LOM, a Locally Oriented Metric which Improves Accuracy in Classification Problems

Julio Revilla Ocejo, Evaristo Kahoraho Bukubiye
Dpt. of Industrial Technologies
University of Deusto
Bilbao, Spain
jrevilla@deusto.es, kahoraho@deusto.es

***Abstract*— New tools for computer automatic reasoning, case-based reasoning or data mining, require powerful artificial intelligence techniques to provide both, a high rate of precision in their predictions, and a collection of similar past experiences that could be applied in the actual scenario. Algorithms based on Nearest Neighbors, Support Vector Machines, etc. often provide an accurate solution for classification problems, but they depend on how the similarity is measured. For this task, most of the experts employ a Euclidean metric that equally weights all attributes of the case (which is unlikely in the real world). In addition, it is well known that a correct metric choice should improve their prediction abilities and avoid the curse of dimensionality. In this paper, we present a new metric for those algorithms. It replaces the traditional Euclidean approach with a new riemannian metric that "enlarges" the space parallel to the frontier of separation between classes, thus improving classification accuracy.**

Keywords- metrics; k-NN; SVM; Riemannian; Dijkstra.

I. INTRODUCTION

In a classification problem [1], a "case" is characterized by a set of numerical, ordinal and/or nominal values formed by its P attributes. It belongs to one of the J possible classes $\{y_j\}$. Classification algorithms based on empirical data have at their disposal N "training cases", which consist of values of the attributes and their classes $\{ \boldsymbol{x}_n, y_n \}$, $n = 1 \ldots N$. The target of these algorithms is assigning a new case to the correct class.

In current case-based reasoning (CBR), data mining tools (DM), etc. nearest neighbor classification algorithms (k-NN) are frequently used to implement the similar cases search phase. This kind of algorithms seeks for the cases closest to the new one to determine its class by majority voting. Internally, they usually employ a Euclidean metric to measure the "distance" (dissimilarity) among cases, but this metric does not behave particularly well in the border of separation between two classes [2][3][4].

We introduce, in this paper, a better metric for these algorithms, the LOM metric. It attempts to adjust locally the perception of which points are close to a given one. It adapts the measurement of distances in such a way that they are shorter in the direction parallel to the tangent hyperplane to the border of separation between classes, and gets larger in the perpendicular direction. The decision function, which provides the border of separation between classes, is estimated in a pilot trial and obtained by a traditional classification algorithm. LOM has only two free parameters, which can be optimally tuned through a cross-validation (CV) procedure.

This paper is organized as follows: Section 2 relates actual and past research in this area. Section 3 describes the LOM metric and its properties. In Sections 4 and 5, the specifics of the LOM metric, working in conjunction with a support vector machine (SVM) decision function, are explained. Section 6 contains the first results, which prove the merit of the algorithm. In the end, a section of conclusions and the work to be addressed in a near future is included.

II. METRICS USED IN SIMILARITY MEASUREMENTS

The vast majority of the algorithms used in automated reasoning based on previous cases, sensor fusion, DM and other typical tasks of artificial intelligence (AI), in one way or another, base its calculations on some kind of measure of similarity/dissimilarity between objects. In most scenarios, this fact usually remains unnoticed (since is regularly used the Euclidean metric as a default). Euclidean metric considers of equal relevance the values of the different attributes. It ignores in what area of the attributes space the case is located and whether the cases close to it belong or not to the same class.

To distinguish between similar objects, humans weight some attributes more than others; the features chosen to classify an object (and their relevance) depend on what they see at first glance. Definitively, humans do not employ a Euclidean metric.

A. Relevant previous studies on simmilarity metrics

Numerous studies have suggested that in the vicinity of the separation surface of two classes, equidistant distance curves (isolines) "should be enlarged" in the direction tangent to the surface and "shortened" in the perpendicular to it. Far away from this border, the Euclidean metric can be considered a sufficiently good choice.

The algorithm LAMANNA of Carlotta Domeniconi and her team [2][3], proposes to employ the boundary of separation between classes, provided by a SVM, to determine the most relevant local directions in the vicinity of a point. This paper introduces very interesting aspects, as using the gradient of the decision function as an indicator of the direction of greater relevance in the classification. However, their algorithm does not lead to a metric, since the distance does not meet properties, such as the triangular inequality or the symmetry ones. Neither is it clear that their distance will lead to positive definite (PD) kernels and, therefore, it is not guaranteed that it could be used by the common optimization algorithms employed in the SVMs.

Weinberger et al. [5] optimize a Mahalanobis metric using semidefinite programming (SDP). Their algorithm provides

 ISBN: 978-1-61208-386-5

acceptable results and leads to a global metric for the entire attributes space. It does also optimize a lot of parameters (P^2) so the final metric does not have easily interpretable properties. In similar approaches, other researchers [6][7][8] have attempted to optimize the Mahalanobis matrix using different techniques (SDP to optimize the separation between pairs of cases, average of weighted covariance matrices, etc.).

Goldberg et al. [9] proposed the NCA algorithm that adjusts the elements of a linear transformation matrix, minimizing the probability of classification error in a random selection of cases close to the current one. It results in a global metric, but the minimization method is subject to be trapped in local minima; and can incur in overfitting. In general, the use of metrics which involve matrices with real elements does not allow their main directions to change in different regions of the space of attributes, just as we propose in this paper.

Following a completely different approach, several authors have tried to improve the metric used to calculate the distances in SVMs based on RBF kernels. Chan et al. [10] proposed to modify the radius of the RBF depending on the density of cases in that region. It is a very simple approach and does not orient the metric in any special direction.

Amari and Wu [11][12] were pioneers in pointing out the relevance of the kernel choice in SVM performance. They proposed to use conformal transformations for RBF kernels, which widen the spatial resolution in the vicinity of the surface of separation between classes. Wu et al. [13] developed a conformal transformation in the space of the features, based on the support vectors (SV) obtained in a previous iteration.

Williams et al. [14] chose a different function to implement the conformal transformation depending on the value of the separation function provided by a SVM (whose value for different points in the attributes space provides the criteria for class separation).

All conformal transformation previously related, use a prior calculation of the border of separation between classes and then "widen" the metric in the associated Riemann space. None of them justify in detail why this would improve the separability of the points in those areas. They neither orient the metric in the direction of class separation.

III. THE PROPOSED METRIC

What does that a point is "close" to another mean? This is a fundamental question to define a metric. In a two-class problem, in the vicinity of a point in the attributes space, some directions point to areas where there are many elements of a certain class, while other directions point to areas with elements of mixed classes. To determine these directions is of fundamental relevance when delineating a flawless metric.

The aim of our actual research is to define a metric that possess two groups of directions in the P-dimensional attributes space:

- One direction is perpendicular to the hypersurface separating both classes. The gradient of the separation function could be a valid method to determine it.
- The set of all directions perpendicular to the former, i.e., all those contained in the hyperplane perpendicular to the gradient.

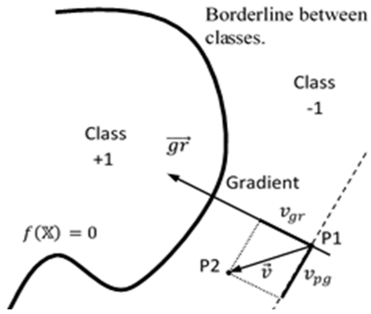

Figure 1. Main directions for the metric.

Different weights can be assigned to each of these two groups of directions. We propose to calculate the "distance" between two very close points, based on a weighted sum of the squared norms of the projections of the segment joining the two points on each of these two sets of directions. Hence, the formula proposed to calculate the distance is:

$$d^2 = \frac{v_{gr}^2}{r_m^2} + \frac{v_{pg}^2}{r_M^2} \tag{1}$$

where:

- $\vec{v}$ is the vector joining the two points (P1-P2 in Fig.1) whose distance is to be evaluated.
- $\overrightarrow{gr}$ is the normalized gradient vector at point P1.
- v_{gr} is the norm of the projection of $\vec{v}$ on the direction of the gradient of the hypersurface of class separation. It can be easily computed by $v_{gr} = \vec{v}.\overrightarrow{gr}$.
- v_{pg} is the norm of the projection of $\vec{v}$ on the hyper-plane perpendicular to the gradient. $v_{pg}^2 = \|\vec{v}\|^2 - v_{gr}^2$.
- r_m is the "minor radius". It reflects that in the direction of the gradient, the length of the projection will contribute to a larger distance between points.
- r_M is the "major radius". It weights the projection in the plane perpendicular to the gradient. Any separation in this direction will contribute to a lesser extent to the distance between points.

In order to relate r_m and r_M to the separation function, they will be defined by the following formulas:

$$\begin{aligned} r_m &= r/\left(1 + \tau\, e^{-f(\mathbb{X})^2}\right) \\ r_M &= r.\left(1 + \tau\, e^{-f(\mathbb{X})^2}\right) \end{aligned} \tag{2}$$

where:

- r defines the general scale for the metric.
- τ is a parameter which shrinks/amplifies the minor/major radius as the base point approaches to the border of separation between classes.
- $f(\mathbb{X})$ is the function that evaluates the membership of a point $\mathbb{X}$ in the attribute's space to one or another class. On the boundary between classes, its value is zero and as the point moves away, takes positive or negative values.

When the base point is close to the border of separation between classes, $f(\mathbb{X})$ is zero, and therefore, the major radius will be $(1 + \tau)^2$ times the minor radius. In those areas of the attributes space where $f(\mathbb{X})$ returns a large positive or negative value, the minor and major radius are nearly equal (see Fig.2).

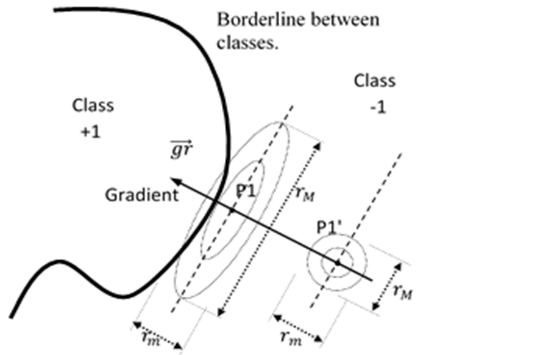

Figure 2. Equidistant distance curves (isolines) for the points P1 and P1'.

Therefore, the metric depends on just two parameters that are adjustable for each scenario: r and τ.

A. Metric properties

A differential element (line element) in a *P*-dimensional Euclidean space could be represented by:

$$ds = [dx_1, dx_2, \dots, dx_P] \tag{3}$$

The normalized gradient of the decision function for a certain point of the attributes space is:

$$gr = [gr_1, gr_2, \dots, gr_P] \tag{4}$$

Thus, the projection of the differential element in the direction of the gradient:

$$ds_{gr} = \langle ds, gr \rangle = \sum_{i=1}^{P} gr_i . dx_i \tag{5}$$

And its squared norm ds_{gr}^2 in matrix form:

$$[dx_1 \;\; dx_2 \;\; \dots \;\; dx_P] \begin{bmatrix} gr_1^2 & gr_1.gr_2 & \dots & gr_1.gr_P \\ gr_1.gr_2 & gr_2^2 & \dots & gr_2.gr_P \\ \dots & \dots & \dots & \dots \\ gr_1.gr_P & gr_2.gr_P & \dots & gr_P^2 \end{bmatrix} \begin{bmatrix} dx_1 \\ dx_2 \\ \dots \\ dx_P \end{bmatrix} \tag{6}$$

The squared norm of the projection of the differential element on the plane perpendicular to the gradient: ds_{pg}^2

$$ds_{pg}^2 = ds^2 - ds_{gr}^2 =$$

$$[dx_1 \, dx_2 \dots dx_P] \begin{bmatrix} 1 - gr_1^2 & -gr_1.gr_2 & \dots & -gr_1.gr_P \\ -gr_1.gr_2 & 1 - gr_2^2 & \dots & -gr_2.gr_P \\ \dots & \dots & \dots & \dots \\ -gr_1.gr_P & -gr_2.gr_P & \dots & 1 - gr_P^2 \end{bmatrix} \begin{bmatrix} dx_1 \\ dx_2 \\ \dots \\ dx_P \end{bmatrix} \tag{7}$$

From these results, it is possible to express a differential element in the LOM metric, given by (3), as:

$$ds^2 = [dx_1 \;\; dx_2 \;\; \dots \;\; dx_P] \; G \begin{bmatrix} dx_1 \\ dx_2 \\ \dots \\ dx_P \end{bmatrix} \tag{8}$$

$$G = \begin{bmatrix} \left(\frac{1}{r_m^2} - \frac{1}{r_M^2}\right) gr_1^2 + \frac{1}{r_M^2} & \left(\frac{1}{r_m^2} - \frac{1}{r_M^2}\right) gr_1.gr_2 & \cdots & \left(\frac{1}{r_m^2} - \frac{1}{r_M^2}\right) gr_1.gr_P \\ \left(\frac{1}{r_m^2} - \frac{1}{r_M^2}\right) gr_1.gr_2 & \left(\frac{1}{r_m^2} - \frac{1}{r_M^2}\right) gr_2^2 + \frac{1}{r_M^2} & \cdots & \left(\frac{1}{r_m^2} - \frac{1}{r_M^2}\right) gr_2.gr_P \\ \dots & \dots & \dots & \dots \\ \left(\frac{1}{r_m^2} - \frac{1}{r_M^2}\right) gr_1.gr_P & \left(\frac{1}{r_m^2} - \frac{1}{r_M^2}\right) gr_2.gr_P & \cdots & \left(\frac{1}{r_m^2} - \frac{1}{r_M^2}\right) gr_P^2 + \frac{1}{r_M^2} \end{bmatrix} \tag{9}$$

Decomposing G into $G_1 + G_2$:

$$G_1 = \left(\frac{1}{r_m^2} - \frac{1}{r_M^2}\right) \begin{bmatrix} gr_1^2 & gr_1.gr_2 & \dots & gr_1.gr_P \\ gr_1.gr_2 & gr_2^2 & \dots & gr_2.gr_P \\ \dots & \dots & \dots & \dots \\ gr_1.gr_P & gr_2.gr_P & \dots & gr_P^2 \end{bmatrix} \tag{10}$$

$$G_2 = \frac{1}{r_M^2} I_P$$

Analyzing this G matrix, it can be concluded that G is the sum of a resultant matrix of a dyadic product: G_1, with an scalar matrix G_2. G_1 has only one non-zero eigenvalue equal to $\left(\frac{1}{r_m^2} - \frac{1}{r_M^2}\right) \|gr\|^2 \geq 0$. The P eigenvalues of G_2 are all equal to $\frac{1}{r_M^2} > 0$.

According to the Weyl's theorem, which states that if A and B are two symmetric matrices of dimension $P_x P$ with eigenvalues $\lambda_1(A) \leq \lambda_2(A) \leq \dots \leq \lambda_P(A)$ and $\lambda_1(B) \leq \lambda_2(B) \leq \dots \leq \lambda_P(B)$ respectively, and if the eigenvalues of the matrix resulting from the sum $A + B$ are: $\lambda_1(A+B) \leq \lambda_2(A+B) \leq \dots \leq \lambda_P(A+B)$, it is true that $\forall \; i \; 1 \dots P$:

$$\lambda_i(A+B) \geq \begin{cases} \lambda_i(A) + \lambda_1(B) \\ \lambda_{i-1}(A) + \lambda_2(B) \\ \dots \\ \lambda_1(A) + \lambda_i(B) \end{cases} \qquad \lambda_i(A+B) \leq \begin{cases} \lambda_i(A) + \lambda_P(B) \\ \lambda_{i+1}(A) + \lambda_{P-1}(B) \\ \dots \\ \lambda_P(A) + \lambda_i(B) \end{cases}$$

From the left side inequality of the expressions above; as all the $\lambda_i(A) \geq 0$ and $\lambda_i(B) > 0$, it is concluded that all the eigenvalues of the matrix G of this metric are positive, therefore the matrix is defined positive, and its rank is P.

Alternatively to the Weyl's theorem, eq. (9) and (10) can be interpreted as the regularization of the singular matrix G_1 by means of the sum of a scalar matrix G_2.

Spaces in $\mathcal{R}^P$ in which the differential distance is measured using an expression of the form (8), where G is symmetric, differentiable at least twice, and its determinant is nonzero, are called Riemann spaces, and G is its fundamental or metric tensor (covariant tensor of second order).

B. Distance between two points

The length of an arc of a curve between two points is calculated by:

$$L = \int_{\lambda_i}^{\lambda_f} ds = \int_{\lambda_i}^{\lambda_f} \sqrt{\sum_{i,j} g_{ij} dx^i dx^j} \tag{11}$$

performing a parametric integration along the geodesic that connects both points.

In any metric, the distance between two points should be measured by the shortest route. A geodesic is the curve that, for two points sufficiently closed, its length is minimal among all the curves joining these two points.

From (11), and using the calculus of variations, it is shown that a geodesic should fulfil the following differential equation:

$$\ddot{x}_k + \sum_{i,j} \Gamma_{ij}^k \dot{x}_i \dot{x}_j = 0 \tag{12}$$

The derivatives are with respect to (w.r.t.) the parameter of integration and the Γ_{ij}^{k} are the 2nd kind Christoffel symbols:

$$\Gamma_{ij}^{k} = \frac{1}{2} \sum_{l=1}^{n} g^{kl} \left(\frac{\partial g_{jl}}{\partial x_i} + \frac{\partial g_{il}}{\partial x_j} - \frac{\partial g_{ij}}{\partial x_l} \right) \tag{13}$$

where:

- g_{ij} are the elements of the covariant metric tensor.
- g^{ij} are the elements of contravariant tensor. This tensor is computed by inverting the matrix G (which is always possible since G is nonsingular).

IV. The LOM Metric When Using a Separation Function from a SVM

Consider a support vector machine as the prior classification algorithm. The expression of the classification decision function is:

$$f(\mathbb{X}) = \sum_{i=1}^{nSV} \alpha_i \, K\left(\mathbb{X}, \mathbb{X}_{SV_i}\right) + b \tag{14}$$

where:

- nSV is the number of support vectors (SV).
- α_i is the weight of the i^{th} SV.
- $\mathbb{X}$ is the new point to be classified.
- $\mathbb{X}_{SV_i}$ is the i^{th} SV.
- $K\left(\mathbb{X}, \mathbb{X}_{SV_i}\right) = e^{-\frac{\left\|\mathbb{X}-\mathbb{X}_{SV_i}\right\|^2}{\sigma^2}}$ is the RBF kernel function that is applied on the $\mathbb{X}$ point and the i^{th} SV
- b is the independent term that adjusts the decision function to be 0 in the frontier between classes.

Now, the different terms that are used in (9) are derived. The gradient (un-normalized), and its derivative w.r.t. x_k:

$$Gr_j = \frac{\partial f(\mathbb{X})}{\partial x_j} = \sum_{i=1}^{nSV} \alpha_i \, K\left(\mathbb{X}, \mathbb{X}_{SV_i}\right) . \left(\frac{-2}{\sigma^2}\right) \left(\mathbb{X}_{x_j} - \mathbb{X}_{SV_{i_{x_j}}}\right) \tag{15}$$

$$\begin{aligned} \frac{\partial Gr_j}{\partial x_k} = &\sum_{i=1}^{nSV} \alpha_i \, K\left(\mathbb{X}, \mathbb{X}_{SV_i}\right) \left(\frac{-2}{\sigma^2}\right) \\ &* \left[\left(\frac{-2}{\sigma^2}\right) \left(\mathbb{X}_{x_j} - \mathbb{X}_{SV_{i_{x_j}}}\right) \left(\mathbb{X}_{x_k} - \mathbb{X}_{SV_{i_{x_k}}}\right) + \delta_{jk} \right] \end{aligned} \tag{16}$$

The normalized gradient, and its derivative w.r.t. x_k:

$$gr_j = \frac{Gr_j}{\sqrt{\sum_{i=1}^{P} {Gr_i}^2}} \tag{17}$$

$$\frac{\partial gr_j}{\partial x_k} = \frac{ \frac{\partial Gr_j}{\partial x_k} \sqrt{\sum_{i=1}^{P} {Gr_i}^2} - \frac{ Gr_j \left(\sum_{i=1}^{P} Gr_i \frac{\partial Gr_i}{\partial x_k} \right) }{ \sqrt{\sum_{i=1}^{P} {Gr_i}^2} } }{ \sum_{i=1}^{P} {Gr_i}^2 } \tag{18}$$

According with the LOM metric definition (2):

$$\frac{\partial \left(\frac{1}{r_M^2}\right)}{\partial x_j} = \frac{-2}{r_M^3} \frac{\partial r_M}{\partial x_j} \qquad \frac{\partial \left(\frac{1}{r_m^2} - \frac{1}{r_M^2}\right)}{\partial x_j} = \frac{2}{r_M^3} \left(\left[\frac{r_M}{r}\right]^4 + 1 \right) \frac{\partial r_M}{\partial x_j} \tag{19}$$

$$\frac{\partial r_M}{\partial x_j} = -2\, r \tau e^{-f^2(\mathbb{X})} f(\mathbb{X}) Gr_j = -2(r_M - r) f(\mathbb{X}) Gr_j \tag{20}$$

Also, it is possible to calculate the derivatives of the elements of the metric tensor w.r.t. the different coordinates:

$$\begin{aligned} \frac{\partial g_{ij}}{\partial x_k} = &\frac{2}{r_M^3} \left(\left[\left(\frac{r_M}{r}\right)^4 + 1 \right] gr_i \, gr_j - \delta_{ij} \right) \frac{\partial r_M}{\partial x_k} \\ &+ \left(\frac{1}{r_m^2} - \frac{1}{r_M^2} \right) \left(gr_i \frac{\partial gr_j}{\partial x_k} + gr_j \frac{\partial gr_i}{\partial x_k} \right) \end{aligned} \tag{21}$$

With all the above results, it is possible to calculate the Christoffel symbols and the geodesic that connects two points.

V. Distance Between Points in the LOM Metric

There are two major approaches to calculate distances based on a metric that varies locally: to approximate the integration of (12), or employ an algorithm that evaluates distances for a grid of points and then, choose the shortest path between origin and destination.

A. Distance calculated by integrating the geodesic

The integration of (12) can be accomplished by many methods. Perhaps the simplest one is:

- Discretize the path between points in a finite number of points: $\boldsymbol{x}^m,\ 0 \le m \le M$. The initial and final points $\boldsymbol{x}^0$ and $\boldsymbol{x}^M$, are fixed points. x_k^m is the k^{th} coordinate of the $\boldsymbol{x}^m$ point.
- Set the equivalent differences equation at each point:

$$\begin{aligned} x_k^m \leftarrow &\frac{\left(x_k^{m+1} + x_k^{m-1}\right)}{2} \\ &+ \frac{1}{8} \sum_{i,j} \Gamma_{ij}^{k}(\boldsymbol{x}^m) \left(x_i^{m+1} - x_i^{m-1}\right)\left(x_j^{m+1} - x_j^{m-1}\right) \end{aligned} \tag{22}$$

- Iterate until the equation is satisfied for all the points within a preestablished error.
- The resulting points profile the geodesic between the initial and final points.

B. Distance calculated by the shortest path

To integrate the differential equation of the geodesic does not guarantee finding the shortest path between two points. A solution that guarantees it, is to search for the shortest path by a well-known algorithm as the one proposed by Dijkstra [17].

VI. Measures of Performance

A two attributes synthetic problem is defined. It resembles an ill-behaved, non-linearly separable classification problem, with sufficient complexity to be interesting. It could be easily reproducible and the number of cases may vary at will in a repeatable way. Thus, a two class problem with six hundred cases for each class is generated:

- The first class consists of data taken from three independent Gaussian distributions with means:

  $$\begin{bmatrix} 0 \\ 2 \end{bmatrix}, \ \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \ \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

  and covariance matrices:

  $$\begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix}, \ \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}, \ \begin{bmatrix} 1 & -\sqrt{1/2} \\ -\sqrt{1/2} & 1 \end{bmatrix}$$

- The second class is formed with an equal amount of data, but taken from only one Gaussian distribution with mean and covariance: $\begin{bmatrix} 0 \\ 2 \end{bmatrix}, \ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

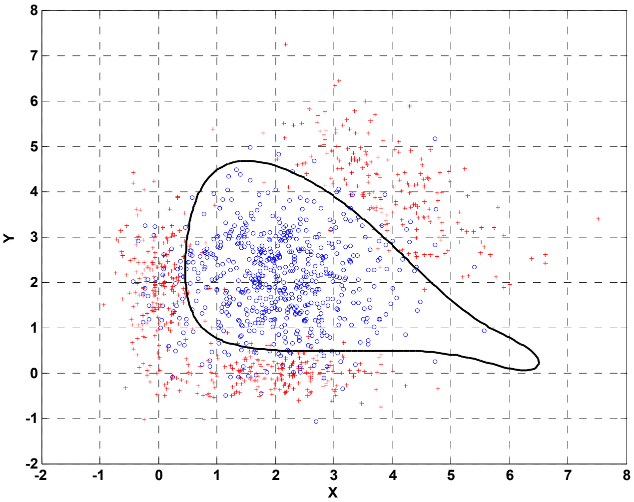

Figure 3. Graphical representation of the synthetic problem. The round points belong to the class -1, the crosses to the class +1.

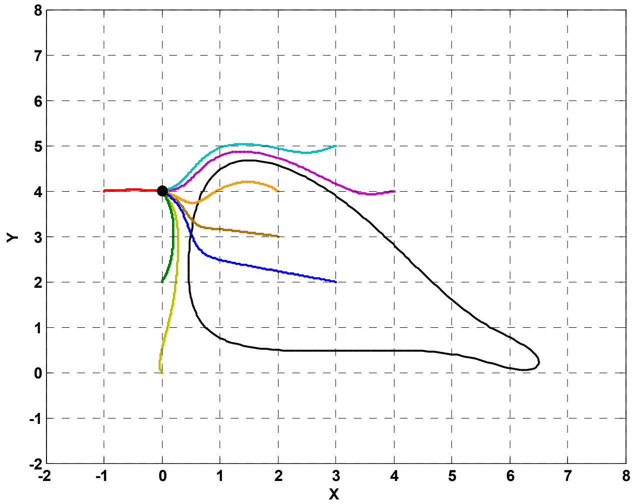

Figure 4. Geodesic curves to reach different points from (0,4) according with the LOM metric.

Fig. 3 provides a graphical representation of these six hundred cases. It also includes a closed curve that depicts the class separation function obtained from a RBF-SVM (with parameters C=4, γ=2).

One of the advantages of a synthetic problem is that it is possible to calculate the expected error for classifying a new case. In our problem, the Bayes error is 10.22%.

By a ten-fold cross-validation technique, the parameters of a SVM based on a RBF kernel are adjusted. The optimal parameters for this SVM are: C = 4, γ = 2, showing an average error in the classification of 10.75%. Therefore, it is only at 0.53% of what can be considered the optimal classification.

A. Geodesic curves in the LOM metric for this test bed

Using the decision curve provided by the SVM, the LOM metric can be defined. Then, it is possible to calculate the distances from an arbitrary point (in Fig. 4 is the point [0,4]) to any other points in the plane by integrating the differential equation of the geodesic (22).

It can be clearly seen, that the traditional straight lines between points in a Euclidean metric have been replaced by curves that follow profiles similar to the decision function to reach their destinations.

This means, that the shorter paths among points prefer to surround the surface of separation between classes (instead of crossing it). Therefore, under this metric, the points that are located in paths parallel to the boundary show greater similarity, and smaller distance, to the case to be classified.

The three main drawbacks of this technique are:

- A geodesic trajectory is not guaranteed to be the shortest path.
- The time required for calculating the distance among all the cases of the problem. A differential equation must be integrated for each pair of points.
- The geodesic trajectory may vary depending on the initial path considered for the integration of the differential equation.

B. Dijkstra's shortest path between two points

In this research, we have implemented a variation of the Dijkstra's algorithm that uses a priority queue to accelerate calculations and allows diagonal paths between the elements of the grid. The results can be seen in Fig. 5. The results agree with those obtained by integrating the geodesic equation.

In this case, it is possible to calculate the distance between one point and the rest of the points in the space of attributes; and it is also easy to draw the contour of the isolines for a given point.

Thus, the objective has been achieved, the isolines of the metric are not simple circles, nor ellipsoids fixed at the origin (as it happens in the current most advanced algorithms), but curves better adapted to each problem. It can be seen that trips to relatively close or distant points according to the Euclidean metric, become larger or smaller distances according in which direction is travelled from the starting point.

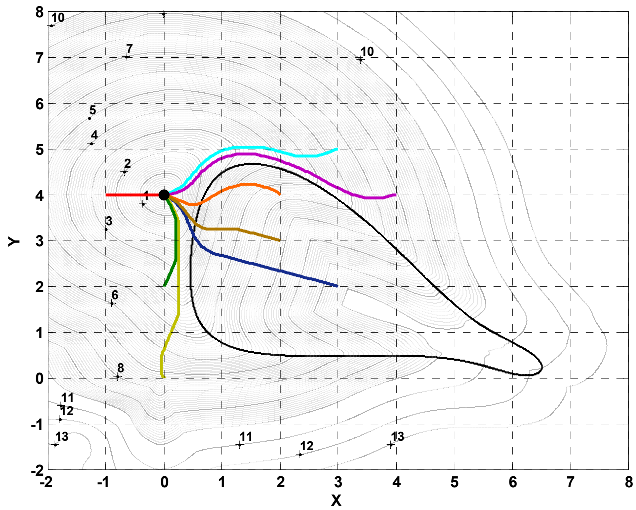

Figure 5. Shortest paths to reach different points from (0,4) using the LOM metric. It can be seen also the isolines that join equidistant points.

 ISBN: 978-1-61208-386-5

C. Using the LOM metric in the k-NN algorithm

We prepared four sets of 180, 90, 60 and 30 cases of the two class problem described in Section VI.A to train four different k-NN classifiers to test the performance of the new metric; and another test set (independent of those enumerated above) containing over 6000 cases. The aim was to show not only that the classification accuracy improves when using the LOM metric, but also that this is true when either the number of training cases is high or low.

Therefore, different SVMs for each problem were trained, and then the k-NN algorithm was applied (using both the Euclidean and LOM metrics) to classify each of the 6,000 test cases. In this study, to calculate the distance between two points with the LOM metric, the Dijkstra technique was used (we also employed different edge lengths to study the effect of degeneration of distance measurements when this parameter got larger). In Table 1, and in the four graphs of Fig. 6, the results obtained are shown.

In each of the four scenarios (180, 90, 60 and 30 cases), a significant increase of classification accuracy is achieved with the k-NN algorithm which employs the new LOM metric (compared with the results obtained with the k-NN or RBF-SVM algorithms using the Euclidean metric).

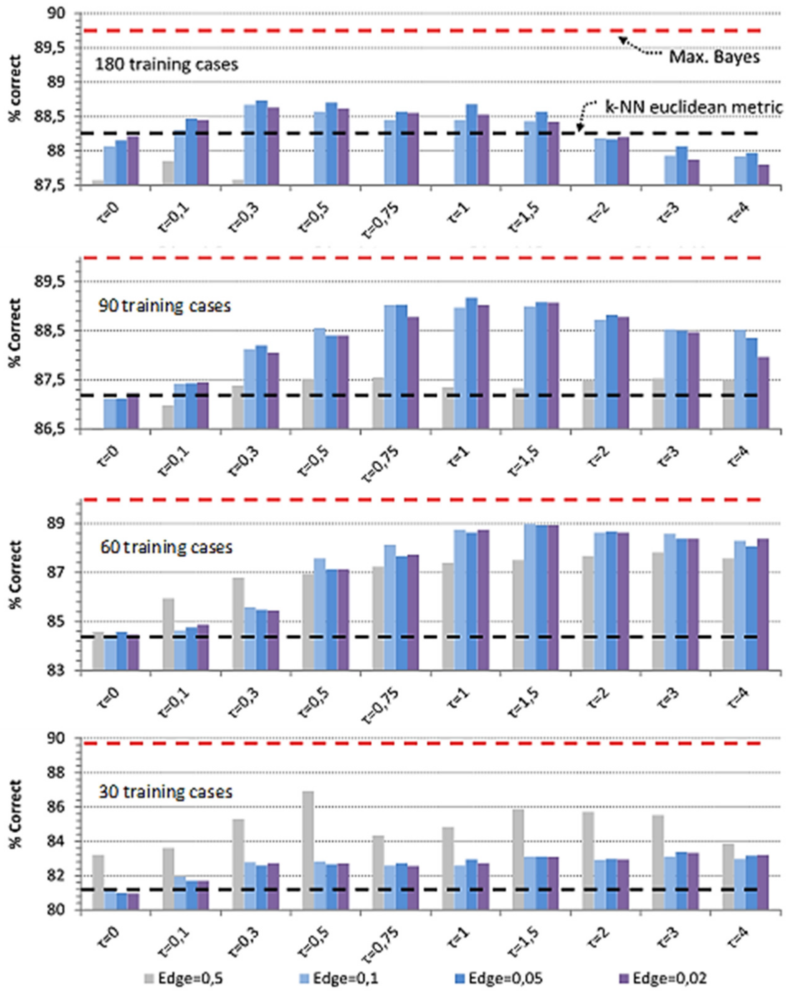

Figure 6. Results of the k-NN classification for the four problems.

TABLE I. SUMMARY OF CLASSIIFICATION ACCURACY

Problem	k-NN euclidean metric	SVM	k-NN LOM metric
180 cases	88.25% (k=13)	88.54%	88.65% (τ=0.2, k=13)
90 cases	87.23% (k=3)	87.53%	89.12% (τ=1.25, k=7)
60 cases	84.40% (k=5)	85.08%	88.98% (τ=1.25, k=13)
30 cases	81.23% (k=1)	82.40%	83.33% (τ=3, k=2)

VII. CONCLUSION AND FUTURE WORK

The main objective of the LOM metric has been achieved; it does not only enlarge the space parallel to the decision function, but it also provides a metric whose isolines are not ellipses centered at the initial point that extend along all the space, but curves which adapt locally to the profile of the decision function.

As it is frequently remarked in the literature, the shortest path between points in a Riemannian metric is a geodesic, but not every geodesic is the shortest path. Several times in this research, we have found two different geodesics to reach the same point. This is due to the different initial path used, and the attractors that govern the integrations of the differential equations. Under this point of view, it is preferable to use the Dijkstra's algorithm.

However, one of the main drawbacks of Dijkstra's algorithm is the time required to calculate the distance between points. Its order of complexity is $O\big((|E| + |V|)log|V|\big)$ using a priority queue, being $|E|$ the cardinality of edges and $|V|$ the same measure for the vertices.

Analyzing the results obtained with the k-NN algorithm, some aspects become relevant:

- It is not necessary to explore different values for the *r* parameter (for k-NN, *r* is just a scale factor).
- The range of the τ parameter for which significant improvements are obtained is wide. This is because the LOM metric is derived from a well-founded theoretical concept.
- Even for edges of 0.1 units, the distance calculations by the Dijkstra's algorithm are correct enough (the range of the attributes in each dimension is between -1 and 6). Shorter edges do not provide a significant improvement of accuracy and only increase calculation time.

One aspect for further research is the study of the degradation of Dijkstra's algorithm when used in spaces with more dimensions than two. But, perhaps the most relevant area of improvement is to define a simplification of the LOM metric that allows its use in multidimensional spaces with reasonable time and memory constraints. We are currently working in one of them and our first results are very promising.

As a conclusion, calculation of distances using the LOM metric is seen as an alternative for those algorithms that use this measurement in their decision making. It increases the classification accuracy and reduces the curse of dimensionality. The main drawback of the LOM metric is that more extensive calculations are needed to obtain the distances. Currently, we are working to minimize this inconvenience.

References

[1] R. Duda, P. Hart, and D. Stork, "Pattern Classification" 2nd Ed. John Wiley, New York, 2000.

[2] C. Domeniconi, D. Gunopulos, and J. Peng, "Large Margin Nearest Neighbor Classifiers", IEEE transactions on Neural Networks, vol.16, 2005, pp. 899-909.

[3] J. Peng, D. R. Heisterkamp, and H. K. Dai, "LDA/SVM Driven Nearest Neighbor Classification", IEEE Transactions on Neural Networks, vol.14, 2003, pp. 940-942.

[4] J. Revilla and E. Kahoraho, "BTW: a New Distance Metric for Classification", Proc. of the International Symposium on Distributed Computing and Artificial Intelligence, DCAI 2012, 2012, pp. 701-708.

[5] K. Q. Weinberger and L. K. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification", Journal of Machine Learning Research, Vol.10, 2009, pp. 207-244.

[6] S. Shalev-Shwartz, Y. Singer, and A. Y. Ng, "Online and Batch Learning of Pseudo-metrics". Proc. of the 21st Int. conference on machine learning, Banff, Canada, 2004, pp. 94-101.

[7] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. "Learning a Mahalanobis metric from equivalence constraints". Journal of Machine Learning Research 6, 2006, pp. 937-965.

[8] N. Shental, T. Herz, D. Weinshall, and M. Pavel, "Adjustment learning and relevant component analysis", Proc. of the 7th European conference on computer vision, London, UK, 2002, pp. 776-790.

[9] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood Component Analysis", Advances in neural information processing systems vol. 17, 2005, pp. 13-520.

[10] Q. Chang, Q. Chen, and X. Wang, "Scaling Gaussian RBF Kernel Width to Improve SVM Classification", Int. Conf. on neural networks and brain, ICNN&B '05 vol. 1, 2005, pp. 19-22.

[11] S. Amari and S. Wu, "Improving Support Vector Machine Classifiers by Modifying Kernel Functions", Neural Networks vol.12, 1999, pp. 783-789.

[12] S. Wu and S. Amari, "Conformal Transformation of Kernel Functions: a Data-Dependent Way to Improve Support Vector Machine Classifiers", Neural processing letters, vol. 15, 2002, pp. 59-67.

[13] G. Wu and E. Chang, "Adaptive Feature-space Conformal Transformation for Imbalances-data Learning", 20th Int. Conf. on Machine Learning (ICML-2003), 2003, pp. 816-823.

[14] P. Williams, S. Li, J. Feng, and S. Wu, "Scaling the Kernel Function to Improve Performance of the Support Vector Machine", Advances in Neural Networks, ISNN'05, 2005, pp. 831-836.

[15] F. Fernandez and P. Isasi, "Local Feature Weighting in Nearest Prototype Classification", IEEE Transactions on Neural Networks, vol.19, 2008, pp. 40-53.

[16] Y. Zhang, H. Zhang, N. M. Nasrabadi, and T. S. Huang, "Multi-metric Learning for Multi-sensor Fusion Based Classification", Information Fusion, vol. 14, 2013, pp. 431-440.

[17] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, "Introduction to Algorithms", MIT Press, Massachusetts. 2000.