Learning Computationally Efficient Metrics for Large Scale Person Identification

Victor Hamer* and Pierre Dupont

UCLouvain - ICTEAM/INGI/Machine Learning Group, Place Sainte-Barbe 2, B-1348 Louvain-la-Neuve, Belgium.
*victor.hamer@uclouvain.be

Abstract. Nearest neighbor (NN) classifiers rely on a distance metric either *a priori* fixed or previously estimated through metric learning. When such metric learning occurs, a natural objective is to minimize the classification errors of the NN classifier. This learning procedure is however commonly performed without any regard to the computational efficiency of the NN classifier at test time. In this work, we propose to formulate the metric learning problem as a multi-objective trade-off between classification performance and computational efficiency at test time. This is illustrated here in the context of person identification. Specifically, a Mahalanobis metric learning scheme is cast as a convex optimization problem over a set of positive semi-definite matrices, and solved through projected gradient descent. Experimental results are presented on semi-artificial data, representative of a profile-based person identification task at a large scale ($\geq 10^6$ individuals). Our method shows a significant improvement of the search efficiency of a NN classifier, compared to standard soft-margin maximization metrics. In presence of hard time and space constraints, it leads to a drastic enhancement of the identification performance.

Keywords: Metric learning · Person identification · Large scale · Nearest neighbor search · Time efficient metrics

1 Introduction

From its emergence until now, metric learning has focused primarily on improving the classification performance of machine learning algorithms that use it. Examples are numerous in the context of nearest neighbor classifiers, for which the loss functions to be minimized on the training set only contain terms that aim at reducing the number of classification errors on test data [8,9,10,21,22,23]. One major issue with such nearest neighbor methods is the increasing decision time with the size of the model, which is essentially the number of stored training examples. Several approaches have been proposed to reduce this decision time in various contexts [2,3,5,6,14,19,24]. These techniques require a predefined metric that may strongly influence their decision speed-up. With a continuously increasing volume of available data, it becomes critical to reduce this decision time even more. We propose to include the test speed-up objective directly into the learning of an adequate metric. This approach is evaluated here in the context of large scale person identification with exact nearest neighbor methods. Metrics with a reduced decision time can be learned, but they suffer from a drop of identification performance. These competing objectives are here balanced to optimize the identification performance while satisfying time (and space) constraints. Our approach to person identification is based on generic features assumed to be representative of an individual profile. In particular, it is not specific to a computer vision methodology as a person can be represented by an arbitrary set of features (*e.g.* strings, zip code number, age, ...) not restricted to image characteristics. The concept of large scale person identification is further introduced below, together with nearest neighbor methods and how one usually restricts their decision time.

Given a large set of noisy examples, representing different individuals, the goal of person identification is to determine to which known person a new example belongs, whenever such person exists. Person identification can be seen as a particular case of extreme classification [18] where the number of labels is huge ($> 10^6$). In such a context, time and space constraints have to be imposed, both for the learning and the test decision procedures.

Formally, the input space X defines the domain from which are drawn the examples representing individuals. The set $P \subset X$ includes the examples belonging to known individuals. Without loss of generality, we assume here that P contains exactly one example per individual. The person identification problem can be tackled with a **fixed-radius nearest neighbor** (NN) algorithm. Given a query $\mathbf{q} \in X$, a *decision threshold* $\theta \in \mathbb{R}^+$ (also called *radius*), a learned distance metric $D : X \times X \to \mathbb{R}^+$ and a (large) set of stored examples P, the identification algorithm works in two steps. First, all examples nearer than θ to $\mathbf{q}$: $P_\theta = \{\mathbf{p} \in P : D(\mathbf{q}, \mathbf{p}) \leq \theta\}$ are found. Then, the label (a specific individual) of the nearest example to $\mathbf{q}$ inside P_θ is predicted. When P_θ appears to be empty, the special label *unknown* is predicted, which means that the test person (*i.e.* the query) is not recognized as someone in the known population P. In the sequel, we say that *two examples match* when their relative distance is smaller than the decision threshold θ.

Three types of errors can arise. A false rejection (FR) happens when a known person is not identified. A false acceptance (FA) occurs when an individual not belonging to the known population is wrongly identified. Finally, a false classification (FC) arises when a known individual is recognized but as somebody else. The false classification rate can be shown to be bounded by the false acceptance rate. As a consequence, learned metrics can be evaluated using the following F_1 score, based only on the false rejection rate FRR and the false acceptance rate FAR:

$$F_1 \triangleq \frac{2 * (1 - FRR) * (1 - FAR)}{(1 - FRR) + (1 - FAR)} \tag{1}$$

The larger this F_1 score the better. A smaller decision threshold θ implies a higher FRR and a lower FAR, and conversely. The threshold θ^* maximizing the F_1 score is a natural objective to be found.

The optimal threshold θ^* and the specific error rates are related to the number of known individuals $n = |P|$ (also called the *population size*) as follows. The FRR is the probability that P_θ becomes empty, $Pr(P_\theta == \emptyset)$, when the query example $\mathbf{q}$ represents a known individual. This probability $Pr(P_\theta == \emptyset)$ can be approximated[1] by $Pr(D(\mathbf{p}, \mathbf{q}) > \theta)$, where $\mathbf{p} \in P$ is the stored example representing the same individual as the test example $\mathbf{q} \in X$. Consequently, the FRR does not depend on the population size n. To the contrary, the FAR increases[2] with n. As a result, the optimal threshold θ^* decreases with the population size n. Our proposed metric learning method focuses on **large** scale person identification systems. The proportion of misclassified pairs of examples (false match/reject) is thus expected to be particularly low. Otherwise it would become impossible to identify correctly anyone among a very large population.

The first step of the decision algorithm described above requires to find all data points that are within a certain threshold θ from the test (or query) point. Such a *range query* is typical of NN methods, which generally make use of the triangular inequality to prune the search space [2,3,6,14,24]. This pruning is itself dependent on the metric used. Consequently, the time performance of NN methods could highly depend on the metric. Numerous schemes for nearest neighbor metric learning have been proposed [8,9,10,21,22,23]. However, they focus on the classification performance of the classifier rather than its computational efficiency at test time.

In this paper, a Mahalanobis [13] metric learning method is proposed for nearest neighbor algorithms balancing between identification performance and computational efficiency at decision time. We first define $d(\mathbf{x}_i, \mathbf{x}_k)$ as the distance vector of the data points $\mathbf{x}_i$ and $\mathbf{x}_k$. The attribute $d(\mathbf{x}_i, \mathbf{x}_k)_j$ is equal to the distance between the attribute j of these two data points, denoted by $x_{i,j}$ and $x_{k,j}$. A Mahalanobis metric is of the form

$$D_M(\mathbf{x}_i, \mathbf{x}_k) = \sqrt{d(\mathbf{x}_i, \mathbf{x}_k)^\top M d(\mathbf{x}_i, \mathbf{x}_k)} \qquad (2)$$

and is defined by the matrix M to be learned. A common constraint is to impose M to be PSD, denoted by $M \succeq 0$. Our method does not require the input data to be made of real vectors. Vectors of any data type are allowed as long as distance

[1] Formally, $Pr(\theta == \emptyset)$ could differ from $Pr(D(\mathbf{p}, \mathbf{q}) > \theta)$ as a non-empty P_θ might not include the example of the test individual. This situation would lead to a false classification, rather than a false rejection. Nevertheless, the chance of this situation to occur is negligible, as it is proportional to the product of FRR and FAR, which are both assumed to be (very) low in realistic settings.

[2] The probability of a FA increases with the number of known individuals $n = |P|$, as a single false match among them is enough for a FA to occur. For sufficiently small error rates, it can be shown that $FAR(n, \theta) \approx 1 - (1 - Pr_{FM}(\theta))^n \approx n * Pr_{FM}(\theta)$ with $Pr_{FM}(\theta)$ the probability of false match with θ as threshold, *i.e.* the probability that two random examples from different individuals would match.

functions between pairwise attributes exist and verify the triangular inequality. The existence of such functions, coupled with the constraint $M \succeq 0$, ensures that D_M verifies the triangular inequality.

Our focus is on learning a metric in possibly non-Euclidean metric spaces. A distance in such a space, for example the Levenshtein distance[3] [17], is generally heavier to compute than Euclidean distances. We will therefore assume that the time taken by the search for neighbors is dominated by the number of distances to compute. AESA algorithms [14,15,20] specifically try to minimize the number of such distance computations through the prior definition of a set $B \subset X$ of *prototypes*. The distance between every prototype $\mathbf{b} \in B$ and every stored point $\mathbf{p} \in P$ is pre-computed. Given a query $\mathbf{q}$ and a decision threshold θ, the distance between an example $\mathbf{p}$ and the query, $D_M(\mathbf{p}, \mathbf{q})$, needs not be computed if and only if:

$$\exists \mathbf{b} \in B : |D_M(\mathbf{q}, \mathbf{b}) - D_M(\mathbf{b}, \mathbf{p})| > \theta. \tag{3}$$

In such a case, $\mathbf{p}$ is said to be pruned or filtered. Different approaches try to prune jointly subsets of examples by storing them in a tree data structure [2,3,6,14,24]. All these methods rely on the filtering condition described in equation (3).

A time constraint, to be satisfied at test time, imposes a maximal threshold θ_{max} because equation (3) implies that a larger decision threshold θ offers a weaker pruning. The actual threshold to be used is equal to $min(\theta^*, \theta_{max})$, which is sub-optimal in terms of classification performance when $\theta_{max} < \theta^*$. The maximal threshold, θ_{max}, also decreases with the size n of the population since the pruning required to meet a given time constraint needs to be stronger. Metrics allowing for a better pruning are able to use higher decision thresholds and, hence, provide a better identification performance when the optimal threshold, θ^*, cannot be used.

Section 2 presents related works on metric learning. Our method balancing between identification performance and computational efficiency is proposed in section 3. Section 4 details the experimental results obtained on semi-artificial profile-based data.

2 Related works in metric learning

Most nearest neighbor metric learning techniques aim at satisfying different types of constraints on the training set [4]:

- Must link constraint: $S = \{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i$ and $\mathbf{x}_j$ should be similar$\}$.
- Cannot link constraint: $D = \{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i$ and $\mathbf{x}_j$ should be dissimilar$\}$.
- Relative constraint: $R = \{(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)\} : \mathbf{x}_i$ should be more similar to $\mathbf{x}_j$ than to $\mathbf{x}_k\}$.

These constraints are typically associated to loss functions to be minimized. The set S contains pairs of points with the same label (*i.e.* representing the

[3] Evaluating this distance has an $O(k * l)$ time complexity, where k and l are the respective length of the input strings.

same individual in our context) and the set D contains pairs of points with different labels. The set R contains triplets of points, only the first two ones sharing the same label. Must link constraints simply impose to points having the same label to be close to each other while cannot link constraints enforce points with different labels to be far away. MMC [23] maximizes the sum of distances between points in D while constraining the sum of squared distances between points in S. ITML [8] minimizes the violation of soft constraints for each pair of points in S or in D: points in S should be closer than a given distance value, while points in D should be farther than another specific distance value.

Relative constraints (notably used in [21,22]) enforce points sharing the same label than a given point $\mathbf{x}$ to be closer to $\mathbf{x}$ than other points with different labels. This type of constraints fits well with the classical k-NN decision procedure. Indeed, they can ensure that, for each training point, impostors are farther away than points sharing the same label. For instance, LMNN specifies relative constraints such that the k-nearest points, to each training point, all share the same label [22]. NCA is an alternative method that tries to maximize the leave-one-out accuracy of a stochastic nearest neighbor classifier [10]. In a similar fashion, MCML tries to collapse each class into a single point, while pushing away, as far as possible, the points labeled with other classes [9]. The loss functions of these three methods are such that the cost of having two points with different labels close to each other depends on the presence or absence of points with the same label in their neighborhood.

Despite their success in metric learning, the above methods are poorly suited to person identification precisely because of their use of relative constraints. Indeed, for person identification, a specific decision threshold θ is typically used because the decision procedure must include the possibility for an individual to be rejected. If one considers similar points sharing the same label, without any impostor in their neighborhood, bringing them close to each other is not enforced by relative constraints. The net result is that such similar points can stay far from each other. This situation would increase their probability of rejection since, to avoid the use of additional meta-parameters, θ is fixed globally for the whole population. Similarly, points not sharing the same label as a given point $\mathbf{x}$ should be pushed far from it, even if $\mathbf{x}$ has nearest neighbors with the correct label. Indeed, if an unknown individual is represented by an example too close to $\mathbf{x}$, it could be falsely accepted independently of the neighbors of $\mathbf{x}$. Consequently, our proposed method only relies on *must* and *cannot link* constraints.

In this paper, we focus on **exact** nearest neighbor search, as opposed to *approximate* techniques [7,11]. When using exact fixed-radius NN, an example can only be pruned if, provably, it cannot be closer to the test example than the threshold θ. In contrast, approximate NN methods have a non-zero probability to prune points that should be kept. Approximate methods can offer a considerable reduction of the decision time with only a small drop of identification performance [25]. For example, Optimized Kernel Hashing OKH [11] uses a set of prototypes and associates to every data point a hashcode, based on linear combinations of distance values between the point and the prototypes. At test time, points with the nearest hashcodes to the query point hashcode are identified. Next, only the real distances between them and the query point are actually computed. An interesting perspective of this work would be to study the possibility to learn a metric that aims at decreasing even further the OKH decision time.

3 Learning Computationally Efficient Metrics

We detail in this section our proposed method to learn a metric balancing classification performance and computational efficiency at test time. The sets S and D contain pairs of points, respectively with the same and different labels, which specify *must* and *cannot link constraints*. The competing objectives in our appoach are presented below.

- Reduce the number of misclassifications through **increasing class separability**. This is implemented by favoring points in S to be close to each other, while points in D should be far from each other. More precisely, the distance probability density function (PDF) of points in S, denoted by f_S, and the PDF of points in D, denoted by f_D, have to be separated by a large margin. Figure 1 represents both densities, with f_S in blue (left) and f_D in red (right). The more these respective densities are separated the better.
- **Increase the pruning efficiency** in order to meet a prescribed time constraint to process a query. This is achieved by flattening f_D, while keeping f_S concentrated on a small domain. Indeed, the filtering condition for a stored point $\mathbf{p}$, a query $\mathbf{q}$ and a set of prototypes B is $\exists \mathbf{b} \in B : |D_M(\mathbf{q}, \mathbf{b}) - D_M(\mathbf{b}, \mathbf{p})| > \theta$. A sharp f_S allows the use of a smaller decision threshold θ, which in turn improves the filtering. A flatter f_D means that the distances between examples from distinct individuals have a larger variance. In such a case, the absolute difference $|D_M(\mathbf{q}, \mathbf{b}) - D_M(\mathbf{b}, \mathbf{p})|$ increases resulting in a stronger pruning.

Reducing the extent of f_S benefits to both objectives, on the training set. In contrast, getting a larger class separation and a flatter f_D are competing objectives. Let us consider, for instance, the limit scenario in which the distances between points from the same individuals are 0 everywhere, while the distances between examples of distinct individuals are all assumed to be equal to some (arbitrarily large) constant $c > 0$. Formally, $f_S(x) = \delta(x)$ and $f_D(x) = \delta(x - c)$ where δ is the Dirac function. This scenario would lead to a perfect classification, for any threshold $0 < \theta < c$. However, $\forall \mathbf{q}, \mathbf{b}, \mathbf{p} : (\mathbf{q}, \mathbf{b}), (\mathbf{b}, \mathbf{p}) \in D : |D_M(\mathbf{q}, \mathbf{b}) - D_M(\mathbf{b}, \mathbf{p})| = |c - c| = 0 < \theta$. Consequently, pruning would be impossible and no time constraint could be met when the population gets very large.

Our method can be seen as an extension to the work by Mignon and Jurie who seek to maximize class separability through soft margin maximization[4] [16]:

[4] These authors use a squared distance, $D_M^2(\mathbf{p}_i, \mathbf{p}_j)$, for convenience.

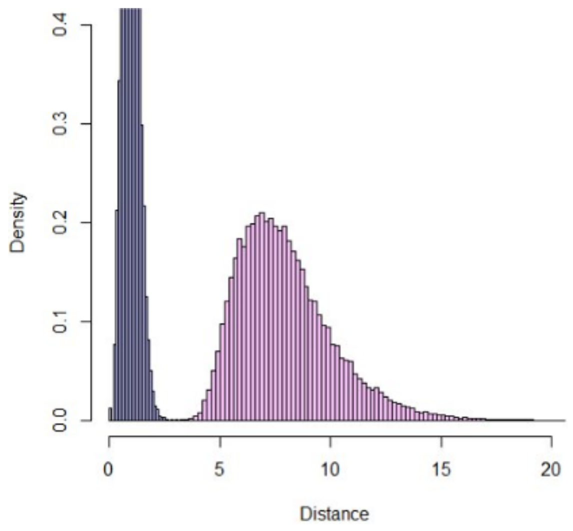


Fig. 1: Probability density functions for a person identification problem according to a metric quantifying the distances between individuals. The blue (left) curve corresponds to distances between examples of the same individual (f_S). The red (right) curve represents the distances between two examples of different individuals (f_D). Both densities slightly overlap in this example.

$$M^* = \operatorname*{argmin}_{M \succeq 0} [\sum_{(\mathbf{p}_i,\mathbf{p}_j)\in\{S\cup D\}} \max(0, y_{ij}(D_M(\mathbf{p}_i,\mathbf{p}_j) - 1))] \tag{4}$$

with $y_{ij} = 1$ if the pair of points comes from the same individual ($(\mathbf{p}_i, \mathbf{p}_j) \in S$), and $y_{ij} = -1$ otherwise ($(\mathbf{p}_i, \mathbf{p}_j) \in D$). These authors rely on a popular approximation to the loss $h(x) = \max(0, x)$ known as the generalized logistic function $l_\beta(x) = \frac{1}{\beta}\log(1 + e^{\beta x})$, such that standard gradient descent approaches can be used. The generalized logistic function tends to the original loss when β, a sharpness parameter, gets larger: $\lim_{\beta\to\infty} l_\beta(x) = h(x)$. Similarly, when $\beta \to \infty$, $l_\beta(x-1) \propto e^{\beta*(x-1)}$ in the region $x < 1$. For a large scale person identification to be effective, the PDFs f_S and f_D, estimated on the training set, are not expected to have a large overlapping mass around $x = 1$. In other words, $l_\beta(x-1)$ and $l_\beta(1-x)$ are expected to be in their non-linear regime, for $x < 1$ and $x > 1$, respectively. The minimization problem (4) can thus be approximated by (5), as long as β is chosen sufficiently large.

$$M^* = \operatorname*{argmin}_{M \succeq 0} [\frac{1}{|S|} \sum_{(\mathbf{p}_i,\mathbf{p}_j)\in S} e^{-\beta*(1-D_M(\mathbf{p}_i,\mathbf{p}_j))} + \frac{1}{|D|} \sum_{(\mathbf{p}_i,\mathbf{p}_j)\in D} e^{-\beta(D_M(\mathbf{p}_i,\mathbf{p}_j)-1)}] \tag{5}$$

In the above objective, the distance PDFs f_S and f_D are separated[5] around 1. Distance values close to 1 are penalized more heavily. The magnitude of this penalty is controlled by the meta-parameter β, which is fixed in order to balance between poor separation ($\beta \to 0$) and strong overfitting ($\beta \to \infty$). More importantly, the meta-parameter β cannot be tuned to maximize the pruning efficiency, as it controls jointly the widths of f_S and f_D.

Our multi-objective of class separability and pruning efficiency, can be reached through the control of the sharpness of f_S and f_D, independently of each other. This can be formalized through the optimization problem (6), in which the original β meta-parameter has been replaced by k_s and k_d, respectively associated with f_S and f_D.

$$M^* = \operatorname*{argmin}_{M \succeq 0} \left[\frac{1}{|S|} \sum_{(\mathbf{p}_i, \mathbf{p}_j) \in S} e^{-k_s * (1 - D_M(\mathbf{p}_i, \mathbf{p}_j))} + \frac{1}{|D|} \sum_{(\mathbf{p}_i, \mathbf{p}_j) \in D} e^{-k_d (D_M(\mathbf{p}_i, \mathbf{p}_j) - 1)} \right] \tag{6}$$

In the limit case, when $k_d \to \infty$, the only distance value between distinct individuals that matters is the smallest one. The class separability is then maximized. While decreasing k_d, the remaining mass of f_D starts playing an increasing role. Larger distance values are increasingly pushed towards the right, which causes f_D to flatten and improves the pruning efficiency. Doing so, however, the relative importance of small distance values decreases, which is detrimental to class separability. A larger k_s value favors a sharper f_S, which offers a better class separability and a stronger pruning as well. Yet, this is only true on the training set and a too large k_s would lead to overfitting, in the same fashion as a too large β would lead to in problem (5).

Unlike problem (6), problem (4) is designed to minimize the number of misclassifications for a specific threshold, here $\theta = 1$. This threshold is given a special status as the loss used in problem (4), $max(0, x - 1)$, changes its behavior at $x = 1$ (it becomes linear, like its approximation l_β). Doing so in our context has no theoretical justification because the used threshold varies with the population size.

Problem (6) is convex and can be solved through projected gradient descent. At each gradient step, this iterative procedure projects the current estimate M to the set of PSD matrices ($M \succeq 0$) while minimizing the Frobenius norm of the difference before and after the projection [23]. This projection can be implemented by decomposing the current solution $M = X^T \lambda X$, with λ a diagonal matrix representing the spectrum of M. The projected solution M' is computed as $M' = X^T \lambda' X$, where the diagonal matrix λ' contains the respective thresholded eigenvalues: $\max(\lambda_i, 0)$. This gradient descent algorithm scales linearly with the number n of training examples.

[5] One could use any arbitrary positive cut-off value instead of 1, since what matters is the separability between f_S and f_D, not the actual scale of the distance values. See figure 1 as an illustrative example.

4 Experiments

Our method has been tested on semi-artificial data. Each individual is represented by 5 string attributes (first, last and middle names, affiliations), a phone number, the day, month and year of birth, as well as 6 categorical attributes (sex, marital status, country, race, work class, education). Actual names and affiliations were extracted from publicly available repositories such as the UCLouvain repository [1]. The day and month of birth were generated randomly following a uniform distribution. The categorical attributes and the year of birth were extracted from the `adult` dataset of the UCI Machine Learning Repository [12]. These 7 attributes were extracted jointly for each example to maintain their possible dependence. The original data includes 30,000 individuals represented by a profile of 15 attributes.[6] Random perturbation is applied to obtain noisy examples from the original data. Such a noise is intended to represent the possible fluctuations of recording individual profile (*e.g.* mispelling of names, encoding errors, ...).

A training set with $|S| = |D| = 250,000$ is used for learning metrics. A test set with $|S| = |D| = 250,000$ is used for visual representation of the distance PDFs. However, the error rates are reported on far larger test sizes to reliably estimate the FAR for large population sizes ($> 10^6$). To avoid any bias, the original 5,000 individuals used to generate the training set are different from the 25,000 individuals in the test set. Figure 1 actually represents the distance PDFs, f_S and f_D, obtained on the test set. In this case, no metric learning was used but rather the Mahalanobis distance (see equation (2)) with M being fixed to a scaled identity matrix $M = \frac{1}{15}I$ (as there are 15 features).

Using a validation set, it was observed that $k_s = k_d \approx 20$ provides the best separability between the distance PDFs f_S and f_D while controlling for possible overfitting. Subsequently, k_s is kept fixed ($k_s = 20$) and test results are reported in figure 2 while varying k_d values. Depending on the specific k_s, k_d setting considered, the range of values of the distances between examples can vary. For illustrative purposes, the learned metrics have been arbitrarily scaled such that the threshold at which their FRR is equal to 0.001 is always $\theta = 1.5$. Such a scaling does not influence the F_1 score as it does not depend on the range of distance values but rather on the possible overlap between PDFs. Figure 2 illustrates that, when k_d decreases, the densities f_S and f_D get less separated which results in higher error rates. At the same time, the density f_D flattens which implies a more efficient pruning.

The optimal F_1 score is reported in Figure 3a for various metrics learned with population sizes of up to 2 millions individuals. The lower the better since $1 - F_1^*$ is actually plotted. This would be the F_1 score of these metrics if no time con-

[6] 30,000 individuals is considered to be enough to simulate a population of millions of individuals. Indeed, an arbitrary number of examples can be generated through random perturbation of the original data. This leads to a reliable estimation of the FRR. Moreover, with 30,000 points, it is possible to form $\frac{30{,}000^2}{2} = 450M$ pairs of points, which is sufficient to estimate the FAR correctly.

straint were present. As expected, a higher k_d gives a better optimal F_1 measure. The optimal thresholds θ^* for these metrics are depicted in figure 3c (dashed). A higher k_d implies a higher optimal threshold θ^*. Indeed metrics learned with a higher k_d have a lower FAR. One can then afford the use of higher decision thresholds θ. In presence of a maximal time budget, the optimal identification performance of these metrics can hartdly be achieved due to the substantially higher optimal thresholds ($\theta_{max} < \theta^*$) and their low pruning efficiency.

The amount of pruning obtained with the learned metrics, for a threshold $\theta = 1.5$ and up to 500 prototypes can be seen on figure 3b. Clearly, a small k_d is more efficient. This better pruning allows these metrics to use larger maximal thresholds θ_{max} when a time constraint is involved. Nonetheless, there is no additional gain when going below $k_d = 1$, even if f_D still gets wider. This can be explained by the fact that unfiltered points are the points that are the closest to the test point. If the separation between the two curves narrows, the closest points are closer, and hence may be harder to filter. This is supported by the fact that a $k_d < 1$ still provides gain for a low number of prototypes. On figure 3c, the maximal thresholds are displayed for a maximal time budget of 7,500 distance computations and a maximum of 500 prototypes, compared to the optimal thresholds (dashed). Metrics with a good separation have higher optimal thresholds θ^* but lower maximal ones θ_{max}. For both reasons, the time constraint becomes limiting a lot sooner.

It is seen on figure 3a that a higher k_d offers better identification performance. However, when adding a maximal time budget, the results are radically different as depicted on figure 4. The dashed line represents the performance after the time constraint imposed a threshold lower than the optimal one ($\theta_{max} < \theta^*$). It turns out that $k_d = 20$ is the best choice only for populations up to 100k individuals. Between 100k and 225k, $k_d = 10$ provides the best results. Then $k_d = 5$ between 225k and 610k. Finally, $k_d = 2.5$ has the lowest error rates from 610k to 2M. This shows well that the best suited metric depends on the size of the population and the time constraint involved. [7]

[7] Technically, it also depends on the number of prototypes used. The difference of θ_{max} is increased when fewer prototypes are used, which could result from an additional space constraint.

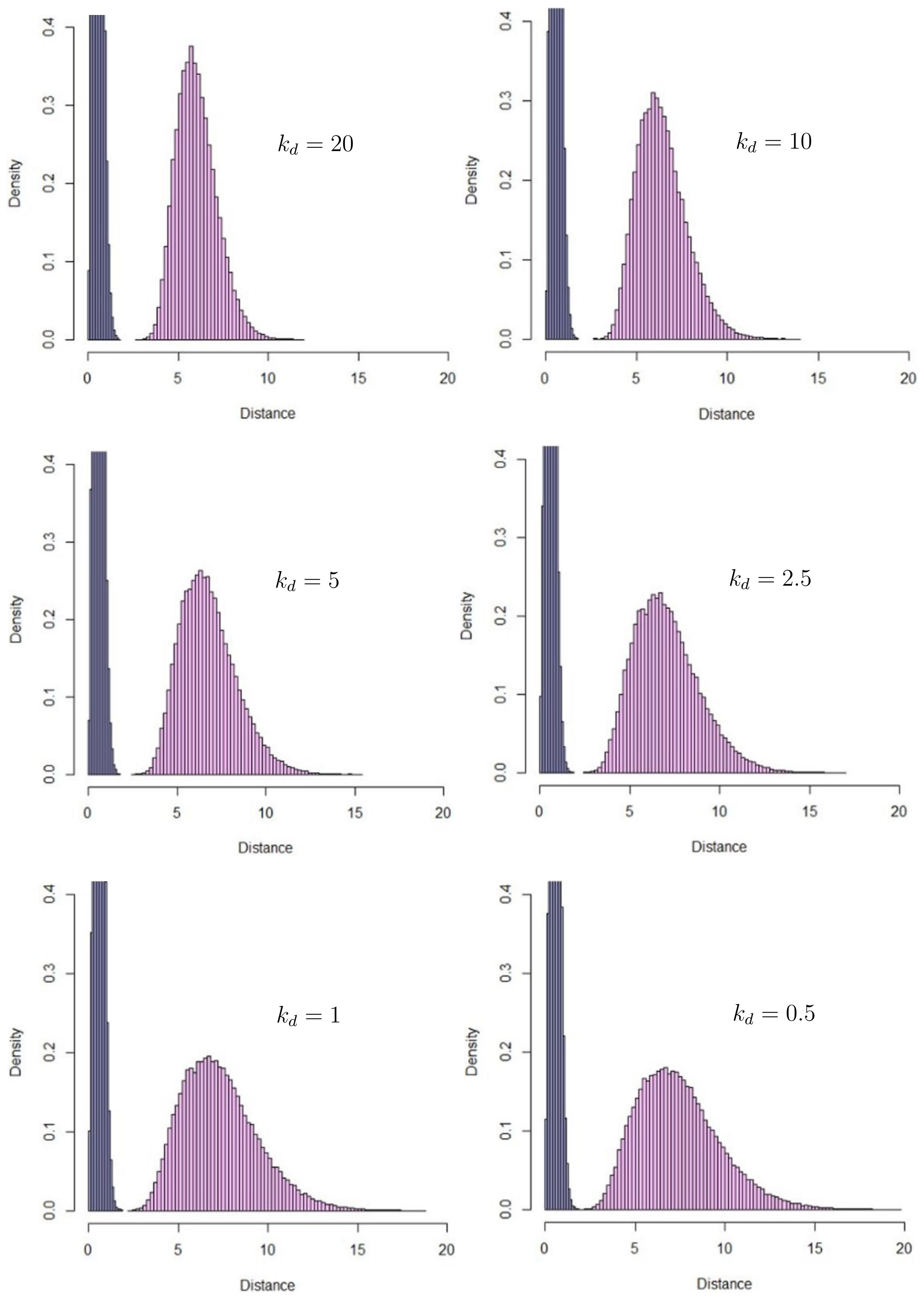
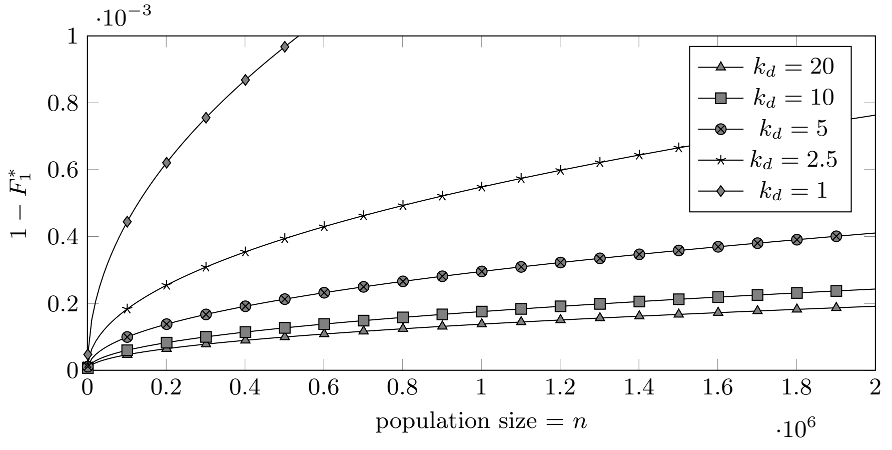
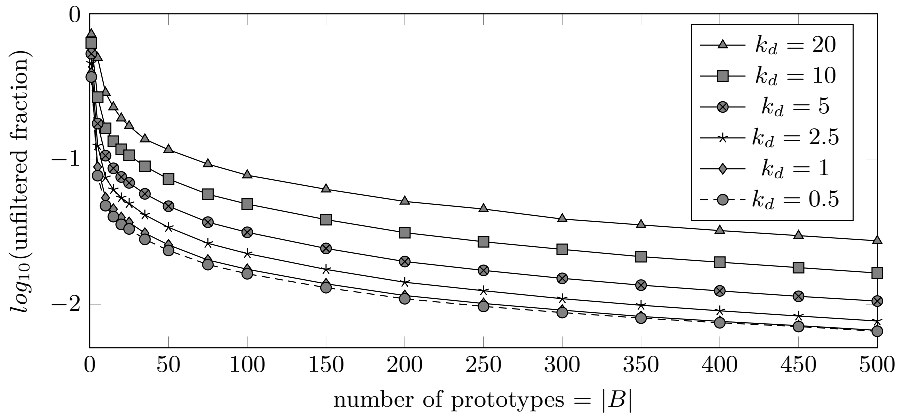


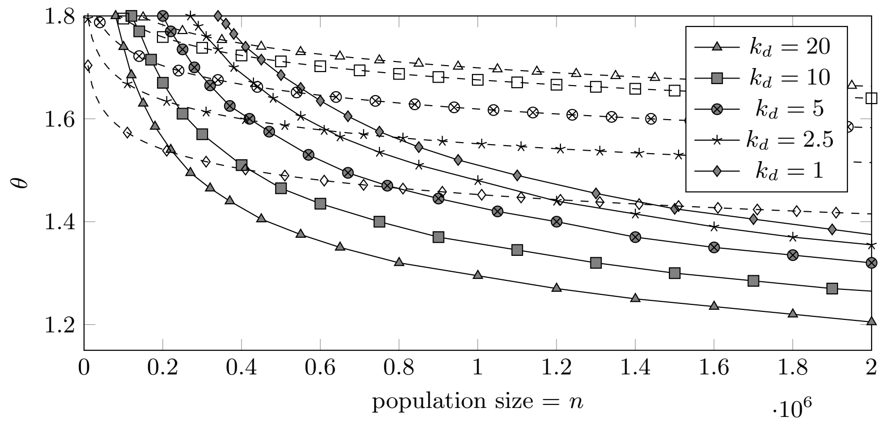
Fig. 2: Test set results for $k_s = 20$ and different k_d values. In blue (left) is plotted the distance PDF corresponding to the set of similar points S (f_S). In red (right) is the distance PDF of the set of dissimilar points D (f_D).



(a)



(b)



(c)

Fig. 3: Optimal identification F_1 score of the metrics learned with different k_d values (the lower the better as $1 - F_1^*$ is plotted), and $k_s = 20$ (a). Unfiltered fraction of examples, for the different metrics, with up to 500 prototypes (θ=1.5) (b). Maximum (plain) and optimal (dashed) thresholds of these metrics, for a time constraint of 7,500 distance computations and a maximal number of 500 prototypes(c).

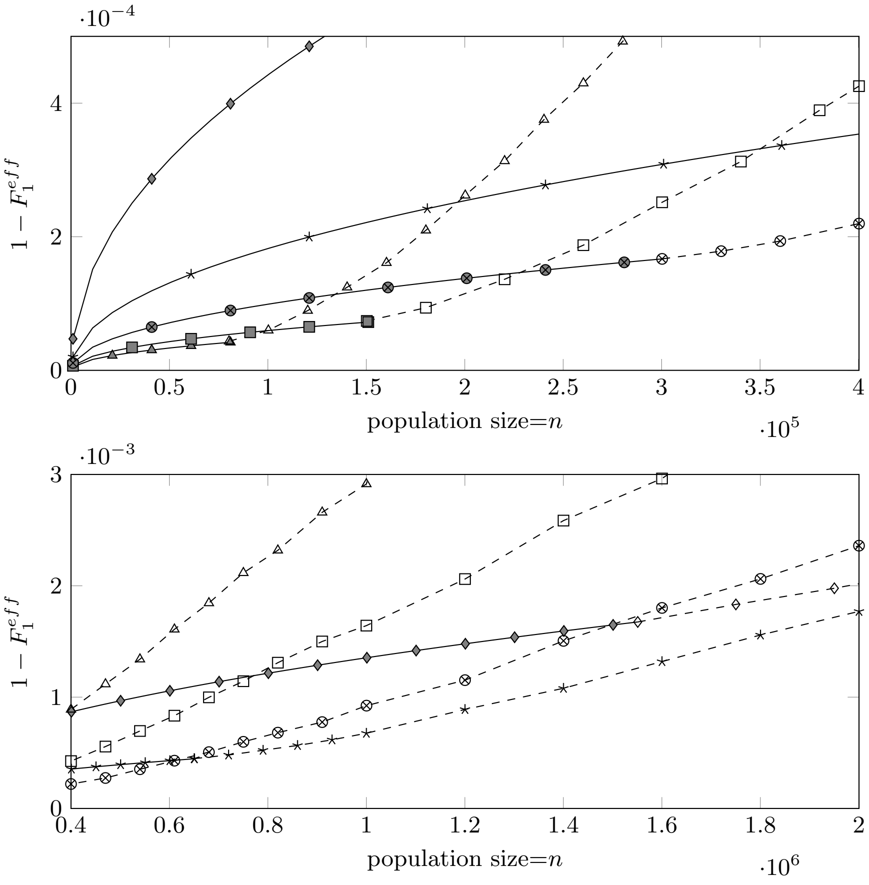


Fig. 4: Effective F_1 score of the metrics given a maximal time budget of 7,500 distance computations, for population sizes up to 2M. The lower the better as $1 - F_1^{eff}$ is plotted. The various curves correspond to the same k_d values as depicted in figure 3, and $k_s = 20$. Plain lines refer to settings where θ^* can be used while satisfying the maximal number of distance computations. Dashed lines refer to settings where the used threshold has to be smaller than θ^* to satisfy the time constraint at the cost of decreasing the identification performance ($\theta_{max} < \theta^*$).

5 Conclusion and perspectives

We address here a person identification task with a fixed-radius nearest neighbor method. For large sets (typically $\geq 10^6$) of stored individuals, we argue that it is crucial to learn metrics that lead to a small decision time. The search efficiency of exact nearest neighbor techniques is dictated by a pruning step that drastically reduces the number of distances to be computed. Taking this pruning step into account, our metric learning approach looks for a trade-off between classification performance and computational efficiency of a NN classifier at test time. This is done through the optimization of a Mahalanobis distance metric over a set of positive semi-definite matrices, and solved by projected gradient descent. The proposed approach is compared to soft-margin maximization metrics which offer the best identification performance for small populations. Yet, for larger populations, for which time and space constraints at decision time play a significant role, pruning efficient metrics are shown to be best suited and are actually closer to the optimal ones in terms of identification performance. This is assessed here on semi-artificial profile-based data, representative of a population of up to two million individuals.

The metric learning approach introduced here could be used, at least in principle, in other settings. For instance, part of our future work concerns Optimized Kernel Hashing [11], an approximate nearest neighbor technique also relying on a predefined metric.

Our current algorithm to solve the convex multi-objective metric learning problem scales relatively poorly in the number of dimensions of the input space ($O(d^3)$, with d the dimensionality of the input space X). This is due to the projection step of the projected gradient descent used here. Nonetheless, it can be adapted to handle high dimensional data, following the method proposed by Mignon and Jurie [16]. Instead of projecting the current solution to the set of PSD matrices, the search space is restricted to be PSD by design, while mapping the high dimensional input space into a lower dimensional one. One could also study the applicability of the proposed method to other identification tasks, such as person identification through computer vision or face verification.

References

1. UCL repertoire. [https://uclouvain.be/fr/repertoires#]
2. Baeza-Yates, R., Cunto, W., Manber, U., Wu, S.: Proximity matching using fixed-queries trees. In: Annual Symposium on Combinatorial Pattern Matching. pp. 198–212. Springer (1994)
3. Baeza-Yates, R., Navarro, G.: Fast approximate string matching in a dictionary. In: String Processing and Information Retrieval: A South American Symposium, 1998. Proceedings. pp. 14–22. IEEE (1998)
4. Bellet, A.: Tutorial on metric learning. Department of Computer Science Viterbi School of Engineering University of Southern California (2013), http://researchers.lille.inria.fr/abellet/talks/metric_learning_tutorial_CIL.pdf
5. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Communications of the ACM **18**(9), 509–517 (1975)
6. Burkhard, W.A., Keller, R.M.: Some approaches to best-match file searching. Communications of the ACM **16**(4), 230–236 (1973)
7. Cao, Y., Qi, H., Zhou, W., Kato, J., Li, K., Liu, X., Gui, J.: Binary hashing for approximate nearest neighbor search on big data: A survey. IEEE Access **6**, 2039–2054 (2018)
8. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: Proceedings of the 24th international conference on Machine learning. pp. 209–216. ACM (2007)

9. Globerson, A., Roweis, S.T.: Metric learning by collapsing classes. In: Advances in neural information processing systems. pp. 451–458 (2006)
10. Goldberger, J., Hinton, G.E., Roweis, S.T., Salakhutdinov, R.R.: Neighbourhood components analysis. In: Advances in neural information processing systems. pp. 513–520 (2005)
11. He, J., Liu, W., Chang, S.F.: Scalable similarity search with optimized kernel hashing. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1129–1138. ACM (2010)
12. Lichman, M.: UCI machine learning repository. [https://archive.ics.uci.edu/ml/datasets/adult] (2013)
13. Mahalanobis, P.C.: On the generalised distance in statistics. In: Proceedings National Institute of Science, India. vol. 2, pp. 49–55 (Apr 1936), http://ir.isical.ac.in/dspace/handle/1/1268
14. Micó, L., Oncina, J., Carrasco, R.C.: A fast branch & bound nearest neighbour classifier in metric spaces. Pattern Recognition Letters **17**(7), 731–739 (1996)
15. Micó, L., Oncina, J., Vidal, E.: An algorithm for finding nearest neighbours in constant average time with a linear space complexity. In: Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on. pp. 557–560. IEEE (1992)
16. Mignon, A., Jurie, F.: Pcca: A new approach for distance learning from sparse pairwise constraints. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 2666–2672. IEEE (2012)
17. Navarro, G.: A guided tour to approximate string matching. ACM computing surveys (CSUR) **33**(1), 31–88 (2001)
18. nski, K.D.: Extreme classification: Machine learning with millions of labels. [http://www.cs.put.poznan.pl/kdembczynski/pdf/extreme-classification-uam-2017.pdf] (May 24, 2017)
19. Omohundro, S.M.: Five balltree construction algorithms. International Computer Science Institute Berkeley (1989)
20. Ruiz, E.V.: An algorithm for finding nearest neighbours in (approximately) constant average time. Pattern Recognition Letters **4**(3), 145–157 (1986)
21. Schultz, M., Joachims, T.: Learning a distance metric from relative comparisons. In: Advances in neural information processing systems. pp. 41–48 (2004)
22. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. In: Advances in neural information processing systems. pp. 1473–1480 (2006)
23. Xing, E.P., Jordan, M.I., Russell, S.J., Ng, A.Y.: Distance metric learning with application to clustering with side-information. In: Becker, S., Thrun, S., Obermayer, K. (eds.) Advances in Neural Information Processing Systems 15, pp. 521–528. MIT Press (2003), http://papers.nips.cc/paper/2164-distance-metric-learning-with-application-to-clustering-with-side-information.pdf
24. Yianilos, P.N.: Data structures and algorithms for nearest neighbor search in general metric spaces. In: Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 311–321. SODA '93, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (1993), http://dl.acm.org/citation.cfm?id=313559.313789
25. Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., Tian, Q.: Scalable person reidentification: A benchmark. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1116–1124 (2015)