Abstract—Tangent vectors are one of the best tools for learning variability in handwritten digits. Many research works indicate that tangent vectors provide a significant improvement of accuracy especially when used with SVM classifiers. However, since they are based on the use of affine transformations they substantially extend the runtime. In addition, the user should adequately select the transformations in order to highlight the variability of data. The present work aims to exploit accuracy improvement of tangent vectors while reducing the runtime. Therefore, we investigate the use of tangent vectors that are a priori extracted from training data. The idea is to substitute each pattern by its Tangent Vector Mahalanobis (TVM) distances with respect to all classes. Then, a SVM is trained over TVM values, which contain a priori knowledge and have a smaller size than digit images. Experiments performed on USPS database showed that the proposed approach improves recognition accuracy and allows a huge reduction in the runtime.

Keywords—handwritten digit recognition, SVM, tangent vectors.

I. INTRODUCTION

In handwritten digit recognition, classical descriptors such as statistical moments and Fourier transform cannot deal with variations of writing style and tools [6]. More recently, tangent vectors were specifically used for learning variability in order to construct classifiers with invariant decisions. Invariance is established with respect to some affine transformations such as rotations and translations where tangent vectors of each pattern correspond to the derivative of these transformations. Various handwriting recognition research works highlighted the contribution of these descriptors for improving the accuracy of SVM classifiers. Recall that, invariance learning was introduced to SVMs (which are the most commonly used classifiers for handwriting recognition) with the VSV (Virtual Support Vectors) method proposed by DeCoste and Schölkopf [2]. The VSV generates virtual samples from support vectors in order to enforce invariance around the decision boundary. Thenafber, Haasdonck and Keysers [3] employed tangent vectors through the tangent distance with SVM kernels which are based on distance calculation. Thereby, various forms of the tangent distance such as two-sided and one-sided were efficiently used with RBF and Negative distance kernels. Furthermore, Pozdnoukhov and Bengio employed a tangent vector proximity measure with RBF kernel [8]. In all previous works, tangent vectors were computed by performing a set of affine transformations the reason for which the computation cost is very important. Simard et al [9], reported that the tangent distance is about 12 times slower than the conventional Euclidian distance. On the other hand, tangent distance kernels can be 30 times slower than conventional kernels. More recently, Keysers et al [5], proposed a fast approach, which generates tangent vectors from the training data. Tangent vectors incorporate invariance knowledge into Mahalanobis distance throughout class covariance matrices. This improved the performance of Bayesian classifiers with 2% in overall accuracy.

The present work investigates the applicability of such tangent vectors with SVM classifiers. The idea consists of substituting each pattern by a feature vector that is composed of its Tangent Vector Mahalanobis (TVM) distances with respect to all classes. Then, a SVM classifier is trained over these features which contain all invariance knowledge. The main advantage of this approach is the needless of computing tangent vectors for each pattern, which is a time consuming task. Also, the size of data is significantly reduced. Specifically, for handwritten digit recognition which is the task at hand, the size decreases from the digit image to the number of classes.

The rest of this paper is arranged as follows. In section 2 we briefly present SVM classifiers. Section 3 summarizes the use of tangent vectors with SVMs and shows how the TVM is used to train SVMs. Experimental results obtained for USPS database are reported in section 4 while the main conclusions are given in the last section.

II. SUPPORT VECTOR MACHINES

Support Vector Machines (SVMs) are formulated to construct binary classifiers. From a set of labelled training
patterns, defined by: \((x_n, y_n) \in \mathbb{R}^M \times \{ \pm 1 \}\), \((M: \text{data dimension})\) and \(\{n = 1, \ldots, N_c\}\) where \(N_c\) is the number of samples per a class \(c\). For a set of functions \(f: \mathbb{R}^M \rightarrow \{ \pm 1 \}\), SVMs seek the function \(f\) that allows the minimal generalization error \([1]\) \([10]\). The selection of the appropriate \(f\) is achieved by minimizing an upper bound on the generalization error while maximizing the margin between the two classes. Therefore, data are classified according to:

\[
f(x) = \text{sign} \left( \sum_{i=1}^{SV} v_i \alpha_i k(x_i, x) + b \right)
\]  

(1)

Where \(b\) is a bias. The optimal hyperplane corresponds to \(f(x) = 0\). \(SV\) is the number of support vectors which are training data whose Lagrange multipliers \(\alpha_i\) are different to zero. The kernel \(k(\cdot, \cdot)\) is any mathematical function, which respects Mercer’s conditions \([2]\). For pattern recognition, the Radial Basis Function (RBF) kernel provides commonly the best performances.

\[
k(x_1, x_2) = \exp \left( -\frac{1}{2\sigma^2} d(x_1, x_2) \right)
\]  

(2)

\[
d(x_1, x_2) = \|x_1 - x_2\|^2
\]  

(3)

\(\sigma\) is user-defined.

Extension of SVMs for multiclass problems can be done through various approaches \([4]\). For a \(C\)-class problem the One-Against-All (OAA), which is the earliest multi-class implementation, performs \(C\) binary SVMs in order to separate iteratively each class from all the others. Since OAA requires a large training time, the One-Against-One approach with the DDAG decision function is commonly used \([9]\). The DDAG performs \(C(C-1)/2\) SVMs each of which separates two classes. Presently, the DDAG is employed to perform multiclass SVMs.

III. TANGENT VECTORS WITH SVMS

The use of a priori knowledge to allow the invariance was introduced to SVMs classifiers through the VSV approach which generates virtual samples from support vectors to enforce invariance around the decision boundary \([2]\). Indeed, invariance learning is based on the idea of learning small or local transformations, which leave the class of data unchangeable. To define these transformations, let \(\vec{x}(\beta)\) denotes a transformation of a pattern \(x\) that depends on a set of parameters \(\beta = [\beta_1, \cdots, \beta_L] \in \mathbb{R}^L\). The linear approximation of \(\vec{x}(\beta)\) using a Taylor expansion around \(\beta = 0\) can be written as \([5]\):

\[
\vec{x}(\beta) = x + \sum_{l=1}^{L} \beta_l \cdot v_l + \sum_{l=1}^{L} \phi(\beta_l^2)
\]  

(4)

\(v_l\) are tangent vectors corresponding to partial derivatives of the transformation \(\vec{x}\) with respect to \(\beta_l = [l = 1, \cdots, L]\) so that:

\[
\frac{\partial \vec{x}(\beta)}{\partial \beta_l} \bigg|_{\beta_l = 0} = 0
\]  

(5)

Since terms of second order and higher in (4) are neglected, \(\vec{x}(\beta) = x\) for \(\beta = 0\) while for small values the transformation does not change the class membership of \(x\). Hence, the linear approximation represents the transformations such as translation, rotation and axis deformations by one prototype and its corresponding tangent vector. Recall that tangent vectors were initially introduced through the tangent distance that can be expressed as:

\[
TD(x_1, x_2) = \min_{\beta, \beta'} \left( x_1 + \sum_{l=1}^{L} \beta_l \cdot v_l - (x_2 + \sum_{l=1}^{L} \beta'_l \cdot v'_{l}) \right)
\]  

(6)

Furthermore, several forms of tangent distance were used with distance-based kernels. Since the calculation of tangent vectors requires prior knowledge about the transformations, Keysers et al., have shown that tangent vectors can be estimated from the training data \([5]\). For a set of classes \(\{c = 1, \ldots, C\}\) with training data \(x_{nc} \{n = 1, \ldots, N_c\}\), tangent vectors maximizing the dissimilarity between classes are chosen such that \(\{\Sigma^{-1/2} \cdot v_{cl}\}\) are the eigenvectors with largest eigenvalues of the matrix:

\[
\Sigma^{-1/2} S_c \left( \Sigma^{-1/2} \right)^T
\]  

(7)

\(\Sigma: \) Covariance matrix computed from the entire training set.

\(S_c: \) Class dependent scatter matrix given by:

\[
S_c = \sum_{n=1}^{N_c} (x_{nc} - \mu_c) (x_{nc} - \mu_c)^T
\]  

(8)

\(\mu_c: \) mean of the class \(c\).

Each tangent vector refers to a transformation or specific variability knowledge. Besides, the number of tangent vectors should be a priori chosen by the user.
In the present work, we propose a framework to employ such tangent vectors with SVM classifiers. The ultimate objective is to avoid at once the need of a priori knowledge about transformations and the calculation of tangent vectors for each pattern. Notice that in front of conventional approaches such as VSV and tangent distance in which tangent vectors are pattern dependent (i.e. each pattern has its own tangent vectors), in this approach they are class dependant. So, once tangent vectors are determined, they are incorporated into the covariance matrix of their class which is written as:

\[
\tilde{\Sigma}_c^{-1} = \Sigma^{-1} - \frac{1}{1 + \gamma^2} \sum_{l=1}^L v_{cl} \cdot v_{cl}^T \Sigma^{-1} 
\]

\(\gamma\): user defined parameter.
\(\tilde{\Sigma}_c\): tangent vector-based covariance matrix for the class \(c\).

The Tangent Vector Mahalanobis (TVM) distance of a pattern \(x\) with respect to a class \(c\) becomes as:

\[
TVM(x|c) = (x - \mu_c)^T \tilde{\Sigma}_c^{-1} (x - \mu_c) \tag{10}
\]

Hence, for each pattern, we calculate a feature vector \(P(x) = \{TV(x|1), \cdots, TV(x|C)\}\) that is constituted by its TVM with respect to all classes. Derived feature vectors are used to train the SVM where the distance \(d(x_1, x_2)\) of the RBF kernel becomes as:

\[
d(x_1, x_2) = \|P(x_1) - P(x_2)\|^2 \tag{11}
\]

Notice that benefits of using feature vectors are not only the implicit incorporation of invariance knowledge into SVM but also the reduction of the data size. Specifically, for handwritten digit recognition the data size decreases from the digit image to a vector of 10 components (which is the number of classes). This makes the kernel evaluation as well as the training of stage faster.

IV. EXPERIMENTAL RESULTS

The validity of the proposed approach is investigated using the well-known USPS database. This database contains gray level digit images of numeral classes segmented to a size of 16×16 pixels and partitioned into 7291 training samples and 2007 test samples. Many test samples are corrupted so that even human cannot classify correctly; some of them are depicted in Figure 1. The optimization algorithm adopted for training SVMs is the Sequential Minimal Optimization (SMO) which provides two main practical advantages [7]. On one hand it tolerates some violations of mercer’s conditions and on the other hand, it is computationally faster than other optimization algorithms such as chunking and decomposition.

All experiments were performed on a 798 MHz Intel Core2 CPU. In a first time, several tests were conducted to find the best choices for SVM parameters which were fixed at \(\sigma^2 = 5\) and 10 for the regularization parameter.

Performance evaluation of the proposed schemes is carried out comparatively to the results obtained for original data (i.e. digit images). Since the number of tangent vectors that are used in TVM estimation is user defined, we evaluated the behavior of SVM according to the number of tangent vectors. The results are depicted in figure 2 which plots error rates of SVM achieved from 1 up to 100 tangent vectors. First, we remark that the error rate decreases proportionally to the number of tangent vectors (\(v_l\)), when this latter varies from 1 to 30. However, above 40 \(v_l\), the error rate becomes increasing which reflects a compromise between the error rate and the number of tangent vectors. In fact, the best result is obtained for 30 tangent vectors. Table 1 summarizes the corresponding results in terms of error rate, training time and recognition speed (RS). The recognition speed is expressed by the number of recognized characters per second. For comparison, the results obtained for original data using the same SVM architecture are given. From table 1, it is easy to see that the TVM improves the SVM performance by more than 1% in error rate. Moreover, the inspection of training times shows that it accelerates the training of SVM to 133 times. This huge acceleration is due to the data size which goes from 256 (16×16) with digit images to 10 with the TVM measures. In addition, the training of TVM features seems simpler because the SMO does not need a large number of optimization passes. This leads to a smaller number of support vectors (#SV) where the TVM-SVM requires 1/3 SV compared to the SVM trained over original data. Thereby, since the decision function is directly related to the number of support vectors, the TVM increases the recognition speed from 11 characters per second to 143 characters per second.

---

[Image 1. Testing samples with their alleged classes from USPS database]
TABLE I. PERFORMANCE EVALUATION FOR TVM-SVM

<table>
<thead>
<tr>
<th>Method</th>
<th>SVM</th>
<th>TVM-SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error rate (%)</td>
<td>5.24</td>
<td>4.14</td>
</tr>
<tr>
<td>Training time (S)</td>
<td>121720</td>
<td>910</td>
</tr>
<tr>
<td>RS</td>
<td>11</td>
<td>143</td>
</tr>
<tr>
<td>#SV</td>
<td>118</td>
<td>35</td>
</tr>
</tbody>
</table>

V. CONCLUSION

This paper investigated the use of class-dependant tangent vectors with SVM classifiers for handwritten digit recognition. The principle idea consists of substituting digit images by their dissimilarity measures with respect to all classes. Dissimilarities are based on Mahalanobis distance in which the covariance matrix is adapted by the class tangent vectors extracted from training data. This approach avoids the need of prior knowledge about transformations as well as the calculation of tangent vectors for each pattern. In addition, it significantly reduces the size of data. Experiments conducted on USPS database indicated that the SVM trained over TVM can improve the recognition accuracy to 1%. Moreover, in front of conventional tangent vector approaches (such as tangent distance) which extend the training time of SVMs, the TVM accelerates both training and recognition stages of SVMs. As a future work, tests on other SVM kernels and other databases are important to confirm again the validity of the TVM-SVM.

REFERENCES