# Geometric mean metric learning - Project report for Machine Learning class

Rémi Coudert

May 21, 2017

## 1 Introduction

In Machine Learning, a great number of algorithms need to use some metric to compute distances between examples. The most famous and useful ones are certainly the $l_1$ (Manhattan), $l_2$ (Euclidian) and $l_\infty$ distances, used in many applications. These distance functions all have their pros and cons, for example the $l_1$ norm is not differentiable whereas the Euclidian distance uses the square root, inducing computation cost and errors.

A distance between two $d$ dimensional points is a function $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ satisfying :

$$d(x, y) \geq 0$$

$$d(x, y) = d(y, x)$$

$$d(x, y) = 0 \Leftrightarrow x \equiv y$$

$$d(x, y) \leq d(x, z) + d(z, y)$$

The goal of metric learning is to be able to learn a specific metric for a given problem, that is finding the best possible metric that suits the data a training time such that it is still as good as possible at test time. Metric Learning was first introduced by Xing et al.[6] and consists in learning the $d \times d$ matrix $A$ such that the learnt distance $d$ is :

$$d_A(x, y) = \sqrt{(x-y)^T A (x-y)}$$

Over the years many metric learning algorithms were introduced such as LMNN[5] or ITML[2]. During ICML 2016, Hab., Hos. and Suv. presented a new metric learning algorithm called Geometric Mean Learning[7].

## 2 Common tools in metric learning and GMML specific prerequisites

In this section, I'll introduce some useful mathematical/theoretical tools used in metric learning and GMML in particular.

Let :

$$S = \{(x_i, x_j) \mid x_i \text{ and } x_j \text{ are in the same class}\}$$

$$D = \{(x_i, x_j) \mid x_i \text{ and } x_j \text{ are in different classes}\}$$

$$A_0 \text{ be the } d \times d \text{ prior knowledge matrix}$$

Hab. et al. defined as well the operator

$$\#_t \ : \ A \#_t B = A^{1/2}(A^{-1/2} B A^{-1/2})^t A^{1/2}, \ t \in [0, 1]$$

representing the points on the geodesic curve joining the matrices $A$ ans $B$.

The trace operator $tr(.)$ on square $d$ dimensional matrices is defined as $tr(A) = \sum\limits_{0 \leq i \leq d} A_{ii}$

$SPD$ matrices are symmetric positive definite matrices, i.e. symmetric matrices with positive eigen-values.

## 3 Geometric Mean Learning formulation and solution

The main idea behind GMML was first to define the objective function :

$$\sum_{(x_i,x_j)\in S} d_A(x_i,x_j) + \sum_{(x_i,x_j)\in D} d_{A^{-1}}(x_i,x_j)$$

The function was then simplified using the trace :

$$\min_{A\succ 0} \sum_{(x_i,x_j)\in S} tr(A(x_i-x_j)(x_i-x_j)^T) + \sum_{(x_i,x_j)\in D} tr(A(x_i-x_j)(x_i-x_j)^T)$$

Define :

$$S' = \sum_{(x_i,x_j)\in S} (x_i-x_j)(x_i-x_j)^T$$

$$D' = \sum_{(x_i,x_j)\in D} (x_i-x_j)(x_i-x_j)^T$$

being the similarity and dissimilarity matrices respectively.

The authors then prove the existence and find the formulation for a closed form solution of the objective function thanks to a recent paper[1] :

$$A = \begin{cases} S'^{-1}\#_{1/2}D & \text{When } S' \text{ is SPD} \\ (S+\lambda A_0^{-1})^{-1}\#_{1/2}(D+\lambda A_0) & \text{When } S' \text{ is PSD} \end{cases}$$

The second case is the regularized version, similar to many machine learning algorithms' regularization, $\lambda$ being the regularization parameter.

## 4 Some Other Metric Learning Algorithms

- Large Margin Nearest Neighbours (LMNN) : This Algorithm uses the KNN principle of looking at close neighbours and updates the metric such that similar points become closer and dissimilar points are pushed away.
- Information Theoretic Metric Learning (ITML) : ITML minimizes the differential relative entropy between two multivariate Gaussians under constraints on the distance function.
- Sparse Determinant Metric Learning (SDML) : An efficient sparse metric learning in high-dimensional space via L1-penalized log-determinant regularization

## 5 Implementation and Experimental protocol

The implementation was done in Python 3, with regular packages, the package metric-learn used to have an implementation of LMNN, ITML and SDML[4], and scikit-learn.

### 5.1 Datasets

The datasets used are :

- Iris, a famous dataset listing iris species and their characteristics. The classification task is to find a given iris' specie with its characteristics.
- Voice. This Dataset aggregate voice informations such as average pitch or peak frequency, the goal being to decide whether the person is Male or Female.
- Diabetes. This Dataset provides several medical information along with a classification of patients having diabetes or not.

### 5.2 Experimental protocol

The computation is done in two parts, the learning of the matrix $A$ associated with the metric, and the classification itself done with KNN and scikit-learn.

In the case of LMNN, singular matrices can't be used at all (no regularization) so testing with the voice dataset wasn't possible.

The data is first divided in a training and testing part, $A$ is learned with the training set and the KNN algorithm learns on it too. The test data is then fitted by the KNN algorithm with $k = 3$ to compute the frequency of classification errors.

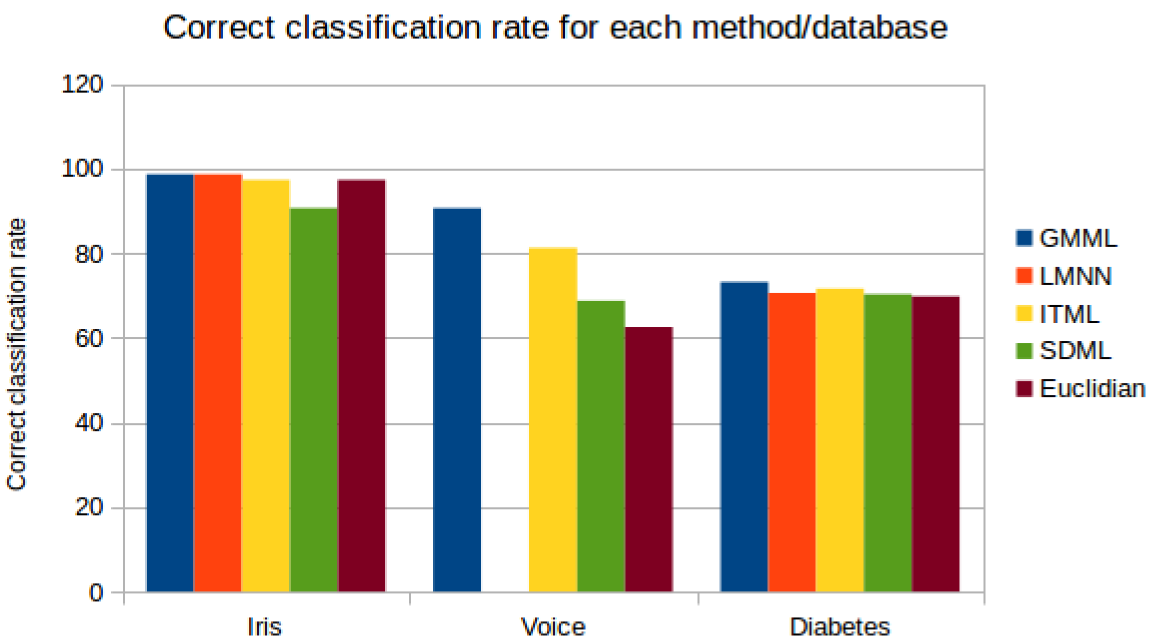
The voice dataset is hard to compute so the tests are done on a fraction of it.

For the case of GMML, the test is done for several values of $t$ $(0.1, ..., 0, 9)$ and $\lambda$ $(0.1, ..., 0.9)$, to output the single best case.

To compare, a standard KNN with the euclidian distance is computed as well.

# 6 Results

Figure 1: Results



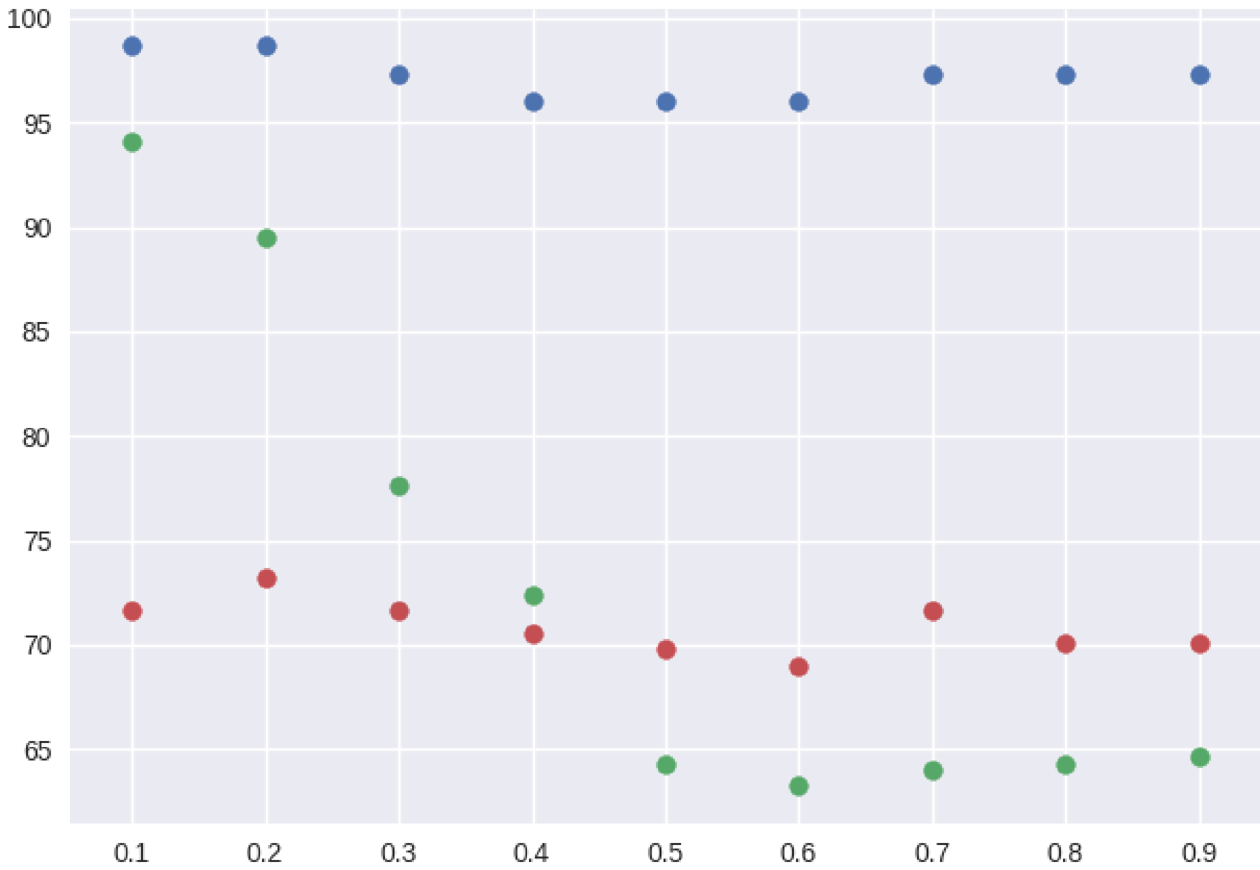

As shown in Figure1, for the 3 datasets, GMML is at least as good as other methods if not better (in the case of voice).

Figure 2 shows the influence of the parameters $t$ on the results of GMML for each dataset. We can see that $t$ influences the results by approximately 5% for iris and voice and roughly 30% for diabetes, which is non-negligible.

# 7 Conclusion

As it can be seen the results for GMML are quite good. This method has the great advantage of having a closed-form solution which is remarkable. Nevertheless, the computation is quite expensive since it involves complex matrix operation such as inversion and square root.

Figure 2: GMML correct classification rate with different values of t, blue : iris, green : voice, red : diabetes




The authors mention that there exists several methods for fast computation of geodesics on SPD matrices such as Cholesky-Schur method and scaled Newton methods[3].

I should mention that the authors tested on more datasets and with different algorthms. Their result were similar to mine except for one dataset GMML being behind. The relative small size of the iris dataset makes it not very good but it permitted me to have a quick test.

More tests could have been done with more algorthms, more/bigger datasets, but the protocol stays the same so it would not imply big changes in the code.

There are a lot of parameters involved in GMML and other algoritms, thus a really complete test is really hard to do very well since we should be evaluating the influence of each paramater for each algorithm.

## References

[1] Rajendra Bhatia and John Holbrook. "Riemannian geometry and matrix geometric means". In: *Linear algebra and its applications* 413.2-3 (2006), pp. 594–618.

[2] Jason V Davis et al. "Information-theoretic metric learning". In: *Proceedings of the 24th international conference on Machine learning*. ACM. 2007, pp. 209–216.

[3] Bruno Iannazzo. "The geometric mean of two matrices from a computational viewpoint". In: *arXiv preprint arXiv:1201.0101* (2011).

[4] Guo-Jun Qi et al. "An efficient sparse metric learning in high-dimensional space via l 1-penalized log-determinant regularization". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM. 2009, pp. 841–848.

[5] Kilian Q Weinberger, John Blitzer, and Lawrence Saul. "Distance metric learning for large margin nearest neighbor classification". In: *Advances in neural information processing systems* 18 (2006), p. 1473.

[6] Eric P Xing et al. "Distance metric learning with application to clustering with side-information". In: *NIPS*. Vol. 15. 505-512. 2002, p. 12.

[7] Pourya Zadeh, Reshad Hosseini, and Suvrit Sra. "Geometric mean metric learning". In: *International Conference on Machine Learning*. 2016, pp. 2464–2471.