MLBoost Revisited: A Faster Metric Learning Algorithm for Identity-Based Face Retrieval

Romain Negrel, Alexis Lechervy, Frédéric Jurie

MLBoost Revisited: A Faster Metric Learning Algorithm for Identity-Based Face Retrieval

Romain Negrel
romain.negrel@unicaen.fr
Alexis Lechervy
alexis.lechervy@unicaen.fr
Frederic Jurie
frederic.jurie@unicaen.fr

Normandie Univ, UNICAEN,
ENSICAEN, CNRS
France

Abstract

This paper addresses the question of metric learning, *i.e.* the learning of a dissimilarity function from a set of similar/dissimilar example pairs. This domain plays an important role in many machine learning applications such as those related to face recognition or face retrieval. More specifically, this paper builds on the recent MLBoost method proposed by Negrel *et al.* [25]. MLBoost has been shown to perform very well for face retrieval tasks, but this algorithm relies on the computation of a weak metric which is very time consuming. This paper demonstrates how, by introducing sparsity into the weak projectors, the convergence time can be reduced up to a factor of $10\times$ compared to MLBoost, without any performance loss. The paper also introduces an explicit way to control the rank of the so-obtained metrics, allowing to fix in advance the dimension of the (projected) feature space. The proposed ideas are experimentally validated on a face retrieval task with three different signatures.

1 Introduction

This paper focuses on the task of *identity-based face retrieval*. This has been a very dynamic research field over the past five years, raising many interesting challenges and producing a variety of interesting methods. Identity-based face retrieval heavily depends on the quality of the similarity function used to compare faces. Instead of using standard or handcrafted similarity functions, the most popular way to address this problem is to learn adapted metrics from sets of similar/dissimilar example pairs. It is usually equivalent to projecting face signatures into an adapted (possibly low-dimensional) space in which similarity can be measured with the Euclidean distance. For large scale applications, the dimensionality of this subspace should be as small as possible to limit the storage requirements, while the projection should also be fast to compute. Interestingly, the Euclidean metric fulfill the second requirement, which explains why producing face representations adapted to the Euclidean metric is interesting. However, such representations are usually of large size. Several methods have been proposed to learn projection matrices reducing the size of the signatures while preserving the performance. This paper builds on such approaches.

More precisely, this paper proposes two improvements over MLBoost – MLBoost [25] is a supervised metric leaning method based on boosting – one of the state-of-the-art Mahalanobis metric leaning methods. These two contributions are:

- The introduction of a new way to compute the weak metrics at a lower computational cost;
- The introduction of a new approach to control the rank of the learned metrics, allowing to fix the dimensions of the low-dimensional space in which the images are represented.

The rest of the paper is as follows: after reviewing some metric learning techniques in Section 2 and giving more details on MLBoost [25] in Section 3, the proposed contributions are presented in Section 4. Section 5 compares the proposed method with state-of-the-art competitors and shows its benefits.

2 Related Works

During the last decades, many Metric Learning (ML) approaches have emerged and have been used in diverse applications such as tracking, image retrieval, face verification, person re-identification, *etc*. ML also plays an important role in many machine learning, pattern recognition or data mining techniques as learning metrics from data is usually better than designing hand crafted metrics. In practice, not only should the metric be good in terms of performance, but also it has to be fast, not memory demanding and computationally scalable.

The literature on ML is too vast to be fully covered here, and the interested reader is referred to the recent book of Bellet *et al*. [2]. We can, however, mention a few of the most notable approaches such as: DDML [15], RBML [20], Structural ML [41], PCCA [22], rPCCA [40], LMNN [37], LDML [14], ITML [11], KISSME [17], RS-KISSME [34], SML [7], MLBoost [25]. Most of these supervised approaches learn a distance or a similarity function based on the Mahalanobis distance. The Mahalanobis distance between $\mathbf{x}_i$ and $\mathbf{x}_j \in \mathbb{R}^D$ is defined as:

$$D_{\mathbf{W}}(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{W}(\mathbf{x}_i - \mathbf{x}_j), \tag{1}$$

where $(\mathbf{x}_i,\mathbf{x}_j)$ denotes the pair of samples to compare and $\mathbf{W} \in \mathcal{M}_{D\times D}$ is a positive semi-definite matrix. The seminal work of [39] estimated $\mathbf{W}$ by solving a convex quadratic programming problem, by satisfying constraints defined by some given training pairs.

However guaranteeing the positive semi-definiteness of $\mathbf{W}$ is computationally expensive. To reduce this cost, several works suggested to factorize $\mathbf{W}$ as $\mathbf{W} = \mathbf{L}\mathbf{L}^\top$ with $\mathbf{L} \in \mathcal{M}_{D\times d}$. In this case, $\mathbf{W}$ is by construction a positive semi-definite matrix and $\mathbf{L}$ defines an implicit projection matrix ($\mathbf{y}_i = \mathbf{L}^\top \mathbf{x}_i$). thus, it is possible to impose rank constraints to regularize the model and learn a smaller feature space ($d << D$).

In the following, we denote by $(\mathbf{p}_{1i},\mathbf{p}_{2i}) \in \mathcal{P}$ the set of positive pairs (two samples belonging to the same class) and by $(\mathbf{n}_{1j},\mathbf{n}_{2j}) \in \mathcal{N}$ the set of negative pairs (two samples belonging to different classes). We also write $\mathrm{D}_{\mathbf{L}}$ instead of $D_{\mathbf{L}\mathbf{L}^\top}$, for simplicity.

In [3], Bellman highlighted the phenomenon called the *curse of dimensionality*: when the dimensionality of the feature space increases, the data representation becomes sparse. In general, this sparsity is problematic, in particular for any method that requires statistical significance. This is why a lot of ML techniques have proposed to reduce the dimension of the data space [10, 12, 33, 35]. For example, [23] and [24] proposed different (unsupervised and supervised) methods to reduce the dimension of large-size descriptors (from thousands to millions dimensions). PCCA [22, 40] proposed to learn a matrix $\mathbf{L}$ used to project the signatures into a low-dimensional space where the distance between similar pairs are smaller than those of dissimilar pairs. To do this, the authors suggested to solve the following optimization problem:

$$\arg\min_{\mathbf{L}} \sum_{\mathcal{P}} \ell_\beta \left(\mathrm{D}_{\mathbf{L}} \left(\mathbf{p}_{1i}, \mathbf{p}_{2i} \right) - 1 \right) + \sum_{\mathcal{N}} \ell_\beta \left(1 - \mathrm{D}_{\mathbf{L}} \left(\mathbf{n}_{1i}, \mathbf{n}_{2i} \right) \right) + \lambda \left\| \mathbf{L} \right\|_F^2, \tag{2}$$

with $\ell_\beta(x) = \frac{1}{\beta} \log\left(1 + \exp\left(\beta x\right)\right)$, where β and λ are two hyper-parameters.

Tuning these hyper-parameters is not an easy task and is application dependent. Interestingly, several methods don't use any hyper-parameters. This is the case of the KissMe method, introduced in [17], formulating the learning problem as a likelihood-test between two Gaussian distributions (one for similar and one for dissimilar pairs). Consequently, it is easy to compute $\mathbf{W}$ such as:

$$\mathbf{W} = \Sigma_{\mathcal{P}}^{-1} - \Sigma_{\mathcal{N}}^{-1}, \tag{3}$$

with $\Sigma_{\mathcal{P}} = \sum_{\mathcal{P}} (\mathbf{p}_{1i} - \mathbf{p}_{2i})(\mathbf{p}_{1i} - \mathbf{p}_{2i})^\top$ and $\Sigma_{\mathcal{N}} = \sum_{\mathcal{N}} (\mathbf{n}_{1j} - \mathbf{n}_{2j})(\mathbf{n}_{1j} - \mathbf{n}_{2j})^\top$. Despite this method is very fast and is not requiring any hyper-parameters, it cannot guarantee that the metric is positive-definite (*i.e.*, distances are not necessarily positive). The authors proposed to project $\mathbf{W}$ on the cone of positive semi-definite matrices when $D_{\mathbf{W}}$ is not exactly a metric.

Recently, several researchers investigated the use of Boosting algorithms [30] for ML. Boosting algorithms are interesting as they do not have, in general, any hyper-parameters and are not prone to overfitting [29]. Strong metrics can be obtained by combining several weak metrics (generally rank-one metrics) to solve an optimization problem with triplet-wise constraints [6, 21, 31, 32]. Negrel *et al.* [25] introduced MLBoost, showing how to learn a boosted metric using pairwise constraints only, in a fast and scalable way.

Several Boosting methods have been developed with computational and storage efficiency in mind. A first strategy is to reduce the computational cost for learning weak learners. This is rather natural as, in boosting, it is better to have simple weak classifiers; a good example is the Haar basis functions introduced in [36]. A second strategy consists in evaluating less or using less weak learners. In [36], a cascade approach is introduced to reduce the average number of weak classifiers evaluated during the test stage. FloatBoost [19] uses a backtracking mechanism: in the training phase, after each iteration of AdaBoost, some weak classifiers are removed. As the number of weak classifiers selected does not change, the time required to compute the metric is controlled. Furthermore, removing some weak classifiers allows to remove the bad ones, improving both convergence and performance. Finally, using a fixed number of weak learners [1, 13] or updating the weak learners after their selection [27] have been studied a lot in the tracking literature.

In this paper, we propose two contributions for reducing the learning cost. First, we propose a novel fast weak ML algorithm; second, we add rank constraints on the strong metric, allowing us to fix the maximal dimension of the so-produced feature space, even when the number of boosting iteration increases.

Algorithm 1 Efficient MLBoost implementation

procedure MLBOOST($\mathbf{X}, \mathcal{P}, \mathcal{N}, itersMax$)
  $t \leftarrow 1$
  $\mathbf{L}^{(1)} \leftarrow \varnothing$
  Initialize weights: $\forall i,\ u_i^{(1)} = 1/|\mathcal{P}|$; $\forall j,\ v_j^{(1)} = 1/|\mathcal{N}|$.
  repeat
    Compute weak metric $\mathbf{z}^{(t)}$ with equation (4).
    Choose the best $\alpha^{(t)}$ with equation (6).
    if $\alpha^{(t)} \leq 0$ **then**
      break
    $\mathbf{L}^{(t+1)} \leftarrow \left[\mathbf{L}^{(t)}, \sqrt{\alpha^{(t)}}\mathbf{z}^{(t)}\right]$.
    Update weights $u_i^{(t+1)}$ and $v_j^{(t+1)}$ with equations (7).
  until $t < itersMax$
  return L

3 Boosted Metric Learning (MLBoost)

This section briefly summarizes the recent MLBoost approach – an efficient technique allowing to learn metrics with Boosting – such as introduced in [25]. MLBoost learns a decomposition of a Mahalanobis-based metric $\mathbf{L}$. Like other boosting techniques, MLBoost combines the weak learners obtained at each iteration to form a strong classifier.

At the beginning, all the pairs are initialized with the identical weights ($u_i^{(1)} = 1/|\mathcal{P}|$ for positive pairs and $v_j^{(1)} = 1/|\mathcal{N}|$ for negative pairs). The weak metric $D_{\mathbf{z}^{(t)}}$ is then obtained by solving the following optimization problem:

$$\begin{aligned} \mathbf{z}^{(t)} = & \arg\max_{\mathbf{z}} \mathbf{z}^\top \mathbf{A}^{(t)} \mathbf{z}, \\ & \text{s.t. } \|\mathbf{z}\|_2 = 1, \text{with} \end{aligned} \tag{4}$$

$$\mathbf{A}^{(t)} = \sum_{\mathcal{N}} v_j^{(t)} \left((\mathbf{n}_{1j} - \mathbf{n}_{2j})(\mathbf{n}_{1j} - \mathbf{n}_{2j})^\top \right) - \sum_{\mathcal{P}} u_i^{(t)} \left((\mathbf{p}_{1i} - \mathbf{p}_{2i})(\mathbf{p}_{1i} - \mathbf{p}_{2i})^\top \right). \tag{5}$$

We note that solving problem (4) is equivalent to the computation of the eigenvector corresponding to the largest eigenvalue of $\mathbf{A}^{(t)}$. Once the weak metric $D_{\mathbf{z}^{(t)}}$ is computed, the algorithm chooses the best weights $\alpha^{(t)}$ by solving the following problem:

$$\alpha^{(t)} = \arg\min_{\alpha} \left(\sum_{\mathcal{P}} u_i^{(t)} \mathrm{e}^{\alpha \left(\mathrm{D}_{\mathbf{z}^{(t)}}(\mathbf{p}_{1i}, \mathbf{p}_{2i}) \right)} \right) \left(\sum_{\mathcal{N}} v_j^{(t)} \mathrm{e}^{-\alpha \left(\mathrm{D}_{\mathbf{z}^{(t)}}(\mathbf{n}_{1j}, \mathbf{n}_{2j}) \right)} \right). \tag{6}$$

At the end of each iteration, the weights of the training pairs are updated by:

$$u_i^{(t+1)} = \frac{u_i^{(t)} \mathrm{e}^{\alpha^{(t)} \mathrm{D}_{\mathbf{z}^{(t)}}(\mathbf{p}_{1i}, \mathbf{p}_{2i})}}{w_{\mathcal{P}}^{(t)}}, \forall i \qquad v_j^{(t+1)} = \frac{v_j^{(t)} \mathrm{e}^{-\alpha^{(t)} \mathrm{D}_{\mathbf{z}^{(t)}}(\mathbf{n}_{1j}, \mathbf{n}_{2j})}}{w_{\mathcal{N}}^{(t)}}, \forall j. \tag{7}$$

The different steps of this algorithm are summarized in Algorithm 1.

MLBoost is robust to overfitting and is free of any hyper-parameters. However, one of its drawbacks is that the final size of the so-obtained feature space can be very large. Furthermore, computing the weak learners is very expensive.

Algorithm 2 Low cost weak metrics

procedure LCWEAKMETRIC($\mathbf{X}, \mathcal{P}, \mathcal{N}, \mathbf{u}^{(t)}, \mathbf{v}^{(t)}, J$)
    Select randomly a subset I of $J < D$ indices $I = \{i_1, \cdots, i_J\}$.
    $\mathbf{X}' \leftarrow \mathbf{X}_I$.
    Compute $\mathbf{A}^{(t)} \in \mathcal{M}_{J \times J}$ with equation (5) and $\mathbf{X}'$.
    Solve equation (4), *i.e.* compute $\mathbf{z}'^{(t)}$, the first eigenvector of $\mathbf{A}^{(t)}$ matrix.
    Create a vector of zero entries: $\mathbf{z}^{(t)} \in \mathbb{R}^W$.
    Set the value of $\mathbf{z}^{(t)}$ in indices by I: $\mathbf{z}_I^{(t)} \leftarrow \mathbf{z}'^{(t)}$.
    return $\mathbf{z}^{(t)}$

4 Faster MLBoost

Our contribution for faster MLBoost is twofold: first, we introduce a new way of building weak learners; second we propose a better way to control the rank (and consequently the dimension of the signature) of the Mahanalobis matrix. The two contributions are presented in the two following subsections.

4.1 Producing weak metrics at lower cost

As explained previously, MLBoost relies on the computation of a weak metric, which is computationally expensive. This cost depends on two parameters: the dimensionality of the input features and the numbers of positive and negative pairs. More precisely, the weak metric is computed in two steps: first, matrix $\mathbf{A}^{(t)}$ is computed using equation (5); second, the Rayleigh quotient of equation (4) is obtained by computing the first eigenvector of $\mathbf{A}^{(t)}$. These two steps have (at least) a quadratic complexity with respect to size of the signature and hence become intractable for large signatures.

In order to reduce the computational cost of the weak metric, we propose to sparsify the weak metric projectors. We do it by arbitrarily setting some of the components of the projectors to zero, allowing to consider only the dimensions of the signatures corresponding to the non-null dimensions of the projectors. These non-null components of the weak metric projectors are randomly selected and uniformly distributed. Algorithm 2 summarizes this strategy. For clarity purposes, we introduce τ, *i.e.* the ratio of non-zero dimensions defined as: $\tau = J/D$, where $J < D$ is the number of non-null components and D is the size of the descriptors. The ratio τ can be seen as the proportion of the non-null components.

The so-computed sparse weak metrics are weaker than those of [25] and more boosting iterations are necessary to reach convergence. However, in the end, the speedup of each iteration is so important that the overall learning time is drastically reduced. We can explain the overall gain by the fact that sampling only a few components reduces the time required to learn the weak classifiers in a quadratic way. On the other hand, we observe that the components are correlated, explaining why keeping only a fraction of them does not result in a strong degradation of the performance. In addition, the proposed random sampling ensures more diversity than optimally selecting the components (*e.g.* using PCA).

4.2 Explicitly Controlling the Rank of MLBoost

As discussed in the related work section (Section 2), controlling the dimensionality of the image signatures is very interesting for practical reasons. This can be done by controlling the rank of the Mahalanobis matrix. As MLBoost adds a new weak metric at each iteration, the rank of the Mahalanobis matrix is increased, iteration by iteration. The only way to control the rank is then to fix the number of boosting rounds, *e.g.* to $\mathbf{T}^{(t)} = \left[\mathbf{L}^{(t)}, \sqrt{\alpha^{(t)}}\mathbf{z}^{(t)}\right]$ which is inconsistent with the general principle of Boosting (*i.e.* the combination of lots of weak learners to obtain a strong learner).

We argue, in this paper, that a better way to control of the rank ($\mathrm{rank}(\mathbf{W}) \leq R$) consists in adding an extra projection at each iteration. This projection is done in two steps: (i) we approximate the current Mahalanobis metric by a Mahalanobis metric with a rank lower or equal to R; (ii) we compute the best weighting of the new metric before using it in the boosting process.

The current Mahalanobis metric $\mathrm{D}_{\mathbf{T}^{(t)}}(\cdot,\cdot)$ can be approximated by solving:

$$\mathbf{P}^{(t)} = \operatorname*{arg\,min}_{\mathbf{P}\in\mathcal{M}_{W\times R}} \sum_{ij} \left(\mathrm{D}_{\mathbf{T}^{(t)}}(\mathbf{x}_i,\mathbf{x}_j) - \mathrm{D}_{\mathbf{P}}(\mathbf{x}_i,\mathbf{x}_j)\right)^2, \tag{8}$$

where $\mathbf{x}_i$ denotes the training samples.

This problem (Eq. (8)) is a standard Multi-Dimensional Scaling (MDS) problem [9]. Moreover as the Mahalanobis metric (1) can be seen as a Euclidean metric in the reduced subspace, then we solve this problem easily by using a Principal Component Analysis (PCA) in the reduced subspace:

$$\mathrm{Cov}(\mathbf{Y}) = \mathbf{V}\Lambda\mathbf{V}^{\top}, \tag{9}$$

with $\mathbf{Y} = \mathbf{T}^{(t)\top}\mathbf{X}$ and $\mathbf{X}$ the matrix containing the training pairs (vectors of differences), $\mathbf{V}$ the eigenvectors of the covariance, and Λ the diagonal matrix containing eigenvalues of the covariance matrix. The optimal matrix $\mathbf{P}^{(t)}$ is computed by combining the current Mahalanobis matrix $\mathbf{T}^{(t)}$ with the R eigenvectors corresponding to the largest eigenvalues $\mathbf{V}_{\{1,\cdots,R\}}$:

$$\mathbf{P}^{(t)} = \mathbf{T}^{(t)}\mathbf{V}_{\{1,\cdots,R\}}. \tag{10}$$

In this case, $\mathbf{P}^{(t)}$ is the best R-dimensional approximation of $\mathbf{T}^{(t)}$. However, it is not possible to directly replace $\mathbf{T}^{(t)}$ by $\mathbf{P}^{(t)}$ in the next steps of MLBoost. As in the first boosting step, we need to compute the weights of the metric:

$$\mathbf{L}^{(t+1)} = \sqrt{\alpha_2^{(t)}}\mathbf{P}^{(t)}, \tag{11}$$

where $\alpha_2^{(t)}$ denotes the weights. Indeed, at the end of each boosting iteration, weighting the training pairs makes the previous weak metric performing as well as a random metric. To compute α_2, we solve (via line search) the following problem:

$$\alpha_2^{(t)} = \operatorname*{arg\,min}_{\alpha} \left(\sum_{\mathcal{P}} \mathrm{e}^{\alpha\left(\mathrm{D}_{\mathbf{P}^{(t)}}(\mathbf{p}_{1i},\mathbf{p}_{2i})\right)}\right)\left(\sum_{\mathcal{N}} \mathrm{e}^{-\alpha\left(\mathrm{D}_{\mathbf{P}^{(t)}}(\mathbf{n}_{1j},\mathbf{n}_{2j})\right)}\right). \tag{12}$$

Finally, we update the weights of the training pairs as follows:

$$u_i^{(t+1)} = \frac{\mathrm{e}^{\mathrm{D}_{\mathbf{L}^{(t+1)}}(\mathbf{p}_{1i},\mathbf{p}_{2i})}}{w_{\mathcal{P}}^{(t+1)}}, \forall i \qquad v_j^{(t+1)} = \frac{\mathrm{e}^{-\mathrm{D}_{\mathbf{L}^{(t+1)}}(\mathbf{n}_{1j},\mathbf{n}_{2j})}}{w_{\mathcal{N}}^{(t+1)}}, \forall j \tag{13}$$

with $w_{\mathcal{P}}^{(t+1)}$ and $w_{\mathcal{N}}^{(t+1)}$ the normalization factors chosen such that $\sum u_i^{(t+1)} = 1$ and $\sum v_i^{(t+1)} = 1$.

Sign.	Method	Final Dim.	n= 1	n= 10	n= 20	n= 50	n= 100
LBP	-	9860	31.9	53.7	60.5	68.8	74.7
	PCA	16	10.2	24.8	34.5	44.7	55.3
		32	16.5	34.5	44.7	55.3	66.0
		128	28.4	46.6	54.6	65.7	72.1
		512	31.2	51.5	59.6	67.4	74.7
		585	36.4	57.7	64.3	74.2	79.7
	KissMe	-	24.3	53.6	59.5	69.7	78.3
	MLBoost	585	**40.2**	**60.8**	**66.7**	**74.9**	**81.1**
AlexNet	-	4096	78.3	92.2	94.8	97.2	97.9
	PCA	16	53.7	82.7	89.1	94.3	96.7
		32	70.7	90.5	92.4	96.2	97.6
		128	75.7	91.7	94.6	96.9	98.1
		383	78.7	92.7	94.8	97.4	**98.3**
		512	78.7	92.4	94.8	97.4	**98.3**
	KissMe	-	76.6	92.4	95.3	96.9	97.8
	MLBoost	383	**81.3**	**93.9**	**96.0**	**97.9**	**98.3**
VGG-Face	-	4096	89.6	96.9	97.4	**98.1**	98.3
	PCA	16	75.7	91.3	93.9	96.2	97.6
		32	87.7	95.0	95.7	97.2	97.9
		128	91.3	96.5	97.2	97.6	98.3
		191	**91.5**	96.5	97.2	97.6	98.6
		512	91.3	96.7	96.9	97.6	98.3
	KissMe	-	90.1	96.7	97.2	97.6	**98.8**
	MLBoost	191	**91.5**	**97.2**	**97.9**	**98.1**	98.3

Table 1: Baseline performance of 3 types of descriptors with (i) Euclidean metric (ii) Euclidean metric after PCA reduction (iii) KissMe [17] (iv) MLBoost [25].

Sign.	Final Dim.	n= 1	n= 10	n= 20	n= 50	n= 100
LBP	1226	41.8	61.4	68.6	75.4	80.9
AlexNet	1128	81.8	94.3	95.7	97.9	98.1
VGG-Face	538	**91.7**	**96.5**	**97.4**	**98.3**	**98.8**

Table 2: Performance of MLBoost with low-cost weak metrics ($\tau = 5\%$), for the three types of signatures.

Sign.	Final Dim.	n= 1	n= 10	n= 20	n= 50	n= 100
LBP	16	18.7	43.5	52.7	64.3	74.0
	32	31.4	57.0	63.1	72.1	77.3
	128	36.4	54.8	62.9	71.6	77.5
	512	**38.5**	**58.6**	**63.6**	**74.0**	**79.2**
AlexNet	16	60.0	97.9	91.3	93.4	94.8
	32	73.5	92.2	95.3	**97.6**	98.1
	128	78.0	93.9	**95.7**	**97.6**	97.9
	512	**79.0**	**94.1**	**95.7**	**97.6**	**98.3**
VGG-Face	16	82.0	94.1	96.7	97.6	98.6
	32	89.4	96.2	97.4	98.1	98.6
	128	90.8	95.7	97.2	98.1	**98.8**
	512	**92.4**	**96.7**	**97.6**	**98.3**	98.6

Table 3: Performance of MLBoost with low-cost weak metrics ($\tau = 5\%$) and rank constraints ($R \in \{16, 32, 128, 512\}$).

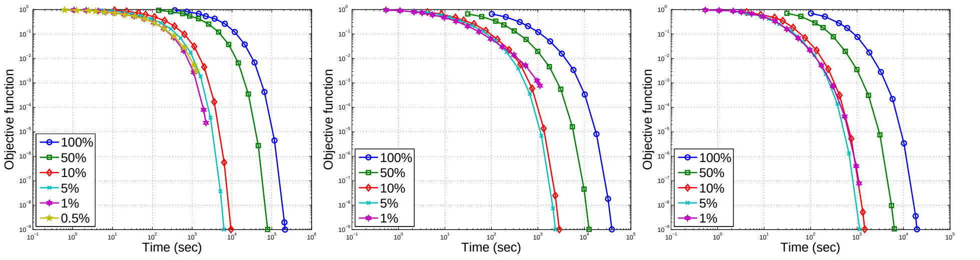
5 Experiments

The two contributions of this paper are experimentally evaluated on the identity-based face retrieval task, *i.e.* given a face query, the objective is to find a face of the same person in a set of known-identity face images and hence predict the identity of the query face. The criterion used to evaluate the performance is the one used in [4, 25], *i.e.*, the mean k-call@n (such as defined in [8]), with $k = 1$. for $n \in \{1, 10, 20, 50, 100\}$.

Datasets and learning pairs. We use the aligned version [38] of the Labeled Faces in the Wild (LFW) database by Huang *et al.* [16]. It contains more than 13,000 images of over 4,000 different persons. In our experiments, we use the same set of images/queries as [4, 25]. Only the identities having at least five examples are used; the others are not used during the learning of metrics nor during their evaluations. This results in a subset of 5,985 images of 423 different persons. The query set is composed of one image of each identity while the training set contains the remaining images. To learn the metrics, we build a set of similar pairs and a set of dissimilar pairs in such a way that all the identities are used equally.

Image descriptions. We evaluate the methods with three types of image signatures: (i) LBP [26]: we use the same signatures as in [4, 25] (signatures of 9860 dimensions). (ii) AlexNet descriptors [18]: we use the same descriptors as Bhattarai *et al.* [5](signatures of 4096 dimensions). (iii) VGG-Face CNN descriptors [28]: we use the publicly available source code[1] (signatures of 4096 dimensions).

[1] http://www.robots.ox.ac.uk/~vgg/software/vgg_face/



(a) LBP (b) AlexNet (c) VGG-Face

Figure 1: Objective as the function of the accumulated time spent on learning the weak metrics, for different values of τ. τ is the parameter fixing the ratio of non-zeros dimensions in the low-cost MLBoost weak metric.

MLBoost learning parameters. We learn the metrics using $2^{17} \approx 131,000$ positive and negative examples pairs. Boosting is stopped when the objective function is lower than 10^{-9} or the maximum number of iterations is reached, *i.e.*, 2048 iterations. To evaluate the metric learned with MLBoost, we project the signatures on the projectors $\mathbf{y}_i = \mathbf{L}^{*\top}\mathbf{x}_i$ and we normalize (ℓ_2) the reduced signatures $\mathbf{y}'_i = \mathbf{y}_i/\|\mathbf{y}_i\|$. We then use the Euclidean metric to compare the queries with the images of the test set.

Baseline results. We use as a baseline the performance obtained with: (i) raw signatures (without metric learning) / Euclidean distance; (ii) signatures reduced by PCA; (iii) KissMe [17]; (iv) MLBoost [25]. The results are reported in Table 1, which compares the performance obtained with the three types of signatures (LBP, AlexNet and VGG-Face). The performances are given in terms of the percentage of the mean 1-call@n. To learn the metric with KissMe, we use the signatures reduced to 128 dimensions with PCA, and we use only $2^{14} \approx 16000$ positive and negative pairs (setting giving the best performance).

We can see that the recent CNN signatures provide much better performance than LBP. We also note that for AlexNet and VGG-Face, PCA can improve the performance (for 128-d or more projections). We can finally see that the metric learned with MLBoost constantly improves the performance, for all types of signatures.

5.1 Low cost weak metric performance

To analyze the effects of our low-cost weak metric on the convergence speed and metric performance, we learn the metrics for the different types of signatures and for various ratios of non-zeros dimensions $\tau \in \{100\%, 50\%, 10\%, 5\%, 1\%\}$. We note that $\tau = 100\%$ is equivalent to the original MLBoost of [25]. Figures 1(a), 1(b) and 1(c) illustrate the convergence of the algorithm for the different ratios of non-zeros components. The vertical axis corresponds to the objective function while the horizontal axis corresponds to the accumulated time spent on computing the weak metrics during boosting. We see that for any type of signatures, the overall time spent in computing the weak metrics before the objective function reaches 10^{-9} is significantly reduced. For a ratio of 5% of non-zeros dimensions, the total time is at least divided by a factor of 10. Table 2 gives the performance of the metrics learned with our low-cost weak metric with 5% of non-zeros dimensions. The performance is very similar to those of the weak metric proposed in [25] (see Table 1). However, the dimension of the final

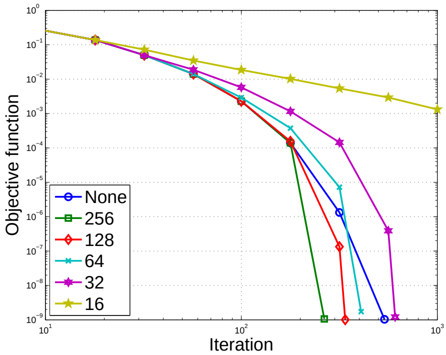


Figure 2: Effect of the rank constraints on MLBoost as a function of the number of iterations

signature is larger, due to the larger number of iterations needed reach convergence.

5.2 Adding Rank Constraints

In this section, we focus on the evaluation of our second contribution, *i.e.* the method proposed to limit the rank of the Mahalanobis matrix. We perform these experiments with our low-cost week metric with 5% of non-null components ($\tau = 0.05$), for the following rank constraint: $R \in \{16, 32, 64, 128, 256, 512\}$. Figure 2 illustrates the convergence of the algorithm (using LBP signatures) for the different rank constraints. The vertical axis corresponds to the objective function while the horizontal axis corresponds to the number of boosting iterations. The blue curve shows the convergence of MLBoost without rank constraints. We see that for strong rank constraints (*e.g.*, $R = 16$), the convergence speed is reduced. However, for $R = 64$, $R = 128$ and $R = 256$, we note that we need fewer iterations to converge than without the rank constraint. We report, in Table 3, the performance given by the metrics learned with MLBoost combined with our low-cost weak metric and the rank constraint. We see that the performance increases with R. In comparison to the original MLBoost (see Table 1), and for any type of signature, we always obtain better performance. The conclusion is that not only is the proposed method faster, but it is also better in terms of performance.

6 Conclusions

This paper introduces two improvements to the state-of-the-art MLBoost method [25]. The first one addresses the prohibitive computational cost required to learn weak metrics in the presence of high-dimensional signatures. The second contribution allows us to limit the rank of the Mahalanobis matrix and, thus, to fix the dimension of the final signatures. The proposed experimental validation not only show a more than 10× speedup but also a significant improvement of the performance. In addition, the paper shows that the size of the final signature can significantly be reduced with only a small loss in performance.

Acknowledgments

The authors acknowledge the support of the French Agence Nationale de la Recherche (ANR), under grant ANR-12-SECU-0005 (project PHYSIONOMIE).

References

[1] Shai Avidan. Ensemble tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):261–271, February 2007. ISSN 0162-8828.

[2] Aurélien Bellet, Amaury Habrard, and Marc Sebban. *Metric Learning*. Morgan & Claypool Publishers, 2015.

[3] Richard E Bellman. *Adaptive control processes: a guided tour*. Princeton university press, 2015.

[4] Binod Bhattarai, Gaurav Sharma, Frederic Jurie, and Patrick Pérez. Some faces are more equal than others: Hierarchical organization for accurate and efficient large-scale identity-based face retrieval. In *European Conference on Computer Vision (ECCV) Workshops*, pages 1–13, 2014.

[5] Binod Bhattarai, Gaurav Sharma, and Frédéric Jurie. Cp-mtml: Coupled projection multi-task metric learning for large scale face retrieval. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] Jinbo Bi, Dijia Wu, Le Lu, Meizhu Liu, Yimo Tao, and Matthias Wolf. AdaBoost on low-rank psd matrices for metric learning. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2617–2624. IEEE, 2011.

[7] Qiong Cao, Yiming Ying, and Peng Li. Similarity metric learning for face recognition. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2408–2415. IEEE, 2013.

[8] Harr Chen and David R Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 429–436. ACM, 2006.

[9] Michael A.A. Cox and Trevor F. Cox. *Handbook of Data Visualization*, chapter Multidimensional Scaling, pages 315–347. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[10] John P. Cunningham and Zoubin Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research*, 16:2859–2900, 2015.

[11] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 209–216, New York, NY, USA, 2007. ACM.

[12] Imola K. Fodor. A survey of dimension reduction techniques. Technical report, Technical report, Lawrence Livermore National Laboratory, 2002.

[13] Helmut Grabner and Horst Bischof. On-line boosting and vision. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '06, pages 260–267, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2597-0.

[14] Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek, and Cordelia Schmid. Face recognition from caption-based supervision. *International Journal of Computer Vision*, 96(1):64–82, 2012.

[15] Junlin Hu, Jiwen Lu, and Yap P. P. Tan. Discriminative deep metric learning for face verification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1875–1882, 2014.

[16] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[17] Martin Koestinger, Martin Hirzer, Paul Wohlhart, Peter M Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2288–2295. IEEE, 2012.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] Stan Z Li and ZhenQiu Zhang. Floatboost learning and statistical face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1112–1123, 2004.

[20] Venice E. Lion, Jiwen Lu, and Yongxin Ge. Regularized bayesian metric learning for person re-identification. In *ECCV Workshop on Visual Surveillance and Re-Identification*, 10 2014.

[21] Meizhu Liu and Baba C. Vemuri. A robust and efficient doubly regularized metric learning approach. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV*, pages 646–659, 2012.

[22] Alexis Mignon and Frédéric Jurie. Pcca: A new approach for distance learning from sparse pairwise constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2666–2672. IEEE, 2012.

[23] Romain Negrel, David Picard, and Philippe-Henri Gosselin. Web scale image retrieval using compact tensor aggregation of visual descriptors. *IEEE MultiMedia*, 20(3):24–33, March 2013.

[24] Romain Negrel, David Picard, and Philippe-Henri Gosselin. Dimensionality reduction of visual features using sparse projectors for content-based image retrieval. In *IEEE International Conference on Image Processing*, pages 2192–2196, Paris, France, October 2014.

[25] Romain Negrel, Alexis Lechervy, and Frederic Jurie. Boosted metric learning for efficient identity-based face retrieval. In *British Machine Vision Conference*, volume 13, pages 1007–1036, 2015.

[26] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[27] Toufiq Parag, Fatih Porikli, and Ahmed Elgammal. Boosting adaptive linear weak classifiers for online learning and tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[28] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[29] Lev Reyzin and Robert E. Schapire. How boosting the margin can also boost classifier complexity. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 753–760, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2.

[30] Robert E Schapire and Yoav Freund. *Boosting: Foundations and algorithms*. MIT press, 2014. ISBN 0262526034.

[31] Chunhua Shen, Alan Welsh, and Lei Wang. Psdboost: Matrix-generation linear programming for positive semidefinite matrices learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1473–1480. Curran Associates, Inc., 2009.

[32] Chunhua Shen, Junae Kim, Lei Wang, and Anton Van Den Hengel. Positive semidefinite metric learning using boosting-like algorithms. *The Journal of Machine Learning Research*, 13(1):1007–1036, 2012.

[33] C. O. S. Sorzano, J. Vargas, and A. Pascual Montano. A survey of dimensionality reduction techniques, 2014.

[34] Dapeng Tao, Lianwen Jin, Yongfei Wang, Yuan Yuan, and Xuelong Li. Person re-identification by regularized smoothing kiss metric learning. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(10):1675–1685, 2013.

[35] Laurens.J.P. van der Maaten, Eric. O. Postma, and H. Jaap van den Herik. Dimensionality reduction: A comparative review. Technical report, Technical report, Tilburg University, 2009.

[36] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, pages 511–518, 2001.

[37] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[38] Lior Wolf, Tal Hassner, and Yaniv Taigman. Similarity scores based on background samples. In *Computer Vision–ACCV 2009*, pages 88–97. Springer, 2009.

[39] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance Metric Learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, Vancouver, Bristish Columbia, December 2002.

[40] Fei Xiong, Mengran Gou, Octavia Camps, and Mario Sznaier. Person re-identification using kernel-based metric learning methods. In *Computer Vision–ECCV 2014*, pages 1–16. Springer, 2014.

[41] Gang Yuan, Zhaoxiang Zhang, and Yunhong Wang. Enhancing person re-identification by robust structural metric learning. In *Image and Graphics (ICIG), 2013 Seventh International Conference on*, pages 453–458. IEEE, 2013.