \( S_{ch} \) in order to avoid the issue of singularity. Since RCA is a metric learning algorithm, it cannot handle supervised learning directly. In the following experiments, we first apply RCA to transform the feature space and then apply the same baseline classifiers as before to predict the testing data. One drawback for this algorithm is that it can only work with positive constraints.

The other approach we compared with is a constrained Gaussian mixture model [25] which incorporates both positive and negative pairwise constraints into a GMM model using EM algorithm. In the following, we call this algorithm pairwise Gaussian mixture model (PGMM), where more details can be found in the work done by Shental et al. [25]. To apply PGMM for classification, we chose two Gaussian mixtures to model the positive data and three mixtures to model the negative data. The number of mixtures is
determined by using cross validation in the training set and picking the best configuration from 1 mixture to five mixtures. Similar to the RCA algorithm, an identity matrix $C_15I$ was added to the covariance matrix for the purpose of regularization. Finally, the posterior probability for a testing example being positive $P(y=1|x)$ can be computed from the data likelihoods $P(x|y=+1)$ and $P(x|y=-1)$.

6.4 Performance Evaluation

The first series of experiments compare the effectiveness of the proposed pairwise learning algorithms using different types of pairwise constraints as well as the baseline classifiers shown in Fig. 6. Three different curves are plotted in each subgraph, indicating the performance of the CPKLR, CPSVM, and NPKLR algorithms. From Fig. 6, we can observe that the classification error can be considerably reduced even with a small number of constraints. By comparing the performance of pairwise learning algorithms in different settings, we find that NPKLR usually outperforms the algorithms using convex loss functions (namely, CPKLR and CPSVM), because the nonconvex pairwise loss functions can provide correct decision boundaries which cannot be done by the convex pairwise loss functions. However, the performance improvement comes at a price of higher computational intensity. Since the computational time of each iteration in solving NPKLR is similar to that of solving CPKLR, the overall computational time of NPKLR is $N_{EM}$ times higher than the time of CPKLR if $N_{EM}$ denotes the number of EM iterations in NPKLR. Moreover, it shows that sometimes NPKLR tends to degrade its performance especially when a large number of constraints are incorporated. This can be explained by the fact that as more constraints introduced, the surface of the nonconvex objective function becomes more “bumpy” and, thus, NPKLR is more likely to get trapped in a local minimum instead of reaching the global optimum. In contrast, both the CPKLR and CPSVM algorithms achieve a relatively smaller performance boost than NPKLR. However, the improvement is usually more stable than NPKLR. Among these two learning algorithms with convex pairwise loss functions, their performances are close to each other but on average the CPKLR algorithm is slightly superior to the CPSVM algorithm.

Along another direction, it is also useful to compare the classification performances across various constraint types. As can be seen, learning with temporal constraints is effective in the single day data set but unable to get any improvement in the multiple day data set. This is partially due to the diverse color representation in the multiday video sequences. It degrades the effectiveness of temporal constraints which cannot capture long term relations between image examples. However, active constraints, if available from users, can be more effective to reduce the error in both data sets. Moreover, the combination of both constraints often produces a higher performance. For the first data set when using NPKLR, it reduces the error rate from 18 percent down to 6 percent with

5. Note that the results reported in this work are slightly different from that reported in the conference version, because we modified the performance measures from the accuracy with regards to silhouette images (used in the conference version) to the accuracy with regards to tracking sequences as a more reasonable criterion.
20 pairs of both types of constraints. For the second data set, it again reduces the error rate from 22 percent down to 12 percent with 40 pairs of both type of constraints.

In Fig. 7, we compare the performance of the CPKLR algorithm with two baseline algorithms as mentioned before, i.e., the RCA algorithm and PGMM using the same amounts of pairwise constraints. In this experiment, we adopted kernel logistic regression (KLR) as the underlying classifier except for PGMM. A combination of temporal and active constraints is applied for each learning algorithm. Because RCA can only take the positive constraints as input, another curve is depicted for PKLR algorithm with the presence of only positive constraints. The results show that our algorithm achieves a superior performance to both the RCA algorithm and the PGMM even without negative constraints. The degrading performance of PGMM suggests that our data might violate the model assumption of the gaussian mixture model. Also, it corroborates the advantage of the proposed discriminative framework which requires fewer assumptions about the underlying distributions. On the other hand, the results also demonstrate the usefulness of incorporating negative constraints.

Fig. 8 examines the sensitivity of the proposed learning algorithms to the change of pairwise factor, i.e., \( \mu \) in (2). We set the pairwise factor to be one of the following values: 0.01, 0.1, 1, 10, and 100. Each learning curve in Fig. 7 represents the accuracy of the CPKLR algorithm using a specific pairwise factor. As we can see from the results, the change across various pairwise factors is not significant, especially when \( \mu \geq 0.01 \) the difference between their accuracies is within 2 percent. We can conclude that the proposed algorithms are fairly insensitive to the choice of different pairwise factor.

Finally, Fig. 9 analyzes the performance of using three different sampling strategies described in Section 6.2, i.e., MIN, MAX, and COM. We can observe that all these sampling strategies can boost the classification performance over the baseline by using additional pairwise constraints. Specifically, the MAX and MIN sampling strategies provide more significant improvement than the COM strategy. This might be related to the fact that MAX/MIN strategies impose constraints between training examples and testing examples, while the COM strategy only consider the coupling between testing examples.

7 CONCLUSION

We have presented a discriminative classification framework which can directly model the decision boundary with labeled data as well as additional pairwise constraints without explicitly estimating the underlying data distribution. Two families of pairwise loss functions, i.e., convex and nonconvex pairwise loss functions, were investigated and three pairwise learning algorithms were derived by plugging in the hinge loss and the logistic loss functions. The
experiments with two surveillance video data sets demonstrated the proposed pairwise learning algorithms could achieve considerable improved performance with pairwise constraints, compared to the baseline classifier which uses labeled data alone and a majority voting scheme. The proposed algorithms also outperformed the RCA algorithm and the Gaussian mixture model with constraints when the same number of pairwise constraints are used. A comparison among the proposed algorithms showed that the algorithms with nonconvex loss functions could achieve a higher classification accuracy but the algorithms with convex loss functions are more efficient and robust.

In this work, we mainly focus on developing new pairwise learning algorithms and leave the exploration of more advanced visual features to future research. Other future work includes incorporating different types of noisy multimodal pairwise constraints, such as constraints from face recognition and speaker identification. It would be interesting to study how these noisy 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 different ways, they can be complementary to each other. It may be possible to apply the proposed learning framework in a new distance metric learned from other algorithms.

**APPENDIX A**

**PROOF OF PROPOSITION 1**

**Proposition 1.** If the pairwise loss functions $L'(f_1, z f_2)$ is convex, it cannot have correct decision boundaries.

**Proof.** Assume to the contrary that there exists a convex loss function $L'(f_1, z f_2)$ which have correct decision boundaries. Based on our definition, if $L'$ has correct decision boundaries, when $f_1$ and $z f_2$ have the same sign we will have at least one point $(f_1, z f_2)$ satisfying $L'(f_1, -z f_2) = L'(-f_1, z f_2) < L'(0, 0)$. According to the definition of convexity, we have

$$L'(0, 0) \leq L'(f_1, -z f_2) + L'(-f_1, z f_2) < L'(0, 0).$$

This leads to a contradiction. This completes the proof. □

**APPENDIX B**

**PROOF OF PROPOSITION 2**

**Proposition 2.** If there is at least one point $(f_1, f_2)$ satisfying $L'_\text{nonconv}(f_1, z f_2) \neq L'_\text{nonconv}(0, 0)$ when $\text{sgn}(f_1) = \text{sgn}(z f_2)$ and another point when $\text{sgn}(f_1) \neq \text{sgn}(z f_2)$, the following pairwise loss function is commutable, even and has correct decision boundary

$$L'_\text{nonconv}(f_1, z f_2) = LL(f_1) + LL(z f_2) - LL(f_1 + z f_2),$$

where $LL(x) = L(x) + L(-x)$ and $L(x)$ can any convex function.

**Proof.** First, we can show that

$$LL(2z) = LL(z) + LL(-z) = L(x) + L(-x) = LL(x)$$

when $z \in \{-1, +1\}$. Therefore, $L'_\text{nonconv}$ is commutable because

$$L'_\text{nonconv}(f_1, z f_2) = LL(f_1) + LL(z f_2) - LL(f_1 + z f_2)$$

$$= LL(z f_1) + LL(f_2) - LL(z f_1 + f_2)$$

$$= L'_\text{nonconv}(f_2, z f_1).$$

Second, $L'_\text{nonconv}$ is even because

$$L'_\text{nonconv}(f_1, z f_2) = LL(f_1) + LL(z f_2) - LL(f_1 + z f_2)$$

$$= LL(f_1) + LL(-z f_2) - LL(-f_1 - z f_2)$$

$$= L'_\text{nonconv}(-f_1, -z f_2).$$

Finally, we need to prove $L'_\text{nonconv}$ can provide correct decision boundaries. Let us first consider the case when $\text{sgn}(f_1) = \text{sgn}(z f_2)$. Without loss of generality, let us assume $f_1 + z f_2 > \{f_1, z f_2\} > 0$. Then, we have the following inequality

$$LL(f_1) + LL(z f_2)$$

$$\leq \frac{f_1 LL(f_1 + z f_2) + z f_2 LL(0)}{f_1 + z f_2} + \frac{z f_2 LL(f_1 + z f_2) + f_1 LL(0)}{f_1 + z f_2}$$

$$= LL(f_1 + z f_2) + LL(0),$$

using the fact that $LL$ is also convex function because the sum operation preserves the convexity. Therefore, we have

Fig. 9. The classification error of the CPKLR algorithm against number of constraints using three different sampling strategies as described in Section 6.2. (a) is reported in the single day data set and (b) is reported in the multiple day data set.
\[
L'_{\text{nonconv}}(f_2, z f_1) = LL(f_1) + LL(z f_2) - LL(f_1 + z f_2) \\
\leq LL(0) = L'_{\text{nonconv}}(0, 0).
\]

According to our condition, there exists a point \((f_1, f_2)\) satisfying \(L'_{\text{nonconv}}(f_2, z f_1) < L'_{\text{nonconv}}(0, 0)\). So, \(L'_{\text{nonconv}}\) can provide correct decision boundaries when \(\text{sgn}(f_1) = \text{sgn}(z f_2)\).

The proof is similar for the case of \(\text{sgn}(f_1) \neq \text{sgn}(z f_2)\). Without loss of generality, let us assume \(f_1 > f_1 + z f_2 > 0 > z f_2\). Then, we have the following inequality,
\[
LL(f_1) + LL(z f_2) = f_1 LL(f_1) - z f_2 LL(z f_2) + \frac{z f_2 LL(f_1) + f_1 LL(z f_2)}{f_1 - z f_2} \\
\geq LL(f_1 + z f_2) + LL(0).
\]

Therefore, we have
\[
L'_{\text{nonconv}}(f_2, z f_1) = LL(f_1) + LL(z f_2) - LL(f_1 + z f_2) \\
\geq LL(0) = L'_{\text{nonconv}}(0, 0).
\]

So, \(L'_{\text{nonconv}}\) can provide correct decision boundaries when \(\text{sgn}(f_1) \neq \text{sgn}(z f_2)\). This completes the proof. □

**APPENDIX C**

**PROOF OF PROPOSITION 3**

**Proposition 3.** \(L'_{\text{conv}}\) is an upper bound of \(L'_{\text{nonconv}}\) when they use the same \(L(x)\). \(L'_{\text{conv}}\) is defined as
\[
L'_{\text{conv}} = LL(f_1 - z f_2) = LL(f_1 - z f_2) + LL(z f_2 - f_1).
\]

**Proof.** According to the convexity of loss function \(LL\), we have
\[
LL(f_1 - z f_2) + LL(f_1 + z f_2) \geq 2LL(f_1),
\]
\[
LL(-f_1 + z f_2) + LL(f_1 + z f_2) \geq 2LL(z f_2),
\]
Summing them together and given \(LL(-f_1 + z f_2) = LL(f_1 - z f_2)\), we can get
\[
L'_{\text{nonconv}}(f_1, z f_2) = LL(f_1) + LL(z f_2) - LL(f_1 + z f_2) \\
\leq L'_{\text{conv}} = LL(f_1 - z f_2).
\]

This completes the proof. □

**ACKNOWLEDGMENTS**

This research was supported in part by the US National Science Foundation (NSF) under Grants No. IIS-0205219, by the US Department of Defense under contract number N0175-03-C-0424 and by the Advanced Research and Development Activity (ARDA) under contract number H98230-04-C-0406. The authors would also like to thank Aharon Bar-Hillel, Tomer Hertz, Noam Shental, Eric Xing, and Stella Yu for their helpful feedback.

**REFERENCES**


Rong Yan received the BE degree in computer science from Tsinghua University, Beijing, in 2001. He is a doctoral candidate at the Language Technologies Institute in Carnegie Mellon University’s School of Computer Science. His research interests include multimedia retrieval, video content analysis, and machine learning. He received the ACM Multimedia Best Paper Runner-Up award in 2004. He is a student member of the IEEE and the IEEE Computer Society.

Jian Zhang received the BS degree in physical oceanography from China Ocean University and the MS degree in computer science from Chinese Academy of Science. He is a doctoral candidate at the Language Technologies Institute in School of Computer Science, Carnegie Mellon University. His research interests include statistical learning, information retrieval, and pattern recognition.

Jie Yang received the PhD degree in electrical engineering from the University of Akron, Ohio, in 1991. He is a senior systems scientist in the Human-Computer Interaction Institute, Carnegie Mellon University. He joined the Interactive Systems Lab in 1994, where he has been leading research efforts to develop visual tracking and recognition system for multimodal human computer interaction. His research interests are multimodal interfaces, computer vision, and pattern recognition.

Alexander G. Hauptmann received the BA and MA degrees in psychology from Johns Hopkins University in 1982, a “Diplom” degree in computer science from the Technische Universität Berlin in 1984, and the PhD degree in computer science from Carnegie Mellon in 1991. He is a senior systems scientist in the Department of Computer Science, Carnegie Mellon University. His research interests are language, speech, and video analysis and retrieval.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.