Image Steganalysis in High-Dimensional Feature Spaces with Proximal Support Vector Machine

Ping Zhong[a], Mengdi Li[b], Juan Wen[b], Yiming Xue[b,1]

[a] *College of Science, China Agricultural University, Beijing, 100083, China*

[b] *College of Information and Electrical Engineering, China Agricultural University, Beijing, 100083, China*

Abstract It has been proved that if correctly regularized, a simple Fisher Linear Discriminant (FLD) or a ridge regression can achieve good detection accuracy for the adaptive image steganographic schemes. In this paper, we present the linear Proximal Support Vector Machine (PSVM) to the image steganalysis. Moreover, we implement the linear PSVM by using the state-of-the-art optimization method Least Square Minimum-Residual (LSMR), and generate a more efficient method named as PSVM-LSMR. In addition, motivated by the Extreme Learning Machine (ELM), we propose a nonlinear PSVM named as PSVM-ELM based on ELM kernel matrix to the image steganalysis. It demonstrates that the linear PSVM can achieve comparable performance with the FLD and ridge regression, and for the large feature sets, its computational time is far more less than that of the FLD and ridge regression. The PSVM-LSMR can further improve the detection accuracy and the training speed of the PSVM, and the performance of it is comparable to that of the Ridge Regression implemented by LSMR (RR-LSMR). Both the PSVM-LSMR and RR-LSMR require the least computational time among all the competitions when dealing with medium or large feature sets. The nonlinear classifier PSVM-ELM has been shown to perform comparably or even better than the FLD and ridge regression for the spatial domain steganographic schemes, and its computational time is apparently less than that of the FLD and ridge regression for the large feature sets. All claims are supported by the experiments with the wide stego schemes and rich steganalysis feature sets in both the spatial and JPEG domains.

Keywords: Proximal support vector machine, extreme learning machine kernel matrix, steganalysis, steganography.

[1]Corresponding author: xueym@cau.edu.cn (Yiming Xue)

1 Introduction

Image steganography is an important convert communication technology that conceals secret messages in images by the means of slight changes in pixel values or DCT coefficients. Currently, the most secure steganographic algorithms are content-adaptive ones, such as HUGO (Pevný et al., 2010), UNIWARD (Holub & Fridrich, 2014), WOW (Holub & Fridrich, 2012) and so on. They tend to hide the secret data in the complicated texture regions and show the excellence anti-detection ability.

With the rapid development of image steganography, steganalysis techniques that are related to detecting the existence of the hidden messages in images also have made great progress. The popular methods consist of extracting the relevant features that help to detect the presence of hidden message, and then designing suitable classifier to separate the classes of cover and stego images. In order to improve the detection performance, the feature dimensions are ever-increasing. As is well known, the state-of-the-art steganalysts are the Spatial Rich Model (SRM) (Fridrich & Kodovskỳ, 2012) and its variants (Holub & Fridrich, 2013; Denemark et al., 2014), which may contain more than 30,000 features. For the large scale and high dimensional training sets, it has been shown that the FLD, ridge regression (Cogranne et al., 2015), and FLD ensemble classifier (Kodovskỳ et al., 2012) are successful classifiers. However, for the widely popular linear Support Vector Machine (SVM) and Gaussian SVM, they are difficult to be trained after the presence of rich media models (Kodovskỳ et al., 2012). Compared with the FLD and ridge regression, the major reason for the difficulty in training these standard SVMs is that they require considerably long computational time to solve a linear or a quadratic program involved. In addition, except the regularization parameter which these machine learning algorithms have, Gaussian SVM also needs to search for another optimal kernel parameter in the training process, and it is time consuming.

Quite different from the standard SVMs, the linear Proximal Support Vector Machine (PSVM) (Fung & Mangasarian, 2001) which has been proposed based on the much more generic regularization networks (Evgeniou et al., 2000) can be fast implemented without of extensive computation. The linear PSVM separates two classes of data points through proximal hyperplanes with the maximum margin. The strong convexity of the formulation leads to the simple proximal code, which is not always the case in the standard SVMs.

Motivated by the PSVM, a simplified nonlinear method referred to Extreme Learning Machine (ELM) (Huang et al., 2012) has been presented for learning single hidden layer feedforward neural networks. ELM has the ability of dealing with the nonlinear feature construction by ELM kernel matrix without the selection of parameters (Huang et al., 2015).

All the methods including the regularized FLD, ridge regression, and the linear PSVM can be interpreted as the least square estimations with l_2 regularization (the regularized FLD corresponds to the ridge regression when the features have zero mean(Cogranne et al., 2015)). The fact that a simple FLD classifier or a ridge regression can achieve good detection accuracy (Cogranne et al., 2015) motivates this paper. It is reasonable to study other regularization methods and make a comparison among their performance. In particular, the regularized FLD and ridge regression get their solutions by inverting a matrix of size $d \times d$ (where d is the number of features), while the linear PSVM can get the solution by inverting an equal or even smaller size of matrix. The potential benefit is reducing training complexity and improving the efficiency. Furthermore, the linear PSVM can be implemented by the fast optimization method LSMR[2] (Fong & Saunders, 2011) to substantially improve the computational efficiency. In addition, we propose a nonlinear PSVM, called PSVM-ELM, based on the ELM kernel matrix by combining the merits of the linear PSVM and ELM. The experiments show that the detection accuracy of the linear PSVM is comparable to that of the FLD and ridge regression, and its computational time is far less than that of the FLD and ridge regression when dealing with the large feature sets. The PSVM-LSMR can further improve the PSVM on both the detection accuracy and computational time, and the performance of it is comparable to that of the RR-LSMR. Both of them require the least computational time among all the competitions when dealing with medium or large feature sets. In addition, it is shown that the detection accuracy of the nonlinear classifier PSVM-ELM is rather good for the spatial domain steganographic schemes. Also, the PSVM-ELM is trained much more efficiently than the FLD and ridge regression on the large feature sets.

The paper is organized as follows. In Section 2, we briefly introduce the regularized FLD and ridge regression. And then the linear PSVM, PSVM-LSMR, and PSVM-ELM are described in Section 3. The experimental results are shown in Section 4, which contains

[2]LSMR function can be downloaded from Stanford University's Systems Optimization Laboratory.

wide set of steganographic schemes and with various steganalysis features. Section 5 concludes the paper.

2 Regularized FLD and Ridge Regression

In this section, we give a brief review of the regularized FLD and ridge regression. More details can be seen in (Cogranne et al., 2015).

Regularized FLD is a classical classification approach. Denote $\boldsymbol{x} \in \mathbb{R}^d$ a row vector of d features extracted from an image. Let

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{C}_{trn} \\ \boldsymbol{S}_{trn} \end{pmatrix} \tag{1}$$

be the matrix of all training samples, where $\boldsymbol{C}_{trn} \in \mathbb{R}^{N_{trn} \times d}$ and $\boldsymbol{S}_{trn} \in \mathbb{R}^{N_{trn} \times d}$ denote the training sets of cover and stego images with their row vectors $\boldsymbol{C}_{trn}^{(i)}$ and $\boldsymbol{S}_{trn}^{(i)}$, respectively. Let the means of two classes be row vectors of size $1 \times d$ denoted $\boldsymbol{m}_c$ and $\boldsymbol{m}_s$, the covariance matrices of two classes be matrices of size $d \times d$ denoted $\boldsymbol{\Sigma}_c$ and $\boldsymbol{\Sigma}_s$, respectively. The regularized FLD yields the separating hyperplane by maximizing the Fisher ratio with the l_2 regularization:

$$\boldsymbol{w} = \arg\min J(\boldsymbol{w})^{-1} + \lambda \|\boldsymbol{w}\|_2^2 \tag{2}$$

where $J(\boldsymbol{w})$ is the Fisher ratio:

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}(\boldsymbol{m}_c - \boldsymbol{m}_s)^T(\boldsymbol{m}_c - \boldsymbol{m}_s)\boldsymbol{w}^T}{\boldsymbol{w}(\boldsymbol{\Sigma}_c + \boldsymbol{\Sigma}_s)\boldsymbol{w}^T} \tag{3}$$

From the training data $\boldsymbol{C}_{trn}$ and $\boldsymbol{S}_{trn}$, the weight vector can be obtained as

$$\boldsymbol{w} = (\widehat{\boldsymbol{m}}_s - \widehat{\boldsymbol{m}}_c)(\widehat{\boldsymbol{\Sigma}}_s + \widehat{\boldsymbol{\Sigma}}_c + \lambda \boldsymbol{I}_d)^{-1} \tag{4}$$

with $\widehat{\boldsymbol{m}}_s = \dfrac{1}{N_{trn}} \sum_{i=1}^{N_{trn}} \boldsymbol{S}_{trn}^{(i)}$, $\widehat{\boldsymbol{m}}_c = \dfrac{1}{N_{trn}} \sum_{i=1}^{N_{trn}} \boldsymbol{C}_{trn}^{(i)}$, $\widehat{\boldsymbol{\Sigma}}_s = \dfrac{1}{N_{trn}-1} \sum_{i=1}^{N_{trn}} (\boldsymbol{S}_{trn}^{(i)} - \widehat{\boldsymbol{m}}_s)^T (\boldsymbol{S}_{trn}^{(i)} - \widehat{\boldsymbol{m}}_s)$, $\widehat{\boldsymbol{\Sigma}}_c = \dfrac{1}{N_{trn}-1} \sum_{i=1}^{N_{trn}} (\boldsymbol{C}_{trn}^{(i)} - \widehat{\boldsymbol{m}}_c)^T (\boldsymbol{C}_{trn}^{(i)} - \widehat{\boldsymbol{m}}_c)$.

Ridge Regression is also known as Tikhonov regularization based on least square estimation. Let $\boldsymbol{y} \in \mathbb{R}^{2N_{trn}}$ be the label vector of training samples from matrix $\boldsymbol{X}$:

$$\boldsymbol{y} = (\overbrace{-1, -1, \cdots, -1}^{N_{trn}}, \overbrace{1, 1, \cdots, 1}^{N_{trn}})^T \tag{5}$$

The goal of the ridge regression is to seek for a weight vector $\widehat{\boldsymbol{w}}$ by minimizing the squared loss between the real labels and estimated values:

$$\widehat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \|\boldsymbol{X}\boldsymbol{w}^T - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{w}\|_2^2 \tag{6}$$

The solution of (6) can be obtained as:

$$\widehat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}_d)^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{7}$$

It has been verified that a simple FLD with l_2 regularization or a ridge regression can achieve almost the same detection accuracy as the wisely used FLD ensemble classifier in steganalysis (Cogranne et al., 2015). With the aid of the advanced optimization algorithm LSMR, the computational time of ridge regression is 10 times smaller than an ensemble classifier. In addition, it has been shown by empirical study that the least square estimation with l_1 regularization, known as LASSO (Least Absolute Shrinkage and Selection Operator), performs worse than these two methods (Cogranne et al., 2015).

3 The Proposed Methods

As briefly shown in the above section, the least square estimations with l_2 regularization may achieve good performance in steganalysis. To this end, we study the following three methods.

Linear PSVM proposed by Fung and Mangasarian (2001) separates data points depending on proximity to one of two parallel separating hyperplanes that are pushed apart as far as possible. The separating hyperplanes are no longer the bounding planes but "proximal" planes, and the effect is reflected in the optimization problem that the inequality constraints of standard SVMs are replaced with equality constraints. In addition, different from standard SVMs, in the objective function of linear PSVM, the square term of bias has been added, which makes the optimization problem strongly convex.

Let us define a diagonal matrix $\boldsymbol{Y} \in \mathbb{R}^{2N_{trn}\times 2N_{trn}}$ with N_{trn} minus ones and N_{trn} plus ones along its diagonal to represent the class of samples from matrix $\boldsymbol{X}$. The distance between two proximal planes is $2/\|[\boldsymbol{w}, b]\|_2$. By maximizing margin and minimizing training

error, the linear PSVM is formulated as

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}(\|\boldsymbol{w}\|_2^2 + b^2) + \frac{\lambda}{2}\boldsymbol{\xi}\boldsymbol{\xi}^T \\ \text{s.t.} \quad & \boldsymbol{Y}(\boldsymbol{X}\boldsymbol{w}^T + b\boldsymbol{e}) + \boldsymbol{\xi}^T = \boldsymbol{e} \end{aligned} \tag{8}$$

where $\boldsymbol{e}$ is a column vector of ones of $2N_{trn}$ dimension; $\boldsymbol{\xi}$ is the slack vector of size $1 \times 2N_{trn}$.

Denote $\overline{\boldsymbol{w}} = [\boldsymbol{w}, b]$ and $\overline{\boldsymbol{X}} = [\boldsymbol{X}, \boldsymbol{e}]$. By replacing the objective function with the constraints, we have

$$\min_{\overline{\boldsymbol{w}} \in \mathbb{R}^{d+1}} \frac{1}{2}\|\overline{\boldsymbol{w}}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{y} - \overline{\boldsymbol{X}}\overline{\boldsymbol{w}}^T\|_2^2 \tag{9}$$

where $\boldsymbol{y}$ is defined in (5). The formula (9) can be viewed as the least square estimation with l_2 regularization in $(\boldsymbol{w}, b)$ space of $\mathbb{R}^{d+1}$.

By Karush-Kuhn-Tucker (KKT) theorem and denote $\boldsymbol{G} = \boldsymbol{Y}[\boldsymbol{X}, \boldsymbol{e}]$, we can solve the optimization problem (8) as follows: When the number of training samples is smaller than the dimension of samples (i.e., $2N_{trn} < d$), the solution is

$$\boldsymbol{w} = \boldsymbol{e}^T(\frac{1}{\lambda}\boldsymbol{I}_{2N_{trn}} + \boldsymbol{G}\boldsymbol{G}^T)^{-1}\boldsymbol{Y}\boldsymbol{X} \tag{10}$$

$$b = \boldsymbol{e}^T(\frac{1}{\lambda}\boldsymbol{I}_{2N_{trn}} + \boldsymbol{G}\boldsymbol{G}^T)^{-1}\boldsymbol{Y}\boldsymbol{e} \tag{11}$$

In contrast, if the number of training samples is lager than the dimension (i.e., $2N_{trn} > d$), by Sherman-Morrison-Woodbury (SMW) formula, we have

$$\boldsymbol{w} = \lambda\boldsymbol{e}^T[\boldsymbol{I}_{2N_{trn}} - \boldsymbol{G}(\frac{1}{\lambda}\boldsymbol{I}_{d+1} + \boldsymbol{G}^T\boldsymbol{G})^{-1}\boldsymbol{G}^T]\boldsymbol{Y}\boldsymbol{X} \tag{12}$$

$$b = \lambda\boldsymbol{e}^T[\boldsymbol{I}_{2N_{trn}} - \boldsymbol{G}(\frac{1}{\lambda}\boldsymbol{I}_{d+1} + \boldsymbol{G}^T\boldsymbol{G})^{-1}\boldsymbol{G}^T]\boldsymbol{Y}\boldsymbol{e} \tag{13}$$

It ensures by (10)-(13) that the solution can obtained by the inverse of the smaller matrix. The decision function of linear PSVM classifier is $f(\boldsymbol{x}) = \text{sign}(\boldsymbol{x}\boldsymbol{w}^T + b)$.

PSVM-LSMR. Note that the linear PSVM is the least square estimations which can be obtained by solving the appropriate system of linear equations. There exist a large numbers of efficient optimization algorithms for solving large linear systems. These algorithms are iterative and related to a stopping criterion either on the solution $\overline{\boldsymbol{w}}$ or the residual $\boldsymbol{y} - \overline{\boldsymbol{X}}\overline{\boldsymbol{w}}^T$. We use the state of the art optimization method LSMR to implement the linear PSVM as the LSMR has the advantages of low computational complexity and low memory requirements. We name the method of solving the linear PSVM with

LSMR as PSVM-LSMR. The PSVM-LSMR consists of two parameters—the regularization parameter λ and the tolerance used in LSMR, which controls the trade-off between computational time and detection accuracy. In the later experiments, we find that the regularization parameter λ is negligible sensitivity to the best detection accuracy, so we fix $\lambda = 10^{-8}$ and search for the optimal tolerance.

PSVM-ELM. In the original nonlinear PSVM (Fung & Mangasarian, 2001), the classifier is given as

$$\widehat{\boldsymbol{\alpha}} = \boldsymbol{e}^T(\frac{1}{\lambda}\boldsymbol{I}_{2N_{trn}} + \widehat{\boldsymbol{G}}\widehat{\boldsymbol{G}}^{\boldsymbol{T}})^{-1} \tag{14}$$

where $\widehat{\boldsymbol{G}} = \boldsymbol{Y}[\boldsymbol{K}, \boldsymbol{e}]$ with $\boldsymbol{K}$ is a kernel matrix. Unlike the linear case, the SMW formula is useless because the kernel matrix $\boldsymbol{K}$ is a $2N_{trn}$ order square matrix. Although the reduced kernel (Lee & Mangasarian, 2001) can be used to reduce the computational complexity, the search of the optimal kernel parameter is still time consuming.

Motivated by the ELM, we propose a nonlinear PSVM through replacing $\boldsymbol{X}$ in (8) with the hidden layer output matrix $\boldsymbol{H}$:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}(\|\boldsymbol{w}\|_2^2 + b^2) + \frac{\lambda}{2}\boldsymbol{\xi}\boldsymbol{\xi}^T \\ \text{s.t.} \quad & \boldsymbol{Y}(\boldsymbol{H}\boldsymbol{w}^T + b\boldsymbol{e}) + \boldsymbol{\xi}^T = \boldsymbol{e} \end{aligned} \tag{15}$$

where

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}(\boldsymbol{C}_{trn}^{(1)}) \\ \vdots \\ \boldsymbol{h}(\boldsymbol{C}_{trn}^{(N_{trn})}) \\ \boldsymbol{h}(\boldsymbol{S}_{trn}^{(1)}) \\ \vdots \\ \boldsymbol{h}(\boldsymbol{S}_{trn}^{(N_{trn})}) \end{bmatrix} = \begin{bmatrix} h_1(\boldsymbol{C}_{trn}^{(1)}) & \cdots & h_L(\boldsymbol{C}_{trn}^{(1)}) \\ \vdots & \vdots & \vdots \\ h_1(\boldsymbol{C}_{trn}^{(N_{trn})}) & \cdots & h_L(\boldsymbol{C}_{trn}^{(N_{trn})}) \\ h_1(\boldsymbol{S}_{trn}^{(1)}) & \cdots & h_L(\boldsymbol{S}_{trn}^{(1)}) \\ \vdots & \vdots & \vdots \\ h_1(\boldsymbol{S}_{trn}^{(N_{trn})}) & \cdots & h_L(\boldsymbol{S}_{trn}^{(N_{trn})}) \end{bmatrix} \tag{16}$$

with $\boldsymbol{h}(\boldsymbol{x}) = [h_1(\boldsymbol{x}), \cdots, h_L(\boldsymbol{x})]$ being the nonlinear feature mapping with L hidden neurons.

According to the KKT optimality conditions, we have

$$\boldsymbol{\alpha}(\frac{1}{\lambda}\boldsymbol{I} + \boldsymbol{Y}(\boldsymbol{H}\boldsymbol{H}^T + \boldsymbol{e}\boldsymbol{e}^T)\boldsymbol{Y}) = \boldsymbol{e}^T \tag{17}$$

Denote $\widehat{\boldsymbol{H}} = \boldsymbol{Y}[\boldsymbol{H}, \boldsymbol{e}]$. If the number of training samples is smaller than the number of hidden neurons i.e., $2N_{trn} < L$, we have the solutions of $\boldsymbol{w}$ and b as follows:

$$\boldsymbol{w} = \boldsymbol{e}^T(\frac{1}{\lambda}\boldsymbol{I}_{2N_{trn}} + \widehat{\boldsymbol{H}}\widehat{\boldsymbol{H}}^T)^{-1}\boldsymbol{Y}\boldsymbol{H}, \qquad b = \boldsymbol{e}^T(\frac{1}{\lambda}\boldsymbol{I}_{2N_{trn}} + \widehat{\boldsymbol{H}}\widehat{\boldsymbol{H}}^T)^{-1}\boldsymbol{Y}\boldsymbol{e} \tag{18}$$

Otherwise, when $2N_{trn} > L$, by SMW formula, we have

$$\boldsymbol{w} = \lambda \boldsymbol{e}^T [\boldsymbol{I}_{2N_{trn}} - \widehat{\boldsymbol{H}}(\frac{1}{\lambda}\boldsymbol{I}_{L+1} + \widehat{\boldsymbol{H}}^T\widehat{\boldsymbol{H}})^{-1}\widehat{\boldsymbol{H}}^T]\boldsymbol{Y}\boldsymbol{H} \tag{19}$$

$$b = \lambda \boldsymbol{e}^T [\boldsymbol{I}_{2N_{trn}} - \widehat{\boldsymbol{H}}(\frac{1}{\lambda}\boldsymbol{I}_{L+1} + \widehat{\boldsymbol{H}}^T\widehat{\boldsymbol{H}})^{-1}\widehat{\boldsymbol{H}}^T]\boldsymbol{Y}\boldsymbol{e} \tag{20}$$

The decision function of PSVM-ELM classifier is $f(\boldsymbol{x}) = \text{sign}(\boldsymbol{h}(\boldsymbol{x})\boldsymbol{w}^T + b)$.

4 Numerical Results

In this paper, the experiments are conducted on BOSSbase 1.01 (Bas et al., 2011), which consists of 10,000 512×512 gray-scale images widely used in steganalysis. Several state-of-the-art steganographic algorithms both in spatial domain and JPEG domain are applied to estimate the detection accuracy of the involved steganalyzers. Table 1 summarizes these embedding algorithms, feature extractors, the number of features (#Feat.), and payload[3].

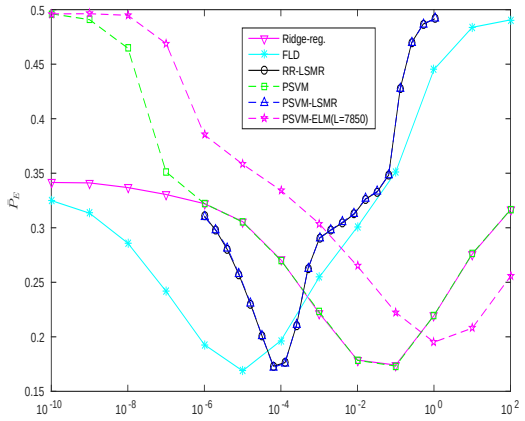
Table 1. The involved steganographic algorithms and feature extractors.

No.	Embedding algorithm	Feature extractor	#Feat.	payload
1	HUGO-BD (Pevný et al., 2010)	SPAM (Pevný et al., 2010)	686	0.2
2	S-UNIWARD (Holub&Fridrich, 2014)	SRMQ1 (Fridrich&Kodovskỳ, 2012)	12,753	0.2
3	WOW (Holub&Fridrich, 2012)	SRM (Fridrich& Kodovskỳ, 2012)	34,671	0.2
4	MiPOD (Sedighi et al., 2015)	maxSRMd2 (Denemark et al., 2014)	34,671	0.2
5	UED (Guo et al., 2012)	$\mathcal{CF}^*$ (Kodovskỳ et al., 2012)	7,850	0.3
6	UED (Guo et al., 2012)	PHARM (Holub&Fridrich, 2015a)	12,600	0.2
7	SI-UNIWARD (Holub&Fridrich, 2014)	DCTR (Holub&Fridrich, 2015b)	8,000	0.4

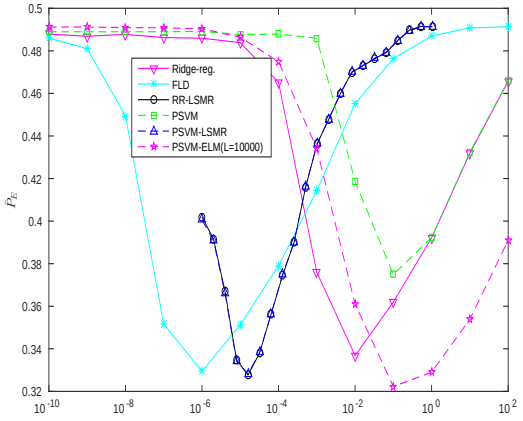
In the experiments, the detection accuracy is measured by $P_E = 1/2(P_{FA} + P_{MD})$, where P_{FA} and P_{MD} are the empirical probability of false alarm and missed detection, respectively. The detection accuracy is averaged over 10 splits on the testing set (a 50/50 split for training and testing was employed). We notice that both PSVM-LSMR and RR-LSMR need to search for two parameters including the regularization parameter λ and the tolerance used in LSMR. In addition, the PSVM-ELM requires to find two parameters—the regularization parameter λ and the hidden neurons L. Along with L increasing, the

[3]All feature extractors and embedding algorithms used are downloaded from the DDE website at http://dde.binghamton.edu/download.

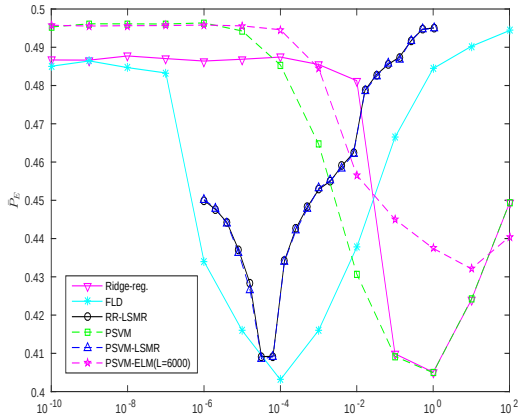
detection accuracy generally becomes better. However, the computational time increases too. The implementation has to be a trade-off between computational time and detection accuracy.



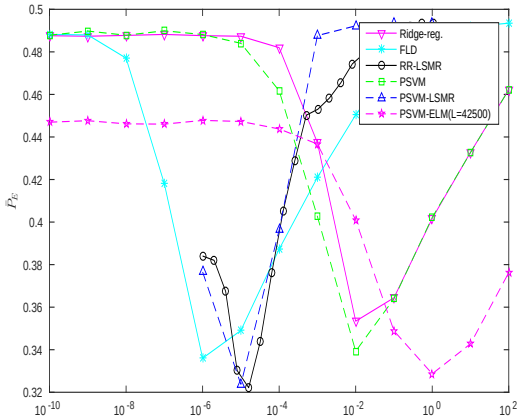
(a) $\mathcal{CF}^*$ features with UED embedding scheme, payload $R = 0.3$



(b) SRMQ1 features with S-UNIWARD embedding scheme, payload $R = 0.2$



(c) DCTR features with SI-UNIWARD embedding scheme, payload $R = 0.4$



(d) SRM features with WOW embedding scheme, payload $R = 0.2$

Figure 1 Values of P_E along with the regularization parameter λ (or tolerance of LSMR) for several embedding schemes and feature sets.

Figure 1 shows the detection accuracy of the FLD, ridge regression, RR-LSMR, and the proposed classifiers. The detection accuracy P_E can be regarded as a function of the regularization parameter λ. Notice that for the RR-LSMR and PSVM-LSMR, the toler-

Table 2. Detection accuracy of the FLD, ridge regression (Ridge-reg), and RR-LSMR compared with the proposed classifiers for various embedding algorithms listed in Table 1. Detection accuracy is measured as total probability of error P_E.

No.	FLD	Ridge-reg	RR-LSMR	PSVM	PSVM-ELM	PSVM-LSMR
1	.4376±.0025	.4379±.0022	.4372±.0030	.4373±.0016	.4191±.0029 (L=1200)	.4441±.0021
2	.3344±.0032	.3406±.0038	.3343±.0032	.3664±.0058	.3276±.0112 (L=10000)	.3325±.0028
3	.3301±.0048	.3375±.0049	.3263±.0029	.3374±.0037	.3335±.0034 (L=42500)	.3277±.0033
4	.3505±.0031	.3553±.0034	.3448±.0032	.3566±.0033	.3505±.0034 (L=42500)	.3444±.0033
5	.1720±.0033	.1787±.0030	.1739±.0023	.1777±.0020	.1970±.0029 (L=7850)	.1732±.0030
6	.4871±.0040	.4950±.0018	.1969±.0137	.3741±.0107	.4305±.0044 (L=12600)	.1924±.0121
7	.4092±.0021	.4114±.0027	.4311±.0036	.4124±.0027	.4369±.0027 (L=6000)	.4310±.0018

ance that controls the stopping criterion of the iterative method was searched since the regularization parameter λ has negligible influence for these two methods. Four different embedding algorithms with different feature sets are illustrated in Figure 1. It can be observed that almost all of the optimal parameters λ lie in $(10^{-10}, 10^{0})$, and we find the best regularization parameter by five-fold cross validation in this range. We also notice that the curves of RR-LSMR and PSVM-LSMR are almost overlapped, and these two methods achieve the best accuracy for tolerance $\approx 10^{-5}$. Since the RR-LSMR and PSVM-LSMR are negligible sensitivity to the regularization parameter λ, we fixed $\lambda = 10^{-8}$ and found the best tolerance from $(10^{-5}, 10^{0})$.

Table 2 lists the detection accuracy of the FLD, ridge regression, RR-LSMR, PSVM, PSVM-ELM, and PSVM-LSMR. All these classifiers were employed with the best regularization parameter λ (or tolerance). It is shown from Table 2 that the linear classifier PSVM performs comparably with the linear classifiers FLD and ridge regression. For the nonlinear classifiers PSVM-ELM, it performs very well for spatial domain steganographic schemes. Especially, it has achieved better detection accuracy than ridge regression for spatial domain steganographic schemes. The performance of the PSVM-LSMR is comparable with that of the RR-LSMR, and both of them have achieved good detection accuracy.

Table 3 shows the comparison results of the computational time of the same classifiers with the same settings. Notice that for feature sets of small or medium dimensionality (up to 8000 features), the FLD is faster than the linear PSVM. However, for the large feature sets, the computational time of the linear PSVM becomes much less than that of FLD.

Table 3. Computation time (in seconds) of the FLD, ridge regression(Ridge-reg), and RR-LSMR compared with the proposed classifiers, same settings as in Table 2. Computations were carried out on a 14-physical-core Intel®Xcon®E5 @ 1.90GHz with RAM 256GB.

No.	FLD	Ridge-reg	RR-LSMR	PSVM	PSVM-ELM	PSVM-LSMR
1	0.1445	1.0125	0.3318	0.2309	0.7880	0.3295
2	49.0227	110.4050	5.3407	21.8980	29.3200	5.3228
3	642.8982	1155.5	18.1730	29.2619	109.1115	18.0362
4	627.5492	1118.7	16.2918	28.9551	109.8685	15.9670
5	18.4487	72.5077	9.6514	20.8580	26.2609	9.5078
6	103.7390	125.3901	5.4252	42.2570	35.6234	5.8454
7	23.7571	61.0150	4.1007	25.4811	18.8528	4.2289

Compared with the ridge regression, the linear PSVM always needs less computational time. Especially, when dealing with the large feature sets, the linear PSVM requires far more less computational time than the ridge regression. In fact, both the FLD and ridge regression have to compute the inversion of a very large matrix of size $d \times d$, which has the complexity $\mathcal{O}(d^3)$. But for the linear PSVM, for comparison, it has a complexity $\mathcal{O}(\min(2N_{trn}, d+1)^3)$. It is also shown that for larger feature sets, the computational time of the nonlinear classifier PSVM-ELM is much less than that of the FLD and ridge regression, and it has the complexity of $\mathcal{O}(\min(2N_{trn}, L+1)^3)$. In addition, the computational time of the PSVM-LSMR is comparable to that of RR-LSMR, and both of them require the least computational time among all the competitors.

5 Conclusion

In this paper, we have proposed three algorithms including the linear PSVM, PSVM-LSMR, and the nonlinear PSVM-ELM for the image stegoanalysis. It has shown that the linear PSVM can achieve the comparable detection accuracy with the FLD and ridge regression, and the computational time of it is far more less than that of the FLD and ridge regression when dealing with the large feature sets. The performance of PSVM-LSMR is comparable to that of the RR-LSMR, and both of them obtain the good detection accuracy with the least computational time among all the competing steganalysts. In addition, the nonlinear classifier PSVM-ELM has been shown to achieve comparable or

even better detection accuracy than that of the FLD and ridge regression for the spatial domain steganographic schemes, and its computational time is much less than that of the FLD and ridge regression for the large feature sets. Further work includes further improving the detection accuracy and reducing the computational complexity.

Acknowledgments

The work is supported by NSFC U1536121.

References

[1] Pevný, T., Filler, T., & Bas, P. (2010). Using high-dimensional image models to perform highly undetectable steganography. In *Infromaton Hidding ser. LNCS*, vol. 6387, pp. 161-177.

[2] Holub, V., & Fridrich, J. (2014). Universal distortion design for Steganography in an Arbitrary Domain. *EURASIP journal on Information Security*, vol. 2014, no. 1, pp. 1-13.

[3] Holub, V., & Fridrich, J. (2012). Designing steganographic distortion using directional filters. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 234-239.

[4] Fridrich, J., & Kodovskỳ, J. (2012). Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868-882.

[5] Holub, V., & Fridrich, J. (2013). Random projections of residuals for digital image steganalysis. *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 1996-2006.

[6] Denemark,T., Sedighi,V., Holub, V., Cogranne, R., & Fridrich, J. (2014). Seletion-channel-aware rich model for steganalysis of digital images. In *International Workshop on Information Forensics and Security (WIFS)*, pp. 48-53.

[7] Kodovskỳ, J., Fridrich, J., & Holub, V. (2012). Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432-444.

[8] Cogranne, R., Sedighi, V., Fridrich, J., & Pevný, T. (2015). Is ensemble classifier needed for steganalysis in high-dimensional feature spaces? *IEEE International Workshop on Information Forensics & security*, pp. 1-6.

[9] Fong, D. C.-L., & Saunders, M. (2011). LSMR: An iterative algorithm for sparse least-squares problems. *SIAM Journal on Scientific Compution*, vol. 33, no. 5, pp. 2950-2971.

[10] Fung, G., & Mangasarian, O. L. (2001). Proximal support vector machine classifiers. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, San Franciso, CA, 2001, pp. 77-86.

[11] Evgeniou, T., Pontil, M., & Poggio, T. (2000). Regularization networks and support vector machines. *Advances in Computational Mathematics*, vol. 13, pp. 1-50.

[12] Huang, G. B., Zhou, H.M., Ding, X.J., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics–part B: Cybernetics*, vol. 42, no. 2, pp. 513-529.

[13] Huang, G., Huang, G. B., Song, S.J., & You, K.Y. (2015). Trends in extreme learning machines: a review. *Neural Networks*, vol. 61, pp. 32-48.

[14] Lee, Y.-J., & Mangasarian, O. L. (2001). RSVM: Reduced support vector machines. In *Proceedings of the First SIAM International Conference on Data Mining*, Chicago, April 5-7.

[15] Bas, P., Filler, T., & Pevný, T. (2011). Break our steganographic system: The ins and outs of orgnizing boss. In *International Workshop on Infoemation Hiding*, pp. 59-70.

[16] Pevný, T. , Bas, P., & Fridrich, J.(2010). Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215-224.

[17] Sedighi, V., Fridrich, J., & Cogranne, R. (2015). Content-adaptive pentary steganography using the multivariate generalized gaussian cover model. *IS&T/SPIE Electronic Imaging conf.*, vol. 9409, pp. 94090H.

[18] Guo, L., Ni, J., & Shi, Y. Q. (2012). An efficient JPEG steganographic scheme using uniform embedding. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 169-174.

[19] Wang, C., & Ni, J. (2012). An efficient JPEG steganographic scheme based on the block entroy of DCT coefficients. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1785-1788.

[20] Holub, V., & Fridrich, J. (2015a). Phase-aware projection model for steganalysis of JPEG images. In *IS&T/SPIE Electronic Imaging conf.*, vol. 9409, pp. 94090T.

[21] Holub, V., & Fridrich, J. (2015b). Low-complexity features for JPEG steganalysis using undecimated DCT. *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 219-228.

[22] Fong D.C.-L. and Saunders M. (2011). LSMR: An iterative algorithm for sparse least-squares problems. *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2950-2971.