Visual Recognition using Mappings that Replicate Margins

Lior Wolf and Nathan Manor
The Blavatnik School of Computer Science
Tel-Aviv University

Abstract

We consider the problem of learning to map between two vector spaces given pairs of matching vectors, one from each space. This problem naturally arises in numerous vision problems, for example, when mapping between the images of two cameras, or when the annotations of each image is multidimensional. We focus on the common asymmetric case, where one vector space $\mathcal{X}$ is more informative than the other $\mathcal{Y}$, and find a transformation from $\mathcal{Y}$ to $\mathcal{X}$. We present a new optimization problem that aims to replicate in the transformed $\mathcal{Y}$ the margins that dominate the structure of $\mathcal{X}$. This optimization problem is convex, and efficient algorithms are presented. Links to various existing methods such as CCA and SVM are drawn, and the effectiveness of the method is demonstrated in several visual domains.

1. Introduction

A great deal of attention is devoted to recognition tasks in which the prediction value is a single label. Such problems are straightforward to benchmark, and they fit well within conventional learning techniques. Some attention was given, however, to the type of tasks that require the prediction of a multidimensional outcome. In [13], efforts to produce a textual descriptors of a video are presented, and in [3] images are mapped to tags and vice versa. In [7] one view of a road scene is compared to another, and in [24] biological data is mapped to images.

Typically, in learning from visual data, the images are converted to vectors for reasons of mathematical and algorithmic convenience. Similarly, for example, textural descriptions, collections of tags, and biological data are often transformed into vectors. Multidimensional perceptual problems, involving images on one side and other forms of structured data on the other, are therefore most readily treated by either recovering a mapping from one vector space $\mathcal{X}$ to another vector space $\mathcal{Y}$ (multidimensional regression), or by mapping the two vector spaces $\mathcal{X}, \mathcal{Y}$ onto one common vector space (as in CCA, see Section. 2). Often, these mappings are recovered based on a set of matching training pairs $\{(x_i, y_i)\}$ such that $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$.

In this work we consider a margin-based solution for the asymmetric mapping problem. Maximum margin methods have proven to be effective for classification, regression, and recently also in metric learning [23]. Here, we recover a mapping $T : \mathcal{Y} \rightarrow \mathcal{X}$ that *replicates or mimics* in the space of the transformed points $T\mathcal{Y}$ the margins that exist in $\mathcal{X}$. Note that we do not *maximize* the margins, since this might distort the space $T\mathcal{Y}$.

By definition, margins require the division of the samples into two groups. When computing the map, we consider many such divisions and resulting margins. Many previous attempts to extend vector-to-vector mappings to use discriminative techniques have assumed that a pair of $\{(x_i, y_j)\}$ such that $i \neq j$ (i.e., the points are not paired in the training set), form an example of an incompatible pair. In many applications we found this assumption to be harmful. The reason might be that since the data is distributed unevenly, some of the presumably non-matching pairs are very similar in nature to the matching pairs.

Since non-matching pairs are typically not provided, and since they cannot be deduced from the data, we seek another source of discriminative information. To compute the transformation T we rely on the existence of a group of hyperplanes that separate the samples in the space of $\mathcal{X}$. These hyperplanes are obtained in various ways in accordance with the application at hand. If such hyperplanes are not available, random hyperplanes are used successfully.

2. Previous work

The problem of learning statistical connection between matching vectors from two vector spaces is well studied. The vectors are often combined to form two matrices X and Y with the same order of columns. Once the model linking the two matrices is learned, several tasks can be performed. For example, decide, given two new vectors x_{new} and y_{new} whether they are matching according to the model. A second task is the multidimensional regression problem, where the vector x_{new} is to be predicted given a vector y_{new}. This second task is a harder one in the sense that by solving it, the first task can be solved, but not the other way around.

Multidimensional regression problems are often solved by extending one dimensional problems such as linear ridge regression [10], Support Vector Regression [20, 14], or Kernel Regression [9]. Sometimes such solutions are applied one coordinate at a time, and sometimes the entire output is predicted at once [19].

While our method is an asymmetric method, and therefore belongs to the family of regression methods, we do not try to find a transformation that maps y to x. Instead, we aim to replicate a specific aspect of the distribution of the points in $\mathcal{X}$ in the transformed points of the space $\mathcal{Y}$. This structure is defined by a set of hyperplanes in $\mathcal{X}$ and multiple labels for each training sample. Often, we take the set of hyperplanes to be uniformly sampled, and the labels to be the projections of the datapoint in $\mathcal{X}$ by these random hyperplanes. In this case, our method can be shown to be related to symmetrical methods such as Canonical Correlation Analysis (CCA).

In CCA [8], two transformations are found that cast samples from $\mathcal{X}$ and $\mathcal{Y}$ to a common target vector-space such that the matching vectors from the two spaces are transformed to similar vectors in the sense of maximal correlation coefficient. Additional constraints enfore the components of the resulting vectors to be pairwise uncorrelated and of unit variance. The CCA formulation is, therefore:

Problem 1. *non-regularized CCA*

$$\max_{U_X, U_Y} \sum_{i=1}^{n} x_i^\top U_X U_Y^\top y_i \ , \quad \textit{subject to}$$

$$\sum_{i=1}^{n} U_X^\top x_i x_i^\top U_X = \sum_{i=1}^{n} U_Y^\top y_i y_i^\top U_Y = I$$

The dimension of the target vector space is typically the minimum of the two dimensions d_M and d_P and shall be denoted by l. Thus U_X is a matrix of dimensions $d_M \times l$ and U_Y is of dimensions $d_P \times l$. For the matching task above, the vectors x_{new} and y_{new} are considered a match if the distance $||(U_X^\top x_{new} - U_Y^\top y_{new_j})||$ is small.

Since oftenly the feature vectors for both vector spaces are of dimensions significantly higher than the number of training samples, statistical regularization must be used to avoid overfitting. A regularized version of CCA suggested by [22] can then be used. Two regularization parameters need to be determined η_X and η_Y, to regularize the computations in each of the vector spaces. Another modification of CCA is kernel CCA (e.g. [2, 27]), which uses the "kernel trick" for capturing non-linear transformations.

CCA minimizes the distances between matching vectors in $\mathcal{X}$ and $\mathcal{Y}$, while disregarding the distances between non-matching vectors. An alternative optimization goal combines the minimization of distances between matching pairs with the maximization of distances between non-matching pairs, thus attempting to achieve maximal discriminative ability. This approach has been developed in [12]. We have extensively tested this approach, and have come to the conclusion that the combined energy (including the maximization of the distances for non-matching pairs) does not outperform CCA when the non-matching pairs are obtained by mixing the matching pairs.

The reason for the lack of improvement is that by using such a procedure many samples that are very similar to matching samples are obtained and labeled non-matching. One solution we have tried is to deliberately sample pairs (x_i, y_j) that are unlike the matching pairs, i.e., we exclude from the sampling of non-matching pairs those pairs (x_i, y_j) for which x_i is similar to x_j or y_i is similar to y_j. This does not seem to help – if the threshold for exclusion is too high, the obtained non-matching pairs are not very informative. If, however, the threshold is set low, the pairs are too similar to matching pairs.

Previous attempts to formulate a max-margin CCA-like problem include the Maximal Margin Robot [16]. Rather than maximizing the sum of correlations, Maximal Margin Robot maximizes the minimal correlation. Robustness to outliers is maintained through the inclusion of slack variables. This formalization produces the following quadratic programming

Problem 2 (Maximal Margin Robot)**.**

$$\min_{T,\xi} \ \frac{1}{2}\|T\|_F^2 + C\mathbf{1}^\top \xi \ , \quad \textit{subject to}$$

$$\forall 1 \leq i \leq n \quad y_i^\top T x_i \geq 1 - \xi_i \quad \xi_i \geq 0$$

Similarly to our method, a transformation T is recovered. However, it is assumed that both $\mathcal{X}$ and $\mathcal{Y}$ are of the same dimension.

The Nearest Neighbor Transfer [24] is a simple alternative that does not require optimization. Given a new vector x_{new}, this simple method chooses out of the training vectors of $\mathcal{X}$ the closest one

$$\arg\min_{i=1}^{n} \|x_i - x_{new}\|$$

and predicts the vector y_i in $\mathcal{Y}$. For the task of selecting a matching vector for x_{new} out of a set of vectors $\{y_{new_k} \in \mathcal{Y}\}_k$, the methods selects the vector most similar to the y_i: $\arg\min_k \|y_{new_k} - y_i\|$.

Many of the above vector to vector mapping techniques are symmetric, i.e., replacing the roles of the two vector spaces does not change the outcome. This is in contrast to the nature of many applications, where often one vector space is much more reliable than the other. Our method, presented in Section 3 maps from the less reliable vector space $\mathcal{Y}$ to the more reliable space $\mathcal{X}$ such that strong distinctions in $\mathcal{X}$ are presented in the mapped data from $\mathcal{Y}$.

3. Problem formulation

We are given n pairs of matching vectors $(x_i, y_i)_{i=1}^n$. For example, $x_i \in \mathcal{X}$ might depict the encoding of an image indexed i in one camera, while $y_i \in \mathcal{Y}$ depicts the a matching frame from another view of the same scene. As mentioned above, our framework is asymmetric and transforms data from the less informative space to the more reliable one. Assume that $\mathcal{X}$ is the reliable vector space and $\mathcal{Y}$ is the less reliable one. For example, wide field of view may be better suited for identifying the scene than a narrow field of view.

Moreover, assume that we have an additional set of k hyperplanes in the space $\mathcal{X}$ each known to provide a good separation of the n samples $x_1, \ldots, x_n$ into two classes, associated with labels $L_{ij} \in \{-1, 1, 0\}$ for the $i - th$ hyperplane and the $j - th$ sample (each hyperplane separates the samples differently). The source of these hyperplanes and labels is application dependent, and in [26] we provide several examples on how to obtain them.

Denote the set of hyperplanes as $w_1, .., w_k$. We look for a transformation $T : \mathcal{Y} \rightarrow \mathcal{X}$, and scalars $b_1, .., b_k$ such that for each i, the hyperplane w_i separates the n samples $Ty_1, .., Ty_n$ around b_i similarly to the labels L_{ij}, i.e., we encourage similarity between L_{ij} and the sign of the projections $w_i^\top Ty_j - b_i$ for all $i = 1..k$ and $j = 1..n$ for which $L_{ij} \neq 0$. Typically, the hyperplanes and the labels are such that $L_{ij} = sign(w_i^\top x_j - b_i^0)$ for some scalars $b_1^0, .., b_k^0$, however, b_i^0 and b_i often differ, and are not part of the input.

Defining the similarity between labels and the projections using correlations one would obtain an extension of CCA, in which an extra set of hyperplanes is used. Alternatively, inspired by the success of large-margin methods such as SVM [4], and LMNN [23], our method minimizes the hinge-loss. In contrast to SVM and LMNN, the margins are not maximized in our framework; Instead, they are replicated, i.e., and we seek to have the same margins in the space of transformed samples $T\mathcal{Y}$ as exist in the space $\mathcal{X}$.

For each of the k separators w_i, $i = 1..k$, we compute or otherwise obtain a goal margin m_i. Typically, this margin is computed in the vector space $\mathcal{X}$ on the training samples x_j as $m_i = \min_{j:L_{ij}>0} w_i^\top x_j - \max_{j:L_{ij}<0} w_i^\top x_j$. Alternatively, any definition of soft margins can be employed, e.g., in the non-separable case.

Using the above definitions, we define the following optimization problem, which yields a mapping $T : \mathcal{Y} \rightarrow \mathcal{X}$ that replicates the margins in $\mathcal{X}$.

Problem 3. *(MRM: Margin Replicating Mapping)*

$$\min_{T, \overrightarrow{b}} \sum_{ij} [m_i - L_{ij}(w_i^\top Ty_j - b_i)]_+$$

This is an unconstrained piecewise linear convex optimization problem that can be rewritten as linear programming by adding slack hinges:

Problem 4. *(MRM as linear programming)*

$$\min_{T, \overrightarrow{b}} \sum_{ij} \xi_{ij} \text{ , } \textit{subject to}$$
$$\xi_{ij} \geq m_i - L_{ij}(w_i^\top Ty_j - b_i)$$
$$\xi_{ij} \geq 0$$

In Sec. 4 we develop a gradient descent algorithm for MRM. Its first iteration (when starting at zero) is in the linear regions of the objective function, and no hinges are active. We show that in this case, there is a deep relation to CCA (see Section 5), therefore, in a sense, the first iteration follows CCA and then evolves.

4. Optimization

For such convex cost functions, the gradient descend method is guaranteed to converge to the global minima, however, this convergence is slow for a large number of separators. To speed up the convergence, we have developed a suitable sample and backtrack method depicted in Figure 1. The method is modeled after the dagging and backtracking modules of boosting methods [18, 5]. In the future, to further speed up the computation, we plan to incorporate within the procedure a column-generation linear-programming module [6].

The reason that the gradient descent methods would converge slowly on Problem 3 is that for every combination of a separator and a sample, there is a term of the form of $\max(*, 0)$, thus, there are many non-differentiability “break” points. During the gradient decent procedure (or more precisely sub-gradient), we cannot have a step size which is larger than the distance to the next break point (to avoid the risk of ’overshooting’), and the progress is therefore slow. In order to reduce the effective number of break points we sample a random subsets of hinge terms - thereby making the gradient step both longer and faster to compute. Each iteration samples a different subset ensuring diversity of search directions and a faster convergence.

5. Connection to other methods

In this section we show that both CCA and SVM are special cases of MRM. Also SVR and MMR are closely related. First, note that unlike most learning algorithms, MRM objective is bounded, thus does not intrinsically require regulation. Yet, regularized MRM can be defined:

Problem 5. *(bounded MRM)*

$$\min_{T,b} \sum_{ij} [m_i - L_{ij}(w_i^\top Ty_j - b_i)]_+ \textit{ w.r.t.}$$
$$\sum_{ij} (w_i^\top Ty_j)^2 \leq 1$$

This is exactly the problem of 3, where the solution is bounded. Since the range of projections T is convex, a

```
function [T,b]=MRM($\{\overrightarrow{y_j}\},\{\overrightarrow{w_i}\},\{m_i\},\{l_{ij}\}$)
Initialize T:=0 and b:=0
Repeat c times
    Select at random for $i=1..p$:
      $j_i^+$,$k_i^+$ s.t. $l_{j_i^+k_i^+}>0$
      $j_i^-$,$k_i^-$ s.t. $l_{j_i^-k_i^-}<0$
    Set $T^* := T$ //to allow backtracking
    Set $dT := \sum_i[(w_{k_i^-}y_{j_i^-}^\top) - (w_{k_i^+}y_{j_i^+}^\top)]$
    Set $\xi_i := m_{k_i^+} - w_{k_i^+}^\top Ty_{j_i^+} - b_{k_i^+}$
    Set $\psi_i := m_{k_i^-} + w_{k_i^-}^\top Ty_{j_i^-} + b_{k_i^-}$
    Set $d\xi_i := m_{k_i^+} - w_{k_i^+}^\top dTy_{j_i^+} - b_{k_i^+}$
    Set $d\psi_i := m_{k_i^-} + w_{k_i^-}^\top \ dTy_{j_i^-} + b_{k_i^-}$
    Let r be a random number in [0,3]
    Repeat $10^r$ times //try both long and short loops
        Minimize $\lambda := min_i[min(\frac{\xi_i}{d\xi_i},\frac{\psi_i}{d\psi_i})]$
            Save argmin: index $i_0$.
            Put $s_0 := +1$ if $\xi$, -1 if $\psi$
        Set $T := T - \lambda \ dT$
        Set $\xi := \xi - \lambda \ d\xi$ and $\psi := \psi - \lambda \ d\psi$
        if $s_0 = +1$
             $w := w_{k_{i_0}^+} \ \ y := y_{j_{i_0}^+}$
        if $s_0 = -1$
             $w := w_{k_{i_0}^-} \ \ y := y_{j_{i_0}^-}$
        Set  $dT$  :=  $dT$  +  $s_0wy^\top$
         $\forall$ $i$ Set  $d\xi_i$  :=  $d\xi_i$  +  $s_0(w^\top w_{k_i^+})(y^\top y_{j_i^+})$
         $\forall$ $i$ Set  $d\psi_i$  :=  $d\psi_i$  $-$  $s_0(w^\top w_{k_i^-})(y^\top y_{j_i^-})$
    End inner loop
     $\forall$ $i$ rebalance $b_i$ via convex 1D optimization.
    If global objective function evaluated on T
    is worse than that of $T^*$ then backtrack by:
         Set $T := T^*$
End main loop
Return T and b
```

Figure 1. Pseudocode of a method for computing the transformation T and biases b in the Margin Replicating Mapping mehtod.

global solution can be found by using a gradient decent algorithm similar to the one of Sec. 4. In our experience the bound is superfluous, however, since the linking to other methods is done asymptotically, such bounding simplifies the arguments.

Unlike MRM, the methods we compare to, namely, CCA (Problem 1), Maximaum Margin Robot (Problem 2) and SVR, do not employ hyperplanes as input. The linking is therefore performed for the case where random hyperplanes are used. The margins are set with accordance to the desired reduction. Specifically, for the random hyperplane case, Problem 5 would reduce to CCA if all m_i approach infinity and would reduce to an algorithm which closely resembles MMR if we set $m_i = 1$. Lastly, MRM reduces to SVM, for a unidimensional $\mathcal{X}$ taking the role of labels.

5.1. connection to SVM and SVR

In the 1D case, where the dimension of $\mathcal{X}$ is 1, w becomes a scalar, and the MRM problem reduces to:

Problem 6. *1D MRM:*
$$\min_{t,b} \sum_j [m - L_j(t^\top y_j - b)]_+$$

Problem 6 is equivalent to problem 7 below when $\gamma = 0$, as can be seen by dividing objective above by the positive (exhaugenic) constant m and substitute $u := \frac{t}{m}$.

Problem 7. *SVM as unconstrained minimization:*
$$\min_{u,b}(\gamma\|u\|^2 + \sum_j [1 - L_j(u^\top y_j - b)]_+)$$

Problem 7 is exactly the optimization problem of SVM, formulated via the hinge loss.

Note that both problems, SVM and MRM, are often bounded even without regularization. Yet, SVM is typically regularized, to make sure the largest margin is selected among all classifiers for which the penalty is minimized. Similarly, MRM can be also be regularized. However, experimentally, we found out that it results in little gain for high dimensional $\mathcal{X}$. In such cases, the penalties for misclassified Ty_j are strong enough even for low norm of T to stabilize the solution. Since we disregard the regularization term of MRM, we obtain an algorithm which has no parameters other than the input data and set of hyperplanes on $\mathcal{X}$. Even in the case of random hyperplanes, T is forced to match the desired structure as occur in Co-Kiring based solutions to SVR [21]. In contrast to semi-parametric methods such as [15], MRM assumes no prior on bias terms b and learn them from input. Of course, such priors, whenever available, might improve the algorithm's performance.

5.2. Connection to CCA

Links between MRM and the non-regularized CCA (problem 1) are presented in [26]. Similarly to CCA, the proposed variant receives zero mean inputs (centralized data). Then, given large m_i's (forcing the hinge terms to become loose), MRM will behave similarly to CCA. Note that the first iteration of MRM ignores all hinges, since the hinge functions are linear near the origin. Thus, the first iteration of MRM acts similarly to the case of $m_i = \infty$. Further iterations refine the produced transformation. This observation was verified on synthetic data [26].

6. MRM modifications

The MRM framework is flexible, allowing for several adaptations. The first two modifications below have been implemented and tested. The weighted MRM typically produces results that are similar to those of the vanilla MRM.

Kernel MRM is often used to speed up experiments involving high dimensional data. The implementation of the third, sparse MRM, is left for future work.

6.1. Weighted MRM

This variant specifies a confidence level for each label L_{ij}. One natural choice of confidence is to use, for a hyperplane w_i and example j the value $L_{ij} = w_i^\top x_j - b_i^0$ as the signed confidence value (recall that b_i^0 is the bias for hyperplane w_i in the space $\mathcal{X}$). The sign of this value $sign(L_{ij})$ is the label, and its absolute value $|L_{ij}|$ a measure of confidence (in practice it is beneficial to trim high confidence values). The following problem is optimized:

Problem 8. *Weighted MRM:*
$$\min_{T,b} \sum_{ij} |L_{ij}|[m_i - sign(L_{ij})(w_i^\top T y_j - b_i)]_+$$

6.2. kernelized MRM

This variant uses the kernel trick to allow for non-linear mappings, and to reduce computational time for high dimensional vector spaces. Let $\phi : X \rightarrow H^X$ and $\psi : Y \rightarrow H^Y$ be two transformations that embed the input samples in dimensional Hilbert spaces. We implicitly recover a transformation $\tau : H^Y \rightarrow H^X$, without performing operations in H^X or in H^Y. Since the recovered T is a linear combination of outer products, it can be represented as products of the form $w_i \psi(y_j)^\top$. During the algorithm run, the transformation T is applied to the datapoints in H^Y, and the results is correlated with w_l, thus obtaining scalars of the form $K_\phi(w_l, w_i) K_\psi(y_j, y_i)$.

We used this kernalization to speed up the linear case when the dimensionality is much larger than the number of examples. While the complexity of each iteration in the Algorithm of Figure 1 (dominated by re-calculations of the objective function) is $O(nkd)$, the complexity of each iteration in the kernelized algorithm is $O(kn)$ if $W^\top W$ and $Y^\top Y$ are precomputed (this precomputation takes $O(dn^2 + dk^2)$).

6.3. sparse MRM

To achieve an intrinsic feature selection, the columns of T might be encouraged to become zero, by binding appropriate cost terms (as suggested by [17]). We wish to eliminate entire columns, and therefore apply these terms to the maximal absolute value of each column:

Problem 9. *minimize over* T,β,ξ,ψ *of*
$$\sum_{ij} |L_{ij}| \xi_{ij} + C \sum_r \psi_r \text{ } w.r.t.$$
$$\forall cr \; \psi_r \geq T_{cr} \, , \forall cr \; \psi_r \geq -T_{cr}$$
$$\forall ij \; \xi ij \geq m_i - sign(L_{ij})[\sum_{rc} T_{rc} y_{rj} w_{ci} - \sum_r \beta_r y_{rj} w_{ri}]$$
$$\forall ij \; \xi_{ij} \geq 0$$

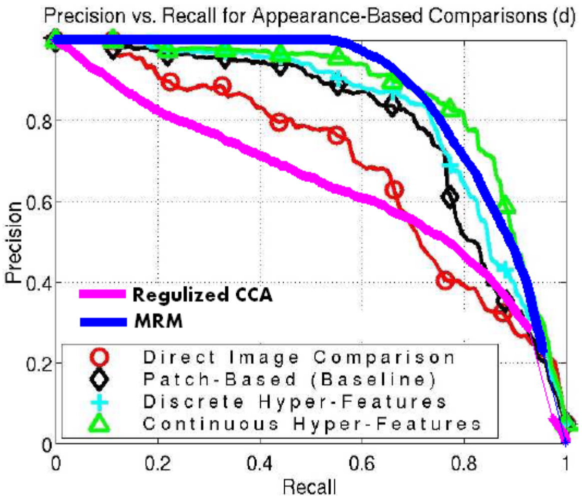


Figure 2. Precision/Recall statistics for matching car images between two cameras. We overlay our results on top of the results of Figure 5 in [7]. MRM performs similarly to the best image-matching technique of [7] eventhough a simple representatoin is used. Regularized CCA (best parameter found) yields poor results with this poorly discriminative data and Linear Ridge-Regression performed even worse (not shown).

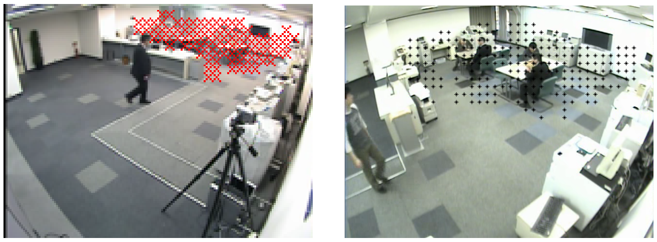
7. Results

We present results for the MRM algorithm for various applications. These include camera to camera mapping, and mimicking the performance of face recognition techniques. Other examples are provided in [26]. Below, 2000 random hyperplanes were used in all examples.

7.1. Matching between cameras

In several applications it is useful to compare the the output of multiple cameras. We employ two such datasets. The first was presented in [7] and contains the output of detected cars in two traffic cameras. The other is collected by us. In the first dataset, the task is to identify car based on a single view from the other camera, while training is on such matched pairs, but of different cars. The training set contains about 120 cars, and the test set contains 50 different car types. This yields a training set with 2922 samples, each is a 1200 dimensional vector, describing the three channels of rescaled 20x20 pixel images. Training time was 17 minutes on a standard PC. Results are comparable to [7], which employs a sophisticated image matching technique. See Fig. 2.

We also received 4 minutes of two synchronized video sequences with partly overlapping fields of view. We used half of the video to apply MRM with random hyperplanes. To test, we randomly picked from the second half of the video one frame from one camera, and 10 frames from the other camera, out of which 9 are random distractors.



(a) (b)

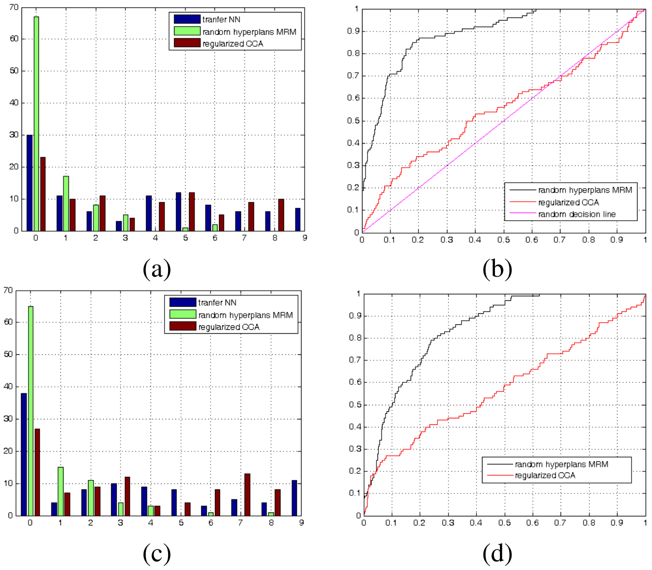
Figure 3. Sample frames from two synchronized videos. Frames from view (b) are mapped to view (a). The views overlap only partly, and are taken at very different angles. The red crosses mark the most prominent location in the view of (a) are pixels in range (i.e. columns in mapping matrix T) that their norm is $>$ 10% of the strongest. These turn out to be at the expected locations within the region viewed by both cameras. We run also the opposite direction and got analog marks in (b). Here algorithm have chosen to focus on the area when vast majority of motion occurs. Note that the two frames here are not matched in time.

Figure 3 depicts two sample frames. In this experiment, both feature spaces were simply 1200 dimentional vectors obtained by resizing the images to 40x30 pixels. Training MRM with 290 frames has taken 2 minutes. On the frame from the camera we used to map from, we mark red crosses to sign the output coordinates whose weight (norm of relevant column in mapping matrix) if at least 10% the maximum (most important pixel in spatial domain). Note that marks are spatially continuous, as desirable, although we did not employ such constraints. Figure 4(a,b) shows the performance of the various algorithms: Nearest Neighbor Transfer (Section 2), regularized CCA, and our MRM method. MRM considerably outperforms all other methods. We have also tried to analyze the opposite (less plausible) direction: map from the wide field of view into the narrow one. Analogous graphs appear in Fig. 4(c,d). As expected results are less impressive, however MRM still outperforms the two other methods.

7.2. Mimicking algorithm performance

One task that is framed naturally in the vector-to-vector learning framework is the task of mimicking an unknown algorithm. We test this application in the context of face recognition, using the LFW database [11]. We wish to recognize face images based on the well-proven LBP feature set [1]. LBP represents each face image as a vector. For the task of comparing two face images and deciding whether they belong to the same person, we consider the vector v_{12} of absolute differences $|v_1 - v_2|$, where v_1 and v_2 are the LBP encodings of these images.

Training linear SVM on this vector, using 9 splits of the LFW benchmark for training and one for testing, and repeating ten times, yields an average performance of 69.5%, which is slightly higher than applying a simple threshold to the Euclidean distance (67.2%) between v_1 and v_2.



(a) (b)

(c) (d)

Figure 4. Performance charts for the video to video matching task. (a) histogram of the ranking of the true frame within the list of distractors as given by each algorithm. As can be seen, MRM typically ranks the highest the correct frame. (b) ROC curves for same vs not-same queries in the same settings, i.e., the accuracy in which each method is able to distinguish between matching pairs and non-matching pairs of frames. (c,d) Performance charts for the video to video matching task, now mapping in the opposite direction.

To further improve performance, we employ the output of more advance algorithms. We employ the raw outputs of the 16 different classifiers employed in [25] to obtain a results of 78.5%. Out of these 16 methods, only one can be directly computed from the vector of absolute LBP differences v_{ij} (the Euclidean distance of LBP descriptors). Using MRM, we map these vectors of absolute differences to the 16D output obtained by the methods of [25]. Thus, the input to MRM contains 4800 samples (pairs of faces, only 8 training splits, since one is used as the background sample set of [25]), each consisting of a vector of 3717 absolute difference and a vector of 16 classifier outputs. Training takes 26 minutes.

We then train SVM on the values of Tv_{ij} and obtain an average performance level of 74.6%, which is not as high as the 16 classifiers together, but is much easier to compute: since both T and the learned SVM are linear, the final classifier is linear as well. CCA and linear ridge regression, applied in a similar setting, both produce results lower than what is obtained with SVM itself (69.5%), for a wide range of regularization parameters.

We have mimicked the performance of the face recognition algorithm based on the raw output of the classifiers (real numbers). If the algorithms are give as a black box, and only the final prediction is given, a confidence level can be estimated empirically, by adding noise to the input data. This is left for future research.

8. Conclusions

The problem of perceptual inference based on pairs of matching vectors is applicable to a wide range of computer vision problems. Despite some effort, the classical regularized CCA algorithm seems to performs better than many of the more recent contributions. This might stem from the addition of discriminative information of non-matching pairs that is either much less relevant than the information obtained by matching pairs, or is obtained using assumptions that are often false.

Here we employ margins, however, unlike previous work this is done by using discriminative information provided in the feature space itself. Another contribution is that we refrain from maximizing the margins in the transformed space, and instead aim to replicate the input margins. Since the input margins are given in one of the two spaces, and replicated in the other, the problem is asymmetric by nature. We note that many of the applications in vector-to-vector learning are indeed asymmetric.

While our method maps one space to the other, we do not perform classical regression analysis. Our optimization framework allows the mapped points to considerably differ from the matching points in the other space. The emphasis is put on similar margins, not similar coordinates, and the optimization computes a new set of bias terms.

Acknowledgments

Lior Wolf is supported by the Israel Science Foundation (grant No. 1214/06), and The Ministry of Science and Technology Russia-Israel Scientific Research Cooperation. This research was carried out in partial fulfillment of the requirements for the Ph.D. degree of Nathan Manor.

References

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *PAMI*, 28(12):2037–2041, Dec. 2006. 6

[2] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *JMLR*, 2003. 2

[3] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In *CVPR*, 2001. 1

[4] G. I. Boser B. and V. V. An training algorithm for optimal margin classifiers. In *COLT*, 1992. 3

[5] T. Bylander and L. Tate. Using validation sets to avoid overfitting in adaboost. In *Artificial Intelligence Research Society Conference*, 2006. 3

[6] A. Demiriz, K. P. Bennett, and J. Shawe-Taylor. Linear programming boosting via column generation. *Mach. Learn.*, 46(1-3):225–254, 2002. 3

[7] A. Ferencz, E. G. Learned-Miller, and J. Malik. Learning to locate informative features for visual identification. *IJCV*, 2008. 1, 5

[8] H. H. Relations between two sets of variates. In *Biometrika, Vol. 28*, pages 321–337, 1936. 2

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction.* Springer, 2001. 2

[10] A. E. Hoerl and R. Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 1970. 2

[11] G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. University of Massachusetts, Amherst, TR 07-49, 2007. 6

[12] D. Lin and X. Tang. Inter-modality face recognition. *ECCV*, pages 13–26, 2006. 2

[13] H.-H. Nagel. Steps toward a cognitive vision system. *AI Mag.*, 2004. 1

[14] B. Schölkopf, A. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000. 2

[15] A. J. Smola, T. T. Frieß, and B. Schölkopf. Semiparametric support vector and linear programming machines. In *NIPS*, 1998. 4

[16] B. T. Szedmak S. and H. D.R. A metamorphosis of canonical correlation analysis into multivariate maximum margin learning. In *ESANN*, 2007. 2

[17] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal Royal Statistics*, 1994. 5

[18] K. M. Ting and I. H. Witten. Stacking bagged and dagged models. In *ICML*, 1997. 3

[19] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector learning for interdependent and structured output spaces. In *ICML*, 2004. 2

[20] V. Vapnik. *Statistical learning theory*. Wiley, 1998. 2

[21] E. Vazquez and E. Walter. Multi-output support vector regression. In *13th IFAC Symposium on System Identification*, pages 1820–1825. Citeseer, 2003. 4

[22] H. D. Vinod. Canonical ridge and econometrics of joint production. *Journal of Econometrics*, 1976. 2

[23] B. J. Weinberger K. and S. L. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, 2005. 1, 3

[24] L. Wolf and Y. Donner. An experimental study of employing visual appearance as a phenotype. *CVPR*, 2008. 1, 2

[25] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Faces in Real-Life Images Workshop in ECCV*, 2008. 6

[26] L. Wolf and N. Manor. Visual recognition using mappings that replicate margins. Extended Technical Report, available at www.cs.tau.ac.il/~wolf, 2010. 3, 4, 5

[27] L. Wolf and A. Shashua. Learning over sets using kernel principal angles. *J. Mach. Learn. Res.*, 4:913–931, 2003. 2