THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 20, Number 2/2019, pp. 197–204

INVARIANT PATTERN RECOGNITION USING SUPPORT VECTOR DATA DESCRIPTION AND TANGENT DISTANCE

Iulian B. CIOCOIU

"Gheorghe Asachi" Technical University of Iasi, Romania
Faculty of Electronics, Telecommunications and Information Technology
E-mail: iciocoiu@etti.tuiasi.ro

Abstract. A gradient-type dynamical system is designed using a data-dependent Lyapunov function constructed using the Support Vector Data Description (SVDD) algorithm. Invariance to standard geometric transformations is inferred by combining SVDD with the tangent distance (TD), which has been shown to yield superior performances against the Euclidean distance in a number of pattern recognition applications. Experimental results using the USPS handwritten characters database and the Olivetti face images database confirm the superiority of the proposed approach over existing solutions.

Key words: gradient-type system, SVDD, tangent distance, invariance.

1. INTRODUCTION

John Hopfield's seminal work in the early 80's triggered a plethora of papers dealing with the subject of *associative memories*. In the training phase, such systems use a diversity of learning procedures in order to store a set of (binary, multi-level, or even continuous) patterns from a given training database. In the recovery phase, the system should deliver previously stored patterns even if adverse input information in terms of noise, missing data, or distortions is provided. Two main approaches have been followed to fulfil this task: a) appropriate information has been used to train *feedforward* networks, speculating their proven generalization capabilities. As such, correct association between (similar or distinct) input and target data may still be valid, despite the fact that corrupted data is provided as input; b) *recurrent* networks that rely on well-established results from nonlinear dynamical systems theory, mainly related to gradient-type systems. The energy landscape, the number, positions, and nature of equilibrium states (or, more generally, stable manifolds) of such systems were especially investigated. Various approaches aim at designing the patterns to be stored as stable equilibria of the systems, while corrupted data should be provided as initial states. The theory guarantees that, under certain conditions, such systems may become globally stable, hence no other (complex) behaviour may occur, except from evolving from any initial state towards a particular stable equilibrium [7].

In order to be used in practical applications, such systems should ideally exhibit no other equilibria except for the (as many as possible) desired ones, their corresponding basins of attraction should be under control, and the addition/elimination of equilibrium states should be performed with minimal redesign of the system.

Pattern recognition applications are often required to exhibit robustness against geometric transformations such as translations, rotations, or scale changes (moreover, character recognition applications should also tackle line thickness and special deformations, while face recognition is very sensitive to illumination variability). Most of the existing solutions rely on various preprocessing algorithms to extract specific invariant features from the data, while special distance metrics may also include the effect of (typically, affine) transformations. Tangent distance (TD) [18] is a well-known example of the later approach, which showed significantly improved performances over the classical Euclidean distance in handwritten character recognition applications.

The present paper extends the results in [3] by merging TD with the construction of a special gradient-type dynamical system, whose associated Lyapunov function is generated using the Support Vector Data Description (SVDD) [19] algorithm. The same basic approach has also been used in a series of papers dealing with clustering and denoising [8, 10-15], but not related to invariant properties. As described next, the proposed solution offers advantages in terms of computational complexity, while still preserving the robustness against geometric transformations, and improving the recognition performances when compared to the Euclidean distance alternative. The paper is organized as follows: section 2 introduces the components of the proposed approach, namely the SVDD algorithm, the proposed dynamical system, and tangent distance. Experimental results for handwritten character recognition and face recognition are given in section 3, while conclusions and directions for future work are finally outlined.

2. ASSOCIATIVE MEMORY DESIGN

2.1. Support Vector Data Description

Support Vector Data Description (SVDD) has been originally introduced as an outlier/novelty detection procedure [19]. It follows the “learning with kernel” paradigm in order to efficiently and flexibly estimate the support of the data distribution. Basically, the method aims at obtaining a spherically shaped boundary around the “normal” data, whose volume should be minimized in order to leave outside potential outliers. More flexible boundaries may be obtained by using the well-known “kernel trick” from SVM theory, namely by mapping the original input data to a feature space after preprocessing with a proper nonlinear function, and then performing SVDD on the mapped data. More specifically, consider $\{\mathbf{x}_i, i = 1{:}\mathrm{N}\}$ a set of vectors whose distribution we seek to obtain. We search for a closed boundary around the given data as a sphere with center **a** and radius *R*. An optimization problem is formulated in order to minimize the radius while including all data within the sphere as [19]:

$$\min R^2, \text{ such that } \left\|\mathbf{x}_i - \mathbf{a}\right\| \le R^2, \ \forall i = 1 \ldots N . \tag{1}$$

Since outliers may be present in the training data, the optimization problem is modified such that the distance may become larger than R^2, but penalties should occur in these situations. The new formulation of the problem now includes the slack variables $\xi_i \ge 0$ and it becomes [19]:

$$\min R^2 + C\sum_i \xi_i , \text{ such that } \left\|\mathbf{x}_i - \mathbf{a}\right\| \le R^2 + \xi_i, \ \xi_i \ge 0, \ \forall i = 1 \ldots N , \tag{2}$$

where the scalar parameter C controls the trade-off between the volume of the sphere and the number of input vectors defined as outliers. The method of Lagrange multipliers is used to solve the problem, and the main results give the expressions of the minimal radius R and sphere center $\mathbf{a}$ as [19]:

$$\begin{aligned} &\mathbf{a} = \sum_i \alpha_i \mathbf{x}_i, \ \alpha_i > 0 \\ &R^2 = \langle \mathbf{x}_k, \mathbf{x}_k \rangle - 2\sum_i \alpha_i \langle \mathbf{x}_i, \mathbf{x}_k \rangle + \sum_{i,j} \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle, \end{aligned} \tag{3}$$

where vectors $\mathbf{x}_i$ for which $\alpha_i > 0$ are called *support vectors* (only those influence the center of the sphere **a**), and $\mathbf{x}_k$ are the subset of the support vectors for which $0 < \alpha_i < C$ (support vectors for which $\alpha_i = C$ are treated as outliers). Parameters α_i are obtained as a result of the optimization procedure.

Since the expression of R^2 only includes scalar products of the support vectors, we may use the classical kernelization approach to increase the flexibility of the boundary in order to accommodate more complex data distributions. More specifically, consider a nonlinear function Φ(.) that maps the original data into a (typically, higher dimensional) feature space. The SVDD approach now seeks a spherically shaped boundary in the new feature space, whose center and radius are easily determined if there exists a kernel function such that scalar products in the original space are conveniently expressed according to $K(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \right\rangle$. For a Gaussian-type kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{e}^{-q\|\mathbf{x}_i - \mathbf{x}_j\|^2}$ the radius is given by [19]:

$$R^2 = 1 - 2\sum_i \alpha_i \, \mathrm{e}^{-q\|\mathbf{x}_k - \mathbf{x}_i\|^2} + \sum_{i,j} \alpha_i \alpha_j \, \mathrm{e}^{-q\|\mathbf{x}_i - \mathbf{x}_j\|^2}, \ \forall \mathbf{x}_k \text{ such that } 0 < \alpha_k < C . \tag{4}$$

It is worth noting that for Gaussian-type kernels the SVDD method is similar (learns the same decision function) as the one-class Support Vector Machines (OC-SVM) approach [17]. More specifically, the minimal sphere enclosing the nonlinearly transformed data is equivalent to finding a maximal margin hyperplane separating the normal points and outlier data.

A number of other approaches for estimating data distribution exist, mostly rooted in statistical theory related to clustering or probability density estimation. Examples include Gaussian processes support functions [10] and Parzen windows, but those suffer from the “curse of dimensionality” and are sensitive to the data density distribution. Extensions of the basic approach have been also reported, aiming at reducing the computational complexity of the optimization process, using the data density information, or including additional regularization parameters related to the relevance of the individual data points [13,14].

2.2. Gradient-Type Associative Memory

Most of the existing associative memory solutions based on recurrent networks exhibit additional spurious equilibria apart from the desired ones, have (too) limited capacity, and offer no simple control over the extent of the basins of attraction around such equilibria. One notable exception is the idea described in [2,3] (originally introduced in [6] to solve a toy-problem classification task) that has been successfully used in a broad range of applications, including soft decision decoding of block codes, face recognition, and handwritten characters recognition. The basic approach relies on defining a gradient-type neural network whose associated Lyapunov function is constructed as a (weighted) sum of individual functions exhibiting good space localization properties. More specifically, we sum up a number of multidimensional Gaussian-type pulses centered on the patterns to be stored, such that those will act as isolated minima of the Lypaunov function (such minima are asymptotically stable states of the gradient-type system [2,3]):

$$\begin{aligned} &\frac{\mathrm{d}x_i}{\mathrm{d}t} = -\frac{\partial V(\mathbf{X})}{\partial x_i}, \quad i = 1 \ldots N \\ &V(\mathbf{X}) = \sum_{m=1}^{M} w_m g_m(\mathbf{X}), \quad g_m(\mathbf{X}) = 1 - \mathrm{e}^{-\frac{d_p{}^{P}(\mathbf{X},\mathbf{X}_m)}{2\sigma_m^2}}, \end{aligned} \tag{5}$$

where $V(\mathbf{X})$ is the Lyapunov function, N is the order of the system, M is the number of memories to be stored, and w_m are (possibly identical) scalar weights. Function $V(\mathbf{X})$ has the appearance of a RBF-type representation, and may be seen as a particular case of a Gaussian process support function estimation procedure [10]. Apart from the solid theoretical support, advantages of the proposed solution include the direct relation between the above equations and the memories to be stored, and an obvious interpretation of the effect of various parameters on the system dynamics. Nevertheless, a clear drawback is the computational complexity, which may become prohibitive if a great number of (high-dimensional) patterns are to be stored.

A superior alternative has been proposed in a series of papers by Lee et al. [8,12,15] that also make use of a gradient-type system, but the definition of the Lyapunov follows the input data distribution modeled by the SVDD algorithm, as in equation (4). Accordingly, the dynamics of the system is given by:

$$\begin{aligned} &\frac{\mathrm{d}x_i}{\mathrm{d}t} = -\frac{\partial V(\mathbf{X})}{\partial x_i}, \quad i = 1 \ldots N \\ &V(\mathbf{X}) = 1 - 2\sum_i \alpha_i \, \mathrm{e}^{-q\|\mathbf{x}-\mathbf{x}_i\|^2} + \sum_{i,j} \alpha_i \alpha_j \, \mathrm{e}^{-q\|\mathbf{x}_i-\mathbf{x}_j\|^2}. \end{aligned} \tag{6}$$

One key advantage of SVDD is that kernel values are evaluated only for the support vectors (which typically represent only a small fraction of the entire training set), while Gaussian processes support functions and Parzen windows require evaluation on the whole database. In order to illustrate the basic idea, a simple example is presented in Fig. 1 (this is the triangle dataset used in [13]). Bi-dimensional training data vectors belonging to 3 distinct classes are used to determine the corresponding support vectors. Contour plots and 3D views of the associated Lyapunov function are given, showing deep minima for each data cluster.

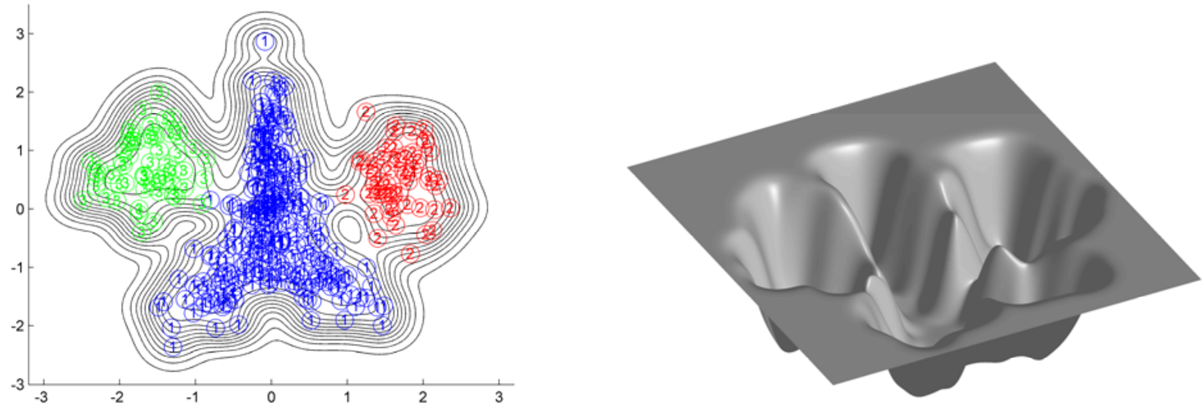

Fig. 1 – 2D training data from 3 distinct classes: contour plots of function $V(\mathbf{X})$ (left); 3D view of the $V(\mathbf{X})$ landscape (right).

Once given the actual values of the support vectors and a_i coefficients, we must first determine the positions of the corresponding minima of function $V(\mathbf{X})$, since those will act as the (stable) equilibrium points of the system. As such, we will use each training data point as an initial state of the system and let it settle towards the equilibrium point whose basin of attraction includes the given initial state. Experiments reported in [8] clearly indicate that the number of the stable equilibria is (sometimes, much) smaller than the dimensionality of the training dataset. Since we are aware of the category each training pattern belongs to, we will be able to label every equilibrium point accordingly. In the testing phase, we set again a test pattern as an initial state of the dynamical system, and assign it the label of the equilibrium point to which the given test point converges to.

2.3. Tangent distance

In order to enhance the pattern recognition performances and infer invariance to geometric transformations to the SVDD-based dynamic associative memory described by equations (6), we propose to replace the Euclidean distance with the TD distance [5,16,18]. Since patterns affected by such transformations define (hopefully, smooth) manifolds in space, the distance between two patterns can now be defined as the minimum distance between their respective manifolds, which should be invariant with respect to the given transformations, as exemplified in Fig. 2.

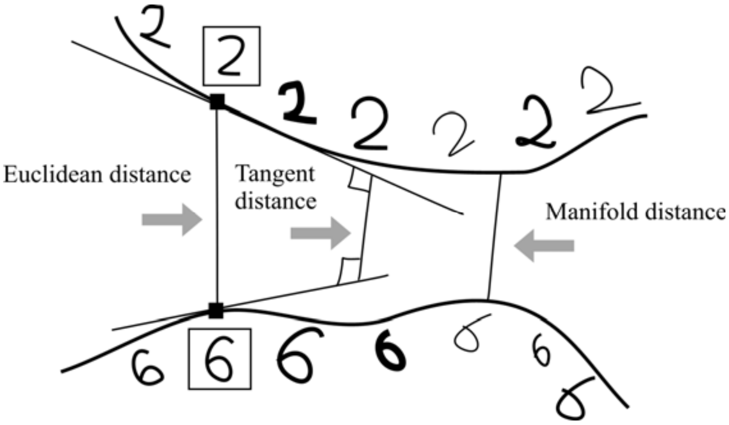

Fig. 2 – Definition of the manifold distance, tangent distance, and Euclidean distance between patterns.

More specifically, when an image is affected by a transformation $t(\mathbf{x},\boldsymbol{\beta})$ that depends on L parameters, the set of all transformed patterns is a manifold of at most dimension L in the pattern space. Due to computational complexity and lack of analytic expressions for the generated manifolds, we must use approximations of the manifold. A common solution is based on a linear combination of the vectors that span the tangent subspace, given by the partial derivatives of $t(\mathbf{x},\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$ [18]:

$$t(\mathbf{x},\boldsymbol{\beta}) \approx \mathbf{x} + \sum_{l=1}^{L} \beta_l \frac{\partial t(\mathbf{x},\boldsymbol{\beta})}{\partial \beta_l} = \mathbf{x} + \sum_{l=1}^{L} \beta_l \, \mathbf{x}_l = \mathbf{x} + \mathbf{T} \cdot \boldsymbol{\beta} . \tag{7}$$

The single and double-sided TD distances are respectively defined as:

$$TD_{SS}(\mathbf{x},\mathbf{y}) = \min_{\boldsymbol{\beta}_x \in R^L} \left\{ \left\| \mathbf{x} + \sum_{l=1}^{L} \beta_{xl} \, \mathbf{x}_l - \mathbf{y} \right\|^2 \right\}$$

$$TD_{DS}(\mathbf{x},\mathbf{y}) = \min_{\boldsymbol{\beta}_x ,\boldsymbol{\beta}_y \in R^L} \left\{ \left\| (\mathbf{x} + \sum_{l=1}^{L} \beta_{xl} \, \mathbf{x}_l) - (\mathbf{y} + \sum_{l=1}^{L} \beta_{yl} \, \mathbf{y}_l) \right\|^2 \right\} \tag{8}$$

While double-sided TD distance may yield marginal classification performance improvements, single–sided TD distance is preferred in practice since it involves a smaller computational cost and the tangent vectors corresponding to the training data can be pre-computed and stored, increasing the speed of the testing phase.

Due to the linear nature of the procedure, we may apply the TD computation not only on the original data, but also on a linearly processed (*e.g.*, compressed) version of it [11]. Denoting by $\mathbf{P} \in \Re^{D \times N}$ the projection matrix, we may write $\mathbf{P} \cdot (\mathbf{x} + \mathbf{T} \cdot \boldsymbol{\beta}) = \mathbf{P} \cdot \mathbf{x} + (\mathbf{P} \cdot \mathbf{T}) \cdot \boldsymbol{\beta} = \tilde{\mathbf{x}} + \tilde{\mathbf{T}} \cdot \boldsymbol{\beta}$, hence the tangent vectors grouped in matrix $\mathbf{T}$ are transformed using the same subspace projection procedure as the original images. The method improves over the solution presented in [3], where a TD-based Gaussian kernel was evaluated on all available training data points.

3. EXPERIMENTAL RESULTS

3.1. Handwritten character recognition

We have performed extensive experiments using the United States Postal Service (USPS) handwritten digits database [20]. It includes 7291 training images and 2007 test images. Each image consists of 16×16 pixels of grayscale values ranging from 0–255. The tangent vectors were computed using MATLAB, starting from a publicly available C implementation [9]. Examples of tangent vectors are presented in Fig. 3. Experiments used the single-sided TD distance and the following setups: a) original images + Euclidean/TD distances; b) PCA-compressed support vectors + Euclidean/TD distances; c) associative memory + PCA-compressed SVDD support vectors + Euclidean/TD distances. In the first scenario (original images), tests were performed using all 7291 training data points. To assess the dependence of the recognition performances on the number of training images, repeated experiments using 60/200/500 images for each digit were performed, while all 2007 available test images were used. The dimension of the projection subspace varied between 20 and 50, capturing more than 90% of the energy of the original images. Results indicated in Fig. 4 show that the TD setups yield superior performances over the Euclidean distance alternative. In Fig. 5 we show examples of patterns that are correctly classified by the TD based approach, while the Euclidean version fails to.

Classification performances are almost insensitive to the dimension of the projection subspace if it exceeds 40. They compare favorably with previously reported results, as indicated in Table 1, and improve when increasing the dimensionality of the training dataset.

3.2. Face recognition

In case of optical character recognition applications as originally introduced in [18], tangent vectors are computed by first smoothing the original images with a proper kernel, then using finite differences (the two steps may be combined by the Sobel operator). Unfortunately, this approach is not appropriate for other classes of images, since the input may not be smooth enough to reliably compute the local tangent vectors.

Table 1

Comparative analysis of classification error rates for USPS database (performances of existing solutions from [3])

Method	**Error rate (%)**
Human performance	2.5
Nearest neighbour classifier (L2)	5.6
Nearest neighbour classifier (TD)	3.54
Support Vector Machines	4
Invariant Support Vector Machines	3
Relevance Vector Machines	5.1
Neural network	4.2
Kernel densities	5.5
Kernel densities + TD	3.7
SVDD + PCA + associative memory + L2	4.98
SVDD + PCA + associative memory + TD	3.98

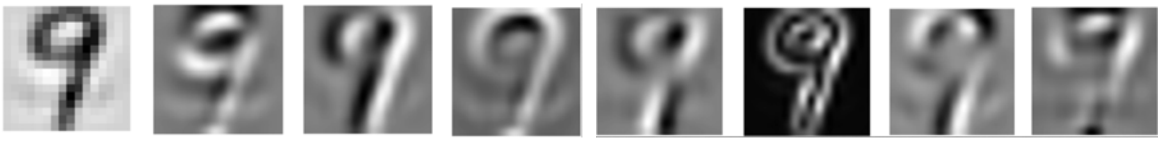

Fig. 3 – Examples of tangent vectors.

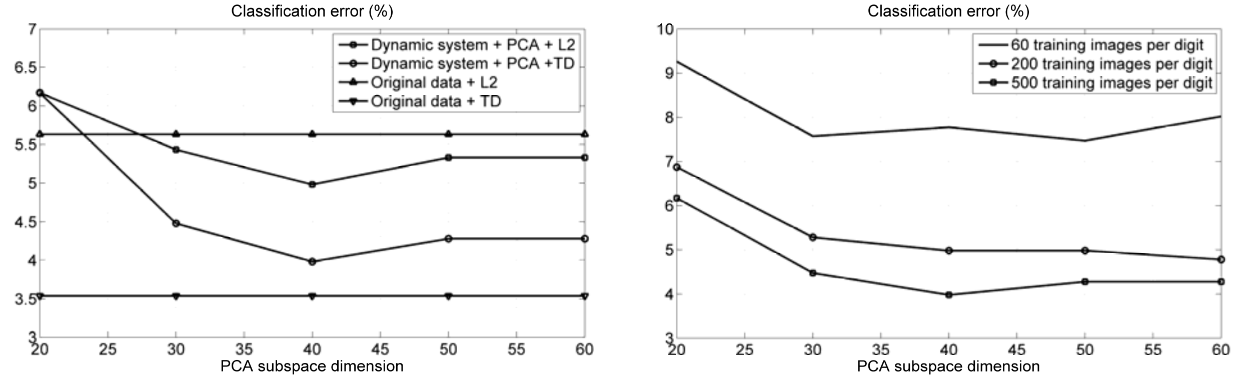

Fig. 4 – Classification error rates vs. the subspace dimension for USPS database (500 training images per digit) (left); classification error rates vs. the dimensionality of the training dataset for USPS database (right).

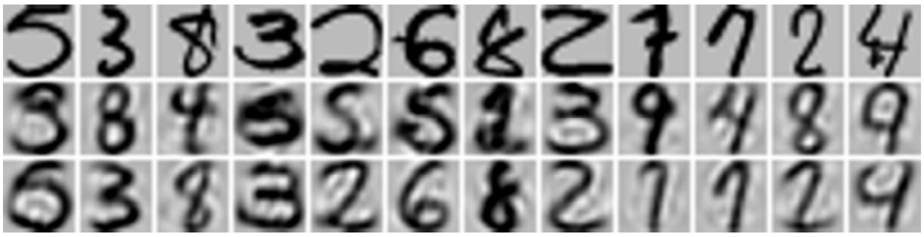

Fig. 5 – SVDD+TD associative memory may yield correct class label when the Euclidean alternative fails to: first row – original test patterns; second row – patterns recovered by the Euclidean version; third row – patterns recovered by the TD-based associative memory.

As a consequence, we may obtain completely unrealistic effects, as illustrated in Fig. 6 in case of face images. As a consequence, we followed the same approach as in [3, 16], and applied controlled affine transformations in order to generate virtual samples to be further used for computing finite differences to approximate the tangent vectors. As pointed out in [12], this method may properly approximate a broader range of geometric transformations than classical smoothing. Only 4 transformations were taken into account, namely left/right horizontal translation (with ± 4 pixels), up/down vertical translation (with ± 2 pixels), scale variation (± 5%), and in-plane rotation (with ± 10º, distinct tangent vectors were computed for rotation to the left and to the right, respectively).

The performances of the invariant associative memory have been tested on the Olivetti database. It comprises 10 distinct images of 40 persons, and includes variations in pose, light conditions, and expression. Each image has 112×92 pixels. We used a training set of 5 images per person, randomly selected from the available 10, and the rest for the testing phase. The original images were downscaled to yield 32×32 pixels by performing a multiresolution decomposition using the Discrete Wavelet Transform, with the additional benefit of providing robustness against face expression variation. Recognition performances (averaged over 20 distinct trials) are given in Fig. 7. The classification performances using TD are clearly better than the Euclidean alternative.

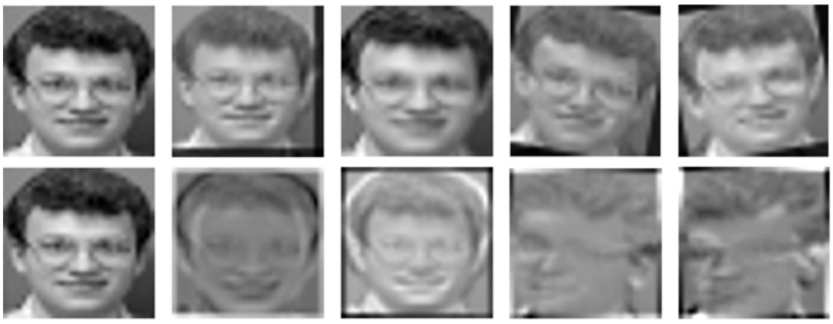

Fig. 6 – Tangent vectors computed by combining smoothing and differentiation in case of rotation and scale variation.

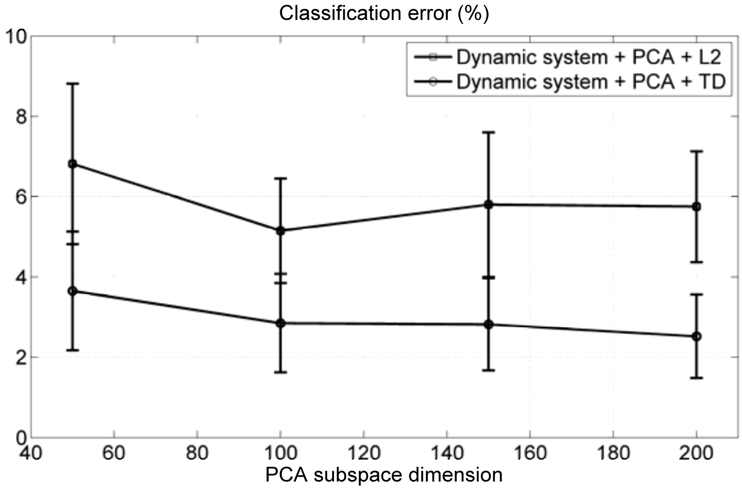

Fig. 7 – Classification error rates for the Olivetti database (%)
(mean values and standard deviations averaged over 20 distinct trials).

4. CONCLUSIONS

Combining tangent distance and SVDD/SVM offers a convenient means of dealing with geometric invariances present in most of pattern recognition applications, while avoiding the excessive cost of alternative solutions such as virtual support vector method, invariant hyperplanes, or kernel jittering [21].

The proposed approach offers an efficient alternative to the method proposed in [3], since the construction of the Lyapunov function may now require kernel evaluations only on (a limited number of) support vectors instead of the entire training dataset. The SVDD data distribution support estimation may be further improved by using recently introduced solutions taking into account local density information [14] or the relevance of the training data points [21]. In case of multi-class classification problems, we could also consider multi-sphere modeling of the underlying training data distribution.

As suggested in [3], the solution may be extended by considering a modular approach relying on multiresolution representations, or by successively visiting distinct equilibrium points in a predefined order.

Table 2

Comparative analysis of classification error rates for Olivetti database (performances of existing solutions from [3])

Method	Error rate (%)
Eigenfaces	10
Pseudo-2D HMM	5
Convolutional Neural Network	3.8
Linear SVM	3
Waveletface + L2	7.5
Discriminant Waveletface + L2	5.5
Discriminant Waveletface + NFL	5
SVDD + PCA + L2	6
SVDD + PCA + TD	4.7
SVDD + PCA + associative memory + L2	5.09
SVDD + PCA + associative memory + TD	2.77

REFERENCES

1. J.T. CHIEN, C.C. WU, Discriminant waveletfaces and nearest feature classifiers for face recognition, IEEE Trans PAMI, **24**, 12, pp. 1644–1648, 2002.
2. I.B. CIOCOIU, *Analog decoding using a gradient-type neural network*, IEEE Trans Neural Networks, **7**, *4*, pp. 1034–1038, 1996.
3. I.B. CIOCOIU, Invariant pattern recognition using analog recurrent associative memories, Neurocomputing, **73**, pp. 119–126, 2009.
4. D. DeCOSTE, B. SCHÖLKOPF, *Training invariant support vector machines*, Machine Learning, **46**, pp. 161–190, 2002.
5. B. HAASDONK, D. KEYSERS, *Tangent distance kernels for support vector machines*, Proc. ICPR, pp. 864–868, 2002.
6. J.Y. HAN, M.R. SAYEH, J. ZHANG, Convergence and limit points of neural networks and its application to pattern recognition, IEEE Trans Syst Man Cyber, **19**, 5, pp. 1217–1222, 1989.
7. M.W. HIRSCH, S. SMALE, Differential equations, dynamical systems, and linear algebra, Academic, 1974.
8. K.H. JUNG, N. KIM, J. LEE, *Dynamic pattern denoising method using multi-basin system with kernels*, Pattern Recognition, **44**, *8*, pp. 1698–170, 2011.
9. D. KEYSERS, *Tangent distance implementation*, available at: http://www-i6.informatik.rwth-aachen.de/~keysers/td/
10. H.C. KIM, J. LEE, *Clustering based on gaussian processes,* Neural Computation, **19**, *11*, pp. 3088–3107, 2007.
11. T. KÖLSCH, D. KEYSERS, H. NEY, R. PAREDES, *Enhancements for local feature based image classification*, Proc. ICPR, pp. 248–251, 2004.
12. J. LEE, D. LEE, An improved cluster labeling method for support vector clustering, IEEE Trans PAMI, **27**, 3, pp. 461–464, 2005.
13. LEE D, LEE J, Domain described support vector classifier for multi-class classification problems, Pattern Recognition, **40**, pp. 41–51, 2007.
14. LEE K.Y, KIM D.W, LEE D, LEE K.H, *Improving support vector data description using local density degree,* Pattern Recognition, **38**, pp. 1768–1771, 2005.
15. D. LEE, J. LEE, Equilibrium-based support vector machine for semi-supervised classification, IEEE Trans Neural Networks, **18**, 2, pp. 578–583, 2007.
16. A. POZDNOUKHOV, S. BENGIO, Tangent vector kernels for invariant image classification with SVMs, Tech Report IDIAP-RR 03-75, 2003.
17. B. SCHÖLKOPF, A.J. SMOLA, *Learning with kernels*, MIT Press, 2002.
18. P.Y. SIMARD, Y.A. LE CUN, J.S. DENKER, B. VICTORRI, *Transformation invariance in pattern recognition – tangent distance and tangent propagation,* Int J Imaging System and Technology, **11**, *3*, pp. 181–194, 2001.
19. D.M.J. TAX, R.P.W. DUIN, *Support vector domain description,* Pattern Recognition Lett, **20**, pp. 1191–1199, 1999.
20. USPS database, ftp://ftp.kyb.tuebingen.mpg.de/pub/bs/dat
21. Z. WANG, D. GAO, Z. PAN, An effective support vector data description with relevant metric learning, Proc. ISNN, LNCS 6064, pp. 42–51, 2010.

Received September 21, 2016