One-Class SVM in Multi-Task Learning

Xiyan He, Gilles Mourot, Didier Maquin & José Ragot
Centre de Recherche en Automatique de Nancy, CNRS UMR 7039 - Université Henri Poincaré, Nancy-I
2, avenue de la forêt de Haye, 54500 Vandœuvre-lès-Nancy, France

Pierre Beauseroy & André Smolarz
Institut Charles Delaunay, STMR CNRS UMR 6279 - Université de Technologie de Troyes
12, Rue Marie Curie, BP 2060, F-10010 Troyes, France

ABSTRACT: Multi-Task Learning (MTL) has become an active research topic in recent years. While most machine learning methods focus on the learning of tasks independently, multi-task learning aims to improve the generalization performance by training multiple related tasks simultaneously. This paper presents a new approach to multi-task learning based on one-class Support Vector Machine (one-class SVM). In the proposed approach, we first make the assumption that the model parameter values of different tasks are close to a certain mean value. Then, a number of one-class SVMs, one for each task, are learned simultaneously. Our multi-task approach is easy to implement since it only requires a simple modification of the optimization problem in the single one-class SVM. Experimental results demonstrate the effectiveness of the proposed approach.

1 INTRODUCTION

Classical machine learning technologies have achieved much success in the learning of a single task at a time. However, in many practical applications we may need to learn a number of related tasks or to rebuild the model from new data, for example, in the problem of fault detection and diagnosis of a system that contains a set of equipments *a priori* identical but working under different conditions. Here "an equipment" may be a simple machine (a pump, a motor, ...), a system (a car, an airplane, ...), or even an industrial plant (nuclear power plant, ...). This may be the case for a car hire system where we have a fleet of vehicles to serve a set of customers. In industry, it is common to encounter a number of *a priori* identical plants, such as in the building or maintenance of a fleet of nuclear power plants or of a fleet of their components. In such cases, the learning of the behavior of each equipment can be considered as a single task, and it would be nice to transfer or leverage the useful information between related tasks (Pan and Yang 2010). Therefore, Multi-Task Learning (MTL) has become an active research topic in recent years (Bi et al. 2008, Zheng et al. 2008, Gu and Zhou 2009, Birlutiu et al. 2010).

While most machine learning methods focus on the learning of tasks independently, multi-task learning aims to improve the generalization performance by training multiple related tasks simultaneously. The main idea is to share what is learned from different tasks (*e.g.*, a common representation space or some model parameters that are close to each other), while tasks are trained in parallel (Caruana 1997). Previous works have shown empirically as well as theoretically that the multi-task learning framework can lead to more intelligent learning models with a better performance (Caruana 1997, Heskes 2000, Ben-David and Schuller 2003, Evgeniou et al. 2005, Ben-David and Borbely 2008).

In recent years, Support Vector Machines (SVM) (Boser et al. 1992, Vapnik 1995) have been successfully used for multi-task learning (Evgeniou and Pontil 2004, Jebara 2004, Evgeniou et al. 2005, Widmer et al. 2010, Yang et al. 2010). The SVM method was initially developped for the classification of data from two different classes by a hyperplane that has the largest distance to the nearest training data points of any class (maximum margin). When the datasets are not linearly separable, the "kernel trick" is used. The basic idea is to map the original data to a higher dimensional feature space and then solve a linear problem in that space. The good properties of kernel functions make support vector machines well-suited for multi-task learning.

In this paper, we present a new approach to multi-task learning based on one-class Support Vector Machines (one-class SVM). The one-class SVM pro-

posed by Schölkopf et al. (2001) is a typical method for the problem of novelty or outlier detection, also known as the one-class classification problem due to the fact that we do not have sufficient knowledge about the outlier class. For example, in the application of fault detection and diagnosis, it is very difficult to collect samples corresponding to all the abnormal behaviors of the system. The main advantage of one-class SVM over other one-class classification methods (Tarassenko et al. 1995, Ritter and Gallegos 1997, Eskin 2000, Singh and Markou 2004) is that it focuses only on the estimation of a bounded area for samples from the target class rather than on the estimation of the probability density. The bounded area estimation is achieved by separating the target samples (in a higher-dimensional feature space for non-linearly separable cases) from the origin by a maximum-margin hyperplane which is as far away from the origin as possible.

Recently, Yang et al. (2010) proposed to take the advantages of multi-task learning when conducting one-class classification. The basic idea is to constrain the solutions of related tasks close to each other. However, they solve the problem via conic programming (Kemp et al. 2008), which is complicated. In this paper, inspired by the work of Evgeniou and Pontil (2004), we introduce a very simple multi-task learning framework based on the one-class SVM method, a widely used tool for single task learning. In the proposed method, we first make the same assumption as in (Evgeniou and Pontil 2004), that is, the model parameter values of different tasks are close to a certain mean value. This assumption is reasonable due to the observation that when the tasks are similar to each other, usually their model parameters are close enough. Then, a number of one-class SVMs, one for each task, are learned simultaneously. Our multi-task approach is easy to implement since it only requires a simple modification of the optimization problem in the single one-class SVM. Experimental results demonstrate the effectiveness of the proposed approach.

This paper is organized as follows. In Section 2, a brief description of the formulation of the one-class SVM algorithm and the properties of kernel functions is first discussed. The proposed multi-task learning method based on one-class SVM is then outlined in Section 3. Section 4 presents the experimental results. In Section 5, we conclude this paper with some final remarks and future work propositions.

2 ONE-CLASS SVM AND PROPERTIES OF KERNELS

2.1 *One-class SVM*

The one-class SVM proposed by Schölkopf et al. (2001) is a promising method for the problem of one-class classification, which aims at detecting samples that do not resemble the majority of the dataset. It employes two ideas of the original support vector machine algorithm to ensure a good generalisation: the maximisation of the margin and the mapping of the data to a higer dimensional feature space induced by a kernel function. The main difference between the one-class SVM and the original SVM is that in one-class SVM the only given information is the normal samples (also called positive samples) of the same single class whereas in the original SVM information on both normal samples and outlier samples (also called negative samples) is given. In essence, the one-class SVM estimates the boundary region that comprises most of the training samples. If a new test sample falls within this boundary it is classified as of normal class, otherwise it is recognised as an outlier.

Suppose that $\mathcal{A}_m = \{\boldsymbol{x}_i\}, i = 1, \ldots, m$ is a set of m training samples of a single class. $\boldsymbol{x}_i$ is a sample in the space $\mathcal{X} \subseteq \mathbb{R}^d$ of dimension d. Also suppose that ϕ is a non-linear transformation. The one-class SVM is predicated on the assumption that the origin in the transformed feature space belongs to the negative or outlier class. The training stage consists in first projecting the training samples to a higher dimensional feature space and then separating most of the samples from the origin by a maximum-margin hyperplane which is as far away from the origin as possible. In order to determine the maximum-margin hyperplane, we need to deduce its normal vector $\mathbf{w}$ and a threshold ρ by solving the following optimization problem:

$$\begin{cases} \min_{\mathbf{w},\boldsymbol{\xi},\rho} & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i - \rho \\ \text{subject to:} & \langle \mathbf{w}, \phi(\boldsymbol{x}_i)\rangle \geq \rho - \xi_i, \quad \xi_i \geq 0 \end{cases} \tag{1}$$

ξ_i are called slack variables, and they are introduced to relax the constraints in some cases for certain training sample sets. Indeed, the optimization algorithm aims at finding the best trade-off between the maximization of the margin and the minimization of the average of the slack variables. The parameter $\nu \in (0, 1]$ is a special parameter for one-class SVM. It is the upper-bound of the ratio of outliers among all the training samples as well as the lower-bound of the ratio of support vectors among all the samples.

Due to the high dimensionality of the normal vector $\mathbf{w}$, the primal problem is solved by its Lagrange dual problem :

$$\begin{cases} \min_{\boldsymbol{\alpha}} & \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j\langle\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)\rangle \\ \text{subject to:} & 0 \leq \alpha_i \leq \frac{1}{\nu m}, \quad \sum_{i=1}^{m}\alpha_i = 1 \end{cases} \tag{2}$$

where α_i are the Lagrange multipliers. It is worth noting that all the mappings ϕ occur in the form of inner products. We need not to calculate the non-linear mapping explicitely by defining a simple kernel function that fulfills Mercer's conditions (Vapnik 1995):

$$\langle\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)\rangle = k(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{3}$$

As an example, the Gaussian kernel $k_\sigma(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}}$ is a largely used kernel among the community. By solving the dual problem with this *kernel trick*, the final decision is given by:

$$f(\boldsymbol{x}) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) - \rho\right) \tag{4}$$

2.2 *Properties of kernels*

In order to exploit the *kernel trick*, we need to construct valid kernel functions. A necessary and sufficient condition for a function to be a valid kernel is defined as follows (Schölkopf and Smola 2001):

Definition 2.1 *Let $\mathcal{X}$ be a nonempty set. A function k on $\mathcal{X} \times \mathcal{X}$ which for all $m \in \mathbb{N}$ and all $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \in \mathcal{X}$ gives rise to a positive definite Gram matrix $\boldsymbol{K}$, with elements:*

$$\boldsymbol{K}_{ij} := k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{5}$$

is called a positive definite kernel.

One popular way to construct new kernels is to build them based on simpler kernels. In this section, we briefly gather some results of the properties of the set of admissible kernels that are useful for designing new kernels. For a detailed description concerning the design of kernel functions, interested readers are referred to (Schölkopf and Smola 2001).

Proposition 2.1 *If k_1 and k_2 are kernels, and $\alpha_1, \alpha_2 \geq 0$, then $\alpha_1 k_1 + \alpha_2 k_2$ is a kernel.*

Proposition 2.2 *If k_1 and k_2 are kernels defined respectively on $\mathcal{X}_1 \times \mathcal{X}_1$ and $\mathcal{X}_2 \times \mathcal{X}_2$, then their tensor product*

$$k_1 \otimes k_2(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_1', \boldsymbol{x}_2') = k_1(\boldsymbol{x}_1, \boldsymbol{x}_1') k_2(\boldsymbol{x}_2, \boldsymbol{x}_2')$$

is a kernel on $(\mathcal{X}_1 \times \mathcal{X}_2) \times (\mathcal{X}_1 \times \mathcal{X}_2)$. Here $\boldsymbol{x}_1, \boldsymbol{x}_1' \in \mathcal{X}_1$ and $\boldsymbol{x}_2, \boldsymbol{x}_2' \in \mathcal{X}_2$.

With these properties, we can now construct more complex kernel functions that are appropriate to our specific applications in multi-task learning.

3 THE PROPOSED METHOD

In this section, we introduce the one-class SVM method for the purpose of multi-task learning. In the context of multi-task learning, we have T learning tasks on the same space $\mathcal{X}$, with $\mathcal{X} \subseteq \mathbb{R}^d$. For each task we have m samples $\{\boldsymbol{x}_{1t}, \boldsymbol{x}_{2t}, \ldots, \boldsymbol{x}_{mt}\}$. The objective is to learn a decision function (a hyperplane) $f_t(\boldsymbol{x}) = \text{sign}(\langle \mathbf{w}_t, \phi(\boldsymbol{x}) \rangle - \rho_t)$ for each task t. Inspired by the method proposed by Evgeniou and Pontil (2004), we make the assumption that when the tasks are related to each other, the normal vector $\mathbf{w}_t$ can be represented by the sum of a mean vector $\mathbf{w}_0$ and a specific vector $\mathbf{v}_t$ corresponding to each task:

$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t \tag{6}$$

3.1 *Primal problem*

Following the above assumption, we can generalize the one-class SVM method to the problem of multi-task learning. The primal optimization problem can be written as follows:

$$\min_{\mathbf{w}_0, \mathbf{v}_t, \xi_{it}, \rho_t} \frac{1}{2}\sum_{t=1}^{T} \|\mathbf{v}_t\|^2 + \frac{\mu}{2}\|\mathbf{w}_0\|^2 + \sum_{t=1}^{T}\left(\frac{1}{\nu_t m}\sum_{i=1}^{m}\xi_{it}\right) - \sum_{t=1}^{T}\rho_t \tag{7}$$

for all $i \in \{1, 2, \ldots, m\}$ et $t \in \{1, 2, \ldots, T\}$, subject to:

$$\langle (\mathbf{w}_0 + \mathbf{v}_t), \phi(\boldsymbol{x}_{it}) \rangle \geq \rho_t - \xi_{it} \tag{8}$$

$$\xi_{it} \geq 0 \tag{9}$$

where ξ_{it} are the slack variables associated to each sample and $\nu_t \in (0, 1]$ is the special parameter of one-class SVM for each task. In order to control the similarity between tasks, we introduce a positive regularization parameter μ into the primal optimisation problem. In particular, a big value of μ tends to enforce the system to learn the T tasks independently whereas a small value of μ will lead the system to learn a common model for all tasks. As in the earlier case of a single one-class SVM, the Lagrangien is formed as:

$$\begin{aligned} &L(\mathbf{w}_0, \mathbf{v}_t, \xi_{it}, \rho_t, \alpha_{it}, \beta_{it}) \\ &= \frac{1}{2}\sum_{t=1}^{T}\|\mathbf{v}_t\|^2 + \frac{\mu}{2}\|\mathbf{w}_0\|^2 + \sum_{t=1}^{T}\left(\frac{1}{\nu_t m}\sum_{i=1}^{m}\xi_{it}\right) - \sum_{t=1}^{T}\rho_t \\ &- \sum_{t=1}^{T}\sum_{i=1}^{m}\alpha_{it}[\langle (\mathbf{w}_0 + \mathbf{v}_t), \phi(\boldsymbol{x}_{it})\rangle - \rho_t + \xi_{it}] - \sum_{t=1}^{T}\sum_{i=1}^{m}\beta_{it}\xi_{it} \end{aligned} \tag{10}$$

where $\alpha_{it}, \beta_{it} \geq 0$ are the Lagrange multipliers. We set the partial derivatives of the Lagrangian to zero and obtain the following equations:

$$\begin{array}{ll} (a) & \mathbf{w}_0 = \frac{1}{\mu}\sum_{t=1}^{T}\sum_{i=1}^{m}\alpha_{it}\phi(\boldsymbol{x}_{it}) \\ (b) & \mathbf{v}_t = \sum_{i=1}^{m}\alpha_{it}\phi(\boldsymbol{x}_{it}) \\ (c) & \alpha_{it} = \frac{1}{\nu_t m} - \beta_{it} \\ (d) & \sum_{i=1}^{m}\alpha_{it} = 1 \end{array} \tag{11}$$

By combining the equations (6), (11)(a) and (11)(b), we have:

$$\mathbf{w}_0 = \frac{1}{\mu}\sum_{t=1}^{T}\mathbf{v}_t \tag{12}$$

$$\mathbf{w}_0 = \frac{1}{\mu + T}\sum_{t=1}^{T}\mathbf{w}_t \tag{13}$$

With these relationships, we may replace the vectors $\mathbf{v}_t$ and $\mathbf{w}_0$ by $\mathbf{w}_t$ in the primal optimisation function (7), which leads to an equivalent function:

$$\min_{\mathbf{w}_t,\xi_{it},\rho_t} \quad \frac{\lambda_1}{2}\sum_{t=1}^{T}\|\mathbf{w}_t\|^2 + \frac{\lambda_2}{2}\sum_{t=1}^{T}\left\|\mathbf{w}_t - \frac{1}{T}\sum_{r=1}^{T}\mathbf{w}_r\right\|^2 + \sum_{t=1}^{T}\left(\frac{1}{\nu_t m}\sum_{i=1}^{m}\xi_{it}\right) - \sum_{t=1}^{T}\rho_t \tag{14}$$

with

$$\lambda_1 = \frac{\mu}{\mu+T} \quad \text{and} \quad \lambda_2 = \frac{T}{\mu+T} \tag{15}$$

We can see that the objective of the primal optimisation problem (7) in the framework of multi-task learning is thus to find a trade-off between the maximisation of the margin for each one-class SVM model and the closeness of each one-class SVM model to the average model.

3.2 *Dual problem*

The primal optimisation problem (7) can be solved through its Lagrangian dual problem expressed by:

$$\max_{\alpha_{it}} -\frac{1}{2}\sum_{t=1}^{T}\sum_{r=1}^{T}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_{it}\alpha_{jr}\left(\frac{1}{\mu}+\delta_{rt}\right)\langle\phi(\boldsymbol{x}_{it}),\phi(\boldsymbol{x}_{jr})\rangle \tag{16}$$

contrainted to :

$$0 \leq \alpha_{it} \leq \frac{1}{\nu_t m}, \quad \sum_{i=1}^{m}\alpha_{it} = 1 \tag{17}$$

where δ_{rt} is the Kronecker delta kernel:

$$\delta_{rt} = \begin{cases} 1, & \text{if} \quad r = t \\ 0, & \text{if} \quad r \neq t \end{cases} \tag{18}$$

We can see that the main difference between this dual problem (16) and that in a single one-class SVM learning (2) is the introduced term $\left(\frac{1}{\mu}+\delta_{rt}\right)$ in the multi-task learning framework. Suppose that we define a kernel function as in equation (3):

$$k(\boldsymbol{x}_{it},\boldsymbol{x}_{jr}) = \langle\phi(\boldsymbol{x}_{it}),\phi(\boldsymbol{x}_{jr})\rangle \tag{19}$$

where r and t are the task index associated to each sample. Taking advantage of the kernel properties presented in section 2.2, we know that the product of two kernels $\delta_{rt}k(\boldsymbol{x}_{it},\boldsymbol{x}_{jr})$ is a valid kernel (Proposition 2.2). Further, the following function:

$$\begin{aligned} G_{rt}(\boldsymbol{x}_{it},\boldsymbol{x}_{jr}) &= (\frac{1}{\mu}+\delta_{rt})k(\boldsymbol{x}_{it},\boldsymbol{x}_{jr}) \\ &= \frac{1}{\mu}k(\boldsymbol{x}_{it},\boldsymbol{x}_{jr}) + \delta_{rt}k(\boldsymbol{x}_{it},\boldsymbol{x}_{jr}) \end{aligned} \tag{20}$$

is a linear combination of two valid kernels with positive coefficients ($\frac{1}{\mu}$ and 1), and therefore is also a valid kernel (Proposition 2.1). We can thus solve the multi-task learning optimisation problem (7) through a single one-class SVM problem by using the new kernel function $G_{rt}(\boldsymbol{x}_{it},\boldsymbol{x}_{jr})$. The decision function for each task is given by:

$$f_t(\boldsymbol{x}) = \text{sign}\left(\sum_{r=1}^{T}\sum_{i=1}^{m}\alpha_{ir}G_{rt}(\boldsymbol{x}_{ir},\boldsymbol{x}) - \rho_t\right) \tag{21}$$

4 EXPERIMENTAL RESULTS

This section presents the experimental results obtained in our analysis. In order to evaluate the effectiveness of the proposed multi-task learning framework, we compare our one-class SVM based multi-task learning method (denoted by MTL-OSVM) with two other methods: the traditional learning method that learns the T tasks independently each with a one-class SVM (denoted by T-OSVM) and the method that uses 1 one-class SVM for all tasks under the assumption that all the related tasks can be considered as one big task (denoted by 1-OSVM).

In our experiments, the kernel of the one-class SVM used for T-OSVM and 1-OSVM is a Gaussian kernel $k_\sigma(\boldsymbol{x}_{it},\boldsymbol{x}_{jr}) = e^{-\frac{\|x_{it}-x_{jr}\|^2}{2\sigma^2}}$. For the proposed multi-task learning method MTL-OSVM, the new kernel is thus constructed based on the Gaussian kernel as presented in equation (20). The optimum values for the two parameters ν and σ of the one-class SVM are determined through cross validation. For the sake of simplicity, we have used a common combination of their values (ν,σ) for all related tasks. In order to ensure the reliability of the performance evaluation, all the results have been averaged over 20 trials each with random draws of training set. As the approaches and comparison are all one-class classification methods, the statistics of both false positive and false negative error rates are reported.

4.1 *Experiment on nonlinear toy data*

We have firstly tested the poposed method on four ($T = 4$) related simple nonlinear classification tasks. The datasets are created according to the following steps. For the first task, each sample is composed of $d = 4$ variables of which the first three are uniformly distributed variables. The fourth variable is set by the relation:

$$x^{(4)} = x^{(1)} + 2x^{(2)} + (x^{(3)})^2$$

The datasets for the other three tasks are then created by adding Gaussian white noises with different amplitudes on the dataset of the first task. The noises are classified respectively as low noise (for Task 2, with an amplitude of about an order of 1% of the first

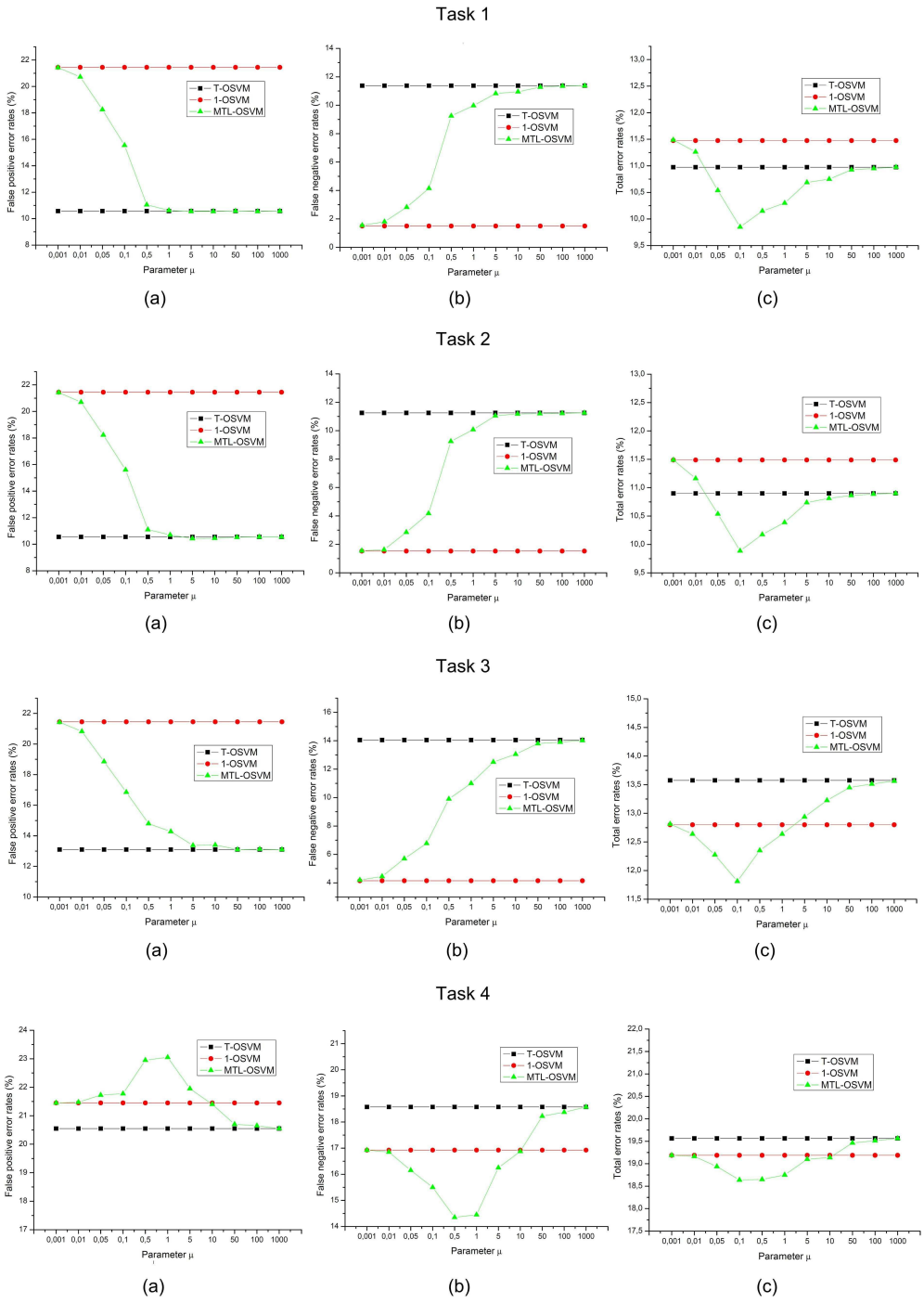

Figure 1: The variation of the average (a) false positive, (b) false negative and (c) total error rates for each task (nonlinear toy data) along with the value of the regularization parameter μ.

dataset amplitude value), medium noise (for Task 3, with an amplitude of about an order of 8% of the first dataset amplitude value) and high noise (for Task 4, with an amplitude of about an order of 15% of the first dataset amplitude value). In order to evaluate the false positive error rates, we have generated a set of negative samples that are composed of $d = 4$ uniformly distributed variables. Therefore, the training set of each task contains only positive samples ($m = 200$) whereas in the test procedure we use the test set of size 400 that contains both positive and negative samples (200 samples for each class). The obtained optimum parameter values of one-class SVM are $(\nu, \sigma) = (0.01, 0.5)$ for this experiment.

Figure 1 illustrates the variation of the average false positive, false negative and total error rates of our multi-task learning method MTL-OSVM for each task along with the value of the regularization parameter μ. The error rates of T-OSVM and 1-OSVM are also presented. We can see that for a very small value of μ, the performance of MTL-OSVM coincides with that of 1-OSVM as if all the tasks were considered as the same task. When the value of μ is very large, the performance of MTL-OSVM is in accordance with that of the traditional independant learning method T-OSVM. With the increase of the value of μ, the behaviors of the first three tasks are similar. The false positive error rate of the MTL-OSVM method tends to decrease whereas its false negative error rate increases. However, for the fourth task, the false positive (false

negative) error rate first increases (decreases) and then decreases (increases) after it reaches the maximum (minimum) value. This behavior may be due to the very high noise that we added to the original dataset. In all, with a good choice of μ, the multi-task framework achieves a better performance in terms of the total error rate when compared to the traditional learning methods.

4.2 *Experiment on textured image data*

We have tested the proposed method on several textured gray-scale images that contain artificial textures generated by using Markov chain models (Smolarz 1997). According to the nature of a texture, we first suppose that the useful information for texture caracterization is included in an isotropic neighbourhood of each pixel. In our experiments we use then the gray levels of a local $d = 5 \times 5$ squared window centered to each pixel as its feature vector. Similar to the previous experiment in Section 4.1, four related tasks are created. The dataset for Task 1 contains samples of size $d = 5 \times 5 = 25$ that are selected randomly from the original single texture source image. The samples for the other three tasks are selected from textured images of the same source as Task 1, but contaminated respectively by low noise (Task 2), medium noise (Task 3) and high noise (Task 4). Negative samples used in the test set are generated by using a different single texture source image. Figure 2 illustrates the single texture source images used for generating the datasets. In each trial, the training set of each task contains $m = 200$ positive samples and the test set is composed of 200 positive and 200 negative samples. The common parameter values of one-class SVM used in this experiment are $(\nu, \sigma) = (0.01, 300)$.

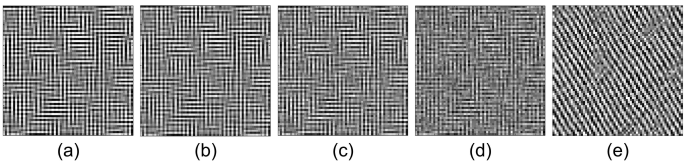

Figure 2: Single texture source images used for generating the datasets. (a) Original texture image for Task 1. (b) Texture image of (a) with low noise, for Task 2. (c) Texture image of (a) with medium noise, for Task 3. (d) Texture image of (a) with high noise, for Task 4. (e) Original texture image for generating negative samples.

Table 1 shows the statistics of the obtained error rates. The corresponding optimum value of μ for MTL-OSVM, which minimises the total error rate, is also presented. According to this table, we can see that the individual learning method T-OSVM has the lowest false positive but a higher false negative. On the contrary, the learning of a single one-class SVM for all tasks (1-OSVM) achieves the lowest false negative at the expense of a higher false positive. The proposed multi-task learning method MTL-OSVM can reach an overall better performance by finding a trade-off between the false positive error rate and the false negative error rate.

Table 1: Error rates (%) of the different methods for all tasks on texture data. FP: false positive error rates, FN: false negative error rates, Total: total error rates.

	Task 1			
	FP	FN	Total	μ
T-OSVM	**3.62** $\pm$ 1.18	27.0 $\pm$ 3.0	15.3 $\pm$ 1.4	–
1-OSVM	27.1 $\pm$ 2.6	**2.70** $\pm$ 1.59	14.9 $\pm$ 1.5	–
MTL-OSVM	14.4 $\pm$ 2.2	9.52 $\pm$ 3.19	**12.0** $\pm$ 1.8	0.1
	Task 2			
	FP	FN	Total	μ
T-OSVM	**4.27** $\pm$ 1.11	28.2 $\pm$ 3.6	16.3 $\pm$ 1.6	–
1-OSVM	27.1 $\pm$ 2.6	**3.27** $\pm$ 2.21	15.2 $\pm$ 1.8	–
MTL-OSVM	14.6 $\pm$ 2.2	10.3 $\pm$ 3.7	**12.4** $\pm$ 2.0	0.1
	Task 3			
	FP	FN	Total	μ
T-OSVM	**7.05** $\pm$ 1.41	28.6 $\pm$ 3.9	17.8 $\pm$ 2.0	–
1-OSVM	27.1 $\pm$ 2.6	**6.62** $\pm$ 3.73	16.8 $\pm$ 2.2	–
MTL-OSVM	15.7 $\pm$ 2.3	14.4 $\pm$ 4.1	**15.0** $\pm$ 2.2	0.1
	Task 4			
	FP	FN	Total	μ
T-OSVM	**27.4** $\pm$ 3.0	34.4 $\pm$ 8.7	30.9 $\pm$ 4.5	–
1-OSVM	**27.1** $\pm$ 2.6	34.0 $\pm$ 8.7	30.5 $\pm$ 4.5	–
MTL-OSVM	29.4 $\pm$ 2.6	**29.0** $\pm$ 9.2	**29.2** $\pm$ 4.5	0.5

The average results of this experiment are depicted in Figure 3. As in the previous experiment on the nonlinear toy data, we can observe the same behaviors of the error rates variations along with the value of the regularization parameter μ. The proposed method MTL-OSVM outperforms the other two methods (T-OSVM and 1-OSVM) for all the tasks. It is worth noting that the optimum value of μ is different for different task. The setting of this parameter is thus very important in order to ensure a good performance. In our experiment we use a validation set to find the optimum value of μ.

4.3 *Discussion*

It is worth noting that in this section only academic experiments on nonlinear toy data with low dimenstional feature space and textured image data with high dimentional feature space are presented. We address the problem of modeling the normal data with the help of one-class SVM method, which is usually considered as an essential step for fault dectection and diagnosis. In each experiment, related tasks are created in order to mimic the application of modeling the normal process behavior of a fleet of plants that are a priori identical but working under different conditions (thus having different noises on the measurements). Here we consider the modeling of the behavior of each plant as a single task and the classification of normal data and anomalies is then performed by the constructed model. The proposed methodologie shows that learning multiple related tasks simultaneously can be beneficial to improve the performance of each constructed model.

One straightforward work is to use the proposed

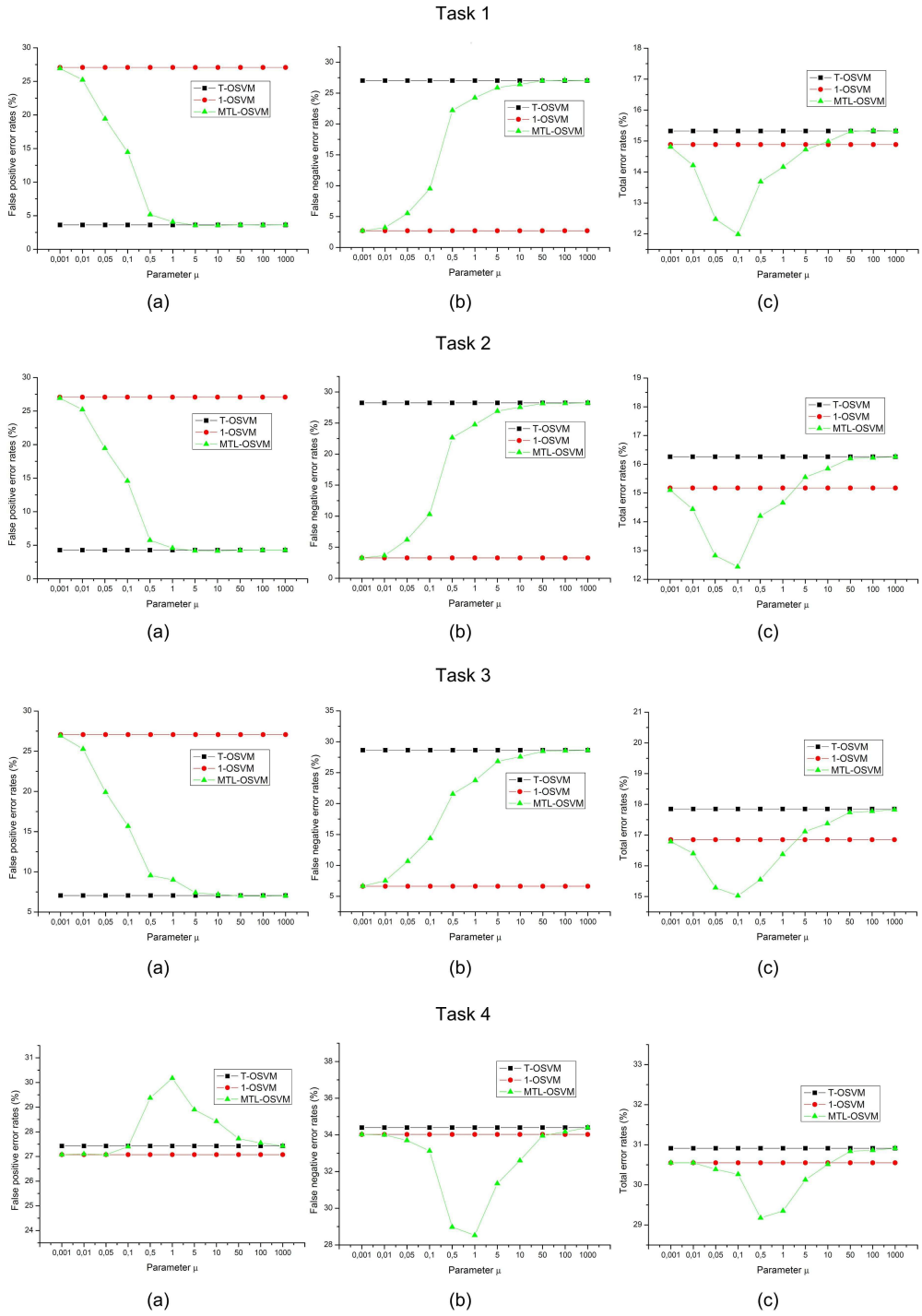

Figure 3: The variation of the average (a) false positive, (b) false negative and (c) total error rates for each task (texture data) along with the value of the regularization parameter μ.

multi-task learning methodologie for fault detection and diagnosis with real industrial data, such as the modeling of the behavior of a fleet of reactor coolant pumps in the nuclear cooling system.

5 CONCLUSION

In this paper we introduced the one-class SVM in the framework of multi-task learning under the assumption that the model parameter values of related tasks are close to a certain mean value. A regularization parameter was used in the optimisation process to control the trade-off between the maximisation of the margin for each one-class SVM model and the closeness of each one-class SVM model to the average model. The design of new kernels in the multi-task framework based on kernel properties significantly facilites the implementation of our method. Experimental validation was made on artificially generated related tasks of one-class classification. The results show that learning multiple related tasks simultaneously can achieve a better performance than learning each task independantly.

In our method we have used a common setting of both one-class SVM parameter and kernel parameter values. One future work is thus to use different parameter values for different tasks. The properties of kernels open a wide range of further developpements

on constructing new kernels for multi-task learning.

6 ACKNOWLEDGEMENT

The authors would like to thank the financial support from GIS 3SGS.

REFERENCES

Ben-David, S. & R. S. Borbely (2008). A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine Learning 73*, 273–287.

Ben-David, S. & R. Schuller (2003). Exploiting task relatedness for mulitple task learning. In *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory and the Seventh Kernel Workshop*, Washington, DC, USA, pp. 567–580.

Bi, J., T. Xiong, S. Yu, M. Dundar, & R. B. Rao (2008). An improved multi-task learning approach with applications in medical diagnosis. In *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, pp. 117–132.

Birlutiu, A., P. Groot, & T. Heskes (2010). Multi-task preference learning with an application to hearing aid personalization. *Neurocomputing 73*, 1177–1185.

Boser, B. E., I. M. Guyon, & V. N. Vapnik (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, Pittsburgh, Pennsylvania, United States, pp. 144–152.

Caruana, R. (1997). Multitask learning. *Machine Learning 28*, 41–75.

Eskin, E. (2000). Anomaly detection over noisy data using learned probability distributions. In *Proceedings of the Seventeenth International Conference on Machine Learning*, San Francisco, CA, USA, pp. 255–262.

Evgeniou, T., C. A. Micchelli, & M. Pontil (2005). Learning multiple tasks with kernel methods. *Journal of Machine Learning Research 6*, 615–637.

Evgeniou, T. & M. Pontil (2004). Regularized multi-task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, USA, pp. 109–117.

Gu, Q. & J. Zhou (2009). Learning the shared subspace for multi-task clustering and transductive transfer classification. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, Miami, Florida, USA, pp. 159–168.

Heskes, T. (2000). Empirical bayes for learning to learn. In *Proceedings of the Seventeenth International Conference on Machine Learning*, San Francisco, CA, USA, pp. 367–374.

Jebara, T. (2004). Multi-task feature and kernel selection for svms. In *Proceedings of the Twenty-first International Conference on Machine Learning*, Banff, Alberta, Canada.

Kemp, C., N. Goodman, & J. Tenenbaum (2008). Learning and using relational theories. In *Advances in Neural Information Processing Systems 20*, pp. 753–760. Cambridge, MA.

Pan, S. J. & Q. Yang (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering 22*(10), 1345–1359.

Ritter, G. & M. T. Gallegos (1997). Outliers in statistical pattern recognition and an application to automatic chromosome classification. *Pattern Recognition Letters 18*, 525–539.

Schölkopf, B., J. C. Platt, J. Shawe-Taylor, A. J. Smola, & R. C. Williamson (2001). Estimating the support of a high-dimensional distribution. *Neural Computation 13*(7), 1443–1471.

Schölkopf, B. & A. J. Smola (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press.

Singh, S. & M. Markou (2004). An approach to novelty detection applied to the classification of image regions. *IEEE Transactions on Knowledge and Data Engineering 16*(4), 396–407.

Smolarz, A. (1997). Etude qualitative du modèle auto-binomial appliqué à la synthèse de texture. In *XXIXèmes Journées de Statistique*, Carcassonne, France, pp. 712–715.

Tarassenko, L., P. Hayton, N. Cerneaz, & M. Brady (1995). Novelty detection for the identification of masses in mammograms. In *Proceedings of the Fourth IEE International Conference on Artificial Neural Networks*, Cambridge, UK, pp. 442–447.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York.

Widmer, C., N. Toussaint, Y. Altun, & G. Ratsch (2010). Inferring latent task structure for multitask learning by multiple kernel learning. *BMC Bioinformatics 11*(Suppl 8), S5.

Yang, H., I. King, & M. R. Lyu (2010). Multi-task learning for one-class classification. In *Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN)*, Barcelona, Spain, pp. 1–8.

Zheng, V. W., S. J. Pan, Q. Yang, & J. J. Pan (2008). Transferring multi-device localization models using latent multi-task learning. In *Proceedings of the 23rd National conference on Artificial intelligence*, pp. 1427–1432.