Information Sciences 181 (2011) 2071–2085

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Personalized mode transductive spanning SVM classification tree

Shaoning Pang [a,*], Tao Ban [b], Youki Kadobayashi [c], Nikola Kasabov [d]

[a] Department of Computing, Unitec Institute of Technology, Private Bag 92025, New Zealand
[b] Information Security Research Center, National Institute of Information and Communications Technology, 184-8795, Tokyo, Japan
[c] Graduate School of Information Science, Nara Institute of Science and Technology, Japan
[d] Knowledge Engineering and Discovery Research Institute, Auckland University of Technology, Private Bag 92006, Auckland 1020, New Zealand

ARTICLE INFO

Article history:
Received 28 February 2009
Received in revised form 27 October 2010
Accepted 1 January 2011
Available online 14 January 2011

Keywords:
SVM aggregating intelligence
Personalized transductive learning
SVM classification tree
Transductive SVMT

ABSTRACT

Personalized transductive learning (PTL) builds a unique local model for classification of individual test samples and is therefore practically neighborhood dependant; i.e. a specific model is built in a subspace spanned by a set of samples adjacent to the test sample. While existing PTL methods usually define the neighborhood by a predefined (dis)similarity measure, this paper introduces a new concept of a knowledgeable neighborhood and a transductive Support Vector Machine (SVM) classification tree (t-SVMT) for PTL. The neighborhood of a test sample is constructed over the classification knowledge modelled by regional SVMs, and a set of such SVMs adjacent to the test sample is systematically aggregated into a t-SVMT. Compared to a regular SVM and other SVMTs, a t-SVMT, by virtue of the aggregation of SVMs, has an inherent superiority in classifying class-imbalanced datasets. The t-SVMT has also solved the over-fitting problem of all previous SVMTs since it aggregates neighborhood knowledge and thus significantly reduces the size of the SVM tree. The properties of the t-SVMT are evaluated through experiments on a synthetic dataset, eight bench-mark cancer diagnosis datasets, as well as a case study of face membership authentication.

1. Introduction

The widely used inductive reasoning approach is concerned with the creation of a model from all training data that represents the available information from the problem space (induction), and then the application of the model to new coming data to predict the property of interest (deduction or generalization). In contrast, the transductive approach, first introduced by Vapnik [30], is defined as a method that estimates the value of an unknown model for a single point in the problem space employing information from testing data, in addition to the training data. While the inductive approach is useful when an global model of the problem is needed in an approximate form, the transductive approach is more appropriate for applications where the focus is not on the overall precision of the model, but rather on each individual case. In this sense, transductive learning effectively fits the cases in clinical and medical applications where interest is on the preciseness of the prediction for an individual patient, as well as information retrieval applications where unlabelled data can likely be used for better text categorization [5].

Unlike an inductive learner which conducts learning only on training data $h_L = L(S_{train}) \rightarrow f$, a transductive learner learns from both training and test data, making predictions for a set of unlabelled data known at the learning time $h_L = L(S_{train}, S_{test}) \rightarrow \ell$, where ℓ is the predicted label of S_{test}. This is also different from semi-label setting inductive learning

* Corresponding author. Tel.: +64 9 8154321x 8392; fax: +64 9 8154338.
E-mail addresses: ppang@unitec.ac.nz (S. Pang), bantao@nict.go.jp (T. Ban), youki-k@naist.ac.jp (Y. Kadobayashi), nkasabov@aut.ac.nz (N. Kasabov).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.01.008

[7], where the learning function is constructed to make predictions on any possible observation $h_L = L(S_{train}, S_{test}) \rightarrow f$. With respect to how the unlabelled data S_{test} is used for transductive learning, two types of transductive learning approaches have been discussed in the literature.

1.1. Transductive learning

In inductive learning, one learns a function that makes predictions on the whole space. Transduction only concerns itself with predicting the values of the function at the test points of interest; thus in transduction, no model is constructed. However, transductive learning shares commonalities with semi-supervised learning (one type of inductive learning) in that both are using information contained in unlabelled data we possess in addition to the labelled training set.

Semi-supervised learning approaches [32,4,7] focus on finer adjustment of the decision rule learned from training samples by also incorporating the knowledge in test samples. First, an inductive learner $h_L = L_i(S_{train})$ is built for an initial decision rule, then the unlabelled samples S_{test} are associated with 'pseudo' labels. The test set with the 'pseudo' labels together with the training set with true labels forms the so-called semi-labelled data. Next, transductive samples are re-sampled from the semi-labelled samples according to a given criterion in order to define a hybrid training set made up of S_{test} and S_{train}. Finally, the resulting hybrid training dataset is used to find more reliable discriminating rules integrating the distribution information presented in all available samples.

Transduction works because the test set can give a nontrivial factorization of the function class. Transductive SVM (TSVM) is a typical model employing testing data for transductive learning [29]. Joachims [14] implemented TSVM for text classification tasks with favorable results reported especially for problems with small training datasets. Later, Chen et al. [8] introduced a new TSVM termed progressive transductive SVM, using pairwise handling of negative and positive samples. Recently, Bruzzone et al. [4] proposed an interesting multi-class TSVM to address ill-posed remote-sensing problems.

1.2. Personalized transductive learning

Personalized transductive learning (PTL) methods approach 'personalized' learning by creating a unique model for each test sample based on its neighboring samples. A typical personalized learning approach usually consists of two steps: (1) *Neighbor-sample filtering*: for each new input sample $\boldsymbol{x}_j \in S_{test}$ that needs to be processed for a prognostic/classification task, its N_j nearest neighbors are selected from the training set to form a neighborhood set $D_j \in S_{train}$. (2) *Regional decision making*: a personalized model M_j is created on D_j to approximate the function value y_j at point $\boldsymbol{x}_j$. PTL is different from traditional lazy learning [3] in that it performs batch learning with the assumption that a complete training dataset is given in advance, whereas lazy learning stores training data for the future use, rather than for constructing a general target function [1].

The simplest PTL model is the k nearest neighbor classifier (kNN) [9]. kNN dynamically creates a prediction model for each test sample by learning from its k nearest neighbors. Another model is Neuro-Fuzzy Inference (NFI) developed recently by Song and Kasabov [27]. NFI constructs a local inference model for every new input vector based on data instances closest to this vector data from an existing database. It differs from kNN at employing fuzzy inference instead of distance comparison for classification decision making. Though NFI is shown outperforming kNN on some benchmark datasets, their generalization abilities are not significantly different because the personalized prediction models for NFI and kNN are based on the same neighbor-sample filtering.

It is worth noting that for data collected from real world applications, the neighborhood directly defined by inter-sample dissimilarities is subjected to high noise or ill-posed conditions. This in turn renders the personalized prediction models unreliable. Fig. 1 illustrates different kinds of neighborhood data distribution patterns. An ordinary inter-sample dissimilarity-based method is unable to approximate such complex neighborhood patterns precisely, because samples adjacent to the query sample might be presented merely as noise in terms of decision making [13].

Thus for better neighborhood modelling, besides the distance metric, it is desirable also to take the classification information/knowledge into consideration.

Motivated by this, we propose a transductive SVM tree (t-SVMT), which implements personalized multi-model cooperative learning in a transductive manner. A t-SVMT defines multi-instance packages termed particles by exploring the discriminative information in the neighborhood of a test sample. The number of samples in a particle is flexible; it follows that a particle may contain just one instance, or as many as half of the full dataset. The discriminative information of a particle is evaluated in advance by an SVM-based cost function. Thus, a t-SVMT is a dynamic structure composed of a series of flexibly-sized knowledgeable units. Note that a t-SVMT differs substantially from TSVM in that it personalizes transductive learning by transductively aggregating a group of inductive SVMs, whereas TSVM performs semi-labelled transductive learning using a single SVM.

1.3. Paper organization

The rest of the paper is organized as follows. Section 2 introduces related researches and the motivation for the proposed PTL learning. Section 3 describes the proposed methodology of a transductive spanning SVM classification tree. Section 4 gives the detail of the proposed t-SVMT training and testing algorithms. Experiments and discussion are given in Section 5. Finally, conclusions and directions for future research are presented in Section 6.

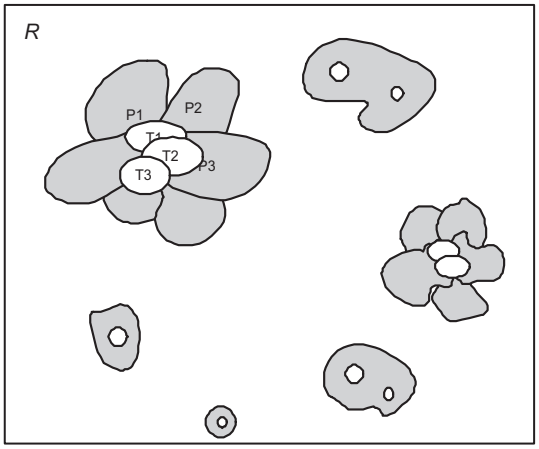

Fig. 1. Illustration of several neighborhood data distribution patterns, where T_i and P_i represent an individual data distribution sub-pattern for testing and training, respectively.

2. Related researches and motivations

2.1. Neighborhood calculation

For personalized transductive learning, a neighborhood is required to be constructed dynamically according to the class data distribution around the test sample. Given a query sample $\boldsymbol{x}$ in data space D, the neighborhood of $\boldsymbol{x}$ is defined as a subset Z_x adjacent to $\boldsymbol{x}$, possibly containing samples from different classes.

Generally, Z_x can be found by selecting samples adjacent to $\boldsymbol{x}$, given a dissimilarity measure defined over pairwise samples,

$$Z_x = \{\boldsymbol{x}, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_k | d(\boldsymbol{z}_i, \boldsymbol{x}) < \theta\}, \tag{1}$$

where $\boldsymbol{z}_i$ represents a neighboring sample, $d(\boldsymbol{z}_i, \boldsymbol{x})$ is the distance between $\boldsymbol{z}_i$ and the query sample $\boldsymbol{x}$, θ is a predefined distance upper-bound controlling the capacity of the neighborhood.

With regard to generalization ability, personalized transductive learning assumes that $f(\boldsymbol{x}, Z_{\boldsymbol{x}})$ approximates the ground truth $f(\boldsymbol{x})$ better than $f(\boldsymbol{x}, D)$ because noise data is removed due to neighboring instance selection. Thus, a personalized transductive function can be learned just from the neighborhood dataset Z_x, rather than from the full dataset D, that is,

$$f(\boldsymbol{x}) = f(Z_x) = f(\boldsymbol{x}, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_k). \tag{2}$$

For example, kNN performs a personalized classification as follows. Given a new instance $\boldsymbol{x}$, kNN creates a k instances neighborhood Z_x based on the given distances metric. It then employs a majority-voting rule on Z_x and assigns x to the class represented by a majority of the samples in Z_x.

In the literature, the above personalized transductive learning has been renovated to build a powerful $f(Z_x)$. As a result, different types of kNN and weighted kNN were proposed to differentiate the importance of different neighbors for better classification or decision making. Though the NFI [27] outperforms kNN in classification, NFI and kNN are eventually the same type of personalized transductive learning, since they both use the same inter-sample dissimilarity measure and select the same neighborhood.

For neighborhood construction, Eq. (1) exclusively relies on inter-sample dissimilarity evaluation. The obtained neighborhood, depending on what kind of dissimilarity measure is used, is presented either as a spherical scope as shown in Fig. 2(b) or a cube as in Fig. 2(c). Such neighborhood constructions are improper when a dynamic neighborhood is required for more accurate personalized modelling. Moreover, the neighborhood obtained from Eq. (1) is merely in the sense of dissimilarity measure, ignoring discriminative knowledge between different classes, which in some cases leads to degeneration in the generalization performance.

2.2. Personalized neighborhood calculation

Here we introduce a personalized $f(Z_x)$ model for multi-model cooperative transductive learning, which transductively solves the problem by cooperative computing over a number of regional models [17].

The neighborhood Z_x is modelled based on the following two principles: (1) class label information is incorporated to define the neighborhood, and (2) a neighborhood is a set of flexibly-sized multi-instance packages, whose data points

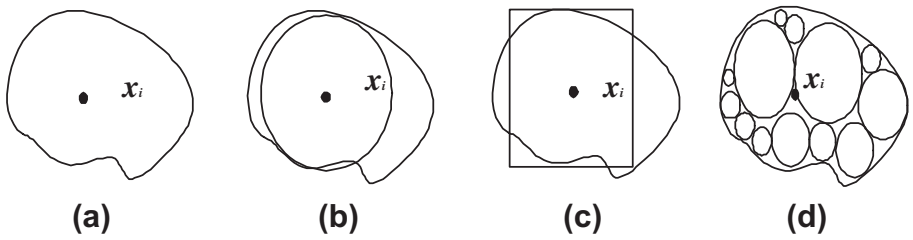

Fig. 2. Illustration of dynamic neighborhood modelling. (a) Truth personalized neighborhood of x_i; (b) spherical neighborhood approximation; (c) cubic neighborhood approximation; (d) the proposed subset aggregated neighborhood.

follow the smoothness assumption [6]: given two instance packages in a high-density region, if one package is neighboring $\boldsymbol{x}$, then it is true for the other package. Mathematically, the proposed personalized neighborhood model is defined as

$$\begin{aligned} & Z_x^* = Z_i \cup Z_j \cup \{\boldsymbol{x}\}, Z_i \leftarrow \wp(D), Z_j \leftarrow \wp(D) \\ \text{Subject to } & d(Z_i, \boldsymbol{x}) < \theta, \quad \text{and} \quad E(Z_i) < \xi, \\ & d(Z_j, \boldsymbol{x}) > \theta, d(Z_i, \boldsymbol{x}) < \theta, \quad \text{and} \quad d(Z_i, Z_j) < \theta, \end{aligned} \tag{3}$$

where θ is a predefined distance upper bound. Z_i is taken as a neighboring package to $\boldsymbol{x}$ by not only physical distance $d(Z_i,\boldsymbol{x})$, but also classification cost $E(Z_i)$ where class label is considered. $d(Z_i,\boldsymbol{x})$ is the distance between Z_i and query sample $\boldsymbol{x}$. $E(Z_i)$ is the regional cost function explained later in Eq. (5). Additionally by the smoothness assumption, Z_j is in the neighborhood of $\boldsymbol{x}$, since Z_i and Z_j are close despite $d(Z_j,\boldsymbol{x}) > \theta$.

It is noticeable that Z_i and Z_j is a flexible-size multi-instance package, which is a subset of D, or sometimes only a single instance. Also, Z_x^* considers the class label information as every Z_i is obtained from a supervised clustering method $\wp(D)$. Therefore, Z_x^* is personalized for x in terms of the membership of the neighborhood.

Applying Eq. (3) to Eq. (2) gives a new personalized transductive model $f(x)'$ based on model aggregation,

$$f'_{\boldsymbol{x}} = f'(Z_x^*) = f(Z_1, Z_2, \ldots, Z_k, \{\boldsymbol{x}\}) = \bigvee_{i=1}^{k} \{f_{Z_i}, Z_i\}, \tag{4}$$

where $\bigvee$ denotes the union between models and f_{Z_i} represents an elementary model (e.g. an SVM) on neighborhood dataset Z_i. Eq. (4), compared to Eq. (2), gives a more flexible and personalized neighborhood modelling. Fig. 2 illustrates the benefits of the new approach. Given that the true personalized neighborhood of instance $\boldsymbol{x}_i$ is as shown in Fig. 2(a). Fig. 2(b) and (c) give two types of neighborhood approximation based on inter-sample dissimilarity evaluation. In contrast, Fig. 2(d) gives an aggregated neighborhood by a set of multi-instance packages represented as circles in the figure. Obviously, Fig. 2(d) presents a more accurate neighborhood approximation than Fig. 2(b) and (c).

2.3. Transductive aggregating system

Under the framework of SVM aggregating intelligence [21], SVM aggregating has been studied for creating SVM-based multi-core cooperative computing techniques towards optimized inductive learning. SVM ensemble is a type of inductive SVM aggregation in the principle of deriving diverse SVM experts to improve the generalization ability of regular SVMs for decision making [31]. SVM ensemble learning assumes that the number of SVMs in the aggregation should be known in advance as prior knowledge despite the number often being difficult to determine in real applications. SVM classification tree (SVMT) [22] is another type of inductive SVM aggregation. It overcomes the difficulty of SVM ensemble by automatically determining the number of SVMs during learning. The disadvantage of existing SVMTs [21,22] is that the spanning of SVMT is completely data-driven (hereinafter DDS_SVMT): the tree grows easily resulting in a large-size SVM decision tree over-fitting to the training data.

Here we present a new SVM aggregating method which transductively constructs an SVMT. The proposed t-SVMT aggregates a set of SVMs adjacent to the test sample and utilizes the knowledge in the neighborhood for better classification. It is capable of preventing the over-fitting of SVMT learning as it aggregates only neighborhood knowledge largely reducing the size of the generated SVM tree. Moreover, the generalization ability of the constructed SVM tree is further enhanced because the neighborhood used is truly personalized and noise-free: every individual SVM in the neighborhood represents a piece of reliable regional knowledge whose relevance to the test sample has been verified in advance.

3. Transductive spanning SVM tree

Towards a multi-model transductive SVM aggregation classifier with high generalization ability, in this section we address three aspects of the model: (1) problem decomposition, i.e. data partitioning, (2) modelling of local knowledge, i.e. SVM training on multi-instance package, and (3) individual SVMs aggregation, i.e. transductive aggregation.

3.1. Data partitioning

In order to create partitions with good discriminative ability, we propose to partition data into a set of physically adjacent subsets with class label information taken into account.

Let $X = \{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}$ be the training set, and the associative labels given by $\{y_1,\ldots,y_N\}$, a new attribute z_i can be formed out of the original attributes $z_i = (\boldsymbol{x}_i, y_i, d_i)$ as the partition label of $\boldsymbol{x}_i$, decomposing X into a set of data partitions $\{Z_1,\ldots,Z_k\}$. A supervised data decomposition can be accomplished by introducing cost function based on a regional SVM approximation,

$$E(Z_k) = \sum_{x\in Z_k}(\boldsymbol{x}-\boldsymbol{x}_k) + \alpha\sum_{\boldsymbol{x}\in Z_k}(f_{svm}(\boldsymbol{x}_k) - f_{svm}(\boldsymbol{x})), \tag{5}$$

where f_{svm} is a standard SVM approximation given later in Eq. (9); $\boldsymbol{x}_k$ is the representative instance of Z_k, computed as the instance closest to the center of Z_k; and α is a balancing weight determined in practice by cross-validation tests. The default value of α is 1.

Given a partitioning function $\wp$ on X with adjustable partitioning scale, Eq. (5) can be used for a supervised partitioning procedure as,

$$\begin{aligned} &[X', g_1,\ldots,g_{ik},\ldots] = \wp(X,\rho),\\ &\text{Subject to}\quad E(g_k) < \xi \quad \text{for any} \quad k \end{aligned} \tag{6}$$

where ρ is the partitioning scale and ξ the threshold of E. As the result of Eq. (6), $g_1,\ldots,g_k$ are the set of partitions selected by optimizing the cost E, and $X' = X - \{g_1\cup,\ldots,g_k\}$ is the complement set.

Eq. (6) can be applied to X with a varying ρ value,

$$\begin{aligned} &[g_1,\ldots,g_i,\ldots] = \wp^n(X,\rho_0),\\ &\text{Subject to}\quad E(g_i) < \xi, \end{aligned} \tag{7}$$

where $\wp^n = \wp(\wp^{n-1}(X_{n-1},\rho_{n-1}),\rho_n)$, $\rho_n = \rho_{n-1} + \Delta$, and Δ is the partitioning scale interval normally set to a small value in the range of $\wp$ definition. For example, for K-means, Δ can be an integer greater-equal to 1.

By Eq. (8), X can be partitioned into a set of data partitions of different sizes $\{g_1,g_2,\ldots,g_i,\ldots\}$, such that $g_1\cup g_2\cup\cdots\cup g_i\cup\cdots\approx D$, and each partition g_i is associated with the partitioning scale ρ, in which g_i is produced.

3.2. SVM particles

To formalize classification as the task of finding partitions with maximum likelihood, a local SVM is required to associate with, and accurately approximate the class distribution of a partition obtained from the above partitioning procedure. An *SVM particle* is defined as a structure that combines the dataset g_i, and the trained SVM f^i_{svm} on g_i,

$$V_i = \{g_i, f^i_{svm}\}. \tag{8}$$

In the case that g_i contains data from two classes, a regular two-class SVM model f_{svm} is applied to f_i,

$$f_{svm} = sign\left(\sum_{i=1}^{l} y_i\,(\boldsymbol{w}_i^T\varphi(\boldsymbol{x}_i) + b_i^*)\right) \tag{9}$$

where φ is the kernel function, l is the number of training samples, and $\boldsymbol{w}_i$ and b_i^* are optimized by

$$\begin{aligned} &\min \quad \frac{1}{2}\boldsymbol{w}_i^T\boldsymbol{w}_i + C\sum_{t=1}^{l}\eta_i^2,\\ &\text{subject to}\quad y_i(\boldsymbol{w}_i^T\varphi(\boldsymbol{x}_t) + b_i) \geqslant 1-\eta_i, \end{aligned} \tag{10}$$

where C is the margin parameter to weight the error penalties η_i, and $\varphi(\cdot)$ is the mapping function implemented by the kernel function.

On the other hand, when Z_i only contains data from one class, a one-class SVM is applied. Following [25], a class SVM function can be modelled as an outlier classifier by labelling samples in Z_i as positive and samples in the complement set $\overline{Z_i}$ as negative. Then the SVM is trained on a dataset $Z_i' = \{\boldsymbol{x}_i | i = 1,\ldots,N\}$, and

$$y_i' = \begin{cases} +1 & \text{if} \quad \boldsymbol{x}_i \in Z_i,\\ -1 & \text{if} \quad \boldsymbol{x}_i \in \overline{Z_i}. \end{cases} \tag{11}$$

The following two types of SVM particles are resultantly obtained: (1) one-class SVM particles $V^{\langle 1\rangle} = \{g, \rho, f_{svm^{(1)}}\}$ for class 1, $V^{\langle 2\rangle} = \{g, \rho, f_{svm^{(2)}}\}$ for class 2, and (2) two-class SVM particle $V^{\langle 2\rangle} = \{g, \rho, f_{svm}\}$. Given a data partition belonging to one class, a one-class SVM is applied to model the data by separating the samples in the partition from the outliers. When the data belong to two classes, satisfying $E(g) > \xi$, a two-class SVM is applied to the partition, and a standard SVM particle is created.

3.3. Personalized mode transductive aggregation

Given a test sample $\boldsymbol{x}$, and a set of SVM particles $\{V_i\}$ derived over S_{train}, the distance between $\boldsymbol{x}$ and a particle Z_i is measured by the normalized Euclidean distance defined as follows,

$$d(\boldsymbol{x}, Z_i)^2 = \left(\boldsymbol{x} - \frac{\sum_{\boldsymbol{x}_k \in Z_i} \boldsymbol{x}_k}{|Z_i|}\right)^T \left(\boldsymbol{x} - \frac{\sum_{\boldsymbol{x}_k \in Z_i} \boldsymbol{x}_k}{|Z_i|}\right), \tag{12}$$

where $|Z_i|$ denotes the cardinal number of the particle.

Then, all selected SVM particles compose the knowledgeable neighborhood for the new input instance $\boldsymbol{x}$, and the decision of classification is made transductively by an aggregation $\hat{f}$ of those SVM particles in the neighborhood as,

$$\hat{f}(\boldsymbol{x}) = \begin{cases} f_{svm^{(1)}} & \text{if} \quad \boldsymbol{x} \rightharpoonup V^{\langle 1\rangle}, \\ f_{svm^{(2)}} & \text{if} \quad \boldsymbol{x} \rightharpoonup V^{\langle 2\rangle}, \\ f_{svm} & \text{otherwise}, \end{cases} \tag{13}$$

where $f_{svm^{(1)}}$ and $f_{svm^{(2)}}$ are one-class SVM decision makers for class 1 and 2, respectively, and f_{svm} is a 2-class SVM decision maker. Taking each SVM particle as a tree node and the partitioning scale ρ as the level of tree structure, $\hat{f}$ can be represented as an SVM tree. Thus, '$\rightharpoonup$' in (13) indicates that $\boldsymbol{x}$ is branched to a certain node of the SVM tree.

Clearly, there may be errors in the classification of the above constructed $\hat{f}$, as $\hat{f}$ may differ from the true classification function f. Thus, a suitably chosen real-value loss function $\mathcal{L} = \mathcal{L}(\hat{f}, f)$ is used to capture the extent of this error. Therefore loss $\mathcal{L}$ is data-dependent:

$$\mathcal{L} = |\hat{f} - y|. \tag{14}$$

As $\hat{f}$ is applied to datasets drawn from the data domain D under a distribution of $\theta(\boldsymbol{x})$. The expected loss can be quantified as,

$$E[\mathcal{L}] = \int_D |\hat{f} - y| \theta(\boldsymbol{x}) d\boldsymbol{x}. \tag{15}$$

Substitute $\hat{f}$ from Eq. (9), and the aggregation risk of SVMT is

$$\mathcal{L} = \sum_{i=1}^{I} |f_{svm_i} - f_i| \varrho_i + \sum_{j=1}^{J} |f_{svm_j^{(1)}} - f_j| q_{1j} + \sum_{k=1}^{K} |f_{svm_k^{(2)}} - f_k| q_{2k}, \tag{16}$$

where f_i, f_j, and f_k are the regional truth classification function. ϱ_i is the probability that the ith particle contains two-class data. $q_r (r = 1, 2)$ represent the probability that a partition contains data from only one class. I, J, and K represent the number of two-class SVM particles, one-class SVM particles for class 1 and class 2, respectively. Here, I, J, and K are determined automatically after $\hat{f}$ is created.

From Eq. (12), the aggregation risk is determined by 2-class SVM and one-class SVM classifiers in all the particles. Assume that every SVM in the aggregation uses the same kernel and penalty parameters, and the probability of the type of the particles determines the aggregation risk. In other words, the data partitioning function $\wp$ raises the risk of SVM aggregation. However, as the loss function of $\wp$ has already been minimized during the process in which the data are partitioned, an optimized SVM aggregation can be achieved as long as the risk from the SVM for each particle is minimized.

4. Proposed t-SVMT algorithm

In this section, the above personalized transductive modelling is interpolated as an algorithm of t-SVMT. First, the training data is divided in a recursive data partitioning procedure. Then, regional knowledge of the input data is approximated as a set of SVM particles. Finally, the selected transductive particles (i.e. particles neighboring to a test instance $\boldsymbol{x}$) are aggregated to a personalized SVM tree model.

4.1. Algorithms description

A t-SVMT can be constructed as follows:

Step 1. Partitioning. The input training data is decomposed and modelled into a set of SVM particles. Algorithm 1 gives the *partitioning* function, where ρ_0 is a predefined initial resolution for the t-SVMT to start analyzing. Default ρ_0 is normally set as a scale that gives the largest-sized data partitions. For example, for K-mean, ρ_0 is set as 2, and for the evolving clustering method (ECM) [16], ρ_0 is set as 0.9. If some prior knowledge of the data is known, e.g. serious class-imbalance and class-overlap, a finer scale is suggested to enable the t-SVMT to analyze data in a higher-resolution space.

In our experiments, we adopt a standard K-mean clustering approach with $\rho = 2$. Note that for a given dataset for classification, if a finer scale (i.e. $k > 2$ in terms of K-means) is used for t-SVMT construction, the training speed of the t-SVMT will become faster since it is easier in a finer-resolution space to find data partitions that are classification optimal. However, the generalization ability to generate the t-SVMT is likely to decrease due to the data being partitioned more than necessary.

Algorithm 1. Partitioning

Function: Partitioning (X_{train}, ρ_0)
X_{train}; /*training dataset*/
ρ; / *initial partitioning scale*/
P; /*output data partition set/
begin
$P = \emptyset$; /* initialize output */
$\rho = \rho_0$; /* initialize partitioning scale*/
if X_{train} is empty
return P; /* Iteration stops when X_{train} is empty */
$[Z_1, \ldots, Z_k]$ = Partition (X_{train}, ρ_0);
for each Z_k {
if all $\boldsymbol{x} \in Z_k$ is in one-class
$P = P \cup [Z_k, \rho]$;
if $E(Z_k) < \xi$/* classification cost function
$P = P \cup [Z_k, \rho]$;
$X_{train} = X_{train} - Z_k$};
if X_{train} size is not decreasing
$\rho = \rho + \triangle$;
Partitioning (X_{train}, ρ);/*zooming in data */
end

Step 2. Spanning transductive SVMT. Given a new input instance $\boldsymbol{x}$, a personalized neighborhood is constructed for $\boldsymbol{x}$ based on the measurement of the physical distance and regional classification cost. Unlike [19], the smoothness assumption in this work is specially applied for refining the obtained neighborhood. This helps to avoid further overfitting of the SVMT due to the sparseness of neighborhood space. Consequently, a personalized t-SVMT $\mathcal{T}_i$ is spanned by transductive SVM particles training followed by SVMT aggregation. Algorithm 2 describes the *t-SVMTtraining* function. In our experiments, two-class SVMs use a linear kernel and one-class SVMs employ an RBF kernel with the parameter adjusted via cross-validation.

Step 3. Testing the constructed t-SVMT $y = \mathcal{T}(\mathbf{x})$. Test sample $\boldsymbol{x}$ is first judged by the test function $\mathcal{T}_0(x)$ at the root node in the SVM tree. Depending on the decision made by the root node, $\mathbf{x}$ will be branched to one of the root node's children. This procedure is repeated until a leaf node or SVM node is reached, then the final classification decision is made for the test sample by a node one-class classifier or a regular SVM classification. Algorithm 3 presents the algorithm of *t-SVMTtesing*.

Algorithm 2. Transductive spanning SVM classification tree

Function: *t*-SVMTtraining (P, x, θ)
$\boldsymbol{x}$; /*a test instance*/
$\mathbf{K}$;/*SVM Kernel for constructing SVM Tree*/
$\mathcal{T}$; /* output *t*-SVMT of $\boldsymbol{x}$ */
begin
$V_t = \emptyset$;/*initialize transductive particle set*/
for each $Z_k \in P${
if $d(Z_k, \boldsymbol{x}) < \theta$
$V_t = V_t \cup [Z_k, \rho_k]$;
else if $d(Z_k, Z_j) < \theta$ and $Z_j \in V_t$/* smoothness assumption */
$V_t = V_t \cup [Z_k, \rho_k]$;}
$\mathcal{T} = \emptyset$;/* initialize *t*-SVMT as a root node */
for each $Z_k \in V_t${
if Z_k is one-class
M_k = Train_SVMone $(Z_k, \mathbf{K})$;/* one-class SVM*/
else
M_k = Train_SVM $(Z_k, \mathbf{K})$;
$\mathcal{T} = \mathcal{T} \cup [Z_k, M_k, \rho_k]$;}/*Add a tree node at levelρ_k*/
end

4.2. Algorithm complexity

Classical SVM algorithms have a time complexity of $O(N^3)$, where N is the number of training points. In the principle of divide and conquer, the large quadratic programming (QP) problem can be split into a series of small possible QP problems to reduce the computational complexity of SVMs to approximately $O(N^{2.1})$ [15].

Algorithm 3. t-SVMT testing

```
Function: t-SVMTtesting (T, x)
  T; /* a constructed t-SVMT*/
  x; /*a testing instance*/
  C; /*output class label*/
begin
1  Current ← 0; /* set root node as the current node */
2  while(SearchNode (T, Current) == Null{/* searching SVMT */
3    Next ← SearchNode (T, Current); /* until a terminal node is reached*/
4    Current ← Next;}
5  if  T_Current is m-class
6    C ← SVMTest(x, K, Current);
7  else
8    C ← SVMoneTest (x, Current); /* one-class SVM*/
9  return C;
end
```

An SVMT decomposes the problem by recursive data partitioning and local SVM approximation (i.e. small QP problem), which has the computational complexity $O(Mlog(M))$ where M is the number of data partitioning and SVM training. This computational complexity leads to a longer training time than single SVM. However, because of the rapid decrease of training points for both data partitioning and local SVM training, SVMT prunes the global QP problem rapidly to small-sized local SVM calculations for most problems. The proposed t-SVMT extracts only personalized neighborhood data partitions. This further reduces the average runtime to $O(M)$.

5. Experiments and discussion

In our experiments, algorithms are implemented on the basis of Simple SVM [18], run on Pentium 4 PC with 3.0 GHZ and 512 Mb RAM. We compare the results of the t-SVMT with those of standard inductive SVM, and previous SVM aggregation methods: SVM ensemble and DDS_SVMT [22]. On the other hand, we also compare the t-SVMT with other transductive methods including transductive SVM (TSVM) from SVMlight [14], kNN, and NFI [27].

5.1. Experimental setup

In the comparison, we set all two-class SVMs with a linear kernel, and 1-class SVMs with a RBF kernel, whose parameter together with the aggregation parameter ξ for SVMTs are determined by cross-validation tests. Also, we tune parameters for other methods under investigation such as kNN and NFI in cross-tests to derive their best performance for each case study. For performance evaluation, we use normally either a standard 10-fold or 5-fold cross-validation, where accuracies are averaged over 10 or 5 runs and at each run, one tenth/fifth of the data is used as testing set and the rest as training set.

In our investigation, we explore the procedure of t-SVMT construction, and examine the effectiveness of the t-SVMT algorithm on discriminability, robustness to class-imbalance, and capability of over-fitting control, respectively.

5.2. Tree investigation

We test the t-SVMT over a synthetic 2D Gaussian dataset with class distribution as shown in Fig. 3(a). The dataset has 1095 instances: 95 are from Class 2, and the remaining 1000 from Class 1. Class 2 here is the imbalanced class with a capacity of approximately one-tenth of Class 1. The two classes overlap at the center of the 2D input domain.

Fig. 3(b) records the data partitioning procedure, where the horizontal axis identifies the scale of data partitioning over the 2D input space (x and y axes), and each circle represents a data particle generated by Algorithm 1. Different sizes of particles are generated during the recursive data partitioning process. Fig. 3(c) resultantly summarizes the distribution of all obtained particles, where particles of different sizes are represented as a set of circles in different sizes and colors. It is noticeable that only a few circles appear at either the Class 1 or Class 2 region, and quite a large number of smaller circles are produced at the region with a clear class mixture. This suggests that the partitioning algorithm automatically zooms in for better resolution at areas with class-overlap. Fig. 3(d) presents the final neighborhood obtained with the smoothness

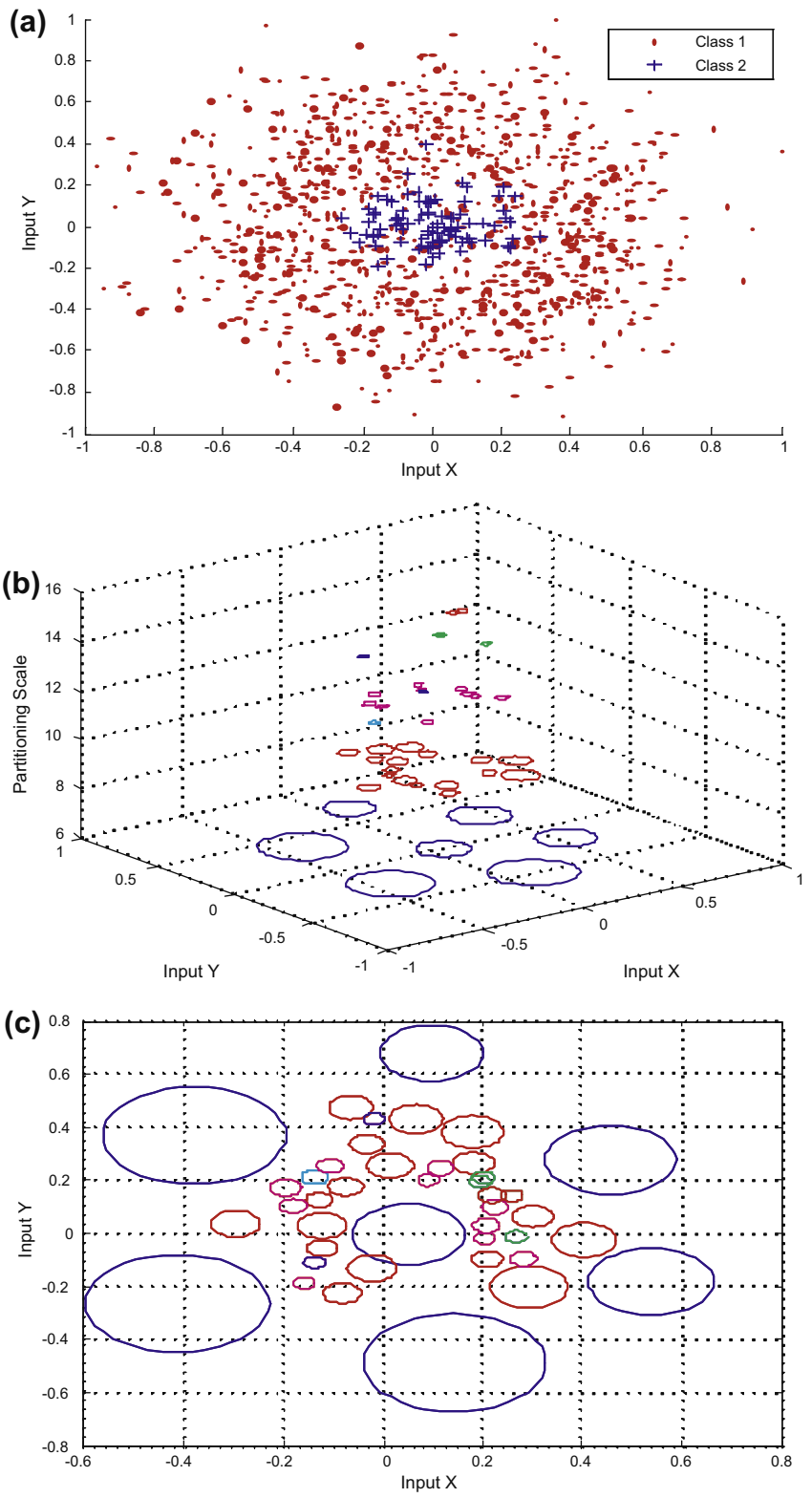

Fig. 3. Illustration of particlizing. (a) Original 2-D Gaussian data; (b) particlizing procedure; (c) obtained data particles.

assumption implemented. As can be seen, the neighborhood, compared to Fig. 3(c), is slightly expanded at the region with class mixture.

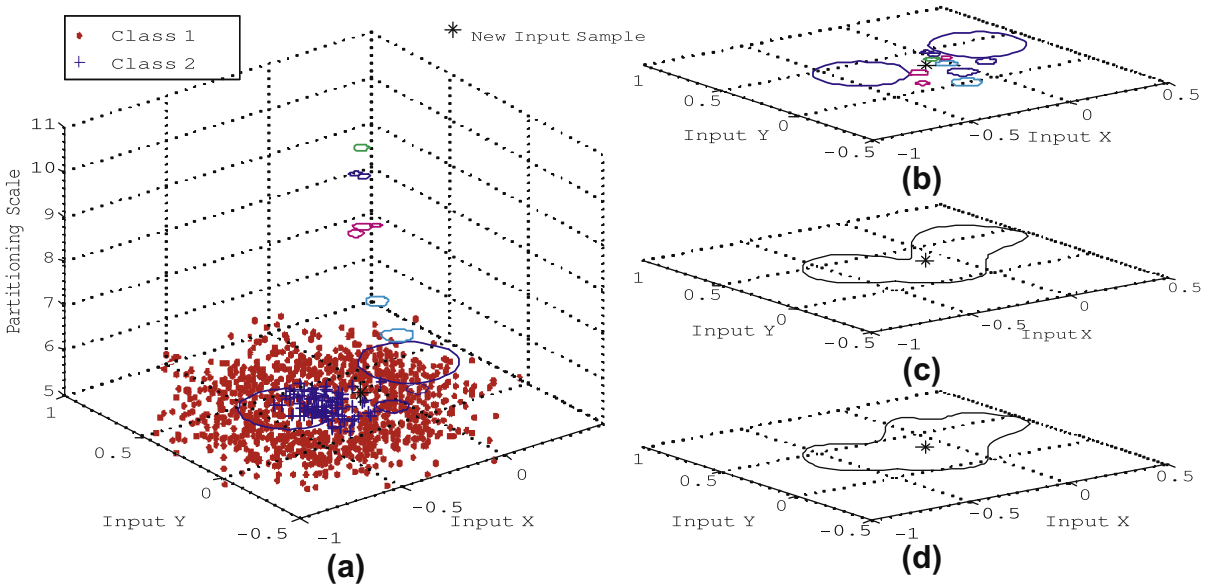

Fig. 4. Illustration of construction of personalised neighborhood in t-SVMT. (a) iterative neighbor knowledge extraction; (b) SVM particles; (c) personalized neighborhood; and (d) personalized neighborhood with smoothness assumption applied.

While exploring the personalized behavior of the t-SVMT, Fig. 4 gives an example of the transductive particle aggregation described in Algorithm 2. For a new instance identified as a '$*$' in Fig. 4, a set of transductive particles are aggregated, constructing an SVM tree as a personalized model for it. Owing to the particle-based neighborhood calculation, the obtained transductive neighborhood is shaped into a personalized area as plotted at the right-bottom of Fig. 4. This is substantially different from the neighborhood from *k*NN and NFI method using a constant circle-shape neighborhood for each test instance. A straightforward advantage of such a personalized neighborhood calculation is that noise instances, if they exist, will be removed in advance by the partitioning algorithm, even if they are physically adjacent to the test instance. This is because that while constructing the SVM tree, every step of data partitioning is under the supervision of an SVM, with the noise instances filtered.

As the result of Algorithm 2, a personalized t-SVMT model is generated for the test instance. Fig. 5 gives four t-SVMT structures for four difference test instances, in which $P1$ represents the root node of the tree, $P2$ is the second-level node, and so on. An SVM node is denoted as an ellipse with an 'SVM' label, and a terminal node is represented as a circle labelled with the class label (class 1 or 2). The structure of SVMTs is versatile because each SVMT serves personally different input instance.

5.3. Discriminability tests on synthetic data

For the data in Fig. 3(a), the t-SVMT is compared on its classification accuracy with three previous transductive methods: *k*NN ($k = 1$), NFI, and TSVM, as well as three inductive SVM methods: inductive SVM (ISVM), SVM Ensemble [21], and inductive SVMT. Additionally, t-SVMT is tested under the condition with (denoted as SA) and without (denoted as N/SA) smoothness assumption implemented for personalized neighborhood construction. Table 1 gives the results from a normal 10-fold cross-validation, where the classification accuracy of class 1 and class 2, and the overall classification accuracy are addressed respectively.

As seen in Table 1, transductive methods generally perform slightly better than inductive methods for the classification of class 2 (skewed class). Among four transductive methods, NFI gives the best class 2 accuracy, at 95.7%, at the cost of decreasing class 1 accuracy to a low of 88.1%; TSVM gives the best general accuracy, at 95.05%, but with quite low class 2 accuracy. The t-SVMT is judged best overall because it achieves the best tradeoff between two-class classification, improving class 2 accuracy to 75.4% while keeping a 95.5% class 1 accuracy. The t-SVMT with SA gives an even better classification of the skewed class 2 in terms of either accuracy or standard division. This indicates that implementation of the smoothness assumption enhances the generalization capability of the t-SVMT.

To further test the capability of the transductive learner for unlabelled data learning (i.e. transductive capability), a second 10-fold cross-validation experiment is conducted reversely using one-tenth of the entire data for training and the remaining data for testing. As seen in the results in Table 2, training with less data causes transductive learners to perform classification generally worse than in the case of Table 1. However, the superiority of the t-SVMT on transductive capability is indicated even more clearly, as the three highest accuracies are all found from the proposed t-SVMT (SA)/t-SVMT (N/SA).

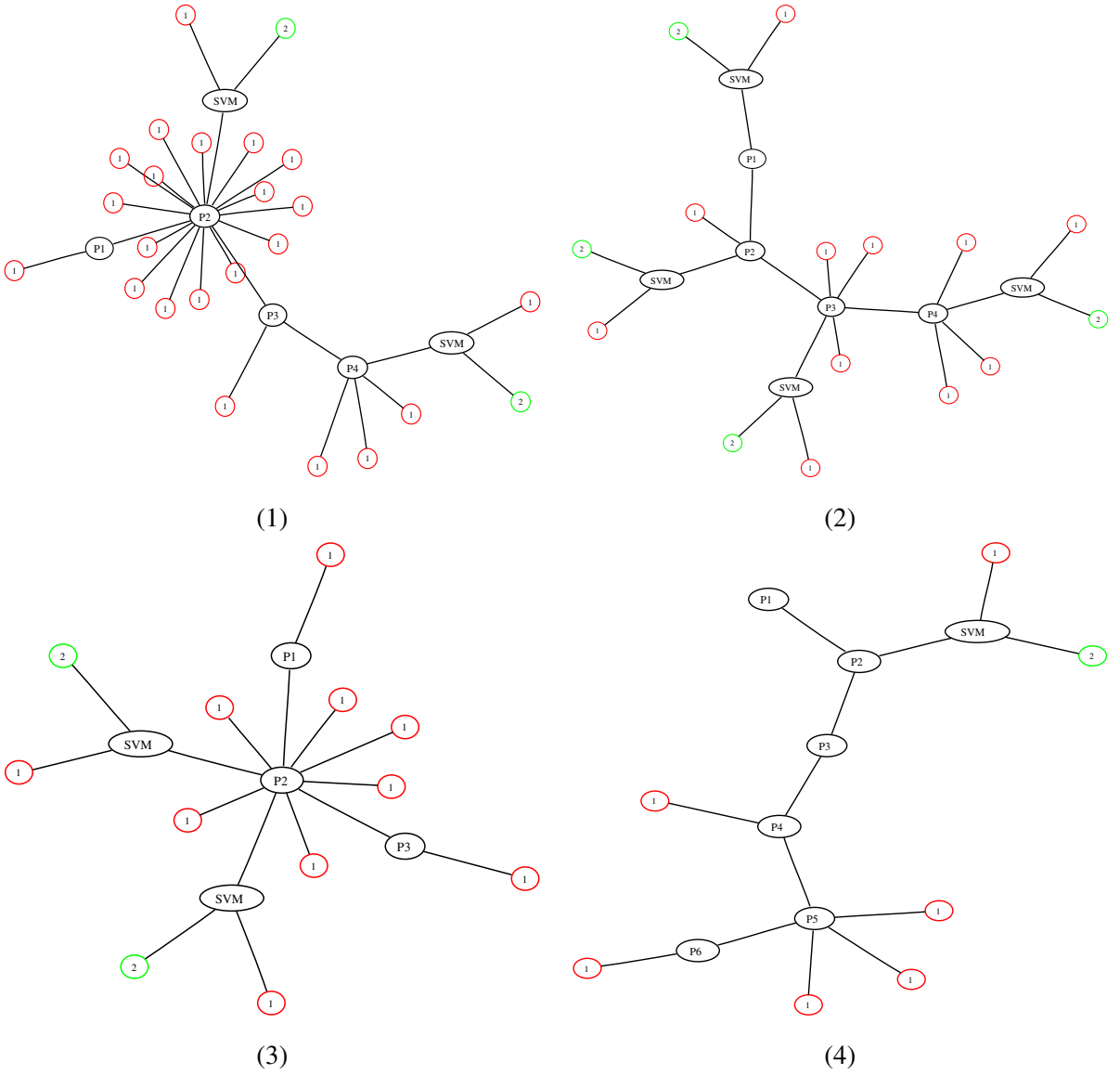

Fig. 5. Four examples of t-SVMT structure.

Table 1
Results of classification for synthetic dataset in Fig. 3, evaluated in standard 10-fold cross-validation.

	Methods	Class 1 Acc.	Class 2 Acc.	General Acc.
Inductive method	ISVM	**100.0** ± 0	0 ± 0	91.3 ± 0
	SVM ensemble	97.4 ± 6.1	56.8 ± 4.0	93.9 ± 5.1
	DDS_SVMT	91.8 ± 1.6	65.3 ± 1.3	89.5 ± 1.5
Transductive method	*K*NN (*K* = 1)	97.6 ± 5.6	61.1 ± 2.7	94.4 ± 3.6
	NFI	88.1 ± 2.4	**95.7** ± 2.9	88.8 ± 2.3
	TSVM	97.6 ± 2.8	69.5 ± 3.8	**95.1** ± 3.4
	t-SVMT(N/SA)	95.5 ± 2.1	75.4 ± 3.1	93.67 ± 2.5
	t-SVMT(SA)	95.5 ± 1.9	78.0 ± 1.1	93.97 ± 1.8

Numbers in boldface indicate the best results.

5.4. Robustness tests on face membership authentication

To evaluate the algorithm's robustness to class-imbalance, we study the face membership authentication (FMA) problem in [22,20]. The membership authentication problem is distinguishing the membership class from the non-membership class

Table 2
Results of classification for synthetic dataset in Fig. 3, evaluated in reverse 10-fold cross-validation, where one-tenth of the whole dataset are for training and the remaining data for testing.

	Methods	Class 1 Acc.	Class 2 Acc.	General Acc.
Transductive method	*K*NN ($K = 1$)	93.64 ± 0.06	33.98 ± 0.19	93.27 ± 0.01
	NFI	85.74 ± 0.02	92.96 ± 0.18	87.26 ± 0.01
	TSVM	94.04 ± 0.03	62.89 ± 0.26	91.29 ± 2.42
	t-SVMT(N/SA)	**94.23** ± 0.02	57.89 ±0.01	91.08 ± 1.63
	t-SVMT(SA)	93.37 ± 0.03	**71.46** ± 0.01	**91.46** ± 1.87

Numbers in boldface indicate the best results.

based on the personal facial images stored in a database. An FMA problem is a typical class-imbalance problem since the membership group is generally much smaller than the nonmembership group. Moreover, it shows a certain kind of sparse property: though the facial images of individuals differ widely, they may belong to the same class (membership or nonmembership); hence some samples in the feature space may have very few neighboring within-class instances.

In this experiment, the same dataset in [22] includes 1355 facial images collected from 270 people (five facial images per person, each image sized at 56×60). Facial features are extracted by applying principle component analysis (PCA), and selecting 100 top energy eigenfeatures. Accordingly, a 5-fold cross-validation is set for performance evaluation (i.e. for each cross-validation, 4 facial images per person are used for training and the other left 1 image for testing). Thus, for the membership authentication of each person, the t-SVMT is constructed over a 1080×100 dataset.

In practice, the running time of the t-SVMT varies over the size of constructed neighborhoods. Table 3 collects the average computational time costs of the t-SVMT and relevant SVM methods on the above FMA dataset in 5-fold cross-validation. As can be seen, t-SVMT runs over 40% faster than SVM*poly* and DDS_SVMT, though it does not perform as fast as regular SVM*linear* or SVM*RBF*. t-SVMT (SA) performs slightly slower than t-SVMT (N/SA) because of the additional SA operations and the increment in the neighborhood size due to SA.

To see how the class-imbalance influences the imbalance of the class accuracy, we gradually decrease the size of the membership class from 135 to 10. Fig. 6 shows the comparison of five methods on the class-distribution imbalance versus

Table 3
Algorithm running time for FMA in 5-fold cross-validation.

Methods	CPU time (s)
SVM*poly*	202.6301
SVM*linear*	21.0988
SVM*RBF*	51.9026
DDS_SVMT	183.4472
t-SVMT(SA)	114.5811
t-SVMT(N/SA)	113.6966

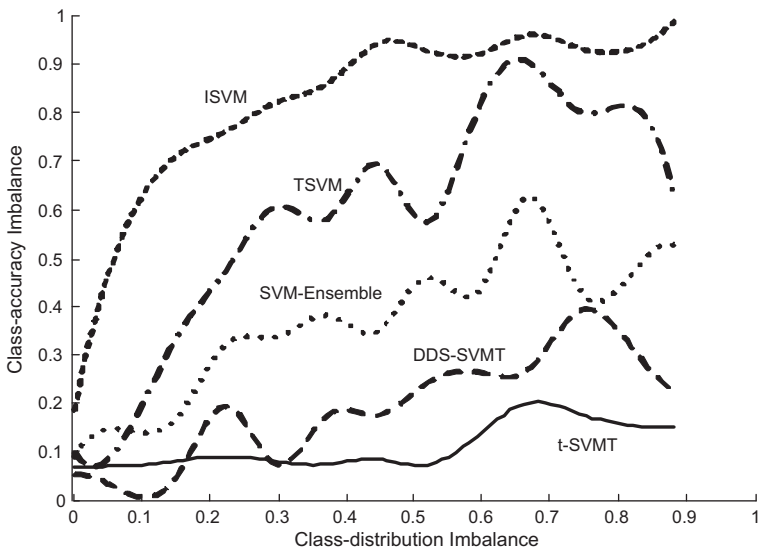

Fig. 6. Class-accuracy imbalance versus class-distribution imbalance. t-SVMT is compared with inductive SVM (ISVM), transductive SVM (TSVM), and SVM Ensemble, and SVMT of data-driven type on face membership authentication.

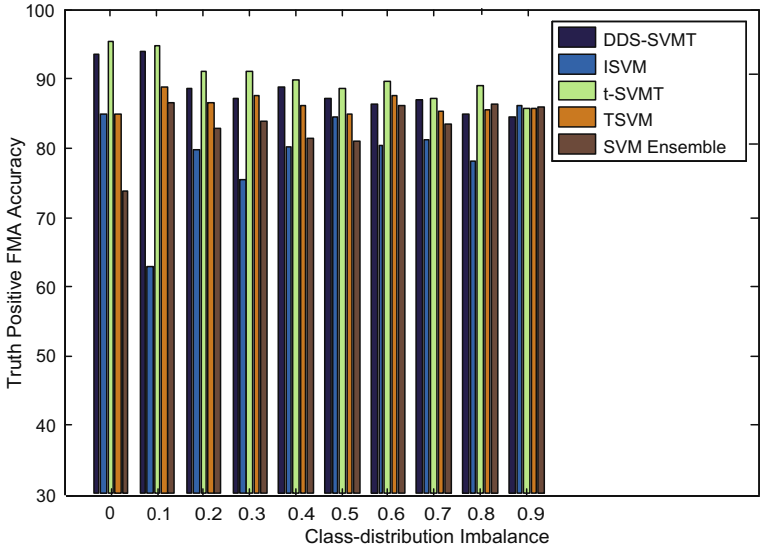

Fig. 7. Comparison of truth positive FMA accuracy under different conditions of class imbalance.

class-accuracy imbalance, where the imbalance value is defined as $1-\frac{\min(class1,class2)}{\max(class1,classs2)}$. As can be seen, as the imbalance of class-distribution increases from 0 $\left(1-\frac{135}{270-135}\right)$ to 0.9615 $\left(1-\frac{10}{270-10}\right)$, the class-accuracy imbalance of inductive SVM shows an immediate, dramatic increase. TSVM does slightly suppress the class-accuracy imbalance, but for most cases the imbalance is above 0.5. SVM ensemble makes the class-accuracy imbalance below 0.5 in most cases, but large fluctuations exist at the later part of the curve. The promising property of DDS-SVMT is that it shows good robustness to class-distribution imbalance, even for class-distribution imbalance greater than 0.6. However the active fluctuation shows potential instability. The proposed t-SVMT shows the best robustness to class-distribution imbalance such that it keeps a smooth curve with the class-accuracy imbalance kept below 0.2, even for the most imbalanced case when the class-distribution imbalance is at the maximum of 0.962. On the other hand, in comparing the performance of face membership authentication, Fig. 7 gives an comparison of the truth positive FMA accuracies under different conditions of class imbalance, where the proposed t-SVMT is seen outperforming single SVMs and SVM ensembles and very competitive with respect to the DDS-SVMT in terms of truth positive FMA accuracy. This indicates that the proposed t-SVMT has an outstanding discrimination, but an even better class-balancing capability than the previous DDS-SVMT.

5.5. Over-fitting tests for cancer diagnosis

To see if t-SVMT can prevent over-fitting for the training dataset, we compared the proposed t-SVMT with DDS_SVMT on the scale of the tree structures – the number of nodes in the tree and the classification performance – the classification accuracy, with 10-fold cross-validation. The experiments were conducted on eight well-known two-class cancer datasets, as listed in Table 4, in which the ovarian cancer dataset is based on proteomics data, others are gene expression data from micro-arrays, and most datasets are biased in terms of the class imbalance ratio (i.e. the ratio of the patients in the two classes). For performance evaluation, leave-one-out (LOO) cross-validation is used for those datasets without separated testing set, otherwise independent validation is performed.

We check the performance of the algorithms under two conditions. For the first case, the raw gene features are used as inputs, and for the second case, 100 genes selected by a standard *t*-test gene selection algorithm are used as inputs. Table 4 lists the details and experimental results of the eight datasets, where the size of SVMT is counted as the average node number over all testing instances. As seen from the table, over-fitting occurs for DDS_SVMT as it yields large tree structures that have about 30 nodes for datasets with less than 80 instances, such as CNS Tumour, Breast Cancer, and Lymphoma (2). In these cases, the classification accuracies are lower than 50%.

On the contrary, the t-SVMT is normally two to three times smaller than DDS_SVMT, but outperforms DDS_SVMT on classification with the same level stability (standard division) for 12 out of the full 16 case studies. By calculating a paired difference *t*-test based on 10-fold cross-validation [10] for each case study, we confirmed that our comparison experiment is statistically significant, and that the proposed t-SVMT is capable of preventing over-fitting by reducing the size of SVMT, while maintaining superior classification accuracy.

6. Discussion and conclusions

In this paper, we introduced a new type of SVM classification tree that performs effective personalized transductive learning for new test instances, implementing a new type of SVM aggregating intelligence for transductive learning.

Table 4
Average classification accuracies for eight cancer datasets, based on 10-fold cross-validation, values are means and standard divisins of 10 runs. Numbers in boldface indicate the best results.

Cancer dataset	Genes/with selection	Labelled data (Cls1/Cls2) Total	Class Imbalance ratio	Unlabelled data	DDS_SVMT %/Tree size	t-SVMT %/Tree size
Lymphoma (1) [26]	7129/100	(19/58)77	19/58 = 0.33	–	**84.4** ± 4.1/15 80.5 ± 5.4/12	77.9 ± 3.8/6 **84.4** ± 3.6/8
Leukemia* [11]	7219/100	(11/27)38	11/27 = 0.41	34	64.7 ± 0/24 **91.2** ± 0/12	**78.3** ± 0/12 **91.2** ± 0/7
CNS tumour [24]	7129/100	(21/39)60	21/39 = 0.53	–	50.0 ± 2.1/34 63.0 ± 2.2/26	**72.0** ± 2.6/8 **78.3** ± 1.2/8
Colon cancer [2]	2000/100	(22/40)62	22/40 = 0.55	–	71.3 ± 2.7/21 75.8 ± 1.4/31	**80.7** ± 2.5/10 **86.5** ± 2.2/9
Ovarian [23]	15,154/100	(91/162)253	91/162 = 0.56	–	**97.3** ± 2.4/13 96.4 ± 4.2/12	75.9 ± 3.0/4 **98.4** ± 4.5/6
Breast * cancer [28]	24,482/100	(34/44)78	34/44 = 0.77	19	52.6 ± 0/38 68.4 ± 0/14	**73.7** ± 0/4 **78.9** ± 0/6
Lymphoma (2) [26]	6431/100	(26/32)58	26/32 = 0.81	–	51.7 ± 1.8/27 58.6 ± 2.4/26	**60.3** ± 1.5/10 **66.7** ± 3.1/15
Lung * Cancer [12]	12,533/100	(16/16)32	16/16 = 1.0	149	64.4 ± 0/15 **77.8** ± 0/12	**75.0** ± 0/8 73.8 ± 0/7

* Independent validation dataset was used for the accuracy evaluation.

The proposed t-SVMT, from the viewpoint of personalized transductive learning, is different from previous PTL methods, such as *k*NN and NFI, in that it uses an SVM particle based knowledgeable neighborhood instead of the simple neighborhood defined by a distance metric. On the other hand, from the viewpoint of SVM aggregation [21], the t-SVMT presents a type of transductive aggregating intelligence following the same divide-and-conquer approach as previous SVMTs. However, the t-SVMT is also different from all previous SVMT aggregating methods in that it does not consider all the knowledge of the overall dataset, but rather only the regional knowledge (i.e. knowledge related to the test instance); therefore the size of the t-SVMT is generally much smaller than other SVMTs.

The advantage of the proposed t-SVMT are summarized as follows: (1) t-SVMT is superior to regular SVM, SVM ensemble, and DDS_SVMT especially in classifying class imbalanced datasets; (2) t-SVMT solves the over-fitting problem of previous SVMTs and constructs an SVMT two to three times smaller than the DDS_SVMT; and thus often outperforms DDS_SVMT in many classification tasks; and (3) t-SVMT uses SVM particle-based knowledgeable neighborhood and presents better personalized intelligence than ordinary transductive methods. This is because SVM particle is a flexibly-sized multi-instance package with noises filtered.

The proposed t-SVMT is only sustainable for two-class problems. In the principle of SVM aggregation, a multi-class SVMT can be constructed by decomposing an m-class task into a series of two-class subtasks; therefore the t-SVMT can be modified to suit multi-class cases by simply extending one-class SVMs to $f_{svm^{(i)}}, 1 \leqslant i \leqslant m$, and the binary SVM f_{svm} of Eq. (9) to a multi-class SVM.

References

[1] Aha et al, A review and empirical comparison of feature weighting methods for a class of lazy learning algorithms, Wettschereck (1997).
[2] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, A.J. Levine, Broad patterns of gene expression revealed by clustering analysis of tumour and normal colon tissues probed by oligonucleotide arrays, Proc. Natl. Acad. Sci. USA 96 (12) (1999) 6745–6750.
[3] G. Bontempi, M. Birattari, Bersini, Lazy learning for local modelling and control design, Int. J. Control 72 (7–8) (1999) 643–658.
[4] L. Bruzzone, M. Chi, M. Marconcini, A novel transductive SVM for semisupervised classification of remote-sensing images, IEEE Trans. Geosci. Remote Sens. 44 (11) (2006) 3363–3373.
[5] M. Ceci, Hierarchical text categorization in a transductive setting, in: ICDM Workshops 2008, pp. 184–191.
[6] O. Chapelle, B. Scholkopf, A. Zien, Semi-Supervised Learning, MIT Press, Cambridge:MA, 2006.
[7] H. Chen, L. Li, J. Peng, Error bounds of multi-graph regularized semi-supervised classification, Inform. Sci. 179 (12) (2009) 1960–1969.
[8] Y. Chen, G. Wang, S. Dong, Learning with progressive transductive support vector machine, Pattern Recogn. Lett. 24 (12) (2003) 845–1855.
[9] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification technique, IEEE Trans. Inform. Theory 13 (1) (1967) 21–27.
[10] T.G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Computation 10 (1998) 1895–1923.
[11] T.R. Golub, Toward a functional taxonomy of cancer, Cancer Cell. 6 (2) (2004) 107–108.
[12] G.J. Gordon, R.V. Jensen, et al, Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma, Cancer Res. 62 (2002) 4963–4967.
[13] Donghai Guan, Weiwei Yuan, Young-Koo Lee, Sungyoung Lee, Nearest neighbor editing aided by unlabeled data, Inform. Sci. 179 (13) (2009) 2273–2282.
[14] T. Joachims, Transductive inference for text classification using support vector machines, in: Proceedings of the Sixteenth International Conference on Machine Learning, 1999, pp. 200–209.
[15] T. Joachims, Making large-scale SVM learning practical, in: B. Schkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 169–184.

[16] N. Kasabov, Evolving connectionist systems, in: Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines, Springer, London, 2002.
[17] J. Kennedy, Russell C. Eberhart, Y. Shi, Swarm Intelligence, Morgan Kaufman, 2001.
[18] Available from: <http://sourceforge.net/projects/simplesvm/>.
[19] S. Pang et al, Spanning SVM tree for personalized transductive learning, in: Proceedings of ICANN 2009, LNCS5768/2009, Springer-Verlag, 2009. pp. 913–922.
[20] S. Pang, D. Kim, S.Y. Bang, Membership authentication in the dynamic group by face classification using SVM ensemble, Pattern Recogn. Lett. 24 (2003) 215–225.
[21] Shaoning Pang, SVM Aggregation: SVM, SVM Ensemble, SVM Classification Tree, IEEE SMC eNewsletter, December 2005. Available from: <http://www.ieeesmc.org/Newsletter/Dec2005/R11Pang.php>.
[22] Shaoning Pang, D. Kim, S.Y. Bang, Face membership authentication using SVM classification tree generated by membership-based LLE data partition, IEEE Trans. Neural Network 16 (2) (2005) 436–446.
[23] E.F. Petricoin, A.M. Ardekani, et al, Use of proteomic patterns in serum to identify ovarian cancer, The Lancet 359 (2002) 572–577.
[24] S.L. Pomeroy, P. Tamayo, et al, Prediction of central nervous system embryonal tumour outcome based on gene expression, letters to nature, Nature 415 (2002) 436–442.
[25] J.C. Schölkopf, J.C. Platt, J.Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the support of a high-dimensional distribution, Technical report, Microsoft Research, MSR-TR-99-87, 1999.
[26] M.A. Shipp, K.N. Ross, et al, Supplementary information for diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning, Nature Med. 8 (1) (2002) 68–74.
[27] Q. Song, N. Kasabov, NFI: a neuro-fuzzy inference method for transductive resoning, IEEE Trans. Fuzzy Syst. 13 (6) (2005) 799–808.
[28] L.J. Van 't Veer et al, Gene expression profiling predicts clinical outcome of breast cancer, Nature 415 (2002) 530–536.
[29] V.N. Vapnik, The Nature of Statistical Learning Theory, second ed., Springer-Verlag, Berlin, Germany, 1999.
[30] V.N. Vapnik, Estimation of dependences based on empirical data, in: Springer Verlag Series in Statistics, Springer Verlag, 1982.
[31] D. Westa, S. Dellanab, Diversity of ability and cognitive style for group decision processes, Inform. Sci. 179 (5) (2009) 542–558.
[32] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, ACL (1995).