

Boosted Metric Learning for Efficient Identity-Based Face Retrieval

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

GREYC, CNRS UMR 6072
Université de Caen Basse-Normandie
France

The performance of many visual recognition algorithms – *e.g.* face verification, person re-identification – heavily depends on the metric used to measure the similarity between input data. Recent works such as have shown how interesting it is to learn an optimal metric for a particular task using Metric Learning (ML). Most approaches learn a Mahalanobis metric based on an objective function whose constraints comes either from a set of labelled examples or, more frequently, from sets of positive (same class) and negative (different class) pairs.

In the recent works that have used metric learning (ML), the works of Shen *et al.* [4] and Bi *et al.* [2], introducing algorithms based on Boosting approach, deserves a particular attention due to the interesting properties they offers: i) they are efficient and scalable, as no semidefinite programming is required, at each iteration only the largest eigenvalue and corresponding eigenvectors are needed, ii) Like AdaBoost, they don't have any parameter to tune and is easy to implement as no sophisticated optimization techniques are involved. It hence contrasts with most of the commonly used ML methods for which hyper-parameters, often introduced for regularization purpose, have to be adjusted by cross-validation.

However, one strong limitation of these approaches [2, 4] is that it requires having triplets for learning the metric *i.e.* constraints expressed under the form $D(x_a, x_b) < D(x_a, x_c)$ where x_a, x_b, x_c are three input vector for which the label is known: positive and negative pairs within a constraint do have to share a common vector. This is a limitation as for most of the verification task only training pairs are available (*e.g.* for person re-identification on the Viper dataset only pairs of same/different persons are provided for training, and thus using such triplets is not possible).

One of the key contribution of this paper is to propose a metric learning approach based on boosting allowing to use pairs of points for training *i.e.* which can use constraints such as $D(x_a, x_b) < D(x_c, x_d)$ while keeping the good properties of [4] (scalability, simplicity, no parameters, *etc.*). Our approach is based on the ranking algorithm RankBoost [3], known to offer 3 particularly interesting features: i) no hyperparameter, ii) great robustness to overfitting (explained with a standard VC-dimensional analysis techniques), and iii) a computational trick for reducing the complexity of the learning step. In the following, we show how to build on RankBoost [3] to efficiently address this metric learning problem.

Moreover, we propose a method for building hierarchical face representations for efficient *identity face retrieval task*. The main idea is to use a ML algorithm for building a tree structured representation of a large set of faces, such that identical-identity faces belongs to the same cluster at each tree level. The hierarchical clustering is built recursively: at each level, a new metric is learned using the faces belonging to a particular tree node which is then used to locally cluster the faces and to create the sub-branches.

The retrieval for a new query face is done by visiting the tree from top to bottom until a leaf is reached; if the current is a non-leaf node, the face is projected into the corresponding subspace and compared with its centroids, which allows to move to the closest child node. When a leaf is reached, the face is compared with all the faces contained in the leaf using the corresponding local metric. The search cost is then equal to the cost of traverses of the tree, plus the cost of faces comparison in the final node. This cost is much lower than of the cost to compared all the faces of the database.

Our method is a extension of the method proposed by Bhattarai *et al.* [1]. We introduced a new method to select the ML training samples in the tree node in order to keep easily the information learned in the metric of parent nodes.

The proposed approach is validated on the *identity face retrieval task*, such as defined by Bhattarai *et al.* [1]. This task consists in retrieving the face(s) of the same person than a query face, from a large database

Method	Dim.	No distractor			+100.000 distractors			+1.000.000 distractors		
		n=1	n=10	n=20	n=1	n=10	n=20	n=1	n=10	n=20
Unreduced	9860	31.9	53.7	60.5	31.4	52.7	59.3	31	50.6	57.2
MLBoost	32	30.7	53.7	62.2	26.2	43.0	48.7	21.5	34.5	38.5
PCCA [1]	32	-	32.1	41.3	-	23.6	28.3	-	13.4	18.2
MLBoost-Tree-d3-Global	32	27.9	50.4	58.6	24.1	39.5	46.1	20.1	31.4	36.2
MLBoost-Tree-d3	32	29.6	54.6	62.2	23.6	39.5	45.2	17.7	31.0	35.2
PCCA-Tree-d3 [1]	32	-	33.3	40.9	-	27.4	32.2	-	18.5	23.9
MLBoost-Tree-d4-Global	32	27.0	49.2	56.7	22.7	37.4	42.3	18.9	30.5	33.6
MLBoost-Tree-d4	32	28.6	52.0	59.3	21.0	35.5	40.9	17.3	26.7	30.5
PCCA-Tree-d4 [1]	32	-	36.2	41.6	-	32.4	36.6	-	30.3	31.7

Table 1: Performance of the proposed method and comparison with PCCA [1].

of faces. The task is evaluated both in terms of scalability and accuracy. We use the Labeled Faces in the Wild (LFW) database with the evaluation protocol, the images/query and the LBP image signature than the one used by Bhattarai *et al.* [1]. The dimension of the output space to 32. The criterion used to evaluate the performance is the mean of k -call@ n , with $k = 1$. We measure the performance for several values of n , *i.e.*, $n \in \{1, 10, 20\}$.

Table 1 compares the performance of the proposed MLBoost algorithm to the performance of the state-of-the art PCCA (Pairwise Constrained Component Analysis) [1] with and without distractors and with and without tree. We restrict ourselves to a binary clustering tree of 3 and 4 levels (denoted with suffix “-d3” and “-d4”). Without the tree, the performance of the proposed approach improves over PCCA by about 20% ($n = 10$).

Searching the faces in only one leaf of the tree affects the overall performance: misclassifications that can happen at any level of the tree cannot be recovered later. This loss increases with the depth of the tree. We can observe this by comparing the performance of the global metric used without and with the tree structure (“MLBoost”) (methods whose names include “MLBoost-Tree-d*-Global”). The loss is of about 1-3% at $n = 1$.

We can also observe that the local metric (the lines denoted as “MLBoost-Tree-d*”) improves the performance of the tree-based structure for the small values of n , compensating the loss induced by the tree (without distractors). Comparing our method with the state-of-the-art method of [1] (lines denoted as “PCCA-Tree-d*”) shows that our methods can perform up to 20% better for some configurations.

In conclusion, we have proposed a ML method base on Boosting with quadrupole. Our method have show better results of the stat-of-the-are method moreover without tuning hyperparameters. Our method to build a tree for efficient *identity face retrieval task* greatly reduced the research cost while preserving the performances.

- [1] 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.
- [2] 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.
- [3] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *The Journal of machine learning research*, 4:933–969, 2003.
- [4] 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.