Non-Euclidean or Non-metric Measures Can Be Informative

Elżbieta Pękalska[1,2], Artsiom Harol[1], Robert P.W. Duin[1], Barbara Spillmann[3], and Horst Bunke[3]

[1] Faculty of Electrical Engineering, Mathematics and Computer Sciences, Delft University of Technology, The Netherlands
[2] School of Computer Science, University of Manchester, United Kingdom
[3] Institute of Computer Science and Applied Mathematics, University of Bern, Switzerland
{e.m.pekalska, a.harol}@tudelft.nl, r.duin@ieee.org, {spillman, bunke}@iam.unibe.ch

Abstract. Statistical learning algorithms often rely on the Euclidean distance. In practice, non-Euclidean or non-metric dissimilarity measures may arise when contours, spectra or shapes are compared by edit distances or as a consequence of robust object matching [1,2]. It is an open issue whether such measures are advantageous for statistical learning or whether they should be constrained to obey the metric axioms.

The k-nearest neighbor (NN) rule is widely applied to general dissimilarity data as the most natural approach. Alternative methods exist that embed such data into suitable representation spaces in which statistical classifiers are constructed [3]. In this paper, we investigate the relation between non-Euclidean aspects of dissimilarity data and the classification performance of the direct NN rule and some classifiers trained in representation spaces. This is evaluated on a parameterized family of edit distances, in which parameter values control the strength of non-Euclidean behavior. Our finding is that the discriminative power of this measure increases with increasing non-Euclidean and non-metric aspects until a certain optimum is reached. The conclusion is that statistical classifiers perform well and the optimal values of the parameters characterize a non-Euclidean and somewhat non-metric measure.

1 Introduction

Many currently available data are non-vectorial by origin. Although some ways exists to represent particular information in a vectorial form, these may be unnatural, of poor quality for the final prediction or very difficult to obtain. Vectorial representations are convenient since there exists a plethora of powerful learning techniques [4]. These are developed in inner product spaces or normed spaces, in which the inner product or norm defines the corresponding metric. On the other hand, if objects contain an inherent, identifiable structure or organization such as contours, shapes, spectra, images or texts, then structural descriptions are advisable. Objects can then be compared by suitable (min-max or edit) distances.

D.-Y. Yeung et al. (Eds.): SSPR&SPR 2006, LNCS 4109, pp. 871–880, 2006.

In other words, a collection of objects can be represented in a relative way, by a vector of dissimilarities (proximities) to a given set of representative examples. This is the so-called dissimilarity (proximity) representation [5,3]. Since proximity can be defined in both quantitative and qualitative contexts, it becomes a natural bridge between structural and statistical pattern recognition.

Kernel methods offer also an alternative to vectorial representations [6]. A kernel K is a (conditionally) positive definite (cpd) function of two variables, interpreted as a generalized inner product, hence similarity, in a Hilbert space $\mathcal{H}$ induced by K. Thanks to the reproducing property of K, the support vector machine (SVM) is built in $\mathcal{H}$ as a linear combination of kernel values to the so-called support vectors. The class of admissible kernels is, however, very limited due to the strong requirement of their being cpd. This is equivalent to stating that the corresponding distance $d(x,y) = \sqrt{K(x,x)+K(y,y)-2K(x,y)}$ is Euclidean for finite kernels [3]. In our terminology, kernels are an example of general proximity representations for which other learning strategies can successfully be applied.

Although proximity measures are widely used for matching and object comparison [1,2,7], classification often relies on assigning a new object to the class of its nearest neighbor. Alternative generalization frameworks exist that handle general proximity measures. They represent dissimilarity information in suitable representation vector spaces [5,8,9,3] or deal with indefinite kernels [10,3]. In case of non-Euclidean or non-metric dissimilarity data, researches usually either rely on the nearest neighbor distances, or choose to constrain/correct the measure to make it obey the metric axioms, e.g. by adding an appropriate constant or using a suitable transformation. In kernel methods, this is equivalent to regularizing the kernel by adding a proper constant to the diagonal.

If a highly non-metric/non-Euclidean measure describes the problem well (as judged by experts), corrections will likely lead to a significant loss of information [11,10,8]. If such deviations are small, they may be neglected as noise. Understanding is, therefore, necessary to identify under which circumstances and to what extent non-metric or non-Euclidean measures are advantageous in statistical learning. We contribute to this issue by presenting an empirical study in which the performance of dissimilarity-based statistical classifiers is related to indices measuring their departure from the Euclidean or metric behavior.

2 Representation Spaces and Classifiers

Assume a training set $T = \{t_1, \ldots, t_N\}$ of N objects and a representation set $R = \{p_1, p_2, \ldots, p_n\} \subseteq T$ of n prototypes. Given a dissimilarity measure d, a dissimilarity representation is an $N \times n$ matrix $D(T,R)$ with the elements $d(t_i, p_j)$. An object $t_i \in T$ is represented by an n-element vector of dissimilarities $D(t_i, R)$. The k-NN rule can directly be applied to such data. While it has good asymptotic properties for metric data, its performance deteriorates for small training (representation) sets. In such cases, alternative learning strategies can be more advantageous. They determine a suitable vector space equipped with the algebraic structure of either an inner product or norm in which the proximity information is represented. In such a vector space, the traditional

learning algorithms can appropriately be adapted. Two simplest approaches are a linear isometric embedding into a pseudo-Euclidean space or the use of the so-called dissimilarity spaces [12,5,3].

In this paper, $D(\cdot, R)$ is interpreted as a data-dependent mapping $D(\cdot, R) : X \to \mathbb{R}^n$ from some initial representation X (such as a vector space, images, strings or graphs) to a vector space defined by R. This is the *dissimilarity space*, in which each dimension $D(\cdot, p_i)$ corresponds to a dissimilarity to a prototype $p_i \in R$. The property that dissimilarities should be small for similar objects (belonging to the same class) and large for distinct objects, gives them a discriminative power. Hence, $D(\cdot, p_i)$ can be interpreted as 'features' and traditional classifiers built in vector spaces can be adapted [9,3]. The simplest are linear and quadratic classifiers, which are weighted combinations of the dissimilarities $d(x, p_i)$ between an object x and the prototypes p_i. The classifiers are optimized on $D(T, R)$, hence on the complete set T, even if only R is used for their representation. They can outperform the k-NN rule since they become more global in their decisions (suppressing the influence of individual noisy examples).

Classifiers. Normal density based (Bayesian) classifiers [4] tend to perform well in dissimilarity spaces [3,5,9]. This especially holds for summation-based dissimilarity measures, summing over a number of components with similar variances. The reason is that such dissimilarities will be approximately normally distributed thanks to the central limit theorem (if one or few variances are dominant, then they will approximate the χ^2 distribution) [3].

For a two-class problem, a quadratic normal density based classifier (NQC), is given by $f(D(x,R)) = \sum_{i=1}^{2} \frac{(-1)^i}{2}(D(x,R)-\mathbf{m}_i)^\mathsf{T} S_i^{-1} (D(x,R)-\mathbf{m}_i) + \log \frac{p_1}{p_2} + \frac{1}{2}\log \frac{|S_1|}{|S_2|}$, where $\mathbf{m}_i$ are the mean vectors and S_i are the class covariance matrices, all estimated in the dissimilarity space $D(\cdot, R)$. p_1 and p_2 are the class prior probabilities. If S_1 and SS_2 are replaced by the average covariance matrix, then a linear classifier is obtained. If the covariance matrices become singular, they need to be regularized. Here, we choose the following regularization $S_i^\kappa = (1-\kappa)S_i + \kappa p_i \operatorname{diag}(S_i)$, $\kappa \in [0,1]$, which leads to the RNQC, i.e. the regularized NQC. In our implementation, the normal-density functions are estimated per class and the final decision relies on the maximum a posteriori probability.

Another useful strategy for dissimilarity data is offered by sparse linear programming machines (LPM), which construct hyperplanes in the corresponding dissimilarity spaces. They are able to automatically determine a prototype set R (or if trained on $D(T, R)$, they may reduce the set R further on) which defines the final classifier. Two variants are here considered: the μ-LPM and the auc-LPM. The μ-LPM is a form of the ℓ_1-SVM with $\mu \in [0, 1)$, where μ is related to the expected classification error [13,9]. The auc-LPM is defined to maximize the area under the ROC curve, as recently proposed in [14].

We also define a new linear classifier, which is a nonnegative least square classifier (NLSQC). Let D denote $D(T, R)$, $R \subseteq T$, $|T| = N$ and $|R| = n$. Consider a two-class problem with the corresponding labels $y_i = +1/-1$. Let $Y_T = \operatorname{diag}(\mathbf{y}^T)$ and $Y_R = \operatorname{diag}(\mathbf{y}^R)$, where $\mathbf{y}^T$ and $\mathbf{y}^R$ are the label vectors for the sets

T and R, respectively. We define our classifier as $f(D(x,R) = sign(h(D(x,R)))$, where $h(D(x,R)) = -\mathbf{w}^{\mathsf{T}} Y_R D(x,R) + w_0$, $w_i \geq 0$, $i = 0, 1, \ldots, n$. (Since w_i are multiplied by y_i^R, so $w_i y_i^R$ can be of any sign.) The classifier will assign 1 to x if $h(D(x,R)) = a > 0$ and –1 if $h(D(x,R)) = a < 0$. By fixing $a = 1$, it yields $y_i^T h(D(t_i,R)) > 1$ for the training objects t_i. The weights are now sought to minimize the sum of square differences $(y_i^T h(D(t_i,R)) - 1)^2$ for all t_i. We formulate the following problem:

$$\mathrm{Min}_{\mathbf{w}} \;\; ||D_{YY} \left[\begin{smallmatrix} \mathbf{w} \\ w_0 \end{smallmatrix} \right] - \mathbf{1}||_2^2, \text{ subject to } \quad w_i \geq 0, \quad i = 0, 1, \ldots, n \qquad (1)$$

where $D_{YY} = [Y_T(-D)Y_R \; -\mathbf{y}^T]$ and $\mathbf{1}$ is a vector of all ones. This can be solved by a standard nonnegative least square method that gives a sparse solution in terms of R. The non-zero weights correspond to the selected subset R' of R. In our case, the sparsity will not be large because of the choice of $-D$ in D_{YY} (or, equivalently, because of non-positive weights $-\mathbf{w}$ in the function h). In this quadratic criterion, $-D$ acts as a similarity (large values in $-D$, hence small distances, indicate large similarity) and requires many objects of R to support the decision boundary. On the contrary, if we choose D instead $-D$ in $h(D(x,R))$, i.e. $D_{YY} = [Y_T D Y_R \; \mathbf{y}^T]$, this will lead to a *very sparse* solution determined by dominating, possibly outlier distances only, hence to a poor discrimination. Non-positive weights $-w_i$ diminish the influence of large distances and shift the 'focus' towards the objects with small distances to the other class.

Equation (1) can be extended to $\mathbf{w}^{\mathsf{T}}(Y_R D^{\mathsf{T}} D Y_R)\mathbf{w}^{\mathsf{T}} + 2\mathbf{w}^{\mathsf{T}} Y_T D Y_R \mathbf{1} + 2\mathbf{1}^{\mathsf{T}} \mathbf{y}^T w_0 + N(w_0^2+1)+1$, in which the first term is the same as in the formulation of a linear SVM defined in a 'feature space' $X = D$. (There, in the dual problem, one would minimize $\frac{1}{2}\mathbf{w}^{\mathsf{T}}(Y_R D^{\mathsf{T}} D Y_R)\mathbf{w}^{\mathsf{T}} - \mathbf{w}^{\mathsf{T}}\mathbf{1}$ given that $\mathbf{w}^{\mathsf{T}}\mathbf{y}^R = 0$ and $w_i \geq 0$ [6].) Such an SVM would work in a *entire* dissimilarity space as it selects the support vectors in the form of $D(t_j, R)$ from T (and not from R)! Hence, a linear SVM in a dissimilarity space is not sparse. The advantage of the NLSQC is that it is a linear function with no additional parameters, which optimizes a square error and is, thereby, competitive to a quadratic classifier. Although it cannot outperform the SVM, it may compete with other LPMs applied to dissimilarity data. These LPMs are usually trained on a complete representation $D(T,T)$ and determine both R and the weights of the classifiers. These representation sets R may be used to train the NLSQC on $D(T,R)$ to enhance the sparsity.

3 Indices Characterizing Data

Assume K classes, $\omega_1, \omega_2, \ldots, \omega_K$ such that $|\omega_i| = n_i$ and $N = \sum_i n_i$. Two indices are defined to reflect the class separability. The first one is $J^1_{\text{sep}} = \frac{\sum_{i=1}^K n_i A_{ii}}{\sum_{i=1}^K n_i/(N-n_i) \sum_{j \neq i} n_j A_{ij}} \in (0,1)$, where A_{ij} is the average dissimilarity between the i-th and j-th classes (hence A_{ii} is the between-class average dissimilarity). The second index focusses on the nearest neighbor distances. $J^2_{\text{sep}} = \frac{1}{K}\sum_{i=1}^K B_i$, where $B_i = \frac{1}{n_i}\sum_{k=1}^{n_i} \frac{\min_{x \in \omega_i} d_{NN}(t_k,x)}{\min_{z \notin \omega_i} d_{NN}(t_k,z)}$ is the average ratio of the nearest neighbor

within-class to between-class distances. The smaller the values, the better separability. Note that if $J^2_{\text{sep}} \approx 1$ or larger, than (on average) the nearest neighbor distances to objects from other classes are similar or smaller than the nearest neighbor distances within the class, hence the 1-NN rule cannot perform well.

Concerning the departure from the Euclidean behavior, it is known that a symmetric $N \times N$ dissimilarity matrix $D = D(T, T)$ has a Euclidean behavior iff the corresponding Gram matrix $G = -\frac{1}{2} J D^{\star 2} J$, where $D^{\star 2} = (d^2_{ij})$ and $J = I - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathsf{T}}$, is positive semidefinite [12,5,3]. It means that all eigenvalues λ_i of G are non-negative. Hence, the magnitudes of negative eigenvalues manifest the amount of deviation from the Euclidean behavior. An indication of such a deviation is given by $J_{\text{eigM}} = |\lambda_{min}|/\lambda_{max}$, i.e. the ratio of the absolute value of the smallest negative eigenvalue to the largest positive one. The overall contribution of negative eigenvalues is estimated by $J_{\text{eigS}} = \sum_{\lambda_i<0} |\lambda_i| / \sum_{j=1}^{N} |\lambda_j|$.

Concerning non-metric aspects, any symmetric D can be made metric by adding a suitable constant γ to all off-diagonal elements of D. In a first attempt, such a constant can be found as $\gamma_0 = \max_{p,q,t} |d_{pq} + d_{pt} - d_{qt}|$ [3]. This estimation is however largely overpessimistic. Starting from this initial γ_0, we find a better estimation of $\gamma \in (0, \gamma_0)$ by an iterative bisection method. Our index is therefore $J_\gamma = \gamma \geq 0$ and it should be judged wrt the actual dissimilarity values. Another way to characterize the deviation from the metric behavior is by J_{ineq} equal to the total number of disobeyed triangle inequalities.

4 Data, Experiments and Results

In our study we use the Chicken Pieces Silhouettes data [15], available from `http://algoval.essex.ac.uk/data/sequence/chicken`. This set consists of 446 binary images from chicken pieces, labeled to one of the five classes,which represent specific parts of the chicken: wing (117 examples), back (76), drumstick (96), thigh and back (61), and breast (96). After edge detection applied to these silhouettes, the edges were approximated by straight line segments of a fixed length L, taking values between 5 and 40 pixels. Since chicken pieces are placed in arbitrary position in an image and mirror symmetry occurs, the line segments may not be the most appropriate. Instead, the sequence of angles between the neighboring segments was chosen as the initial string representation. Additionally, the approximate algorithm of Bunke and Bühler [16] was applied to handle the rotation invariance and axis symmetry. Given such string representations a family of edit distances [17] is considered with fixed insertion and deletion costs equal to some C and a substitution cost of the absolute difference between the angles. Consequently, we deal with an (L, C)-family of edit distance measures parameterized by L and C. In our case, we set $L = 5, 7, 10, 15, 20, 25, 29, 30, 31, 35, 40$ pixels and $C = 45, 60, 90, 120$ (angle degrees), which give rise to 44 different dissimilarity data, in total. The distances were originally asymmetric and are made symmetric by averaging, $d_{ij} = \frac{d_{ij}+d_{ji}}{2}$.

In our classification experiments we perform 50 runs of 2-fold cross-validation. In each run, all objects are first randomly split into the training set T and test set S. Then, classifiers are trained on $D_{L,C}(T, T)$ and tested on $D_{L,C}(S, T)$.

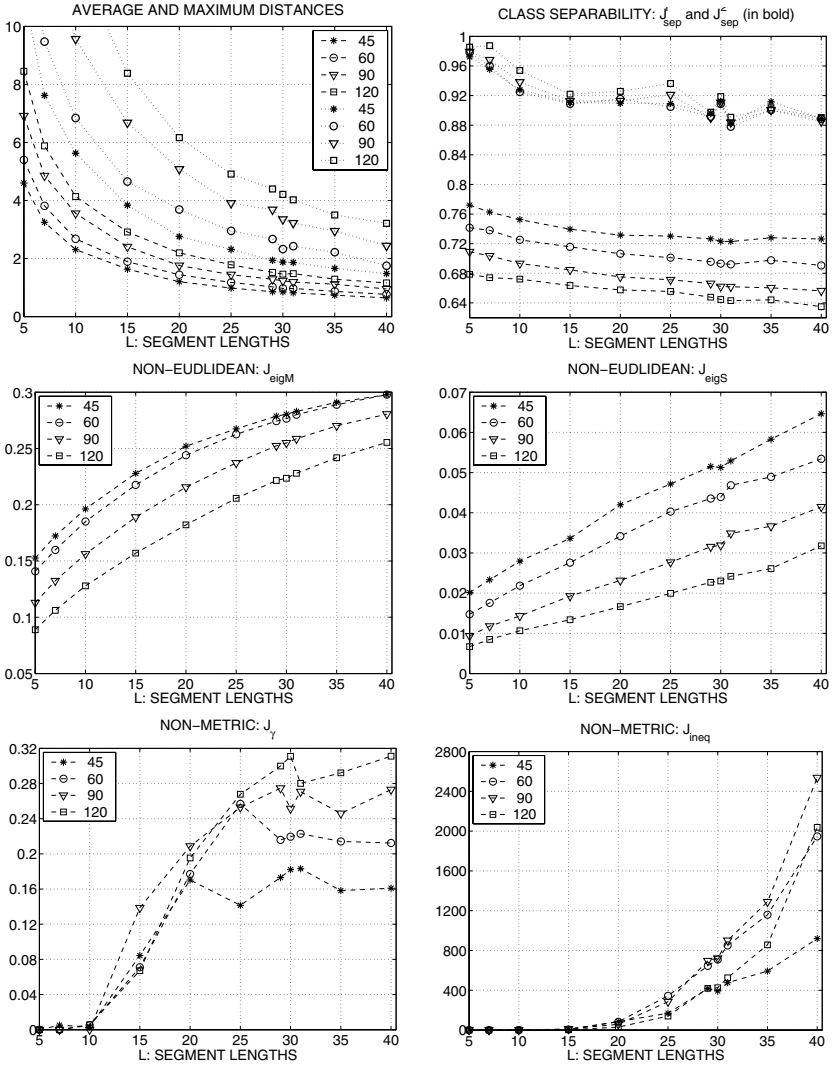

Fig. 1. Indices characterizing dissimilarity data. Legend values refer to C. Markers describing the same value of C are connected by lines to enhance the visibility.

Next, in the second fold, the classifiers are trained on $D_{L,C}(S,S)$ and tested on $D_{L,C}(T,S)$. Finally, the errors, weighted by prior probabilities, are determined. This is repeated 50 times and the results are averaged out. To avoid too large distance values, all dissimilarities are scaled by $\frac{1}{\sqrt{N}}$, where $N = |T|$. The following classifiers are used: the 1-NN and k-NN rules directly applied to the dissimilarity representation $D(T,T)$ (k is optimized in a LOO approach), edited-and-condensed nearest neighbor (CNN) [18], μ-LPM, with $\mu = \max\{0.01, 1.3 \cdot \text{LOO-NN-error}\}$, the auc-LPM (with the trade-off parameter

set to 20) [14], the NSQLC and the RNQC with $\kappa = 0.05$. Additionally, the NSQLC is trained on the representation sets determined by the μ-LPM and auc-LPM, and denoted as the NSQLC(μ) and the NSQLC(auc), respectively. Remember that the LPMs and the NSQLC determine $R \subset T$ and that all (multi-class) linear classifiers are derived in an one-against-all strategy.

The properties of dissimilarity data are characterized by the indices introduced in Sect. 2. These will reflect the character of the dissimilarities, the class separability and the deviation from both Euclidean and metric behaviors. The indices are derived in the same setup as above. Their values are first averaged over two folds in a cross-validation scheme, and then over 50 runs.

Results. The indices defined in Sec. 2 were evaluated on 44 dissimilarity data with varying parameters L and C of the (L, C)-edit distances. By observing the results in Fig. 1, the following conclusions can be drawn:

- The average dissimilarities decrease with growing L. The smaller L, the larger maximal distances. The average and maximal distances grow with increasing C.
- The classification task is difficult since J^2_{sep} takes values close to 0.9 or 1. This means that the NN distances within a class are not much smaller than the NN distances to the objects outside the class. $C = 31$ seems to be optimal. Concerning the J^1_{sep}, the smaller C, the better the separability.
- None of the dissimilarity data set has a Euclidean behavior. The deviation becomes larger with the increasing L and decreasing C, as judged by J_{eigS} and J_{eigM}.
- The (L, C)-edit distances are practically metric up to $L \leq 10$; they become non-metric for larger values of L. The deviation from the metric behavior becomes larger with increasing C and is the smallest for $C = 45$. For $L \geq 30$, the additive constant γ that makes the dissimilarity measure metric roughly equals to $16 - 30\%$ of the average distance.

The classification results are shown in Fig. 2. In general, we observe that the performance of all classifiers improves with the increasing value of L up to a certain optimum and then starts to decrease. Most classifiers perform the best or nearly the best for $L = 30$. Concerning C, the classifiers reach the highest accuracy for $C = 45$ and gradually decrease their performance for larger values of C.

We will now provide the average *total* number of prototypes, i.e. $|R|$, determined by sparse linear classifiers. These numbers, presented as '$\cdot - \cdot - \cdot$', are averaged over C as only minor differences occur. The numbers taking the places of the first, second and third dot refer to $L = 5$, $L = 30$ and $L = 40$, correspondingly. We have: the μ-LPM: 223-123-112, the auc-LPM: 120-86-85, the NLSQC: 217-191-188, the NLSQC(μ): 217-116-106 and the NLSQC(auc): 119-84-83. For the CNN, the condensed sets vary over C and vary from 38 to 45.

The CNN relies on the smaller condensed set R but it performs the worst of all. The auc-LPM needs a relatively small R, but it also does not work well; it cannot compete with the 1-NN and k-NN rules unless $L \leq 15$. Other linear

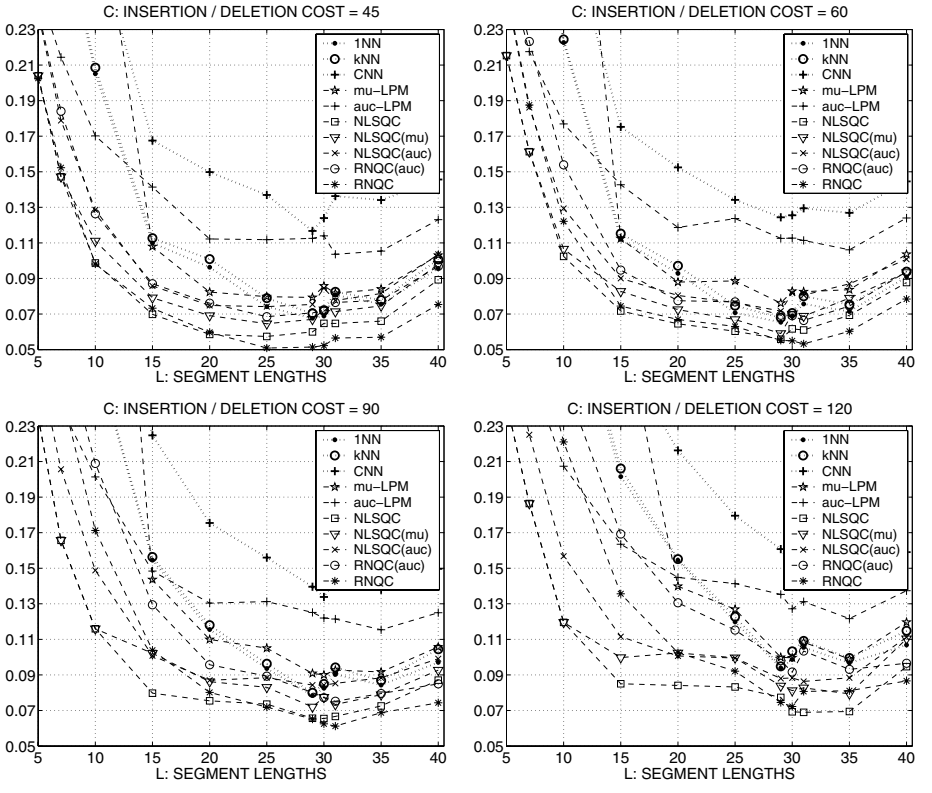

Fig. 2. Average 2-fold cross-validation errors (over 50 runs) for four values of C. Markers describing the same classifier are connected by lines to enhance the visibility. The standard deviations of the average errors are $0.0017 - 0.002$ on average (depending on C) for all classifiers. Their maximal values bounded by 0.0027, except for the μ-LPM, for which the maximal values are 0.011 for $C <= 10$ (where μ-LPM fails). The differences (between the average errors) larger than 0.01 are statistically significant.

classifiers outperform the auc-LPM, expect for the μ-LPM and $L \leq 15$. In general, the μ-LPM does not perform better than the NN rules (with little exceptions) and it deteriorates for $L \leq 15$, which is caused by the fact that the hyperplane cannot be determined (μ-LPM fails) and in our set-up the pseudo-Fisher classifier is automatically trained instead. However, if the representation objects determined by the auc-LPM or the μ-LPM are used to train the NLSQC, the performance drastically increases. The NLSQC(μ) is the third best performing classifier, which provides a good trade-off between the total cardinality of R and the classification accuracy. The representation objects preselected by the auc-LPM seem to make a n over-optimized set for the NLSQC(auc). Interestingly, the performance of the NLSQC(auc) is similar or much better than that of the RQNC(auc). For all C, our NLSQC performs the best or second best, after

the RNQC if $L \geq 30$. Nearly all training objects are, however, needed for the representation. For the RNQC, always holds that $R = T$.

5 Conclusions

In this paper, examples of a parameterized family of (L, C)-edit distances are evaluated for the classification task on chicken pieces silhouettes. The deviation from non-Euclidean behavior grows with increasing L and decreasing C, while the deviation from non-metric behavior grows with both increasing L and C. Linear or quadratic classifiers built in dissimilarity spaces can outperform the direct k-NN rule and reach the optimal (or nearly optimal) results for $L = 30$. Our new linear classifier, the NLSQC, reaches the highest accuracy for most values of L and C. The best overall performance is reached for $L = 30$ and $C = 45$ which gives rise to a highly non-Euclidean and somewhat non-metric dissimilarity data. This is very interesting, since many researchers try to avoid non-metric data and define edit distances as metric measures. Our results suggests that non-Euclidean and/or non-metric distances can be informative and useful in statistical learning. We hope to explore these issues in the future research.

Acknowledgments. This work is supported by the Dutch Organization for Scientific Research (NWO).

References

1. Dubuisson, M., Jain, A.: Modified Hausdorff distance for object matching. In: ICPR. Volume 1. (1994) 566–568
2. Jacobs, D., Weinshall, D., Gdalyahu, Y.: Classification with Non-Metric Dist.: Image Retrieval and Class Representation. TPAMI **22** (2000) 583–600
3. Pękalska, E., Duin, R.: The dissimilarity representation for pattern recognition. Foundations and applications. World Scientific (2005)
4. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Verlag (2001)
5. Pękalska, E., Paclík, P., Duin, R.: A Generalized Kernel Approach to Dissimilarity Based Classification. JMLR **2** (2002) 175–211
6. Schölkopf, B., Smola, A.: Learning with Kernels. MIT Press (2002)
7. Veltkamp, R., Hagedoorn, M.: State-of-the-art in shape matching. Technical Report UU-CS-1999-27, Utrecht University, The Netherlands (1999)
8. Pękalska, E., Duin, R., Günter, S., Bunke, H.: On not making dissimilarities Euclidean. In: S+SSPR. (2004) 1145–1154
9. Pękalska, E., Duin, R., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recognition **39** (2005) 189–208
10. Haasdonk, B.: Feature space interpretation of SVMs with indefinite kernels. TPAMI **25** (2005) 482–492
11. Laub, J., Müller, K.R.: Feature discovery in non-metric pairwise data. JMLR (2004) 801–818
12. Goldfarb, L.: A unified approach to pattern recognition. Pattern Recognition **17** (1984) 575–582

13. Graepel, T., Herbrich, R., Schölkopf, B., Smola, A., Bartlett, P., Müller, K.R., Obermayer, K., Williamson, R.: Classification on proximity data with LP-machines. In: ICANN. (1999) 304–309
14. Tax, D., Veenman, C.: Tuning the hyperparameter of an auc-optimized classifier. In: BNAIC. (2005) 224–231
15. Andreu, G., Crespo, A., Valiente, J.M.: Selecting the toroidal self-organizing feature maps (TSOFM) best organized to object recogn. In: ICNN. (1997) 1341–1346
16. Bunke, H., Bühler, U.: Applications of approximate string matching to 2D shape recognition. Pattern recognition **26** (1993) 1797–1812
17. Bunke, H., Sanfeliu, A., eds.: Syntactic and Structural Pattern Recognition Theory and Applications. World Scientific (1990)
18. Devijver, P., Kittler, J.: Pattern recognition: A statistical approach. Prentice/Hall (1982)