PCCA: A new approach for distance learning from sparse pairwise constraints

Alexis Mignon, Frédéric Jurie

PCCA: A New Approach for Distance Learning from Sparse Pairwise Constraints

Alexis Mignon and Frédéric Jurie
GREYC, CNRS UMR 6072, Université de Caen, France
lastname.firstname@unicaen.fr

Abstract

This paper introduces Pairwise Constrained Component Analysis *(PCCA), a new algorithm for learning distance metrics from sparse pairwise similarity/dissimilarity constraints in high dimensional input space, problem for which most existing distance metric learning approaches are not adapted. PCCA learns a projection into a low-dimensional space where the distance between pairs of data points respects the desired constraints, exhibiting good generalization properties in presence of high dimensional data. The paper also shows how to efficiently kernelize the approach. PCCA is experimentally validated on two challenging vision tasks, face verification and person re-identification, for which we obtain state-of-the-art results.*

1. Introduction

Several computer vision problems rely heavily on the use of distance functions learned from image pairs. Among them, *face verification* and *person re-identification* are two important tasks. While face verification [16, 23] is of deciding whether two face images (*e.g*. Fig. 1) represent the same person or not, person re-identification [9, 15] is that of matching images of persons taken across non-overlapping camera views (*e.g*. Fig. 1).

Two key ingredients of approaches addressing these tasks are (i) the descriptors (or signatures) used to represent images and (ii) the distance function used to compare the signatures. In the present paper, we focus on the latter.

Given the many uncontrolled sources of variations, *e.g*. changes in illumination, human pose and camera properties, it is unlikely that any standard distance function, *e.g*. Euclidean distance, will be adequate for the task even with very relevant image signatures. Hence, many works have successfully addressed the problem by learning *task specific* distance functions [14, 33].

Distance metric learning is a well studied topic. Many widely used learning algorithms, like unsupervised clustering (*e.g*. k-means), Nearest Neighbors and kernel-based classifiers, require a metric over the data input space. Such a metric is not only supposed to reflect the intrinsic properties of the data but also to be adapted to the specific application domain. Therefore, many approaches have tried to learn the distance using domain specific constraints [1, 10, 11, 14, 19, 21, 27, 30, 31, 33].

Figure 1. Sample positive pairs for the LFW dataset [16] (left) and the VIPeR dataset [12] (right).

Although all metric learning methods rely roughly on the intuitive idea that similar data points should be closer than dissimilar points, most of them are not adapted to the tasks we are interested in. Either they assume the training data to be fully annotated (*i.e*. class labels must be given) [10, 11, 19, 27, 30], or they are too domain specific [1, 5, 33], or they suffer from significant loss in performance when the dimensionality of the input space is high or when the amount of available training data is low [7, 14].

In the present paper, we propose a new metric learning algorithm applicable when only a sparse set of pairwise similarity constraints on high-dimensional data points are given for training. In other words we are interested in problems where similarity information is available only for a limited number of pairs of points. We build a low-dimensional space in which the training constraints are respected by minimizing a loss function penalizing distances greater than a threshold for positive pairs and lower than the same threshold for negative pairs. Thus, our method is of loss minimization with regularization (by fixing the dimension of the projection space). Experimental validations on challenging datasets for face verification *e.g*. Labeled Faces in the Wild (LFW) [16] and person re-identification *e.g*. VIPeR [12] validate our approach.

2. Related work

Distance metric learning plays a significant role in pattern recognition and therefore has received a lot of attention. The literature in distance learning can be split into two main categories: *manifold learning*, for which the key idea is to learn an underlying low-dimensional manifold preserving the distance between observed data points (ISOMAP [26] or Multidimensional Scaling [6] are two good representatives of this category), and *supervised or semi-supervised approaches* that try to learn metrics by keeping points of the same class close while separating points from different classes. This paper focuses on the latter.

Following the early work of Xing *et al.* [31], most distance metrics learning approaches learn a Mahalanobis-like distance: $\mathcal{D}^2_M(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^T M(\mathbf{x}-\mathbf{y})$ where M is a positive semi-definite (PSD) matrix satisfying the training constraints. The main advantage is that the optimization of the Mahalanobis matrix can be seen as a constrained convex programming problem which can be solved with existing efficient algorithms. Furthermore, [20] shows that this framework can be extended to non-linear problems using the kernel trick.

However, guaranteeing M is PSD can be computationally expensive. Hence, several works such as [11, 27, 33] factorize M as $M = L^T L$, ensuring the PSD constraint and implicitly defining a (potentially low-dimensional) projection into an Euclidean space which reflects the distance constraints. Our work belongs to this line.

Besides these general approaches, several methods specifically address k-Nearest Neighbors (k-NN) classification. They either introduce constraints on absolute distances between pairs *e.g.* Neighborhood Component analysis (NCA) [11] and Maximally Collapsing Classes method (MCC) [10], or constraints on relative distances such as the Large Margin Nearest Neighbors [30] or its variants *i.e.* Large Margin Component Analysis (LMCA) [27], invariant LMNN [19] and LMNN-R [8]. With these approaches, the k-nearest neighborhood of each point is explicitly inspected and the distance metric is learned in a way that for each training point, the neighbors from others classes are always farther than the neighbors from the same class up to a margin. As pointed out in the introduction, these approaches require the class labels of all the training points, and are thus not adapted to problems for which only pairwise constraints are available.

In contrast, On-line Algorithm for Scalable Image Similarity (OASIS) [5] and Probabilistic Relative Distance Comparison (PRDC) [33] are specifically designed to work with pairwise constraints. However, they make strong assumptions about input data or about the structure of the constraints, making them inapplicable in general (OASIS requires sparse vectors as input data, and, as PRDC, can be used only if each class is represented by at least one pair of similar and one pair of dissimilar points). For tasks such as face recognition with the Labeled Faces in the Wild dataset [16], none of these requirements are satisfied.

Interestingly, the recently proposed Information Theoretic Metric Learning (ITML) [7] and Logistic Discriminant Metric Learning (LDML) [14] are designed to deal with general pairwise constraints. However, they exhibit poor generalization capability when trained with few training data [14, 33]. Furthermore, ITML uses a Kullback-Leibler divergence criterion and a specific optimization scheme [18] to maintain PSD and low-rank properties, and optimizes the full rank matrix (such as LDML). The computational cost as well as the number of parameters to learn increases with the square of the dimensionality of the input space, which makes them impractical for the tasks we address (at least without prior dimensionality reduction) and prone to overfitting. While LDML does not guarantee M to be PSD and does not use any regularization term, its robust probabilistic model based on the logistic function has been shown to perform better than ITML (see [14] for comparisons).

Although our method can be used in the general pairwise constraints case, in contrast with ITML and LDML it can cope with high dimensional data and exhibits good generalization properties even with few training data.

Finally, coming back to the tasks we are interested in (*i.e.* face verification and person re-identification), the most efficient approaches rely on the above-mentioned approaches for distance learning. ITML and LDML are successfully used for face verification in [14, 25] while LMNN-R and PRDC are used for person re-identification in [8, 33] outperforming previous approaches such as RankSVM [24] or the boosting based approach of [13].

3. Pairwise Constrained Component Analysis

This section presents our contribution, the Pairwise Constrained Component Analysis (PCCA).

Without loss of generality, we assume the training data consists of a set of N $d-$dimensional points (*e.g.* visual descriptors), denoted as $X_{N\times d}$, and a small set of c constraints between points of X, denoted $\mathcal{C} = \{(i_k, j_k, y_k) | k = 1,\dots,c\}$. $i_k, j_k \in \{1\dots N\}^2$ are the indices of the two points of the constraint k, and $y_k \in \{-1, 1\}$ indicates whether the points belong to the same class or not. While defining the constraints this way is appropriate for sparse pairwise constraints ($c \sim O(N)$), it can also be used for clustering tasks by having one constraint per pair of training data ($c \sim O(N^2)$).

3.1. Problem formulation

We look for the linear transformation L that maps data points into a low-dimensional space of dimension $d' \ll d$ in which the Euclidean distance satisfies the pairwise (training) constraints $\mathcal{C}$. This mapping transforms $\mathbf{x}$ (original in-

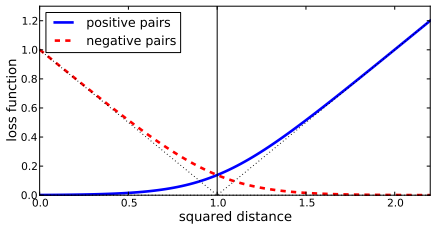

Figure 2. Loss function for pos. (solid) and neg. (dashed) pairs.

put space) into $\mathbf{x}' = L\mathbf{x}$. The square of our distance is then given by

$$D_L^2(\mathbf{x}, \mathbf{y}) = \|L(\mathbf{x} - \mathbf{y})\|_2^2 \tag{1}$$

Learning the distance means computing L such that, for a given threshold t, distances between similar points are smaller than t while those between dissimilar points are greater than t. As t fixes the scale of distances in the low-dimensional space, it can be set to $t = 1$.

Finding the optimal value of L can be done by minimizing the following objective function:

$$\min_L E(L) = \sum_{n=1}^{c} \ell_\beta \left(y_n (D_L^2(\mathbf{x}_{i_n}, \mathbf{x}_{j_n}) - 1) \right) \tag{2}$$

where $\ell_\beta(x) = \frac{1}{\beta} \log(1 + e^{\beta x})$ is the generalized logistic loss function [32]. The objective function thus penalizes the pairs which do not match the required constraints. Our loss function, $\ell_\beta(x)$, is a smooth approximation of the hinge loss $h(x) = \max(0, x)$ to which it converges asymptotically as the *sharpness* parameter β increases *i.e.* $\lim_{\beta \to \infty} \ell_\beta(x) = h(x)$ (see Fig. 2).

Although, the optimization problem in Eq. (2) is not convex with respect to L, since $\ell_\beta(x)$ is convex, it is convex with respect to $M = L^T L$. Consequently, solving the non-convex problem Eq. (2) with respect to L by an iterative gradient descent scheme is guaranteed to converge to a global minimum [17, 2].

Our algorithm has two additional parameters: (i) the dimensionality of the output space d' and (ii) the sharpness parameter β. Note that the function $\ell_\beta(x)$ is always above the hinge loss $h(x) = \ell_\infty(x)$. Thus optimizing on β simultaneously with L, can trivially lower the value of the objective function without improving L. Hence, β is fixed during the optimization, and we estimate β and d' by cross validation.

The problem in Eq. (2) is thus solved using a gradient-descent scheme with line search. The gradient of the objective function $E(L)$ is given as:

$$\frac{\partial E}{\partial L} = 2 \sum_{n=1}^{c} y_n \sigma_\beta(y_n(1 - D_L^2(\mathbf{x}_{i_n}, \mathbf{x}_{j_n}))) L C_n \tag{3}$$

where $\sigma_\beta(x) = (1 + e^{-\beta x})^{-1}$, which corresponds to the usual logistic function for $\beta = 1$, and $C_n = (\mathbf{x}_{i_n} - \mathbf{x}_{j_n})(\mathbf{x}_{i_n} - \mathbf{x}_{j_n})^T$.

Compared to ITML [7] and LDML [14] our formulation presents several advantages. First, the linear mapping to a low-dimensional space is equivalent to imposing a low rank constraint on the Mahalanobis matrix. This helps us in avoiding a regularization term like in ITML and, thus, keeping the number of parameters (to learn) of $O(d)$ instead of $O(d^2)$ (ITML and LDML). In practice, it allows us to work directly with high-dimensional input data, while ITML and LDML must be preceded by a step of dimensionality reduction resulting in loss of information. Finally, as pointed out in section 2, the benefits of LDML are mostly due to its robust probabilistic model. In the following, we show that PCCA implicitly defines a robust probabilistic model also.

3.2. Probabilistic interpretation

The loss function $\ell_\beta(x)$ can be rewritten as $\ell_\beta(x) = -\frac{1}{\beta} \log(\sigma_\beta(-x))$. If the probability p_n that the n-th pair is correctly labeled is given by

$$p_n = \sigma_\beta(y_i(1 - D_L^2(\mathbf{x}_{i_n}, \mathbf{x}_{j_n}))) \tag{4}$$

then $e^{-\beta E(L)} = \prod_{n=1}^{c} p_n$ can be interpreted as the likelihood that all pairs are correctly labeled. Solving the problem given Eq. (2) is thus equivalent to finding the maximum likelihood estimate of L.

3.3. Kernel PCCA

Distance metric learning methods can suffer, when the given data has non linearities, due to the linear form of the Mahalanobis distance. To partially alleviate this limitation, several authors [28, 11, 27, 7] have used the "kernel trick" to map the data vectors into an higher dimensional space without having to explicitly compute the mapping. Only the value of their inner products (precomputed and stored in the kernel matrix) is required.

Following this line, we propose to re-parametrize L by $L = AX^T$, which is equivalent to consider that each row of L is a linear combination of elements of X. Then we have

$$\begin{aligned} D_L^2(\mathbf{x}_i, \mathbf{x}_j) &= \|AX^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 \\ D_L^2(\mathbf{x}_i, \mathbf{x}_j) &= \|A(\mathbf{k}_i - \mathbf{k}_j)\|_2^2 \\ D_L^2(\mathbf{x}_i, \mathbf{x}_j) &= D_A^2(\mathbf{k}_i, \mathbf{k}_j) \end{aligned} \tag{5}$$

where $\mathbf{k}_l = X^T \mathbf{x}_l$ is the l-th column of the kernel matrix $K = X^T X$. An interesting property of this formulation is that the linear and the kernel form of the distance have similar equations. However, learning A the same way as L (*i.e.* using the kernel matrix K instead of X) has bad convergence properties (see Fig. 3). This has been noticed by several authors who instead solve the dual problem [28, 10]. In contrast, we show that the problem can be handled

directly in the primal by using a descent direction equivalent to the gradient of the linear case.

The adaptation rule of the gradient descent in the linear case is:

$$L_{t+1} = L_t - 2\eta L_t \sum_{n=1}^{c} \mathcal{L}_n^t C_n \tag{6}$$

where L_t is the value of the matrix L at iteration t, η the learning rate and $\mathcal{L}_n^t = y_n \sigma(\beta y_n (1 - D_{L_t}^2(\mathbf{x}_{i_n}, \mathbf{x}_{j_n})))$. By writing $L_t = A_t X^T$ and multiplying all terms on the right by X we get:

$$\begin{aligned} A_{t+1} X^T X &= A_t X^T X - 2\eta A_t \textstyle\sum_{n=1}^{c} \mathcal{L}_n^t X^T C_n X \\ A_{t+1} K &= A_t K - 2\eta A_t \textstyle\sum_{n=1}^{c} \mathcal{L}_n^t \Gamma_n \end{aligned} \tag{7}$$

with:

$$\begin{aligned} \Gamma_n &= X^T (\mathbf{x}_{i_n} - \mathbf{x}_{j_n})(\mathbf{x}_{i_n} - \mathbf{x}_{j_n})^T X \\ \Gamma_n &= (\mathbf{k}_{i_n} - \mathbf{k}_{j_n})(\mathbf{k}_{i_n} - \mathbf{k}_{j_n})^T \end{aligned} \tag{8}$$

We then multiply both sides of equation (7) on the right by K^{-1}:

$$A_{t+1} = A_t - 2\eta A_t \textstyle\sum_{n=1}^{c} \mathcal{L}_n^t \Gamma_n K^{-1} \tag{9}$$

$$A_{t+1} = A_t - 2\eta A_t \textstyle\sum_{n=1}^{c} \mathcal{L}_n^t K J_n \tag{10}$$

where $J_n = (\mathbf{e}_{i_n} - \mathbf{e}_{j_n})(\mathbf{e}_{i_n} - \mathbf{e}_{j_n})^T$ and $\mathbf{e}_l$ is the l-th vector of the canonical basis, namely the vector with the l-th element equal to 1 and every other to 0.

We can notice that the adaptation rule in Eq. (9) is identical to the original formula in Eq. (6) except that the gradient is right multiplied by K^{-1}. Matrix K^{-1} acts as a preconditioner for the gradient descent. The same issue has been observed when solving the Support Vector Machine (SVM) problem in the primal [4].

Along with faster (theoretical) convergence, another benefit of this preconditioner is that only two columns of the preconditioned gradient matrix have to be modified at each iteration for each pair (see Eq. (10)) instead of the full matrix, which dramatically speeds up the computation of the gradient. Fig. 3 shows typical evolution of the objective function as a function of the number of iterations, with and without preconditioning. Notice the log-log scales on the axes. While the preconditioned version of the algorithm takes aggressive descending steps, the raw version tends to wander in plateau regions which makes convergence without preconditioning roughly 10 times slower.

4. Experiments

We validate the proposed approach on two different vision tasks, namely *face verification* and *person re-identification*, for which having a good metric is crucial. We use two publicly available challenging datasets. In the following, we first present the datasets and their benchmarking protocols and then present our results.

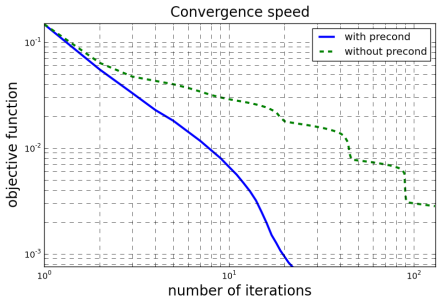

Figure 3. Convergence speed of the PCCA algorithm with and without preconditioning, on the VIPeR dataset using RBF χ^2 kernel. See Section 4 for details.

Datasets. For face verification, we use the popular *Labeled Faces in the Wild* (LFW) dataset [16]. It contains more than 13,000 images of faces collected from the web. 1680 of the pictured people have two or more distinct photos. The database in split in two views. `View 1`, which is given for development purposes (training, validation, model selection, etc.) has 2200 pairs of faces (half positives, half negatives) and a test set of 1000 pairs. `View 2` is for benchmarking only. It consists of 10 folds of 600 pairs each used for cross validation. The average classification score and standard deviation over the 10-folds is reported. We address here the *image-restricted* setting, where the name of the persons are not explicitly given in the training set. Therefore, only the information about whether a pair of images is matched or mismatched is known.

Our experiments on person re-identification use the *Viewpoint Invariant Pedestrian Recognition* (VIPeR) database [12]. It contains 632 pedestrian image pairs (1264 images) and is the largest publicly available dataset for person re-identification. Images are taken from arbitrary viewpoints and different illumination conditions. Each image is scaled to 128x48 pixels and is encoded in RGB. For evaluation purposes, the dataset is split into train and test sets by randomly selecting p persons out of the 632 for the test set, the remaining persons being in the train set. Since there are two images per person, we use them in the train set to provide one positive pair for each person while n^- negative pairs for each person are built by randomly selecting one image of the person and one image of another person. The test set is split into a gallery and a probe set by randomly putting – for each person of the test set – one of the two images to the probe set and the other to the gallery. The process is repeated 10 times and the results are reported as the mean/std values over the 10 folds. Performances are evaluated with the Cumulative Match Characteristic (CMC, see [12] for details), which can be seen as the *recall at* r. CMC score at rank $r = 1$ thus corresponds to the recognition rate. Computing the CMC for $r > 1$ (e.g. $r = 5$) is also important since in real use cases the first retrieved images can be visually inspected by an operator.

Metric	Number of training pairs 600	10,000
ITML-sqrt [14]	76.2 ± 0.5	80.5 ± 0.5
LDML-sqrt [14]	77.5 ± 0.5	83.2 ± 0.4
PCCA-sqrt	82.2 ± 0.4	83.3 ± 0.5
PCCA-χ^2	83.1 ± 0.5	84.3 ± 0.5
PCCA-χ^2_{RBF}	$\mathbf{83.8 \pm 0.4}$	$\mathbf{85.0 \pm 0.4}$

Table 1. Accuracy on LFW with 600 and 10,000 training pairs per fold. See text for details.

Image representations. For comparison, all our experiments on LFW use the SIFT descriptors computed by [14], available on their website. For experiments on VIPeR we use descriptors similar to those proposed by [33] *i.e.* 16 bins color histograms in 3 color spaces (RGB, HSV and YCrCb) as well as texture histograms based on Local Binary Patterns (LBP) computed on 6 non overlapping horizontal strips. All the histograms are independently normalized to unit L_1 norm and concatenated into a single vector.

Explicit Feature Maps and Kernelization. Our image representations for both databases are histograms, so the experiments were all performed using metrics adapted to histograms.

The experiments with linear approaches were performed using element-wise square-rooted histograms and are thus indicated by a *-sqrt* suffix. As shown by [29], this is equivalent to mapping the original histograms into the feature space corresponding to the Bhattacharyya kernel, improving the performance.

Kernel PCCA uses the χ^2 kernel: $K_{\chi^2}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{d} \frac{2x_i y_i}{x_i+y_i}$ for which the corresponding metric distance is: $D^2_{\chi^2}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{d} \frac{(x_i-y_i)^2}{x_i+y_i}$. The generalized radial basis function (RBF) kernel [29] is $K^{\mathrm{RBF}}_{\chi^2}(\mathbf{x},\mathbf{y}) = e^{-D^2_{\chi^2}(\mathbf{x},\mathbf{y})}$, generalizing the Gaussian kernel for non-euclidean spaces.

The experiments based on the χ^2 and the RBF χ^2 kernels are respectively indicated by the use of -χ^2 and -$\chi^2_{\mathbf{RBF}}$ suffices.

4.1. Experiments on face verification

In this section, we use LFW to evaluate the performance of PCCA for different kernels. We also evaluate the impact of the number of training pairs available and compare the performance with state-of-the-art approaches.

Table 1 presents our results and gives comparisons with LDML and ITML (using same SIFT descriptors) with 600 and 10,000 training pairs, for different kernels. Note that for PCCA we fixed $d' = 40$ and $\beta = 3$ for all the following experiments (including those on VIPeR). These values have been observed to give good average results on validation data (`View 1`).

The first column gives the performance for 600 training pairs per fold, where our method obtains 83.8 ± 0.4,

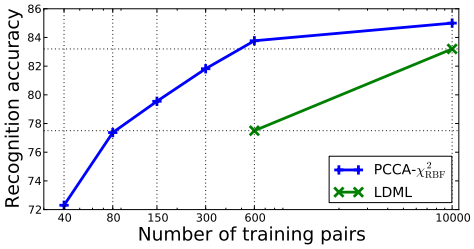

Figure 4. Recognition accuracy of PCCA-χ^2_{RBF} as a function of the number of training pairs for training. Results for LDML [14] with 600 and 10000 pairs are also reported with horizontal lines.

which is significantly better than the $77.5 \pm 0.5\%$ obtained by LDML [14]. ITML gives consistently lower results than both PCCA and LDML.

As said before, LFW has 600 pairs in each of the 10 folds of `View 2`. This is the "official" unrestricted setting for benchmarking with LFW. However, to evaluate the impact of the number of training pairs, we have used different configurations with more and less than 600 training pairs per folds. For producing more than 600 pairs, we had to use the identity of the people depicted in each set to randomly create new pairs. It means we are in this case in the *unrestricted* setting of LFW.

With 10,000 pairs per set, LDML and linear PCCA give comparable results, *i.e.* $83.2 \pm 0.4\%$ and $83.3 \pm 0.5\%$ respectively. Using the χ^2 kernel with PCCA gives a mean accuracy of $84.3 \pm 0.5\%$ and the best performances are obtained with the RBF χ^2 kernel ($85.0 \pm 0.4\%$) which again significantly out-performs LDML.

Fig. 4 shows the performance of PCCA as a function of the number of training pairs per fold, using the RBF χ^2 kernel. The scores for LDML are presented for comparison purposes as reported in [14]. The scores increases quickly for small numbers of training pairs (note the log scale on the horizontal axis) and then keep increasing slowly up to 10,000 pairs, indicating that only a small set of constraint is required to reach optimal scores. With 80 training pairs we already have equivalent performance to LDML with 600 pairs. Our method outperforms LDML with a large margin, especially with small number of training pairs. Note that a strong dimensionality reduction (PCA) must be applied to the input data before using LDML, while PCCA can use the data in their original space.

Furthermore, our results compare remarkably well with other state-of-the-art results on LFW, knowing we use only 9 SIFT descriptors (at 3 scales) for representing faces. The much more elaborate and face-specific LE descriptor [3] gives only $83.4 \pm 0.6\%$ in the restricted paradigm. [22] reports $85.6 \pm 0.5\%$ using very densely sampled local binary patterns combined with a learned cosine similarity measure.

$p = 316$	$r = 1$	$r = 5$	$r = 10$	$r = 20$	$p = 532$	$r = 1$	$r = 5$	$r = 10$	$r = 20$
PRDC $n^- = 1$ [33]	15.66	38.42	53.86	70.09	PRDC $n^- = 1$ [33]	9.12	24.19	34.40	48.55
MCC $n^- = 631$ [33]	15.19	41.77	57.59	73.39	MCC $n^- = 199$ [33]	5.00	16.32	25.92	39.64
ITML $n^- = 631$ [33]	11.61	31.39	45.76	63.86	ITML $n^- = 199$ [33]	4.19	11.11	17.22	24.59
PCCA-sqrt $n^- = 1$	13.48	34.84	49.43	67.18	PCCA-sqrt $n^- = 1$	7.34	21.02	31.30	45.37
PCCA-χ^2 $n^- = 1$	13.67	35.22	49.93	68.20	PCCA-χ^2 $n^- = 1$	7.03	20.32	30.86	45.71
PCCA-χ^2_{RBF} $n^- = 1$	17.02	43.26	58.67	76.36	PCCA-χ^2_{RBF} $n^- = 1$	7.61	22.42	33.40	48.42
PCCA-sqrt $n^- = 10$	17.28	42.41	56.68	74.53	PCCA-sqrt $n^- = 10$	8.44	24.34	35.62	50.07
PCCA-χ^2 $n^- = 10$	17.28	43.64	59.68	76.04	PCCA-χ^2 $n^- = 10$	7.95	24.23	35.73	50.45
PCCA-χ^2_{RBF} $n^- = 10$	**19.27**	**48.89**	**64.91**	**80.28**	PCCA-χ^2_{RBF} $n^- = 10$	**9.27**	**24.89**	**37.43**	**52.89**

Table 2. Cumulative Match Characteristic on VIPeR with $p = 316$ and $p = 532$ persons in the test set, for the to r retrieved images. See text for details.

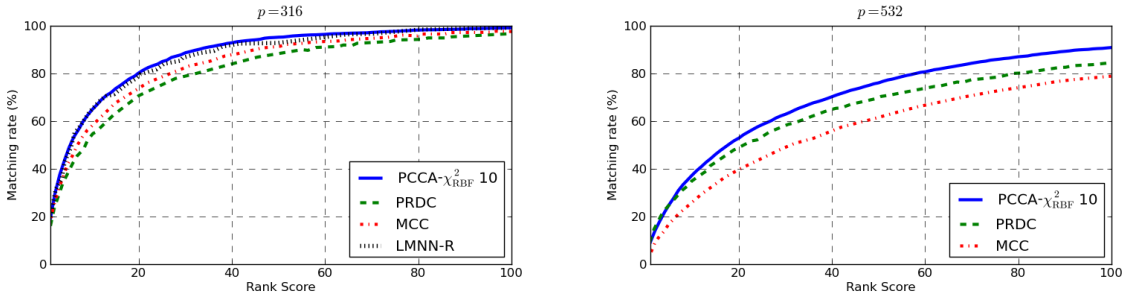

Figure 5. VIPeR dataset: CMC as the function of r, for PCCA and 2 state of the art approaches.

4.2. Experiments on person re-identification

Our experiments on the VIPeR database follow the protocol given in [12]. Table 2 presents our results as well as those obtained by two state-of-the-art approaches, namely PRDC [33] and MCC [10]. Results for ITML, again lower than others, are given for comparison purposes only. For PRDC and MCC, we reproduce the figures given in [33]. The left-hand part of the table reports CMC scores obtained with $p = 316$ persons in the test set (the 316 remaining ones being in the train set), for different kernels and different ratio n^- of negative/positive pairs in the train set. The right-hand part of the table is for $p = 532$ person in the test set (100 being left for the train set).

As expected, the performance of PCCA tends to increase with the complexity of the kernel (e.g. +4% for $r = 1$ and $p = 316$ with χ^2_{RBF}) and, to a smaller extent, with the number of negative pairs used in the train set (e.g. +1% for 9 additional negative pairs, for $r = 1$ and $p = 316$, with the χ^2 kernel).

PCCA always outperforms MCC, even with $n^- = 1$, both for $p = 316$ and $p = 532$, when using the RBF χ^2 kernel. With $n^- = 10$ negative pairs, PCCA outperforms both PRDC and MCC.

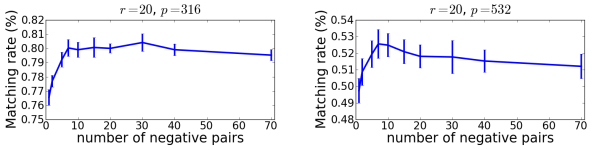

Figure 6. CMC of PPCA as a function of the number of negative pairs used for training. Computed on VIPer at rank $r = 20$, for $p = 316$ and $p = 532$.

Please note that even if PRDC uses only sparse training data (for each person only two pairs are required), PRDC imposes that each person in the training set is involved in at least one positive pair *and* one negative pair. It is therefore not possible to set $n^- = 10$ despite negative examples can be very easily generated. PCCA, for $n^- = 1$, uses equivalently sparse training data while not imposing any constraint on the pairs, which is much more flexible and allows the use of more negative pairs.

[8] has recently proposed LMNN-R , inspired from LMNN [19] with an additional rejection mechanism. As the paper reports only curves and no tables, quantitative comparison is more difficult. In Fig. 5, we have reproduced their results and give CMC scores obtained by LMNN-R [8] for ranks up to 100 with $p = 316$. We also report the scores for PCCA (with RBF χ^2 and $n^- = 10$), PRDC [33] and MCC [10]. PCCA gives similar or slightly better than LMNN-R while requiring much less training data. Indeed, please note that LMNN-R, as MCC, requires to have the identity of each person of the training set while PCCA uses only pairs with same/different annotation.

Finally, Fig. 6 reports CMC of PCCA as a function of the number of negative pairs used for training, both for $p = 316$ and $p = 532$ ($r = 20$). The error bars represent standard error on the mean value over the 10 folds. The scores quickly increase to reach a maximum around $n^- = 7$ and then saturates for $p = 316$ while dropping after $k = 10$ for $p = 532$. From this figure we see that only a very small amount of training data is enough to learn the metrics.

5. Conclusions

We have presented a new method to learn a low-dimensional mapping in which distances between data points complies with a set of sparse training pairwise constraints.

Unlike existing methods, PCCA does not require additional assumptions on the structure of the data or the constraints and can handle natively high dimensional input spaces.

Experiments performed on the Labeled Faces in the Wild and the Viewpoint Independent Person Re-identification datasets proved that PCCA exhibits excellent generalization properties even in the case of sparse high dimensional data and that it performs better than clustering methods requiring the full annotation of the training set. State-of-the-art performances are achieved thanks to the strong regularization induced by the low dimensional projection and the robust underlain probabilistic model.

Aknowledgement. This work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation and by the ANR, grant reference ANR-08-SECU-008-01/SCARFACE.

References

[1] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *J. Mach. Learn. Res.*, 6:937–965, 2005. 1

[2] S. Burer and R. D. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming (series B)*, 95:2003, 2001. 3

[3] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. In *CVPR*, 2010. 5

[4] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19:1155–1178, 2007. 4

[5] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *J. Mach. Learn. Res.*, 11:1109–1135, 2010. 1, 2

[6] T. Cox and M. Cox. *Multidimensional scaling*. Number v. 1. Chapman & Hall/CRC, 2001. 2

[7] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML*, 2007. 1, 2, 3

[8] M. Dikmen, E. Akbas, T. S. Huang, and N. Ahuja. Pedestrian recognition with a learned metric. In *ACCV*, 2010. 2, 6

[9] N. Gheissari, T. B. Sebastian, and R. Hartley. Person reidentification using spatiotemporal appearance. *CVPR*. 1

[10] A. Globerson and S. Roweis. Metric learning by collapsing classes. *NIPS*, 2006. 1, 2, 3, 6

[11] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *NIPS*, 2004. 1, 2, 3

[12] D. Gray, S. Brennan, and H. Tao. Evaluating appearance models for recognition, re-acquisition and tracking. In *PETS*, 2007. 1, 4, 6

[13] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*, 2008. 2

[14] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *ICCV*, 2009. 1, 2, 3, 5

[15] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. J. Maybank. Principal axis-based correspondence between multiple cameras for people tracking. *PAMI*, 28(4):663–671, 2006. 1

[16] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007. 1, 2, 4

[17] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351. 3

[18] B. Kulis, M. Sustik, and I. Dhillon. Learning low-rank kernel matrices. In *ICML*, 2006. 2

[19] M. P. Kumar, P. H. S. Torr, and A. Zisserman. An invariant large margin nearest neighbour classifier. In *ICCV*, 2007. 1, 2, 6

[20] J. T. Kwok and I. W. Tsang. Learning with idealized kernels. In *ICML*, 2003. 2

[21] S. Mahamud and M. Hebert. The optimal distance measure for object detection. In *CVPR*, 2003. 1

[22] H. V. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *ACCV*, 2011. 5

[23] P. Phillips, H. Wechsler, J. Huang, and P. Rauss. The feret database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998. 1

[24] B. Prosser, W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by support vector ranking. In *BMVC*, 2010. 2

[25] Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *BMVC*, 2009. 2

[26] J. B. Tenenbaum, V. Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000. 2

[27] L. Torresani and K. C. Lee. Large Margin Component Analysis. In *NIPS*. Cambridge, MA, 2007. 1, 2, 3

[28] I. W. Tsang, J. T. Kwok, and C. W. Bay. Distance metric learning with kernels. In *ICANN*, 2003. 3

[29] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *PAMI*, 2011 (to appear). 5

[30] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10:207–244, 2009. 1, 2

[31] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS*, 2002. 1, 2

[32] T. Zhang and F. Oles. Text categorization based on regularized linear classification methods. *Information Retrieval*, 4:5–31, 2001. 3

[33] W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by probabilistic relative distance comparison. In *CVPR*, 2011. 1, 2, 5, 6