Multimedia Tools and Applications (2020) 79:25441–25461
https://doi.org/10.1007/s11042-020-09179-9

If-SVM: Iterative factoring support vector machine

Yuqing Pan[1] · Wenpeng Zhai[1] · Wei Gao[1] · Xiangjun Shen[1]

Received: 24 July 2019 / Revised: 31 May 2020 / Accepted: 4 June 2020 /
Published online: 3 July 2020

Abstract
Support Vector Machine (SVM) is widely applied in classification and regression tasks where support vectors are pursued through convex quadratic programming technique due to its effectiveness and efficiency. However, existing studies ignore the importance of training samples when they are fed into the model. In this paper, we propose a novel Iterative Factoring Support Vector Machine (If-SVM) method. Sample factoring is introduced in our proposed model to measure the significance of each data point, where it can effectively reduce the negative impact of trivial or noisy data points. In this way, our proposed model is concentrates on the critical data points falling around the hyperplane. By introducing this iterative factoring of data points into SVM, the classification accuracy of our proposed method is above that of 1.45% than other comparative methods in image recognition datasets. Experimental results on a variety of UCI demonstrate that, our proposed method has superior performances in decreasing the total number of support vectors than the other state-of-the-art SVM methods. More importantly, our further experiments also illustrate that, the classification performance of the state-of-the-art SVM methods can be improved 1.29% by incorporating our sample factoring idea into their models, which demonstrate our idea is a useful tool to improve the state-of-art SVM models.

Keywords Support vector machine (SVM) · Factoring · Iterative · Noisy samples

1 Introduction

Support Vector Machine (SVM) based on statistical learning theory is a machine learning algorithm proposed by Vapnik in [3, 4, 10]. The SVM seeks an optimal separation hyperplane between limited positive and negative sample information, and to find the optimal compromise between the complexity of the model and generalization ability has shown its advantages of effectiveness and efficiency in classification and regression task with support vectors being pursued through convex quadratic programming technique[2, 20, 31].

✉ Xiang-Jun Shen
xjshen@ujs.edu.cn

[1] School of Computer Science and Communication Engineering, JiangSu University, JiangSu, 212013, China

In order to improve the classification performance of SVM, various improved methods were proposed subsequently. One of such the methods applied mutual information (MI) to measure the relevance between two random variables [18, 24, 25], and to estimate the MI between each feature and the given class labels [12, 22]. The weights of each feature estimated by the MI method improve the generalization ability of the traditional SVMs, whereas show bad performance in high dimensions. Therefore, a novel radius-margin-based SVM model for joint learning of feature transformation and the SVM classifier [7, 21, 26–29] was proposed. However, most suffer computational expense and simplified forms of transformation. A central SVM (CSVM) [1, 9, 35] which uses class centers to construct support vector machine was proposed. Euclidean metric criterion extended to Minkowski metric was proposed to directly calculate weight of each feature [5, 16, 17]. Nevertheless, it is difficult to tune the additional parameter.

To the best of our knowledge, the importance of training samples before feeding into a model has not been considering in SVM. It is well known that all samples are assumed to have identical contributions to obtain optimal hyperplane in conventional SVM and its improved methods [8, 15, 19]. However, available training data are often contaminated by noise and outliers in many practical applications. Therefore, the performance of SVM may be dominated by weakly related or even irrelevant samples. A robust support vector machine [11, 23] was proposed, where a general method is able to form an adaptive margin by using the distance between each class of training data center and data points. Lin et al proposed a Fuzzy Support Vector Machine (FSVM) [13, 14] applying the fuzzy membership degree to the training data to relax the influence of outliers. However, the selection of membership function has always been a difficult problem in the fuzzy support vector machine.

From the above discussion, we propose a novel Iterative Factoring Support Vector Machine (If-SVM), where sample factoring is introduced in our proposed model to measure the significance of each data point. It can effectively decrease the negative impact of trivial or noisy data points. Thus, it avoids training the classifier on trivial or noisy samples. Compared with existing weighted SVM methods, we can derive novel better dataset in training. Therefore, the influences of non-critical samples in SVM are decreased. By introducing this iterative factoring data points in SVM, the classification accuracy of our proposed method is above that of 1.45% than other comparative methods in image recognition datasets. We also will extend and apply our idea to other image processing applications in future work, such as image segmentation [30, 32, 34].

1) We introduced a sample factor into the proposed model to measure the significance of each data point. This indicator variable can determine whether a data point is a critical sample or not.
2) We further propose a novel Iterative Factoring Support Vector Machine (If-SVM) method which iteratively evaluates the importance of each sample to reduce the influence of non-critical samples. This significantly decreases negative impacts of trivial or noisy data points on the classifier model.
3) Our further experiments also demonstrate that, the performance of the state-of-the-art SVM methods can also be improved by incorporating our sample factoring idea into their models, which demonstrate our idea is a useful tool to improve the state-of-art SVM models.

The remainder of this paper is organized as follows. Section 2 briefly reviews the basic theory of standard SVM and introduces some improved methods. We present the theoretical

deduction of our proposed algorithm in detail. Next, it is extended into kernel space in Section 3. Experimental evaluation is reported and discussed in Section 4. Finally, we conclude the paper with future work in Section 5.

2 Related work

In order to effectively reduce noise and maximally improve classification accuracy, many researchers proposed some methods to improve the performance of support vector machine. In this section, we briefly review the traditional SVM for classification and present several improved methods.

Suppose the training set of the classification problem is $T = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, $\mathbf{x}_i$ represents the ith training sample, $y_i \in \{-1, 1\}$ stands for the corresponding class label of $\mathbf{x}_i$, with $i = 1, ..., n$. The classic SVM algorithm aims to obtain optimal hyperplane by the following optimization problem

$$\begin{aligned} \min \frac{1}{2} \|\mathbf{w}\|^2 \ + \ C \sum_{i=1}^{n} \xi_{\mathbf{i}} \\ \text{s.t. } y_i (\mathbf{w^T x_i} + b) \ \geq \ 1 - \xi_i, i \ = \ 1, ..., n, \\ \xi_i \geq 0, i \ = \ 1, ..., n, \end{aligned} \tag{1}$$

where $\mathbf{w}$ is a normal vector of hyperplane $\mathbf{w^T x_i} + b = 0$ in the feature space, b is the scalar offset of hyperplane, $\xi_{\mathbf{i}}$ is a slack variable and C is a penalty parameter.

In this way, the optimization problem can be transformed into a convex quadratic programming problem. To solve this quadratic programming problem, we construct a Lagrangian and transform into the dual

$$\begin{aligned} \max \sum_{i=1}^{n} \alpha_i \ - \ \frac{1}{2} \sum_{i=1}^{n} \sum_{\mathrm{j}=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x_i}^T \mathbf{x_j} \\ \text{s.t.} \quad \sum_{\mathrm{i}=1}^{N} \alpha_i \ y_i = 0, \\ 0 \ \leq \ \alpha_i \leq C, i = 1, ..., n, \end{aligned} \tag{2}$$

where $\alpha = [\alpha_1, ..., \alpha_n]^T$ is the vector of nonnegative Lagrange multipliers. The corresponding decision function is $\text{sgn} \sum_{i=1}^{n} y_i \alpha_i \mathbf{x}_i{}^T \mathbf{x} + b$.

Since, traditional classification algorithm of support vector machine assumes each sample vector has the same importance for classification, the discrepancies of training samples were ignored when fed into the model. Therefore, this may affect the classification performance of support vector machine when we predict a sample category. To solve this problem, researchers proposed sample weighted methods, which give large weight to the training samples with high relevance, and small weight to the training samples with low relevance.

Extending the original probabilistic c-means (PCM) algorithm into a kernel space based on kernel methods, Yang developed the KPCM algorithm where partitioned relative values are used as weights for the proposed W-SVM [29]. The weights used in WSVM are generated by kernel-based probabilistic c-means (KPCM) algorithm, the corresponding

optimization problem can be formulated as

$$\max \sum_{i=1}^{n} \alpha_i \; - \; \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j})$$
$$\text{s.t.} \quad \sum_{\mathrm{i}=1}^{N} \alpha_i \; y_i = 0,$$
$$0 \leq \alpha_i \; \leq \; C * u_i, i = 1, ..., n, \tag{3}$$

where $K(\mathbf{x}_i, \mathbf{x}_j)$ can be represented by a dot product in the feature space as $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi \mathbf{x}_i \cdot \varphi \mathbf{x}_j$, the nonlinear mapping function φ maps an input vector $\mathbf{x}$ in the input space $\mathbf{X}$ onto $\varphi \mathbf{x}$ in the feature space F, that is $\varphi : X \rightarrow F$. Note that a weight is assigned to the data point $\mathbf{x}_i$ in (3). Penalty parameter C is a constant, and $C * u_i$ will set different penalty parameters for each training sample. It can be drawn from formula (3) that the larger the $C * u_i$, the smaller the possibility of misclassification of sample $\mathbf{x}_i$.

Cui et al [6] combined an outlier detection approach and adaptive weight value for the training samples. Suppose the weight is u_i and the error ${\xi_i}^2$ is weighted, the optimization problem is presented as follows

$$\min \frac{1}{2} \|\mathbf{w}\|^2 \; + \; C \sum_{i=1}^{n} {\xi_i}^2 u_i$$
$$\text{s.t.} \; \; y_i(\mathbf{w^T x_i} + b) \geq 1 \; - \; \xi_i, i = 1, ..., n, ,$$
$$\xi_i \geq 0, i \; = \; 1, ..., n, \tag{4}$$

where the initial weight is calculated according to the fitting error of each data sample, the larger the fitting error of data sample, the smaller the weight. By this way, the interference of noise and isolated points can be reduced in classification, while the normal samples remain unchanged in classification.

In addition, with regards to features of the sample vector, some features are strongly correlated with the classification, whereas others are weakly correlated or even unrelated. If we do not consider the distinct importance of different features in the classification, then the kernel function may be determined by the weak related or unrelated features leading to performance degradation. Therefore, many improved feature weighted methods were proposed.

Do et al [7] introduced a vector of parameters u_i, which in fact performs feature weighting. The weight u_i will be used to calculation of kernel function. The radius R is bounded with $u_i {R_i}^2 \leq {R_i}^2 \leq \sum_{i=1}^{n} u_i {R_i}^2 \leq 1$. The MR-SVM solves the following convex relaxation problem

$$\frac{1}{2} \sum_{i=1}^{n} \frac{\mathbf{w_i}^2}{u_i} + \frac{C}{\sum_{i=1}^{n} u_i {R_i}^2} \sum_{i=1}^{n} {\xi_i}^2$$
$$\text{s.t.} \quad y_i = \mathbf{w^T x_i} + b + \xi_i, i = 1, ..., n,$$
$$\xi_i \geq 0, i = 1, ..., n, \tag{5}$$

where u_i is a weight for the ith feature and R_i is the radius of dimensions i. This vector u_i weights the different features in the feature space. Under the sparsity constraint, it forces

many trivial features to have a zero weight. Therefore, it can reduce a larger number of trivial features at each iteration.

Wu et al [27] proposed a convex radius-margin-based SVM model for joint learning of feature transformation and SVM classifier. The generalized block coordinate descent method is used to solve the F-SVM model, and the feature transformation is updated by gradient descent. F-SVM introduces a linear transformation matrix A and integrates the radius information, the radius-margin-based model given as follows

$$\begin{aligned} \min \frac{1}{2}\|\mathbf{w}\|^2 {R_i}^2 \; + \; C \sum_{i=1}^{n} {\xi_i}^2 u_i \\ \text{s.t.} \quad y_i = \mathbf{w^T A x_i} \; + \; b + \xi_i, i \; = \; 1, ..., n, \\ \xi_i \geq 0, i \; = \; 1, ..., n, \end{aligned} \tag{6}$$

where the radius R is bounded with $\|\mathbf{Ax_i} - \mathbf{Ax_0}\| \leq 1, \xi_i, i = 1, ..., n$. In this way, the computation of kernel function can avoid being dominated by irrelevant or trivial features.

3 The proposed if-SVM

This section presents our proposed novel Iterative Factoring Support Vector Machine method which can effectively decrease the influence of non-critical samples and improve the classification performance. We introduce a scaling factor in order to measure the significance of each data point. The proposed If-SVM is formulated to in the following section.

3.1 The proposed method

Given a training data set T in feature space, to obtain optimal separating hyperplane, the traditional SVM model in (1) can be designed as follows

$$\begin{aligned} \min \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \; y_i(\mathbf{w^T x_i} d_i + b) \geq 1 - \xi_i, i = 1, ..., n, \\ \xi_i \geq 0, i = 1, ..., n, \\ d_i \geq 0, i = 1, ..., n, \end{aligned} \tag{7}$$

where C is a constant which determines the trade-off between margin maximization and the amount of misclassification. Note that as shown in (7), a sample factoring d_i is introduced in the standard SVM to measure the significance of each data point.

The above optimization problem can be solved by its dual problem. Then we construct the Lagrange function as follows

$$L(\mathbf{w}, b, \xi, d, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$
$$- \sum_{i=1}^{n}\alpha_i y_i(\mathbf{w}^{\mathbf{T}}\mathbf{x_i}d_i + b) - 1 + \xi_i$$
$$- \sum_{i=1}^{n} V_i\xi_i - \gamma(\sum_{i=1}^{n} d_i - 1) - \sum_{i=1}^{n}\varphi_i d_i, \tag{8}$$

By taking the derivative of the Lagrange $L(\mathbf{w}, b, \xi, d, \alpha)$ with respect to parameters, $\mathbf{w}, b, \xi, d, \alpha$ the following dual optimization problem is presented

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i \mathbf{x_i}^T d_i = 0, \tag{9}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n}\alpha_i y_i = 0, \tag{10}$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - V_{\mathrm{i}} = 0, \tag{11}$$

$$\frac{\partial L}{\partial d_i} = -\alpha_i y_i \mathbf{x_i}^T d_i + \gamma - \varphi_i = 0. \tag{12}$$

According to the Karush-Kuhn-Tuker conditions, the following expression can be defined

$$\varphi_i d_i = 0. \tag{13}$$

It can be observed that if Lagrange parameter $\varphi_i \neq 0$ the $d_i = 0$, then we further obtain $\mathbf{w}^{\mathbf{T}}\mathbf{x_i}d_i = 0$, which means $\mathbf{x_i}$ is not a support vector since it is not involved in training the model. From this observation, the physical explanation of sample factoring d_i is an indicator variable to determine whether the data point $\mathbf{x_i}$ is a critical sample or not. That is to say that, if $d_i \neq 0$, α_i has a non-trivial solution with a high probability.

With this explanation of sample factoring, substituting (9)-(12) into the Lagrange (8), yields the following dual optimization problem

$$\min \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T \mathbf{x_j} d_i d_j - \sum_{i=1}^{n}\alpha_i$$
$$\text{s.t.} \quad \sum_{i=1}^{n}\alpha_i y_i = 0, i = 1, ..., n,$$
$$0 \leq \alpha_i \leq C, i = 1, ..., n. \tag{14}$$

For a test sample $\mathbf{x_i}$, its class label can be determined by the following function

$$F(\mathbf{x_i}) = sign(\sum_{i=1}^{n}\alpha_i y_i \mathbf{x_i}^T d_i \mathbf{x} + b). \tag{15}$$

For the nonlinear SVM, the optimization problem can be generalized for nonlinear kernels as follows

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j})d_i d_j - \sum_{i=1}^{n}\alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n}\alpha_i y_i = 0, i = 1, ..., n,$$

$$0 \le \alpha_i \le C, i = 1, ..., n. \tag{16}$$

Finally, the decision function can be generalized for nonlinear kernels as follows:

$$F(\mathbf{x_i}) = sign(\sum_{i=1}^{n}\alpha_i y_i K(\mathbf{x_i}, \mathbf{x_j})d_i d_j + b) \tag{17}$$

3.2 Obtaining sample factors

Since the sample factor d_i is able to indicate whether the data point $\mathbf{x_i}$ is a critical sample or not, which is illustrated in Fig. 1, we obtain the value of sample factor as follows

$$s_i = (\mathbf{w^T x_i} + b) \Big/ \left\|\mathbf{w}^2\right\|, \tag{18}$$

$$d_i = \max(0, 1 - s_i), \tag{19}$$

where s_i is the distance between data point $\mathbf{x_i}$ and the hyperplane. As d_i is an indicator of $\mathbf{x_i}$ being a support vector or not, we use hinge loss distance d_i to evaluate the importance of $\mathbf{x_i}$. If $\mathbf{x_i}$ is close to the hyperplane, we set a larger factor value. On the contrary, if $\mathbf{x_i}$ is far away from the hyperplane, it means that $\mathbf{x_i}$ is less important with a smaller factor value. If $\mathbf{x_i}$ is 0, namely, not a support vector, it will be discarded and have no any impact in the next iteration of modeling. By this way, our proposed If-SVM model is capable of focusing on the critical data falling around the hyperplane, and abandons those data which are far away from the hyperplane. With this sample factoring setting, our If-SVM can obtain

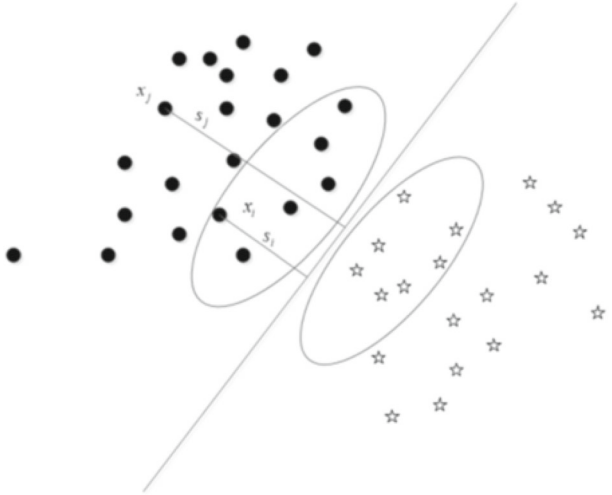

Fig. 1 SVM optimal separation hyperplane

Table 1 The steps of If-SVM

Algorithm 1 Iterative Factoring Support Vector Machine (If-SVM) method.

Input:$T = \{(x_i, y_i)\}_{i=1}^{n}$
Initialize $\{d_i = 1\} \forall i$
Output: If-SVM classification decision model according to eq. (17)
1: While loss and $(\mathbf{w}, b)$not converged do
2: Train the model by using the off ¨ Cthe-shelf-SVM solver
3: Obtain the parameters $\mathbf{w}, b$
4: Update the d_i according to the eq. (18) and eq. (19)
5: Update the dataset $T_{i+1} = T * d_i$
6: End While

better hyperplane with this iterative data reducing model. For conciseness, the main steps of our If-SVM optimization algorithm are listed in Table 1 and Fig. 2. In the initialization stage, we assume d_i=1 and initialize **w,b** using the off-the-shelf SVM solver. Then, d_i is updated according to the (18) and (19) and the dataset is updated by $T_{i+1} = T * d_i$. Our If-SVM algorithm alternates between updating w,b and d_i until convergence. As a result, our proposed model can concentrate on the critical data around the hyperplane to achieve a better classification hyperplane.

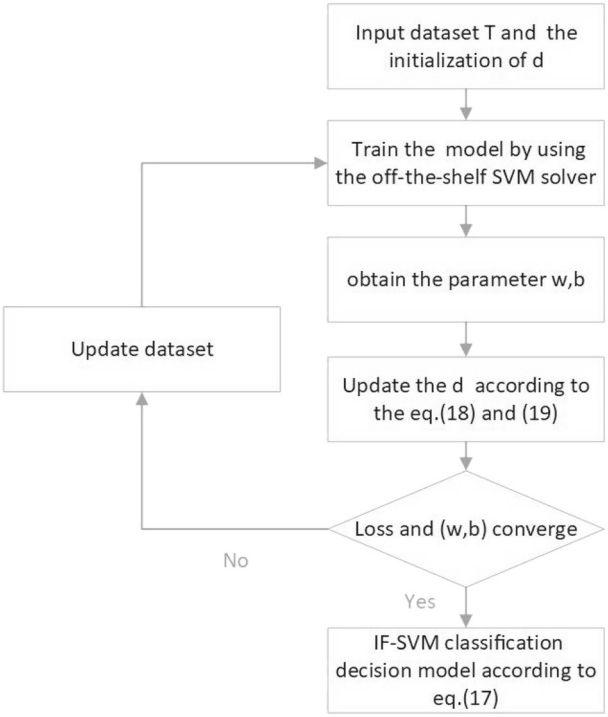

Fig. 2 The flowchart of If-SVM

Table 2 Description of the USPS, MNIST, Extended Yale B and CIFAR-10 datasets used in the experiments

Dataset	Number of samples	Feature dimensions
USPS	9298	256
MNIST	70000	256
Extended Yale B	2414	1024
CIFAR-100	60000	1024

4 Experiments

In this section, we evaluate the performance of our proposed If-SVM method in comparison with several state-of-the-art methods including SVM [33], WSVM [16], RMM [14] and FWSVM [34]. Experiments have been conducted on the 11 UCI data sets as described in Table 3, the LFW database and four large-scale image data sets, USPS, Extended Yale B, CIFAR-10 and MNIST.

4.1 Dataset description

We select 15 publicly available datasets for the evaluation of the performance of our algorithm: UCI, USPS, Extended Yale B, CIFAR-10 and MNIST. 11 of them were taken from the UCI repository. Experiments are conducted on USPS, MNIST, Extended Yale B, CIFAR-10 and UCI datasets as described in Tables 2 and 3. The results for the various methods are shown in Tables 4 and 5 with best results in bold.

The USPS dataset: This dataset consists of handwritten numbers from 0 to 9. The training and testing sets consist of 7291 examples and 2007 examples respectively. Each example has 256 attributes or pixels that describe each number.

The Extended Yale B dataset: The Extended Yale B database consists of 2,414 frontal-face images of 38 subjects. The cropped 192¡Á168 face images were captured under various laboratory-controlled lighting conditions and with different facial expressions. For each subject, half of the images are randomly selected for training (i.e., about 32 images per subject), and the left half for testing.

Table 3 Description of the 11 UCI datasets used in the experiments

Dataset	Number of samples	Feature dimensions
Breast	4770	9
Ionosphere	351	34
Liver	345	6
Musk	6598	166
Parkinsons	195	22
Titanic	3300	5
Wpbc	198	13
Ringnorm	7400	20
Twonorm	7400	20
German	5000	20
Image	7500	18

Table 4 Comparison of the average classification accuracy by linear SVM, linear WSVM, linear RMM, linear FWSVM and linear If-SVM

Dataset	SVM	WSVM	FWSVM	RMM	If-SVM
Breast	70.71	73.62	73.46	72.71	**74.85**
Ionosphere	89.17	90.01	90.64	90.64	**91.24**
Liver	70.00	71.17	71.23	71.70	**73.10**
Musk	91.98	**92.72**	92.28	92.20	92.38
Parkinsons	89.06	90.13	**91.34**	90.01	90.86
Titanic	90.60	91.64	91.91	90.81	**93.67**
Wpbc	79.50	81.31	82.25	82.50	**83.50**
Ringnorm	75.58	75.12	75.71	75.60	**76.35**
Twonorm	96.65	96.94	96.32	96.60	**97.32**
German	76.73	77.30	77.92	77.02	**78.93**
Image	85.09	85.54	85.30	85.14	**85.99**

The CIFAR-100 dataset: The CIFAR-100 dataset consists of 60000 natural color images and has 100 classes. The dataset contains a training set of 50000 images and a test set of 10000 images. Each example has 1024 attributes or pixels that describe each image.

The MNIST dataset: The MNIST dataset comes from the National Institute of Standards and Technology (NIST). The training set consists of 250 different handwritten digits. The training and testing sets consist of 60000 examples and 10000 examples respectively. Each example has 256 attributes or pixels that describe each number.

4.2 Experiment settings

For each dataset, we use the average classification accuracy obtained by 10 runs of the tenfold cross validation (CV) as the performance indicators. In our tenfold CV, the training set of n samples is randomly divided into 10 folds of size n/10. Then, the classifier is trained using 9 folds while the learned classifier is evaluated using the retained test fold. Therefore,

Table 5 Comparison of the average classification accuracy by kernel SVM, kernel WSVM, kernel RMM, kernel FWSVM and kernel If-SVM

Dataset	SVM	WSVM	FWSVM	RMM	If-SVM
Breast	75.71	75.53	75.86	75.65	**76.88**
Ionosphere	91.94	93.11	92.58	92.80	**95.46**
Liver	74.00	75.12	74.73	74.20	**76.45**
Musk	**99.73**	99.31	98.28	98.20	99.65
Parkinsons	85.37	87.01	86.95	86.77	**89.16**
Titanic	90.93	91.24	90.91	91.22	**92.27**
Wpbc	78.00	80.15	82.37	83.00	**83.45**
Ringnorm	98.01	98.15	98.86	98.30	**99.32**
Twonorm	97.81	97.97	98.12	98.00	**98.53**
German	75.50	76.90	78.41	76.59	**78.81**
Image	84.97	84.88	85.13	83.92	**85.43**

all samples in the dataset are used as training set and test set, and each sample is verified once. Finally, the results on the ten test folds are averaged to produce a single estimation. Moreover, the running time of each method is provided according to the ten runs of our ten-fold CV. All the experiments are conducted on a desktop PC with Intel(R) Xeon(R) CPU (3.30 GHz) and 32GB RAM under the MATLAB 2017b programming environment. In experiments, a coarse-to-fine search strategy is adopted for determining the hyper parameters. The grid search method is first adopted for coarse searching, and then the line bisection method is exploited to refine the hyper parameters within a small range. Concretely, we set penalty parameter $C \in \{ 10^{\min:step:\max} \}$ with min=-3, step=1, max =5 in linear If-SVM and $\sigma \in \{ 2^{\min:step:\max} \}$ with min=-10, step=1, max =5 for Gaussian RBF kernel in kernel If-SVM.

4.3 Experimental results on UCI datasets

For each UCI dataset, the If-SVM method is compared with several existing methods, including the LIBSVM, WSVM [29], RMM [21] and FWSVM [33].

1) Evaluation on Linear If-SVM: Table 4 presents the classification accuracy of our proposed linear If-SVM and the competing methods. As we can see in Table 4, If-SVM achieves the best or the second best classification accuracy on 11 UCI data sets. The classification accuracy of If-SVM is 85.29% above that of 2.10% to traditional SVM, that of WSVM 1.15% , that of FWSVM 0.89% , that of RMM 1.21% , Specifically, the improvement of If-SVM over SVM is higher than 3.0% by accuracy on 3 data sets, i.e., Breast, Titanic and Wpbc. We also evaluate the effect of hyper parameter C in linear If-SVM on Wpbc dataset. It can be seen from Fig. 3 that when $C < 0.1$, the accuracy is relatively low. The classification accuracy can be improved along with the increase of C to 10. Nevertheless, the accuracy decreases significantly when $C > 100$.
2) Evaluation on Kernel If-SVM: Table 5 lists the classification accuracy of our proposed kernel If-SVM and the competing methods. As shown in Table 5, Kernel If-SVM achieves the highest classification accuracy on 10 of the 11 data sets among the competing methods. The classification accuracy of If-SVM is 88.67% in excess of 2.13%

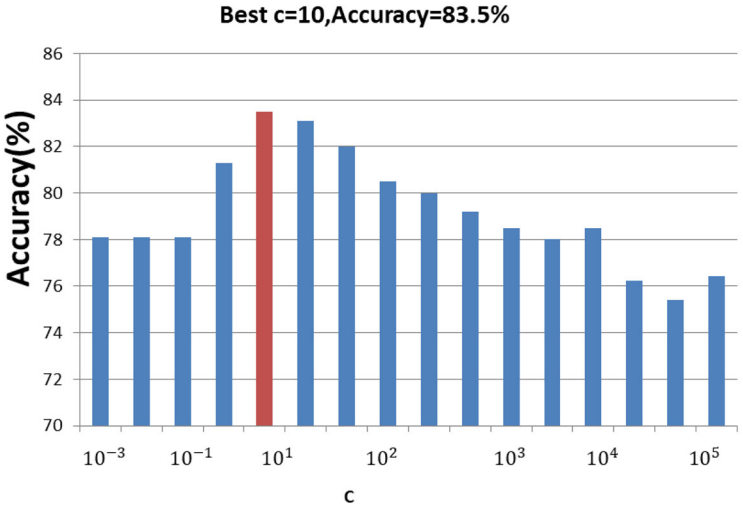

Fig. 3 Parameters C of If-SVM on the Wpbc dataset

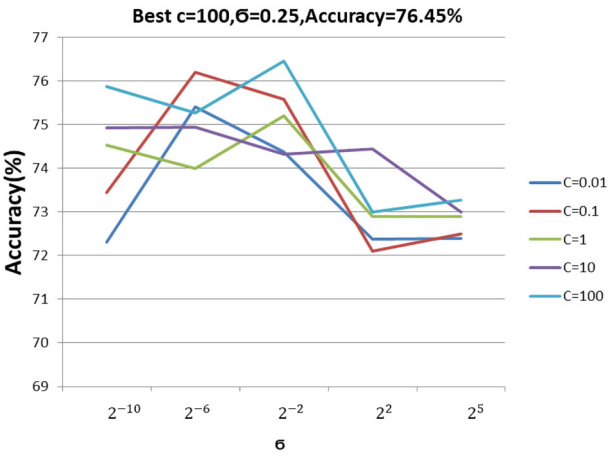

Fig. 4 Parameters analysis on the Liver dataset

to traditional SVM, 1.46% to W-SVM, 1.20% to FWSVM, 1.52% to RMM. Specifically, the improvement of If-SVM over SVM is higher than 3.0% by accuracy on 4 data sets, i.e., Ionosphere, Parkinsons, Wpbc and German. As shown in Fig. 4, we evaluate the effect of hyper parameters using the Liver dataset, including the tradeoff C and the kernel parameter σ in kernel F-SVM. We can see that better accuracy can be obtained by using larger C (e.g., C=100) and smaller σ (e.g., σ =0.25). Similar conclusion can be drawn from other data sets.

4.4 Experimental results on image classification datasets

For each dataset, the If-SVM method is compared with several existing methods, including the LIBSVM,WSVM [29], RMM [21] and FWSVM [33]

1) Evaluation on Linear If-SVM: Table 6 and Fig. 5 presents the classification accuracy and average classification accuracy of our proposed linear If-SVM and the competing methods. As shown in Table 6, If-SVM achieves the best or the second best classification accuracy on USPS, Extended Yale B, CIFAR-100 and MNIST datasets. It is shown in Fig. 5 that the classification accuracy of If-SVM is 76.37% above that of 2.50% to traditional SVM, that of WSVM 1.59% , that of FWSVM 1.13% , that of RMM 1.01% .

Table 6 Comparison of the average classification accuracy by linear SVM, linear WSVM, linear RMM, linear FWSVM and linear If-SVM

Dataset	SVM	WSVM	FWSVM	RMM	If-SVM
USPS	69.05	71.75	67.01	72.15	**72.56**
MNIST	80.02	80.77	81.52	81.30	**82.13**
E Yale B	71.98	72.29	73.01	72.56	**74.04**
CIFAR-100	74.43	74.32	75.43	75.22	**76.76**

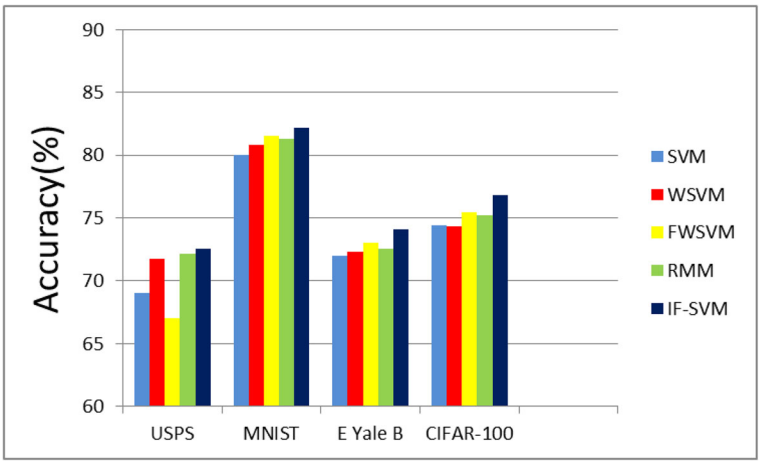

Fig. 5 The average classification accuracy of linear SVM, linear WSVM, linear RMM, linear FWSVM and linear If-SVM on LFW database

2) Evaluation on Kernel If-SVM: Table 7 and Fig. 6 present the classification accuracy and average classification accuracy of our proposed kernel If-SVM and the competing methods. As we can see in Table 6, the average classification accuracy of If-SVM is 83.13% in excess of 2.75% to traditional SVM, 1.83% to W-SVM, 1.09% to FWSVM, 1.73% to RMM. In order to further compare the performance of various methods, we use the box diagrams and line chart to display intuitively. The accuracy of five methods on MNIST dataset is depicted in Fig. 7, which shows that the median accuracy of If-SVM is better than the comparative methods on MNIST dataset. It means that the proposed method can effectively improve the classification accuracy. In addition, the shape of each box diagram of If-SVM is relatively narrow, which is an indication that our method is more stable than the others.

4.5 Experimental results on LFW database

The face recognition method can be evaluated with two test protocols for LFW: the restricted and the unrestricted settings. In our experiments, the performance is evaluated by our tenfold CV on a set of 300 positive and 300 negative image pairs under the restricted settings. The only information available is whether each pair of training images is the same person. We

Table 7 Comparison of the average classification accuracy by kernel SVM, kernel WSVM, kernel RMM, kernel FWSVM and kernel If-SVM

Dataset	SVM	WSVM	FWSVM	RMM	If-SVM
USPS	72.58	73.04	74.05	72.11	**75.73**
MNIST	90.69	94.20	95.26	93.82	**95.84**
E Yale B	79.04	80.19	81.20	79.05	**81.90**
CIFAR-100	76.22	77.76	77.66	76.62	**79.06**

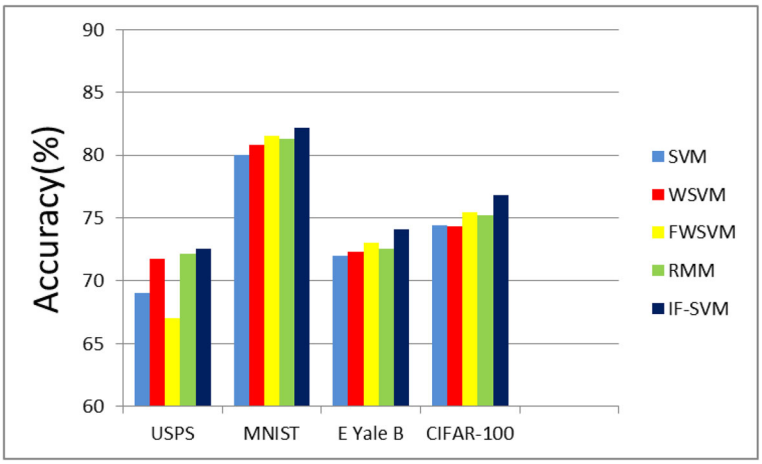

Fig. 6 The average classification accuracy of kernel SVM, kernel WSVM, kernel RMM, kernel FWSVM and kernel If-SVM on LFW database

use SIFT features to extract 128 features at nine fiducial points on three scales, and finally get 3456 dimensional feature vector. However, due to the large scale of dimension and the limitation of computational overhead, we use the PCA for dimensionality reduction to 100. We compare the accuracy of face recognition with other state-of-the-art algorithms, and the results are illustrated in Table 8. From Table 8, the accuracy of our If-SVM is higher than other state-of-the-art methods. The performance of kernel If-SVM is 0.85% higher than that of W-SVM on LFW database. Moreover, kernel If-SVM can still get an improvement of 0.75% over FWSVM. Figure 8 shows the ROC curves of the competing methods. It can be seen from Fig. 8 that If-SVM algorithm has better classification performance to the other algorithm.

The disadvantages and advantages of five methods are listed in Table 9. Compared with the existing SVM and RMM methods, our If-SVM model improves the robustness

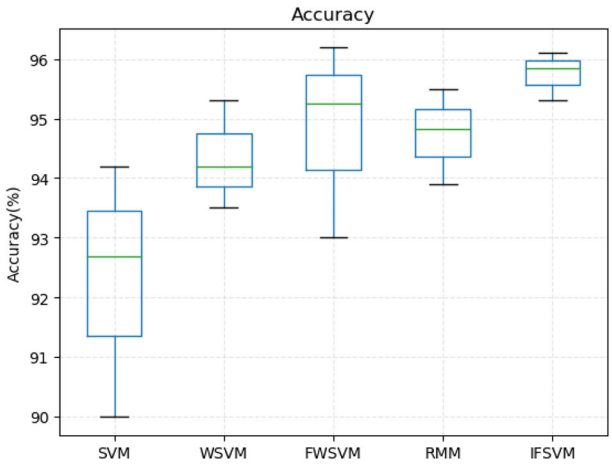

Fig. 7 The accuracy of five methods on MNIST dataset

Table 8 Comparison the average classification accuracy by SVM, W-SVM, RMM, FWSVM and If-SVM in LFW database

SVM methods	Linear	Kernel
SVM	72.71	73.57
W-SVM++	79.50	80.00
RMM	78.33	78.93
FWSVM	79.16	80.35
IFSVM	80.35	81.10

and shows better classification performance by reducing the adverse impact of trivial or noisy data points on the classifier. Unlike the WSVM and FWSVM which are difficult to tune additional parameter, our proposed method has robustness with a roughly parameters searching.

4.6 Applying sample factoring idea into state-of-the-art methods

Moreover, our further experiments also demonstrate that, by applying our sample factoring idea into other state-of-the-art kernel methods, it can also improve the performance of those methods.

In order to clearly show the advantages of our method over the comparative methods, we discuss and analyze the results in this section. From Table 10, it is obvious that the accuracy

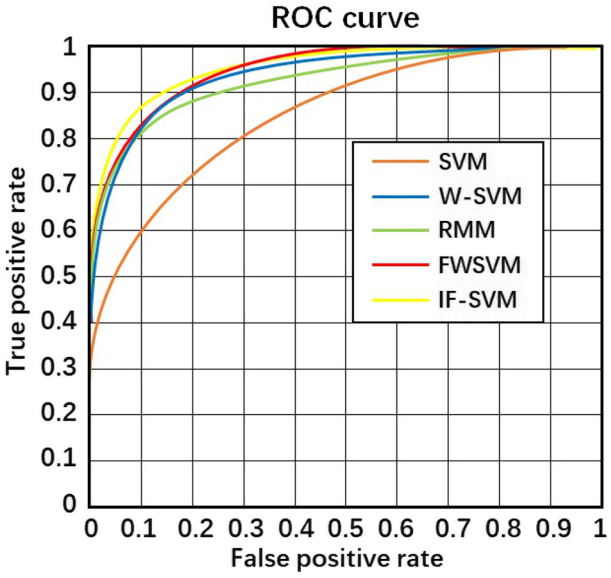

Fig. 8 Roc curves on LFW database

Table 9 Summary of the characteristics of five methods

Methods	SVM	WSVM	RMM	FWSVM	If-SVM
Impact of outliers	High	Normal	High	Low	Low
Classification accuracy	Low	Normal	Normal	Normal	High
Selection of parameter	Normal	Difficult	Normal	Difficult	Normal

of classification has been improved by combining our sample factoring idea with other state-of-the-art kernel algorithm. In the image classification dataset, as we can see the average classification accuracy of If-SVM above that of F-SVM 1.49% , that of R-SVM 1.26% , that of W-SVM 1.72% . This indicates that the samples being extracted by the proposed If-SVM are more discriminatory as compare to those by comparative methods, which indicates that If-SVM is more stable than others. The reason is that our methods can reduce the effect of those imperfect samples through a factor weighting in the model. For a comprehensive illustration, we also present some experimental results on the number of samples and run time in training.

In each step of the iteration, we have to compute the solution of a standard SVM which has a complexity $O(n^2)$, where n is number of training samples. Moreover, we also need to update the factor d^i which has a complexity $O(n^3)$. However, beneficial from our algorithm, it can decrease the number of non-critical samples in SVM, and the number of samples in the training set are used for training decreases dramatically at second iteration. Therefore, time complexity increase is not very high.

In order to prove that our algorithm does not increase the time complexity of the algorithm, we use the proposed method to classify four data sets from UCI. As can be seen from Fig. 9, we reduced the scale of the training set and removed noise samples in the process of iteration. In our proposed method, only 60% of the samples in the training set are used for

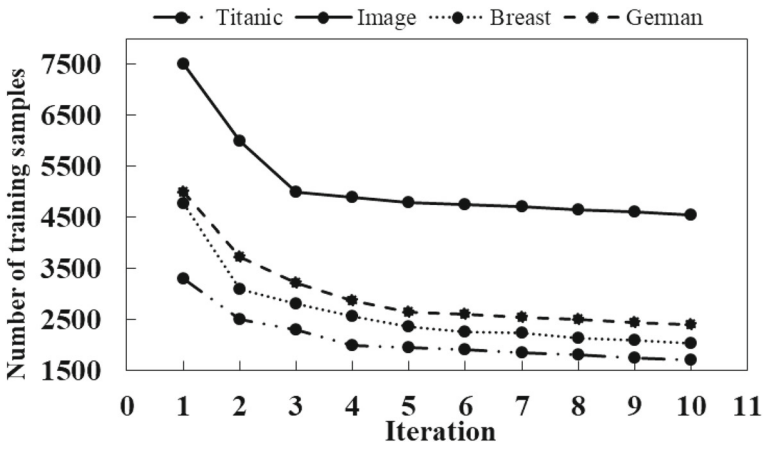

Fig. 9 The number of training samples on the Titanic, Image, Breast and German dataset

Table 10 Comparison the average classification accuracy by F-SVM, F-SVM++, $R-SVM^{+}$ [35], $R-SVM^{+}$++, W-SVM and W-SVM ++(++ means add sample factor idea to the original method

SVM methods	USPS	MINST	E Yale B	CIFAR-100
F-SVM	90.17	97.55	80.47	82.13
F-SVM++	92.27	98.61	82.31	83.12
$R-SVM^{+}$	70.44	94.53	78.62	81.30
$R-SVM^{+}$++	71.10	95.81	81.07	81.95
W-SVM	70.90	93.65	75.05	74.83
W-SVM++	72.25	94.57	79.17	75.32

training. Therefore, the run time of this algorithm has not increased too much (Figs. 10, 11 and 12).

To further prove the effect of our method, we use bar charts to show the run time on the 7 datasets in Fig. 10 to Fig. 13. If-SVM is about three times slower than SVM on Titanic and German dataset. However, benefitted from our proposed algorithm, the number of non-critical samples in SVM is decreased. Therefore, this method is about 70% time slower than SVM in large data sets such as Ringnorm, Musk and Twonorm datasets. And with the increase of training set amount, the effect of our proposed method is more obvious.

F-SVM++, $R-SVM^{+}$++ and W-SVM++ is moderately quicker than original methods without sample factor. From Fig. 11, it is shown that the run time of F-SVM++ and the original methods without sample factor on the 7 datasets in training. F-SVM++ spends less run time on 6 of the 7 data sets than the original methods without sample factor. From Fig. 12, it is shown that the run time of $R-SVM^{+}$++ and the original methods without sample factor on the 7 datasets. $R-SVM^{+}$++ is faster than the original methods without sample factor on 7 data sets, From Fig. 13, it is shown that the run time of W-SVM++ and the original methods without sample factor on the 7 datasets. W-SVM++ achieves the less run

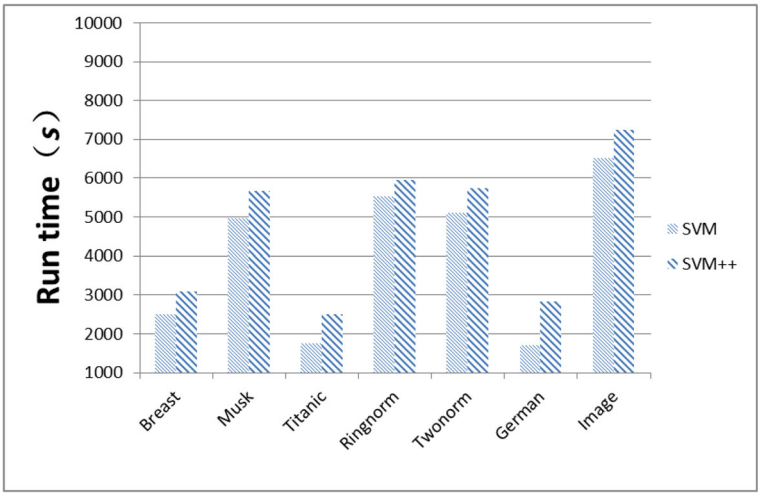

Fig. 10 Comparison of the run time (in seconds, s) of SVM and SVM++(++ means add sample factor idea to the original method

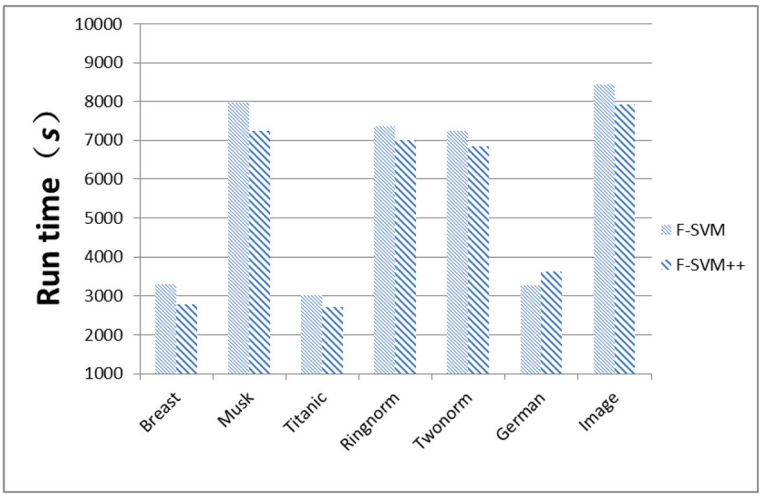

Fig. 11 Comparison of the run time (in seconds, s) of F-SVM and F-SVM++ (++ means add sample factor idea to the original method

time on 5 of the 7 data sets than the original methods without sample factor. Although our proposed method has a slight disadvantage in training time on Titanic, German and Breast datasets. It is evident from Figs. 10 to 13 that F-SVM++, $R-SVM^{+}$++ and W-SVM++ can spend less run time in larger datasets above 2000 samples than original methods without sample factor methods. This reveals that, the proposed method ensures the classification performance of SVM model, effectively reducing the amount of training data, which reduces the run time in training.

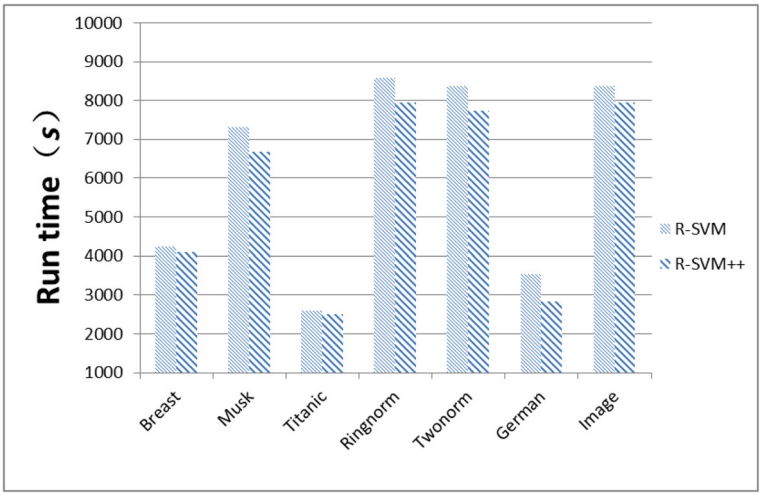

Fig. 12 Comparison of the run time (in seconds, s) of $R-SVM^{+}$ and $R-SVM^{+}$++(++ means add sample factor idea to the original method

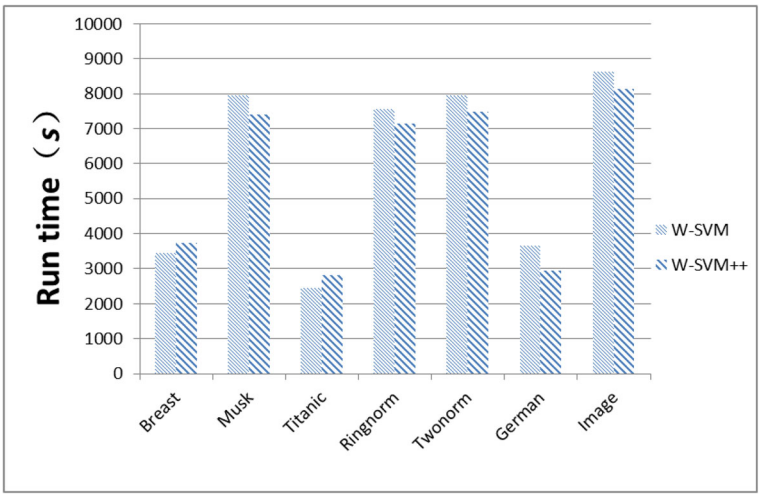

Fig. 13 Comparison of the run time (in seconds, s) of W-SVM and W-SVM ++(++ means add sample factor idea to the original method

5 Conclusion

In this paper, we propose a novel If-SVM method. A sample factoring is introduced in the standard SVM to measure the significance of each data point. It can avoid the classifier being trained by trivial or noisy samples. Therefore the influence of non-critical samples in SVM is decreased. By this way, our proposed model can pay more attention to the critical data which fall around the hyperplane. And it can help to achieve a better classification hyperplane. Experimental results with several UCI datasets show that our If-SVM can decrease the numbers of support vectors and have better classification performance than state-of-the-art SVM methods. Extensive experiments on different image classification datasets demonstrate that our proposed method have advantage of better performances in image classification accuracy among the other comparative SVM methods. Our further experiments also demonstrate that, by applying our sample factoring idea into other state-of-the-art kernel methods, it can also help to improve the performance of those methods. Finally, motivated by the recent success of image segmentation, we will extend and apply our idea to other image processing applications in future work.

Acknowledgements This work was funded in part by the National Natural Science Foundation of China (No.61572240, 61701200).

References

1. Amorim RCD, Mirkin B (2012) Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering. Pattern Recogn 45(3):1061–1075
2. Apasiba Abeo T, Shen XJ, Bao BK, Zha ZJ, Fan J (2019) A generalized multi-dictionary least squares framework regularized with multi-graph embeddings. Pattern Recognit 90:1–11
3. Chaki J, Dey N, Shi F, Sherratt RS (2019) Pattern mining approaches used in sensor-based biometric recognition: a review. IEEE Sens J PP(99):1–1
4. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

5. Cui C, Asari VK (2013) Adaptive weighted local textural features for illumination, expression, and occlusion invariant face recognition. In: Imaging& multimedia analytics in a web& mobile world
6. Cui W, Yan X (2009) Adaptive weighted least square support vector machine regression integrated with outlier detection and its application in qsar. Chemom Intell Lab Syst 98(2):130–135
7. Do H, Kalousis A, Hilario M (2009) Feature weighting using margin and radius based error bound optimization in svms. In: European conference on machine learning& knowledge discovery in databases
8. Fan Y, Tian F, Qin T, Bian J, Liu TY (2017) Learning what data to learn
9. Gao H, Zhuang L, Maaten LVD, Weinberger KQ (2017) Densely connected convolutional networks. In: IEEE Conference on computer vision& pattern recognition
10. Han X, Jinjun W, Ziku W, Guofeng L, Yan W, Juan L (2018) Learning solutions to two dimensional electromagnetic equations using ls-svm. Neurocomputing pp S0925231218305,873–
11. Hu WJ, Song Q (2003) An accelerated decomposition algorithm for robust support vector machines. Circuits Syst II Exp Briefs IEEE Trans 51(5):234–240
12. Kwak N, Choi CH (2002) Input feature selection by mutual information based on parzen window. IEEE T Pattern Anal 24(12):1667–1671
13. Lin CF, de Wang S (2004) Training algorithms for fuzzy support vector machines with noisy data. Pattern Recogn Lett 25(14):1647–1656
14. Lin CF, Wang SD (2002) Fuzzy support vector machines. IEEE T Neural Networ 13(2):464–471
15. Liu Y, Wen K, Gao Q, Gao X, Nie F (2018) Svm based multi-label learning with missing labels for image annotation. Pattern Recognition 78, S0031320318300, 372
16. Min H, Li K, Wang X, Ren F (2015) Facial expression recognition based on histogram weighted hcbp. Journal of Electronic Measurement & Instrumentation
17. Phan AV, Nguyen ML, Bui LT (2016) Feature weighting and svm parameters optimization based on genetic algorithms for classification problems. Appl Intell 46(2):1–15
18. Principe JC (2010) Information theoretic learning renyi¡¯s entropy and kernel perspectives
19. Ren M, Zeng W, Yang B, Urtasun R (2018) Learning to reweight examples for robust deep learning
20. Shen XJ, Dong Y, Gou JP, Zhan YZ, Fan J (2018) Least squares kernel ensemble regression in reproducing kernel hilbert space
21. Shivaswamy PK, Jebara T (2010) Maximum relative margin and data-dependent regularization. J Mach Learn Res 11(1):747–788
22. Shuang C, Partridge D (2004) Feature ranking and best feature subset using mutual information. Neural Comput Appl 13(3):175–184
23. Song Q, Hu W, Xie W (2002) Robust support vector machine with bullet hole image classification. IEEE Trans Syst Man Cybern 32(4):440–448
24. Torkkola K (2003) Feature extraction by non-parametric mutual information maximization. J Mach Learn Res 3(3):1415–1438
25. Wang M, Yang J, Liu GP, Xu ZJ, Chou KC (2004) Weighted-support vector machines for predicting membrane protein types based on pseudo-amino acid composition. Protein Eng Des Sel 17(6):509–516
26. Wu J, Yang H (2017) Linear regression-based efficient svm learning for large-scale classification. IEEE T Neur Net Learn 26(10):2357–2369
27. Wu X, Zuo W, Lin L, Jia W, Zhang D (2018) F-svm: Combination of feature transformation and svm learning via convex relaxation. IEEE T Neur Net Lear PP(99):1–15
28. Xing HJ, Ha MH, Tian DZ, Hu BG (2008) A novel support vector machine with its features weighted by mutual information. In: IEEE International joint conference on neural networks
29. Yang X, Song Q, Cao A (2007) Weighted support vector machine for data classification. In: IEEE International joint conference on neural networks
30. Yu H, He F, Pan Y (2019) A scalable region-based level set method using adaptive bilateral filter for noisy image segmentation. Multimedia Tools Applications 79(10)
31. Yu J, Hong C, Rui Y, Tao D (2017) Multi-task autoencoder model for recovering human poses. IEEE T Ind Electron PP(99):1–1

32. Zhang J, He F, Chen Y (2019) A new haze removal approach for sky/river alike scenes based on external and internal clues. Multimedia Tools and Applications (20)
33. Zhang Q, Dong L, Fan Z, Ying L, Li Z (2011) Feature and sample weighted support vector machine
34. Zhang S, He F (2019) Drcdn: learning deep residual convolutional dehazing networks. Vis Comput, pp 1–12
35. Zhang X (1999) Using class-center vectors to build support vector machines. In: Neural networks for signal processing ix, IEEE signal processing society workshop