Probabilistic Tangent Subspace: A Unified View

Jianguo Lee LIJG01@MAILS.TSINGHUA.EDU.CN
Jingdong Wang WANGJD01@MAILS.TSINGHUA.EDU.CN
Changshui Zhang ZCS@MAIL.TSINGHUA.EDU.CN
Zhaoqi Bian BZQ-DAU@MAIL.TSINGHUA.EDU.CN
Department of Automation, Tsinghua University, Beijing, 100084, P. R. China

Abstract

Tangent Distance (TD) is one classical method for invariant pattern classification. However, conventional TD need pre-obtain tangent vectors, which is difficult except for image objects. This paper extends TD to more general pattern classification tasks. The basic assumption is that tangent vectors can be approximately represented by the pattern variations. We propose three probabilistic subspace models to encode the variations: the linear subspace, nonlinear subspace, and manifold subspace models. These three models are addressed in a unified view, namely Probabilistic Tangent Subspace (PTS). Experiments show that PTS can achieve promising classification performance in non-image data sets.

1. Introduction

In pattern classification, one of the critical factors affecting performance is the pattern variation, which makes it important to consider the inherent variability of patterns. It is desirable to develop invariant classification algorithms that can tolerate small variations of input patterns while keeping the class label unchanged. Recently, designing invariant classification algorithms has become a topic of increasing interest.

One classical invariant algorithm is Tangent Distance (TD) (Simard et al., 1993; Simard et al., 2001). TD assumes there is a sample manifold generated by certain transformations, which can be approximated by the first-order Taylor expansion. The transformation, which TD considers as prior knowledge, is usually a small affine transformation such as scaling, rotation and translation in image objects. TD performs well in Optical Character Recognition (OCR) tasks, and has also been incorporated into many other methods for invariant pattern classification, such as Gaussian Mixture Models (Keysers et al., 2001).

Another typical invariant algorithm is Virtual-Support Vector Machines (V-SVM) (Schölkopf et al., 1996). The key idea of V-SVM is that the virtual examples or virtual data, which are generated by adding known small transformations to support vectors, are added to the training set to retrain SVM. Since these virtual data are near the margin, the retraining procedure will improve the decision boundary of SVM classifier. In practice, V-SVM has been proven to achieve better performance than standard SVM in OCR tasks.

There has also been some other research incorporating invariant prior knowledge into kernel methods (Burges, 1999). Various types of invariant kernels have been proposed in this way, for instance, jittering kernels (DeCoste & Schölkopf, 2002) and TD kernels (Haasdonk & Keysers, 2002).

Although TD and the invariant SVM algorithms have been successfully used in some applications, they have the common limitation, i.e., the invariances must be known as prior knowledge, such as some kinds of small affine transformations. It is not hard to predefine this kind of prior knowledge in image objects by tangent vectors (Simard et al., 2001), but besides image data sets, how could we provide the prior knowledge in more general data sets?

The goal of this paper is to propose a novel and practical way for TD-based invariant classification. The basic assumption is: tangent vectors can be approximately represented by the pattern variations, which are essentially the differences between prototypes and their neighbor samples that are from the same class. This study proposes three subspace models, which

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.

involve the linear subspace, nonlinear subspace, and manifold subspace models, to encode the variations. Subspace models, combined with the corresponding complementary subspaces, are embedded in probabilistic density functions to obtain a good representation of the full tangent space. Therefore, these three subspace models can be addressed in a unified view, namely Probabilistic Tangent Subspace (PTS). In the framework of PTS, the variations have been encoded to represent the tangent space, thus the distance measure derived from PTS becomes an invariant distance measure for pattern classification.

The rest of this paper is organized as follows. Section 2 briefly introduces Tangent Distance and its extension. Section 3 presents three formulations of PTS. Section 4 discusses some issues of PTS and related work. Section 5 presents experimental results on synthetic and real data sets. Finally, conclusions are drawn.

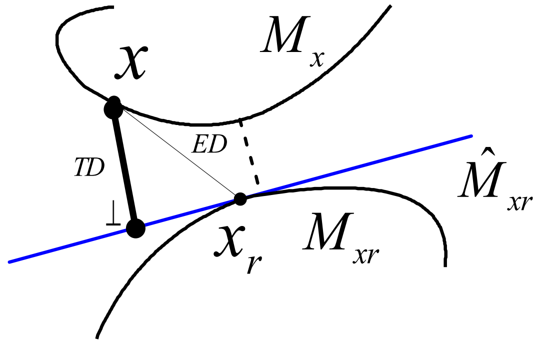

Figure 1. An illustration of TD. The Euclidean Distance (ED) between a test x and a prototype x_r is shown in a thin black line in comparison with the distance (dashed line) between the corresponding manifolds M_x and M_{xr}. The tangent approximation of the manifold at x_r is depicted by a plain line ($\hat{M}_{xr}$), and the corresponding (one-sided) TD is depicted by a thick line.

2. Overview of Tangent Distance

Tangent Distance (TD) is an invariant distance measure. Let $x \in \mathcal{R}^N$ be a pattern and $t(x, \alpha)$ denote the pattern obtained by a small transformation of x depending on a parameter α. TD assumes that $t(x, \alpha)$ has the same class membership as x. The set of transformed patterns, $M_x = \{t(x, \alpha)\} \subset \mathcal{R}^N$, now forms a manifold in pattern space. The transformed pattern $t(x, \alpha)$ can be approximated by a Taylor expansion, i.e.

$$t(x, \alpha) = x + \alpha T + o(\alpha^2),$$

where T is a tangent vector ($T = \partial t(x, \alpha)/\partial\alpha$). TD approximates the manifold M_x by the first-order Taylor expansion: $\hat{M}_x = \{x + \alpha T\} \subset \mathcal{R}^N$. And the one-sided TD is defined as:

$$TD(x, x_r) = \min_{\alpha} \|x_r + \alpha T - x\|, \tag{1}$$

where x is the test pattern, and x_r is the prototype pattern. An illustration of TD is shown in Figure 2.

For small transformations within one category, TD between the test and the prototype is therefore zero with the introduction of tangent vector T. From the perspective of generative models, small transformations of a prototype actually generate samples in its neighborhood. In other words, we could view the differences between prototypes and their neighbor samples that are from the same category as approximation of tangent vectors, which form a set:

$$S = \{z | z = x - x_r, \text{if } c(x) = c(x_r) \text{ and } x \in \mathcal{N}(x_r)\}, \tag{2}$$

where $c(x)$ denotes the class label of sample x, $\mathcal{N}(x_r)$ indicates the neighbor set of prototype x_r, and the size of S is denoted by M ($M = |S|$). This analysis supports our basic assumption that tangent vectors can be approximately represented by the pattern variations.

Intuitively, the set S can be characterized by subspace models. Actually, a similar idea was adopted to propose the Tangent Subspace (TS) algorithm (Hastie et al., 1995). TS performed Singular Vector Decomposition (SVD) on tangent vector set to reduce dimensionality, and thereafter selected prototypes from the training set. However, TS was only employed in OCR tasks, as its meaning was far from well understood.

Since no operable representation of tangent vectors has been given for general data sets, TD's application is seriously restricted. Starting from the basic assumption, as an operable representation of tangent vectors is presented here, TD can be extended to general data classification tasks.

In subspace methods, experience shows that it is not advisable to represent the full tangent space only by principal subspace. A complete representation, which is usually a probabilistic density function, is needed in this work. The strategy by Moghaddam (Moghaddam & Pentland, 1997) was adopted, which divided the full tangent space into two subspaces — principal subspace and its orthogonal complement subspace. Then two marginal Gaussian densities were estimated in these two subspaces, and a complete density function was formed by the product of these two independent marginal densities. Afterwards, an invariant distance measure could be derived from the complete density function.

3. Probabilistic Tangent Subspace

In this section, three formulations are presented under the unified view of PTS, which characterize the tangent vector set as linear subspace, nonlinear subspace, and manifold subspace, respectively.

3.1. Linear Subspace Case: PTS-I

Given training data set $X = \{x_i\}_{i=1}^m \subset \mathcal{R}^N$, PTS-I first forms the tangent vector set S according to (2), then characterizes S by a linear subspace, and finally estimates one Gaussian density function in this space:

$$G(z|S) = \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} \exp\Big\{-\frac{1}{2}(z-\mu)^T\Sigma^{-1}(z-\mu)\Big\}, \tag{3}$$

where μ is the mean vector, and Σ is the covariance matrix.

The exponent term in (3) is a Mahalanobis distance

$$d(z) = (z-\mu)^T\Sigma^{-1}(z-\mu) = \tilde{z}^T\Sigma^{-1}\tilde{z},$$

where $\tilde{z} = z - \mu$. Using the eigenvector matrix U and the eigenvalue matrix Λ of Σ, the Mahalanobis distance can be rewritten in a diagonal form:

$$d(z) = \tilde{z}^T U\Lambda^{-1}U^T\tilde{z} = y^T\Lambda^{-1}y,$$

where $y = U^T\tilde{z}$ is the projection of z into the transformed space. This can also be expressed in terms of a sum:

$$d(z) = \sum_{i=1}^{N}\frac{y_i^2}{\lambda_i}.$$

The full tangent space is divided into principal subspace F and its orthogonal complement subspace $\bar{F}$, where F is spanned by the first p principal components of Principal Component Analysis (PCA) on S. Therefore, the summation can be divided into two independent parts corresponding to F and $\bar{F}$ respectively:

$$d(z) = \sum_{i=1}^{p}\frac{y_i^2}{\lambda_i} + \sum_{i=p+1}^{N}\frac{y_i^2}{\lambda_i}.$$

Note that the first summation can be computed by projecting z into the first p-dimensional principal subspace. The remaining summation can be handled by the same strategy of Moghaddam & Pentland (1997), which approximates $d(z)$ by:

$$\hat{d}(z) = \sum_{i=1}^{p}\frac{y_i^2}{\lambda_i} + \frac{1}{\rho}\sum_{i=p+1}^{N} y_i^2, \tag{4}$$

where ρ is a weight coefficient. And the second summation can also be considered as reconstruction error

$$\varepsilon^2(z) = \sum_{i=p+1}^{M} y_i^2 = \|\tilde{z}\|^2 - \sum_{i=1}^{p} y_i^2, \tag{5}$$

so that it can be computed from principal subspace.

Therefore, we can write out the estimated complete density function based on $\hat{d}(z)$ as the product of two marginal and independent Gaussian densities:

$$\hat{G}(z|S) = \left[\frac{\exp\Big\{-\frac{1}{2}\sum\limits_{i=1}^{p}\frac{y_i^2}{\lambda_i}\Big\}}{(2\pi)^{p/2}\prod\limits_{i=1}^{p}\lambda_i^{1/2}}\right]\cdot\left[\frac{\exp\Big\{-\frac{\varepsilon^2(z)}{2\rho}\Big\}}{(2\pi\rho)^{(N-p)/2}}\right]. \tag{6}$$

The value of ρ can be derived by minimizing the Kullback-Leibler (KL) divergence between the true density $G(z|S)$ and the approximation $\hat{G}(z|S)$:

$$J(\rho) = \int G(z|S)\log\frac{G(z|S)}{\hat{G}(z|S)}\mathrm{d}z, \tag{7}$$

which yields

$$\rho* = \frac{1}{N-p}\sum_{i=p+1}^{N}\lambda_i. \tag{8}$$

Consequently, the Gaussian density $\hat{G}(z|S)$ can be computed only from the principal tangent subspace, while information from the full tangent space could be utilized.

As tangent subspace is assumed to be a uniform subspace, which does not vary with the class label, all the other parameters in $\hat{G}(z|S)$ except $\hat{d}(z)$ are constant after the training phase.

In the testing phase, when a test sample x_t is queried, we project the linear variation $z_r = x_t - x_r,\ x_r \in X$ into the principal tangent subspace, and evaluate the PTS-I distance measure:

$$\hat{d}(x_t, x_r) = \hat{d}(z_r) = \sum_{i=1}^{p}\frac{y_i^2}{\lambda_i} + \frac{\varepsilon^2(z_r)}{\rho}. \tag{9}$$

The prototype x_r, which has the shortest distance to x_t, corresponds to the classification result of x_t.

3.2. Nonlinear Subspace Case: PTS-II

It is possible that tangent space can not be well represented by a linear subspace model. Hence, we consider characterizing the tangent vector set by a nonlinear subspace model with the introduction of kernel trick (Schölkopf & Smola, 2002). The idea is that a kernel is defined over the tangent vector set S: $k(z,z') = \langle\phi(z),\phi(z')\rangle$, and an $M \times M$ Gram matrix K is obtained with the element $K_{ij} = k(z_i, z_j)$. Thus Kernel PCA in this nonlinear case will replace PCA in the linear case. The following work, which is to estimate a Gaussian density in kernel space, is similar to the linear case.

The Gaussian density function in kernel space can be written as

$$G\big(\phi(z)|S\big) = \frac{1}{(2\pi)^{M/2}\,|\Sigma_\phi|^{1/2}} \exp\big\{-\frac{1}{2}(\tilde{\phi}(z))^T \Sigma_\phi^{-1} \tilde{\phi}(z)\big\}, \tag{10}$$

where M denotes the dimension of feature space[1], $\tilde{\phi}(z) = \phi(z) - \mu_\phi$, μ_ϕ is the mean vector, and Σ_ϕ is the covariance matrix in kernel space.

The covariance matrix could undergo a decomposition in the high dimensional space: $\Sigma_\phi = V\Lambda V^T$, where Λ denotes the diagonal matrix of nonzero eigenvalues $\{\lambda_i\}$, and V is the matrix of the corresponding eigenvectors $\{v_i\}$. Thus the so-called *Kernel PCA map* (Schölkopf et al., 1998b) is adopted to simplify the kernel Mahalanobis distance:

$$D(z) = \tilde{\phi}(z)^T \Sigma_\phi^{-1} \tilde{\phi}(z) = \sum_{i=1}^{M} \frac{(v_i \cdot \tilde{\phi}(z))^2}{\lambda_i}. \tag{11}$$

Let $y_i = \langle v_i, \tilde{\phi}(z)\rangle$, where $\{y_i\}_{i=1}^M$ are kernel principal components of z. We obtain

$$D(z) = \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i}. \tag{12}$$

Substituting (11) and (12) into (10), we have

$$G\big(\phi(z)|S\big) = \frac{1}{(2\pi)^{M/2} \prod_{i=1}^{M} \lambda_i^{1/2}} \exp\Big\{-\frac{1}{2}\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i}\Big\}. \tag{13}$$

$D(z)$ is divided into two parts the same as in the linear case, and is approximated by

$$\hat{D}(z) = \sum_{i=1}^{p} \frac{y_i^2}{\lambda_i} + \frac{1}{\rho}\sum_{i=p+1}^{M} y_i^2. \tag{14}$$

The second summation can also be considered as a reconstruction error such that

$$\varepsilon^2(z) = \sum_{i=p+1}^{M} y_i^2 = \big\|\tilde{\phi}(z)\big\|^2 - \sum_{i=1}^{p} y_i^2, \tag{15}$$

and $\|\tilde{\phi}(z)\|^2$ can be computed by

$$\begin{aligned}\big\|\tilde{\phi}(z)\big\|^2 &= \big\|\phi(z) - \mu_\phi\big\|^2 = \big\|\phi(z) - \frac{1}{M}\sum_{i=1}^{M}\phi(z_i)\big\|^2 \\ &= \big|k(z,z) + \frac{1}{M^2}\sum_{i,j=1}^{M} k(z_i,z_j) - \frac{2}{M}\sum_{i=1}^{M} k(z,z_i)\big|.\end{aligned}$$

[1]Although the kernel space has very high dimensionality, the dimension of feature space is bounded by the size of the tangent vector set S.

With (14) and (15), the Gaussian density in kernel space can be approximated by

$$\hat{G}\big(\phi(z)|S\big) = \left[\frac{\exp\big\{-\frac{1}{2}\sum\limits_{i=1}^{p}\frac{y_i^2}{\lambda_i}\big\}}{(2\pi)^{p/2}\prod\limits_{i=1}^{p}\lambda_i^{1/2}}\right]\left[\frac{\exp\big\{-\frac{\varepsilon^2(z)}{2\rho}\big\}}{(2\pi\rho)^{(M-p)/2}}\right].$$

The value of ρ is determined in the same way as in the linear case, which is by minimizing the KL divergence between the true density $G\big(\phi(z)|S\big)$ and its approximation $\hat{G}\big(\phi(z)|S\big)$:

$$J(\rho) = \int G(\phi(z)|S)\log\frac{G(\phi(z)|S)}{\hat{G}(\phi(z)|S)}\mathrm{d}\phi(z).$$

This yields the optimal value

$$\rho* = \frac{1}{M-p}\sum_{i=p+1}^{M}\lambda_i.$$

Since Σ_ϕ and the centered Gram matrix $\tilde{K}/M$ have the same nonzero eigenvalues, $\rho*$ can be simplified by employing the property of matrix:

$$\rho* = \frac{1}{M-p}\Big[trace(\tilde{K}/M) - \sum_{i=1}^{p}\lambda_i\Big]. \tag{16}$$

The testing procedure is similar to the one used in the linear cases. When a test sample x_t is queried, we project $z_r = x_t - x_r,\ x_r \in X$ into the principal kernel tangent subspace and compute the PTS-II distance measure:

$$\hat{D}(x_t, x_r) = \hat{D}(z_r) = \sum_{i=1}^{p}\frac{y_i^2}{\lambda_i} + \frac{\varepsilon^2(z_r)}{\rho}. \tag{17}$$

Similarly, x_t is assigned the same class label as the prototype x_r, which has the shortest distance to x_t.

3.3. Manifold Subspace Case: PTS-III

Consider that the sample x is coming from a certain manifold function $\phi(\cdot)$, TD in (1) can be rewritten as

$$TD_m(x, x_r) = \min_\alpha \|\phi(x) + \alpha T - \phi(x_r)\|. \tag{18}$$

Therefore, similar to the linear case, a manifold tangent vector set S_m can be defined as:

$$\begin{aligned}S_m = \{\underline{z}|\ \underline{z} = \phi(x) - \phi(x_r), &\text{ if } c(x) = c(x_r) \\ &\text{and } \phi(x) \in \mathcal{N}(\phi(x_r))\},\end{aligned} \tag{19}$$

where the size of S_m is also denoted by M.

The set S_m can also be characterized by subspace models. Let's consider the dot product in the corresponding manifold tangent subspace:

$$\begin{aligned}\langle \underline{z}, \underline{z}' \rangle &= \langle \phi(x) - \phi(x_r), \phi(x') - \phi(x_r') \rangle \\ &= \langle \phi(x), \phi(x') \rangle - \langle \phi(x), \phi(x_r') \rangle \\ &\quad - \langle \phi(x_r), \phi(x') \rangle + \langle \phi(x_r), \phi(x_r') \rangle.\end{aligned}$$

By introducing kernel tricks, we obtain

$$\langle \underline{z}, \underline{z}' \rangle = k(x, x') - k(x, x_r') - k(x_r, x') + k(x_r, x_r').$$

Similar to Kernel PCA, an $M \times M$ tangent Gram matrix $\underline{K}$ can be obtained from S_m with the element $\underline{K}_{ij} = \langle \underline{z}_i, \underline{z}_j \rangle$.

The following work is similar to PTS-II: one Gaussian density $\hat{G}(\underline{z}|S_m)$ is estimated from S_m to develop the PTS-III distance measure:

$$\hat{\underline{D}}(\underline{z}) = \sum_{i=1}^{p} \frac{y_i^2}{\lambda_i} + \frac{\varepsilon^2(\underline{z})}{\rho}, \tag{20}$$

where $\{y_i\}$ are the kernel projection of $\underline{z}$ by eigenvectors of the Gram matrix $\underline{K}$, $\{\lambda_i\}$ are corresponding eigenvalues, and the optimal weight coefficient is

$$\rho* = \frac{1}{M-p}\Big[trace(\underline{K}/M) - \sum_{i=1}^{p} \lambda_i\Big]. \tag{21}$$

Since the testing procedure is similar to those in previous cases, it is omitted here.

3.4. Unified Classification Scheme for PTS

To summarize, Table 1 presents a unified classification scheme for the three formulations of PTS. For convenience, the square root of three PTS distance measures is denoted as D_{PTS} in the following.

Table 1. Unified Classification Scheme for PTS

Input: Given training data set $X = \{x_i\}_{i=1}^m \subset \mathcal{R}^N$
Training phase:
S1: Form the covariance matrix or the Gram matrix by the tangent vector set;
S2: Perform (kernel) eigen-decomposition to obtain the principal (kernel) tangent subspace;
S3: Estimate the weight coefficient ρ.
Testing phase, for a test sample x_t:
S4: Project $z_r = x_t - x_r$ or the manifold difference $\underline{z}_r$ into principal tangent subspace;
S5: Compute the reconstruction error $\varepsilon^2(\cdot)$;
S6: Compute the PTS distance measure D^2_{PTS};
S7: Repeat *S4*–*S6* for each prototype;
S8: Return the label of the prototype, which has the shortest PTS distance to x_t, as classification result of x_t.

4. Discussions and Related Work

In this section, we firstly present a geometric interpretation of D_{PTS}, then discuss some issues in PTS, and finally discuss some related work.

4.1. Geometric Interpretation of D_{PTS}

Consider that a manifold M_{xr} is generated by small transformations of a sample x_r, the corresponding tangent space $\hat{M}_{xr}$ is spanned by the pattern variations according to the basic assumption. Then PTS-I is essentially a probabilistic representation of $\hat{M}_{xr}$.

Given a test sample x_t on M_{xr}, PTS-I first projects $z = x_t - x_r$ into $\hat{M}_{xr}$. Denote the projection as y, and its length as $\|y\|$. Thus TD can be approximated by:

$$T\hat{D}^2(x_t, x_r) = \|x_r - x_t\|^2 - \|y\|^2 \approx \varepsilon^2(z),$$

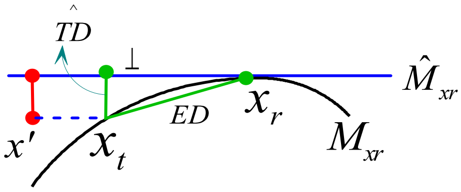

Figure 2. Geometric Interpretation of D_{PTS}.

which is shown in Figure 2.

It seems that $T\hat{D}$ would be enough for classification tasks. However, it is not true in practice. Consider when a test sample x' does not locate on M_{xr} and $c(x') \neq c(x_r)$, $T\hat{D}(x', x_r)$ may have the same length as $T\hat{D}(x_t, x_r)$. This case is illustrated in Figure 2. Therefore, we should judge whether x_t does locate on M_{xr} or not. This could be realized by computing a Mahalanobis Distance (MD) between x_t and x_r in the corresponding principal tangent subspace. In fact, when x_t locates on M_{xr}, $MD(x_t, x_r)$ will be much shorter than the case that x_t does not locates on M_{xr}. Combining MD and $T\hat{D}$ together yields the final D^2_{PTS}:

$$D^2_{PTS}(x_t, x_r) = MD^2(x_t, x_r) + \gamma \cdot T\hat{D}^2(x_t, x_r).$$

Note that the combining coefficient $\gamma = 1/\rho$ is generally much larger than 1 in practice, so that $T\hat{D}$ can play main role in the final decision.

Similar interpretations can be given to PTS-II and PTS-III.

4.2. Several Issues in PTS

The first issue is the capacity of PTS when it is employed to characterize tangent space. Intuitively, this capacity will be affected by the amplitude of the variations. Small variations may be easy to handle. In PTS, it is necessary to predefine the size of neighborhood $\mathcal{N}(x_r)$, which reflects the amplitude of the variations. In our experiments, if each category has very few samples, all the samples in the same category of x_r belong to $\mathcal{N}(x_r)$. Otherwise, a proper size for $\mathcal{N}(x_r)$ should be chosen, such as k-nearest neighbors of prototype x_r. Additionally, the variations are characterized by a uniform subspace in current PTS. However, if the variations are quite different between categories, separate subspaces should be constructed for each category to characterize the corresponding variations more precisely. And if the variations in some categories are too complicated, they should be encoded by multiple subspaces. In our tentative plan, the multiple subspaces based encoder may be the Vector-Quantization PCA (Kambhatla & Leen, 1997).

The second issue is learning prototype samples for PTS. Learning prototype samples is an important subject in instance-based learning. Due to the length limitation, this paper won't get involve in it deeply. In our experiments, if the training set was small, all training samples were used as prototype samples. Otherwise, we simply adopted the methods by Hastie et al. (1994).

Another issue is the computational efficiency of PTS. In fact, the linear case ran very fast in the test procedure, while the nonlinear case and the manifold case took more computational cost as most kernel-based algorithms. However, their computational cost could be greatly decreased by introducing the cache technique, which was used in many SVM software packages.

4.3. Relationship to Bayesian Face Matching

Much research work related to PTS has been conducted. One of the most influential methods is the Bayesian Face Matching (BFM) algorithm (Moghaddam, 2002), which is also known as one of the best face recognition algorithms in the FERET test (Phillips et al., 2000).

Given training face image set $\{x_i\}_{i=1}^{m} \subset \mathcal{R}^N$, BFM defines the image intensity difference $z{=}x_i - x_j$ and the intra-class difference set Ω_I [2] as:

$$\Omega_I = \{z | z = x_i - x_j, \;\; \text{if } i \neq j \text{ and } c(x_i) = c(x_j)\}.$$

The BFM algorithm estimates one Gaussian density function in Ω_I, and uses the estimated intra-class likelihood $\hat{P}(z|\Omega_I)$ as the similarity measure. Succeeding work also develops distance measure that is similar to the PTS-I distance measure (Wang & Tang, 2003).

[2] In fact, Moghaddam's paper does not give this set explicitly, but its description is equal to this definition.

The definition of Ω_I is very similar to the definition of the tangent vector set S. In fact, since the number of training samples per category is usually very small in face recognition tasks, all training samples are used as prototypes and treated as neighbors of each other, which makes PTS-I degrade to the BFM algorithm. In this sense, BFM can be viewed as a special case of PTS-I, and PTS could achieve promising performance in face recognition tasks. Additionally, the PTS framework gives a geometric interpretation of BFM from the perspective of tangent distance.

4.4. Comparison with Invariant Kernels

Besides those invariant kernel methods mentioned previously, there are other two invariant kernel methods, which are very closely related to PTS.

The first work is essentially performing PCA on the tangent vector set as preprocessing (Schölkopf et al., 1998a). Let the tangent covariance matrix be C and $B = C^{-\frac{1}{2}}$. Schölkopf's method transforms input data into a new space by: $x \rightarrow Bx$. Therefore, an invariant kernel was designed for linear SVM:

$$k(x, x') = \langle Bx, Bx' \rangle.$$

Although PTS-I also performs PCA on the tangent vector set, there are at least three big differentiae. First, tangent covariance matrices are derived in different ways. Schölkopf's method derives the tangent covariance matrix from the SVM's decision function, and does not give an operable representation of tangent vectors for general data sets. Contrarily, PTS-I provides an operable representation of tangent vectors directly from TD, and thereafter obtains the tangent covariance matrix. Second, the projection is different. Schölkopf's method projects x to Bx, while PTS-I projects $z = x - x_r$ into the tangent space. Third, their goals are different in performing PCA. Schölkopf's method aims at designing an invariant kernel for SVM, while PTS-I aims at obtaining a complete probabilistic representation of the tangent space.

The other work (Chapelle & Schölkopf, 2002) is substantially a nonlinear extension of Schölkopf's work, which performs Kernel PCA on the manifold tangent vector set to design a new invariant kernel for nonlinear SVM. The differentiae between Chapelle & Schölkopf's work (2002) and PTS-III are similar to those between PTS-I and Schölkopf's work (1998a).

5. Experiments

To evaluate the performance of the proposed three PTS models in general pattern classification tasks, we

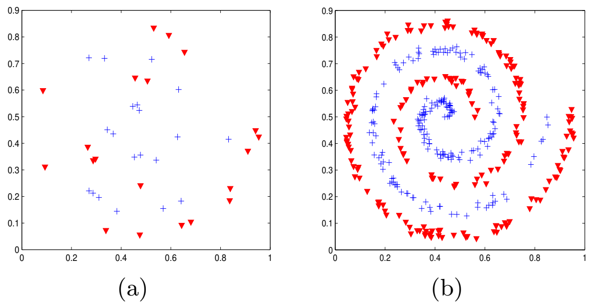

Figure 3. Data sets of interweaving spirals. Sample points in (a) are for training, while samples in (b) are for testing.

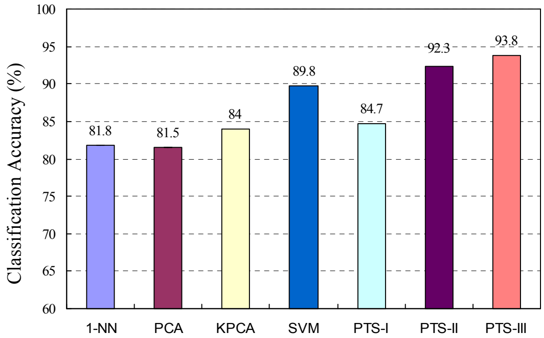

Figure 4. Classification results on synthetic data set.

conducted experiments on both a synthetic data set and two real non-image data sets.

5.1. The Two-Spirals Example

This experiment was conducted on a synthetic two-spiral, two-dimensional and two-class data set, which was partitioned into a training subset and a testing subset. The training subset, shown in Figure 3(a), contained 40 sample points (20 samples in each category). The testing subset, shown in Figure 3(b), contained 400 sample points. The two-spirals example was chosen for this study because its data set contains remarkable variations.

Several other instance-based learning algorithms such as 1-Nearest-Neighbor (1-NN), PCA and Kernel PCA were implemented in comparison with PTS. Note that PCA and Kernel PCA were followed by a 1-NN classifier. One state-of-the-art classification algorithm, SVM, was also implemented for comparison. The dimension of principal subspace was set to 2 for PCA and Kernel PCA, and the size of $\mathcal{N}(x_r)$ was set to 6 for PTS. Kernel parameters were well tuned to achieve the best results for kernel-based algorithms. In this experiment, it turned out that RBF kernel was better for Kernel PCA and SVM, while SIGMOID kernel was better for PTS-II and PTS-III.

Figure 4 shows the results. It is obvious that PTS, especially PTS-II and PTS-III, achieves promising performance and outperforms many other algorithms.

5.2. SPECT Heart Diagnosis Experiments

In this section, PTS was employed in a challenging classification task using the data sets from the UCI Machine Learning Repository: SPECT/SPECTF heart diagnosis data sets. These two data sets are very good for testing machine learning algorithms. Both of them describe diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images. Each contains 267 samples, which are represented by attributes summarizing the original SPECT images. SPECT has 22 binary attributes, whereas SPECTF has 44 continuous attributes.

The task was to diagnose whether each patient or image was normal or abnormal in heart. Both data sets were partitioned into two subsets: a training subset and a testing subset. The training subset contained 80 samples (40 samples in each category), and the testing subset contained 187 samples.

Similar to the synthetic example, 1-NN, PCA, Kernel PCA and SVM were implemented in comparison with PTS. The dimension of principal subspace was set to 10 for PCA, Kernel PCA and PTS, and the kernel parameters were also well tuned. Table 2 shows the results. The known best results of this task, which were achieved by CLIP4 (Cover Learning using Integer Programming) and the ensemble of CLIP4 (Kurgan et al., 2001; Kurgan & Cios, 2002), are also shown in the table for comparison.

Table 2. Classification accuracies (%) on the SPECT task.

Algorithms	SPECT	SPECTF
1-NN	80.2	72.1
PCA	73.3	75.1
KPCA	86.1	89.8
SVM	89.9	93.7
CLIP4	90.4	77.0
PTS-I	71.7	73.2
PTS-II	92.0	94.1
PTS-III	92.5	94.9

The results show that PTS-I did not achieve promising performance in this task, while PTS-II and PTS-III achieved extremely good results, and outperformed the other algorithms. The reason may be that there is strong nonlinearity in these two data sets.

6. Conclusion

This paper addressed three invariant classification algorithms for general data sets under a unified view of Probabilistic Tangent Subspace (PTS). We made one basic assumption: tangent vectors can be approximately represented by the pattern variations. Starting from this assumption, three probabilistic subspace models, which involve the linear subspace, nonlinear subspace and manifold subspace models, are proposed to characterize the variations. Experiments showed that PTS could achieve promising performance in more general data classification tasks.

The main contribution can be summarized as: this study proved that tangent vectors can be approximately represented by the pattern variations. Although PTS was successfully applied in some tasks, there are still many unsolved issues. Future work is planned to address possible solutions. We also hope that our work could provide some illuminating points for other researchers in invariant pattern classification.

Acknowledgments

The authors would like to thank anonymous reviewers for their very helpful comments and suggestions.

References

Burges, C. (1999). Geometry and invariance in kernel based methods. *Advances in kernel methods – support vector learning*. Cambridge, MA: MIT Press.

Chapelle, O., & Schölkopf, B. (2002). Incorporating invariances in nonlinear svms. *Advances in Neural Information Processing Systems 14* (pp. 609–616).

DeCoste, D., & Schölkopf, B. (2002). Training invariant support vector machines. *Machine Learning*, *46*, 161–190.

Haasdonk, B., & Keysers, D. (2002). Tangent distance kernels for support vector machines. *Proc. of the 16th International Conference on Pattern Recognition* (pp. 864– 868).

Hastie, T., Simard, P., & Saeckinger, E. (1995). Learning prototype models for tangent distance. *Advances in Neural Information Processing Systems 7*.

Kambhatla, N., & Leen, T. (1997). Dimension reduction by local principal component analysis. *Neural Computation*, 1493–1516.

Keysers, D., Macherey, W., Dahmen, J., & Ney, H. (2001). Learning of variability for invariant statistical pattern recognition. *Proc. of the 12th European Conference on Machine Learning* (pp. 263–275).

Kurgan, L., & Cios, K. (2002). Ensemble of classifiers to improve accuracy of the CLIP4 machine learning algorithm. *Proc. of the SPIE International Conference on Sensor Fusion: Architectures, Algorithms, and Applications VI* (pp. 22–31).

Kurgan, L., Cios, K., Tadeusiewicz, R., Ogiela, M., & Goodenday, L. (2001). Knowledge discovery approach to automated cardiac SPECT diagnosis. *Artificial Intelligence in Medicine*, *23*, 149–169.

Moghaddam, B. (2002). Principal manifolds and probabilistic subspace for visual recognition. *IEEE Trans. PAMI*, *24*, 780–788.

Moghaddam, B., & Pentland, A. (1997). Probabilistic visual learning for object representation. *IEEE Trans. PAMI*, *19*, 696–710.

Phillips, P., Moon, H., & et al. (2000). The FERET evaluation methodology for face recognition algorithms. *IEEE Trans. PAMI*, *22*, 1090–1104.

Schölkopf, B., Burges, C., & Vapnik, V. (1996). Incorporating invariances in support vector learning machines. *ICANN96 (LNCS, Vol. 1112)*. Springer.

Schölkopf, B., Simard, P., Smola, A. J., & Vapnik, V. (1998a). Prior knowledge in support vector kernels. *Advances in Neural Information Processing Systems 10* (pp. 640–646).

Schölkopf, B., Smola, A., & Müller, K. R. (1998b). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*, 1299–1319.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Simard, P., LeCun, Y., & Denker, J. (1993). Efficient pattern recognition using a new transformation distance. *Advances in Neural Information Processing Systems 5* (pp. 50–58).

Simard, P., LeCun, Y., Denker, J., & Victorri, B. (2001). Transformation invariance in pattern recognition – tangent distance and tangent propagation. *International Journal of Imaging System and Technology*, *11*, 181–194.

Wang, X., & Tang, X. (2003). Unified subspace analysis for face recognition. *Proc. of IEEE International Conference on Computer Vision*.