Is Ensemble Classifier Needed for Steganalysis in High-Dimensional Feature Spaces?

Rémi Cogranne
ICD - LM2S - UMR 6281 CNRS
Troyes University of Technology
Troyes, France
remi.cogranne@utt.fr

Vahid Sedighi and Jessica Fridrich
Department of ECE
Binghamton University
Binghamton, NY 13902-6000
{vsedigh1,fridrich}@binghamton.edu

Tomáš Pevný,
Department of Computers
Czech Technical University in Prague
Praha 6,166 27, Czech Republic
Tomas.Pevny@agents.fel.cvut.cz

***Abstract*—The ensemble classifier, based on Fisher Linear Discriminant base learners, was introduced specifically for steganalysis of digital media, which currently uses high-dimensional feature spaces. Presently it is probably the most used method to design supervised classifier for steganalysis of digital images because of its good detection accuracy and small computational cost. It has been assumed by the community that the classifier implements a non-linear boundary through pooling binary decision of individual classifiers within the ensemble. This paper challenges this assumption by showing that linear classifier obtained by various regularizations of the FLD can perform equally well as the ensemble. Moreover it demonstrates that using state of the art solvers linear classifiers can be trained more efficiently and offer certain potential advantages over the original ensemble leading to much lower computational complexity than the ensemble classifier. All claims are supported experimentally on a wide spectrum of stego schemes operating in both the spatial and JPEG domains with a multitude of rich steganalysis feature sets.**

***Index Terms*—Ensemble classifier, linear classifier, regularization, steganalysis, steganography.**

I. Introduction

The objective of steganography is to hide a secret message within an innocuous looking cover object, such as a digital image, obtaining thus a stego object that can be sent overtly through an insecure channel. The related field that aims at detecting the presence of the hidden message is called steganalysis. Both fields have experienced a rapid development during the previous two decades [1].

Currently, the best detectors for modern steganographic methods are based on supervised learning; the general approach consists of selecting a suitable set of features that can reveal the presence of hidden data, and then training a classifier using supervised machine learning, to distinguish between the classes of cover and stego features. To improve the detection accuracy of modern steganographic schemes, the feature dimension had to be substantially increased: recently proposed rich models may contain more than 30.000 features [4], [5]. This feature dimensionality (together with correspondingly large training sets) make it difficult to train the widely popular Gaussian Support Vector Machine (SVM), a popular choice among steganalysts before the introduction of rich media models.

The FLD ensemble classifier [6] has been proposed as a much more scalable alternative for classifier construction with large training sets and high dimensional feature spaces. This classifier is non-linear because of its employment of the majority voting rule. Recently, it has recently been proposed [7] to replace the majority voting with a statistical test optimal within a multivariate Gaussian model of the base learners' projections to achieve a better control over the error rates and to extend the ensemble training to unknown payload and multi-class steganalysis [8]. However, by doing so, the ensemble became a linear classifier. For binary classification measured using the classical minimal total classifier error under equal priors, the performance of this version of the ensemble was shown to be essentially the same as for the original ensemble. This observation motivated the current paper.

As recognized in [16], sub-space sampling in FLD ensemble can be viewed as a way to regularize the linear classifier. Thus, it makes sense to study other regularizations and compare their performance with the ensemble. The potential benefit is simplifying the training and lowering its complexity and potentially improving the performance when merging qualitatively different feature spaces of highly unequal dimension.

The present paper is organized as follows. In Section II, we provide a brief description of the original FLD ensemble classifier as well as its linear form cast within hypothesis testing theory. Then, in Section III introduces several approaches for regularizing linear classifiers as alternatives to the ensemble. The results of all numerical results appear in Section IV which includes wide spectrum of steganographic methods and with a diverse set of steganalysis features. Section V summarizes the present work and concludes the paper.

II. FLD Ensemble Classifier

To explain the contribution of this paper, in this section we review the original FLD ensemble as well as its recent linear reformulation. We use the following notational conventions. Matrices will be represented with capital bold letters $\mathbf{X}$, vectors are denoted with lower case bold letters $\mathbf{x}$, scalars with lower case letters x, and sets and probability distributions with calligraphic capital letters $\mathcal{X}$. The FLD ensemble classifier was originally proposed [6] as an alternative to support vector machines, a scalable machine learning tool that can be efficiently used to build accurate detectors in high dimensional feature spaces and large training data sets. Since the FLD is a well-known tool, it is only briefly described in this section. The reader is referred to [9] for a more detailed presentation.

Let $\mathbf{f} \in \mathbb{R}^d$ be a (row) vector of d features extracted from one image. Let the training sets of cover and stego image features be matrices of size $N^{\rm trn} \times d$ denoted $\mathbf{C}^{\rm trn}$ and $\mathbf{S}^{\rm trn}$ whose components are $c^{\rm trn}_{n,i}$ and $s^{\rm trn}_{n,i}$, respectively. The FLD assumes that among these two classes, the features are i.i.d. with means $\boldsymbol{\mu}_c$ and $\boldsymbol{\mu}_s$, row vectors of size $1 \times d$, and covariance matrices $\boldsymbol{\Sigma}_c$ and $\boldsymbol{\Sigma}_s$ of size $d \times d$. Among all linear decision rules defined by:

$$\mathcal{C}: \begin{cases} \mathcal{H}_0 \ \text{ if } \ \mathbf{f}\cdot\mathbf{w}^{\rm T} - b < 0 \\ \mathcal{H}_1 \ \text{ if } \ \mathbf{f}\cdot\mathbf{w}^{\rm T} - b > 0, \end{cases} \tag{1}$$

where $\mathbf{f}$ is a feature (row) vector to be classified and b is a threshold, the FLD finds the vector of weights $\mathbf{w} \in \mathbb{R}^d$ that maximizes the following Fisher separability criterion (also called Fisher ratio):

$$F(\mathbf{w}) = \frac{\mathbf{w}(\boldsymbol{\mu}_c - \boldsymbol{\mu}_s)^{\rm T}(\boldsymbol{\mu}_c - \boldsymbol{\mu}_s)\mathbf{w}^{\rm T}}{\mathbf{w}(\boldsymbol{\Sigma}_c + \boldsymbol{\Sigma}_s)\mathbf{w}^{\rm T}}. \tag{2}$$

Few calculations show that the maximization of $F(\mathbf{w})$ from the training data, $\mathbf{C}^{\rm trn}$ and $\mathbf{S}^{\rm trn}$, leads to the following weighting vector $\mathbf{w}$:

$$\mathbf{w} = (\widehat{\boldsymbol{\mu}}_s - \widehat{\boldsymbol{\mu}}_c)\left(\widehat{\boldsymbol{\Sigma}}_c + \widehat{\boldsymbol{\Sigma}}_s\right)^{-1} \tag{3}$$

$$\text{with } \ \widehat{\mu}_{c_i} = \frac{1}{N_{\rm trn}}\sum_{n=1}^{N^{\rm trn}} c^{\rm trn}_{n,i}\ ,\ \widehat{\mu}_{s_i} = \frac{1}{N_{\rm trn}}\sum_{n=1}^{N^{\rm trn}} s^{\rm trn}_{n,i}\ ,$$

$$\widehat{\Sigma}_{c_{n,i}} = \frac{1}{N_{\rm trn}-1}\sum_{n=1}^{N^{\rm trn}}(c^{\rm trn}_{n,i} - \widehat{\mu}_{c_i})(c^{\rm trn}_{n,j} - \widehat{\mu}_{c_j})\ ,$$

$$\text{and } \ \widehat{\Sigma}_{s_{n,i}} = \frac{1}{N_{\rm trn}-1}\sum_{n=1}^{N^{\rm trn}}(s^{\rm trn}_{n,i} - \widehat{\mu}_{s_i})(s^{\rm trn}_{n,j} - \widehat{\mu}_{s_j}).$$

In practice, the inversion of the "between class" covariance matrix, $\widehat{\boldsymbol{\Sigma}}_c + \widehat{\boldsymbol{\Sigma}}_s$, is seldom performed directly but almost always using a regularization by adding $\lambda\mathbf{I}_d$ to improve numerical stability: $\widehat{\boldsymbol{\Sigma}}_c + \widehat{\boldsymbol{\Sigma}}_s + \lambda\mathbf{I}_d$, with $\mathbf{I}_d$ the identity matrix of size $d \times d$. In fact, when the feature-space dimensionality d is of a similar order of magnitude that the number of samples $N^{\rm trn}$, the empirical between-class covariance matrix is often ill-conditioned.

The FLD ensemble is a set of L base learners implemented as FLDs trained on uniformly randomly selected $d_{\rm sub}$-dimensional subsets $\mathcal{F}_1, \ldots, \mathcal{F}_L$ of the feature space. This approach to diversify base learners was firstly used with decision trees in [24]. The ensemble reaches its final decision by fusing the decisions of all L individual base learners using majority voting. The ensemble training scales well w.r.t. the feature dimensionality and the training set size because one can select $d_{\rm sub} << d$. The hyper-parameters $d_{\rm sub}$ and L are determined by a search using either a cross-validation set or by bootstrapping (the latter choice was selected in the original publication [6]).

Recently [7], [8], the FLD ensemble was reformulated within the hypothesis testing theory. In particular, the majority voting rule was replaced with a likelihood ratio. Below, we briefly explain the main idea. Denoting the weight vector of the ith base learner as $\mathbf{w}^{(i)}$, we define $\mathbf{v} \in \mathbb{R}^L$, $\mathbf{v} = (v_1, \ldots, v_L)$, $v_i = \mathbf{f}\cdot\mathbf{w}^{(i)\,\rm T}$, the vector of L projections of the feature vector $\mathbf{f}$ on all L weight vectors, $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(L)}$. An assumption has been made about $\mathbf{v}$ that it follows a multivariate Gaussian (MVG) distribution $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ on cover and stego features. After normalizing the projection vector by $\tilde{\mathbf{v}} = (\mathbf{v} - \boldsymbol{\mu}_0)\boldsymbol{\Sigma}_0^{-1/2}$ and under the shift hypothesis $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1$, $\tilde{\mathbf{v}} \sim \mathcal{N}(0, \mathbf{I}_L)$ and $\tilde{\mathbf{v}} \sim \mathcal{N}(\boldsymbol{\theta}_1, \mathbf{I}_L)$ on cover and stego images. The majority voting decision can thus be replaced with a LRT in the form:

$$\mathcal{L}: \begin{cases} \mathcal{H}_0 \ \text{ if } \ \Lambda^{\rm lr}(\tilde{\mathbf{v}}) < \tau \\ \mathcal{H}_1 \ \text{ if } \ \Lambda^{\rm lr}(\tilde{\mathbf{v}}) > \tau, \end{cases} \tag{4}$$

where τ is a threshold that can be selected, for example, to maximize the test power while satisfying a prescribed false-alarm rate, and

$$\Lambda^{\rm lr}(\tilde{\mathbf{v}}) = \frac{\boldsymbol{\theta}_1\tilde{\mathbf{v}}^{\rm T}}{\|\boldsymbol{\theta}_1\|_2}, \tag{5}$$

is the likelihood ratio. Note that, in contrast with the majority voting, this decision rule makes the classifier linear and the sole difference to the single FLD using all features is how the linear classifier is constructed.

III. From Ensemble to linear classifier

As briefly summarized in the previous section, replacing the majority-voting rule in the FLD ensemble with a LRT turns the ensemble into a linear classifier. And, as shown in [7], [8], this linear classifier can achieve essentially the same performance as the non-linear FLD ensemble [6] at least in classification problems in steganalysis. Indeed, if the decision boundary in high-dimensional rich image models is close to linear, we will not see any large difference between linear and non-linear classifiers. Being aware of this caveat, we hypothesize that at least in current classification problems in digital image steganography, it appears that linear classifiers can achieve essentially the same classification accuracy as the original FLD ensemble employing majority voting. It thus becomes meaningful to ask whether there exist simpler approaches based on the FLD that use alternative methods for

its regularization that might offer advantages over the original ensemble, such as a lower training complexity.

To this end, we study the following four linear classifiers: an ℓ_2 regularization of the reciprocal Fisher ratio, ridge regression that is a least square estimation with ℓ_2 regularization, an alternative implementation of ridge regression using LSMR optimization method [11], and a least square estimation with ℓ_1 regularization, known as LASSO.

ℓ_2 regularization of the Fisher ratio. This is achieved by replacing the maximization of the Fisher ratio (2) with

$$\mathbf{w} = \arg\min F(\mathbf{w})^{-1} + \lambda \|\mathbf{w}\|_2^2. \tag{6}$$

It is shown in Appendix that this ℓ_2 regularization leads to the same weight vector $\mathbf{w}$ as the vector obtained using Tikhonov regularization of the between-class covariance matrix (3):

$$\mathbf{w} = (\widehat{\boldsymbol{\mu}}_s - \widehat{\boldsymbol{\mu}}_c) \left(\widehat{\boldsymbol{\Sigma}}_c + \widehat{\boldsymbol{\Sigma}}_s + \lambda \mathbf{I}_d \right)^{-1}.$$

Ridge-regression, also referred to Tikhonov regularization based on least square estimation [10]. To formally describe this classifier, let us denote the matrix of all training samples:

$$\mathbf{X} = \begin{pmatrix} \mathbf{C}^{\text{trn}} \\ \mathbf{S}^{\text{trn}} \end{pmatrix},$$

and, similarly, let us define the label vector $\mathbf{y} \in \mathbb{R}^{2N^{\text{trn}}}$ that represents the class of the samples from matrix $\mathbf{X}$:

$$\mathbf{y} = \Big(\underbrace{-1 -1 -1 \ldots -1 -1}_{N^{\text{trn}}} \underbrace{11 \ldots 11}_{N^{\text{trn}}} \Big)^{\mathrm{T}}.$$

The ridge regression aims at finding a weighting vector $\mathbf{w}_{\text{rr}}$ that minimizes the squared error between the label $\mathbf{y}$ and the linearly estimated label:

$$\mathbf{w}_{\text{rr}} = \underset{\mathbf{w} \in \mathbb{R}^d}{\arg\min} \|\mathbf{y} - \mathbf{X}\,\mathbf{w}^{\mathrm{T}}\|_2^2 + \lambda \|\mathbf{w}\|_2^2. \tag{7}$$

Few calculations show that, see for instance [10], that an explicit solution of Eq. (7) is given by:

$$\mathbf{w}_{\text{rr}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda \mathbf{I}_d)^{-1} \mathbf{X}^{\mathrm{T}} \mathbf{y}.$$

Interestingly, the weighting vector given by the ridge regression corresponds to the weighting vector of the regularized FLD provided that the features have a zero-mean (see Appendix A). We note however that, because we did not center the features, the solution of the ridge regression does not correspond to the weighting vector of the regularized FLD.

Solving linear least square with LSMR. Note that all optimization problems above can be obtained by solving the appropriate system of linear equations. The main advantage of this approach is the fact that there exist numerous optimization methods for solving large linear systems efficiently. These methods are iterative and are associated with a stopping criterion either on the solution $\mathbf{w}_{\text{rr}}$ or the residual $\mathbf{y} - \mathbf{X}\,\mathbf{w}'_{\text{rr}}$. To this end, we implemented the ridge regression using a large linear system optimization method called LSMR [11] due to its low computational complexity and low memory requirements.[1]

[1]LSMR (Least Square Minimun-Residual) function can be downloaded from Stanford University's Systems Optimization Laboratory.

We note that this approach needs to find two parameters – the regularization parameter λ and the tolerance used in LSMR, which controls the trade off between computational efficiency and optimality of the found solution. In this paper, we simply fix $\lambda = 10^{-8}$ (as we saw negligible sensitivity w.r.t. this parameter) and search for the tolerance to obtain the best detection accuracy.

LASSO regularization, Equations (7) and (6) can be viewed as ℓ_2 regularized optimizations. The machine learning community frequently uses an ℓ_1 regularization because it has the added benefit of producing sparse solutions (solutions where only some items of $\mathbf{w}$ are non-zero), thus identifying a sufficient set of features for linear classifiers (ℓ_1 regularization is also called the Least Absolute Shrinkage and Selection Operator (LASSO)). In this work, we used ℓ_1 in Eq. (7), which leads to the following minimization problem:

$$\arg\min \|\mathbf{y} - \mathbf{X}\,\mathbf{w}^{\mathrm{T}}\|_2^2 + \lambda \|\mathbf{w}\|_1. \tag{8}$$

While there is no analytic solution to the LASSO regularization problem (8), efficient convex optimization methods can be applied [25].[2]

IV. Numerical results

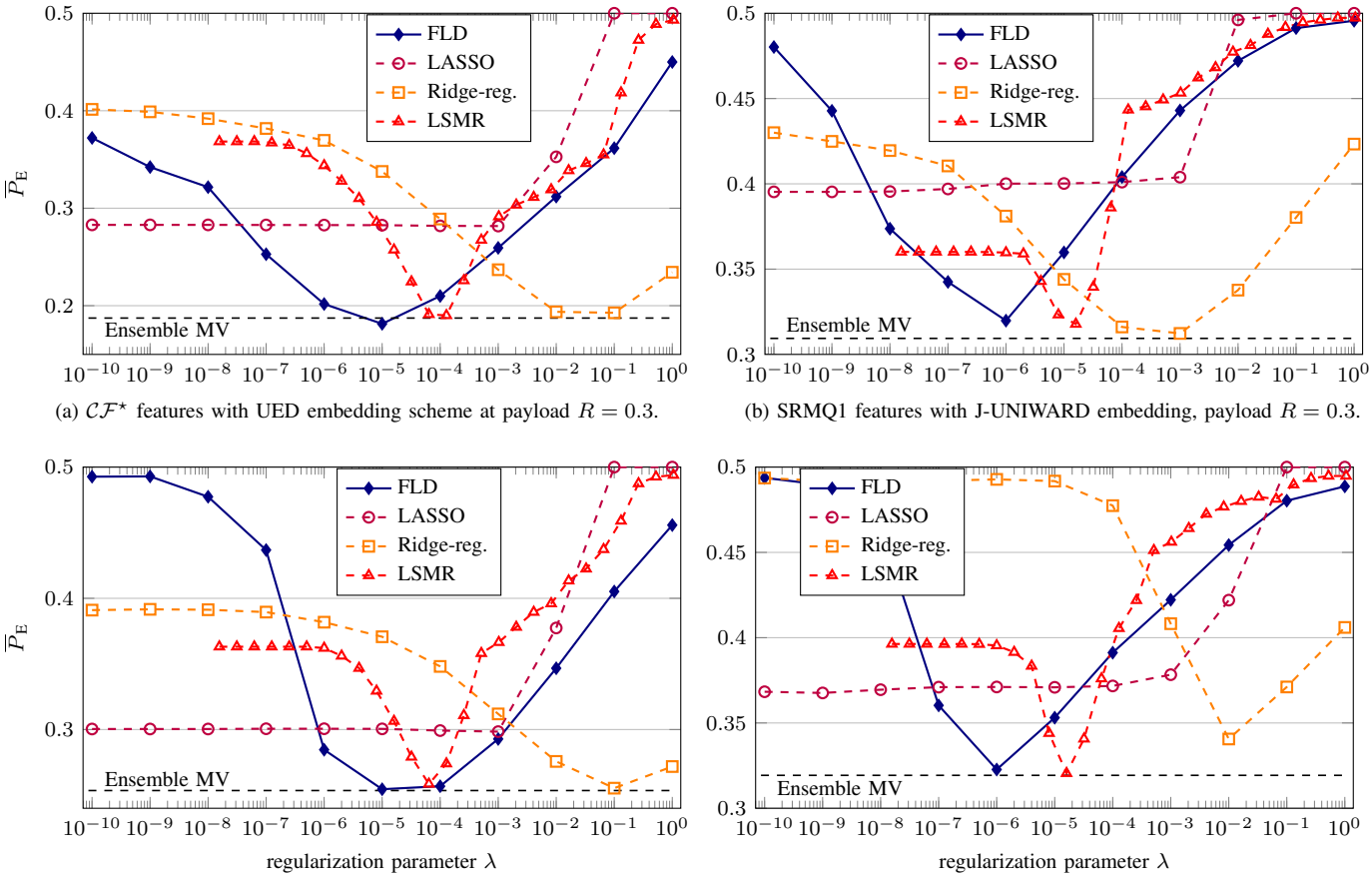
Before discussing the numerical results, we briefly present the common core of all experiments.

All results presented in this paper are obtained on BOSSbase 1.01 [12] of $10,000$ 512×512 gray-scale images. For generality, both spatial domain and JPEG domain steganographic schemes have been used together with spatial domain and JPEG domain feature sets.[3] For spatial domain, four embedding algorithms have been used, namely, HUGO [13] with bounding distortion (HUGO-BD), Wavelet Obtained Weights (WOW) [14], Spatial UNIversal WAvelet Relative Distortion (S-UNIWARD) [2], and the recent scheme based on statistical detectability [15], [17]. For steganalysis, we used four spatial domain feature sets: the second-order Subtractive Pixel Adjacency Matrix (SPAM) [3] of dimensionality 686, the Spatial Rich Model (SRM) [4] as well as its selection-channel-aware version (maxSRM) [18], both made of $34,671$ features, and the version with a single quantization (SRMQ1) containing $12,753$ features.

For the JPEG domain, we used both non-side informed algorithms and side-informed algorithms. For the first type, we used nsF5 [19], Entropy-Based Steganography (EBS) [20], Uniform Embedding Distortion (UED) [21], and JPEG domain UNIWARD [2],referred to as J-UNIWARD. Three algorithms have been used for side-informed JPEG steganography: Perturbed Quantization (PQ) [19], the side-informed version of EBS (SI-EBS) [20], and side-informed UNIWARD, SI-UNIWARD [2]. Five feature set that target JPEG domain embedding have been used: $\mathcal{CF}^\star$ [6], of dimension $7,850$, the Cartesian-calibrated JPEG Rich Model (CC-JRM) [5] used alone and used in union with SRMQ1, referred to as JSRM [5],

[2]In the present paper we used Matlab ® lasso function.

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



(a) $\mathcal{CF}^\star$ features with UED embedding scheme at payload $R = 0.3$.

(b) SRMQ1 features with J-UNIWARD embedding, payload $R = 0.3$.

(c) CC-JRM features with SI-EBS embedding scheme, payload $R = 0.3$.

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

Fig. 1: Evolution of P_E as a function of the regularization parameter λ (or tolerance of LSMR) for several embedding algorithms and several feature sets.

whose dimensionality is $35,263$, the recent Discrete Cosine Transform Residual (DCTR) [22], consisting of $8,000$ features based on undecimated DCT coefficients, and the PHase Aware pRojection Model (PHARM) [23], made of $12,600$ features.

In this paper, the detection accuracy is measured as the total probability of error under equal Bayesian priors $P_E = {}^1\!/_2\,(P_{FA} + P_{MD})$, with P_{FA} and P_{MD} the empirical probability of false alarm and missed detection respectively. The detection accuracy is also always averaged over 10 splits on the testing set (a 50/50 split for training and testing was used).

First, Figure 1 contrasts the detection accuracy of the proposed linear classifier with the ensemble classifier. We note that the detection accuracy P_E is plotted as a function of the regularization parameter λ. Note that, as explained in the previous section, for the ridge-regression, that uses the LSMR large linear system optimization, we search for the tolerance, which controls the stopping criterion of the iterative minimization search, since the regularization parameter λ has negligible influence for this methods. Four different cases with different steganographic algorithms and feature sets are presented in Figure 1. The conclusions that follow can be observed across other combinations of feature sets and embedding algorithms. First, note that the LASSO performs poorly in general but has the important advantage to be rather insensitive to the regularization parameters: the detection accuracy is almost always at its best for $\lambda \in (10^{-15}, 10^{-5})$. Similarly, we note that both the FLD classifier and the ridge regression with a large system optimization can achieve roughly similar performance as the ensemble classifier. We also note that the FLD classifier almost always achieves the best detection accuracy for $\lambda \approx 10^{-6}$; this fact can be used when searching for the best regularization parameter in cross-validation. On the other hand, the regularization parameter for which the the ridge regression, as well as its approximation, achieve the lowest P_E depends on the feature set and the steganographic algorithm.

Except for LASSO, for which we set $\lambda = 10^{-10}$, the optimal value of the regularization parameter (or tolerance for LSMR) was determined by a simple grid search. For the large linear system optimization for ridge regression, because it has a very low computational complexity, we modified the code so that it outputs the solutions obtained for various tolerance values. This allowed us to keep the same computational complexity as the smallest tolerance value, which we set

TABLE I: Detection accuracy of the ensemble classifier compared with the proposed linear classifiers for various embedding schemes. Detection accuracy is measured as total probability of error P_E.

Embedding algorithm / feature set	Ensemble classifier [6]	FLD classifier	Ridge Regression	LSMR Optimization [11]	LASSO
HUGO-BD [13], $R = 0.2$ / SPAM [3]	$.4409 \pm .0021$	$.4414 \pm .0017$	$.4409 \pm .0021$	$.4405 \pm .0027$	$.4598 \pm .0015$
S-UNIWARD [2], $R = 0.2$ / SRMQ1 [4]	$.3283 \pm .0033$	$.3364 \pm .0029$	$.3450 \pm .0211$	$.3342 \pm .0029$	$.3671 \pm .0038$
WOW [14], $R = 0.2$ / SRM [4]	$.3196 \pm .0031$	$.3289 \pm .0021$	$.3402 \pm .0039$	$.3267 \pm .0023$	$.3694 \pm .0051$
MiPOD [15], $R = 0.2$ / maxSRMd2 [18]	$.3237 \pm .0038$	$.3321 \pm .0036$	$.3343 \pm .0101$	$.3307 \pm .0029$	$.3669 \pm .0048$
UED [21], $R = 0.3$ / $\mathcal{CF}^\star$ [6]	$.1890 \pm .0043$	$.2181 \pm .0823$	$.1909 \pm .0046$	$.1925 \pm .0049$	$.2875 \pm .0047$
J-UNIWARD [2], $R = 0.3$ / SRMQ1 [5]	$.3112 \pm .0045$	$.3197 \pm .0032$	$.3317 \pm .0074$	$.3185 \pm .0023$	$.3950 \pm .0034$
UED [21], $R = 0.2$ / PHARM [23]	$.1742 \pm .0034$	$.4950 \pm .0041$	$.4959 \pm .0026$	$.1748 \pm .0024$	$.5000 \pm .0000$
SI-UNIWARD [2], $R = 0.4$ / DCTR [22]	$.4261 \pm .0029$	$.4355 \pm .0196$	$.4292 \pm .0041$	$.4298 \pm .0021$	$.4899 \pm .0135$
SI-EBS [20], $R = 0.3$ / CCJRM [5]	$.2517 \pm .0035$	$.2608 \pm .0025$	$.2614 \pm .0061$	$.2592 \pm .0023$	$.3019 \pm .0042$
SI-UNIWARD [2], $R = 0.3$ / JSRM [5]	$.4582 \pm .0020$	$.4630 \pm .0026$	$.4608 \pm .0030$	$.4616 \pm .0031$	$.4764 \pm .0042$

TABLE II: Computation time (in seconds) of the ensemble classifier compared with the proposed linear classifiers, same settings as in Table I. Computations were carried out on a 16-physical-core Intel® Xeon® E5 @ 2.60GHz with RAM 256GB.

Embedding algorithm / feature set	Ensemble classifier [6]	FLD classifier	Ridge Regression	LSMR Optimization [11]	LASSO
HUGO-BD [13], $R = 0.2$ / SPAM [3]	17.67	0.66	0.76	0.91	3.63
S-UNIWARD [2], $R = 0.2$ / SRMQ1 [4]	123.7	58.75	99.72	7.62	56.71
WOW [14], $R = 0.2$ / SRM [4]	281.7	1270	1132	22.33	130.7
MiPOD [15], $R = 0.2$ / maxSRMd2 [18]	101.6	931.3	951.9	25.87	157.07
UED [21], $R = 0.3$ / $\mathcal{CF}^\star$ [6]	328.4	32.52	30.97	31.06	92.30
J-UNIWARD [2], $R = 0.3$ / SRMQ1 [5]	282.6	87.49	98.04	5.95	53.69
UED [21], $R = 0.2$ / PHARM [23]	279.6	102.9	52.86	6.44	29.79
SI-UNIWARD [2], $R = 0.4$ / DCTR [22]	86.66	27.68	19.87	7.50	23.05
SI-EBS [20], $R = 0.3$ / CCJRM [5]	350.8	381.9	301.8	17.56	177.3
SI-UNIWARD [2], $R = 0.3$ / JSRM [5]	73.62	1280	672.7	32.56	197.7

to 10^{-6} in this paper, as this corresponds to the case with highest number of iterations. A larger value may be used to further decrease the computational time, see Figure 1). The tolerance value for which the P_E is the smallest on the cross-validation subset is used for testing. For the FLD and the ridge regression, the computational time is slightly lower for larger regularization parameters. Hence, we start the one dimensional search with a rather large regularization parameter, typically $\lambda = 10^{-6}$ for the FLD and $\lambda = 10^{-1}$ for ridge regression. However, the implementation has to be a trade-off between computational time and detection accuracy. For the FLD and ridge-regression (using off-the-shelf solver) we wanted to keep the computational time manageable, though sometimes rather important, as the cost of poor convergence in few cases, see for instance the results obtained for UED embedding scheme and PHARM features in Table I. We would like to acknowledge that this implementation of the linear classifier may certainly be largely improved. Nevertheless, the results presented in Tables I and II show that even such simplistic approaches already show very promising results.

Table I compares the detection accuracy of the ensemble classifier (with optimal values of $d_{\rm sub}$ and L), the FLD, the ridge regression, and the large system optimization for ridge regression using the LSMR and the LASSO. All these classifiers were used with the optimally found regularization parameter λ (or tolerance) as described above. Table I shows that for a majority of the cases, the linear classifiers perform slightly worse than the ensemble classifier: the difference in terms of P_E is between 0.05% and 0.8% for ridge regression implemented using LSMR optimization algorithm. Note also that the LASSO always performs significantly worse.

Table II compares the computational time of the same classifiers with the same settings. Note that for feature sets of medium dimensionality (typically up to $15,000$ features) the ensemble classifier always requires a much higher computational time than all its competitors. However, for larger feature sets (see the results with SRM, maxSRM, CC-JRM, and JSRM) the computational time of FLD and ridge regression as well as the memory requirements become prohibitively large since a very large matrix has to be stored and inverted. In fact, most of the computational time of both the FLD and ridge regression come from the inversion of a matrix of size $d \times d$, which has the complexity $\mathcal{O}(d^3)$. Let us recall, for comparison, that the ensemble has a complexity $\mathcal{O}(N^{\rm trn} L d_{\rm sub}^2 + L d_{\rm sub}^3)$. By contrast we note that the ridge regression solved using LSMR iterative optimization methods has a complexity of $\mathcal{O}(N^{\rm it} N^{\rm trn} \times d)$, with $N^{\rm it}$ the number of iterations [11]. This should be contrasted with the loss of detection accuracy.

V. Conclusion

The main contribution of this paper is to show that the power of the FLD ensemble classifier widely used in steganalysis does not come from the non-linearity of the majority voting rule but from the natural regularization process of the ensemble when training on random subsets of features. This paper also shows that, if correctly regularized, a simple FLD classifier or a ridge regression may achieve almost the same performance. However, their naive implementation by using off-the-shelf solvers (e.g. in Matlab) leads to very high complexity for large feature sets. As a remedy, we have demonstrated on ridge regression that state-of-the-art optimization algorithms allow us to achieve almost the same detection accuracy as an

ensemble classifier for a computational time up to 10 times smaller. Further work can be done to further improve the detection accuracy and the computational complexity.

ACKNOWLEDGEMENTS

Vahid Sedighi and Jessica Fridrich were supported by Air Force Office of Scientific Research under the research grant number FA9950-12-1-0124. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of AFOSR or the U.S. Government. The work of Rémi Cogranne was funded in part by the STEG-DETECT program for scholar mobility and by IDENT research grant both from Conseil Rgional de Champagne-Ardenne.

APPENDIX
REGULARIZATION OF FLD

In this appendix, we show that an L_2-regularization of the reciprocal Fisher ratio, which is the optimization problem: $\arg\min_{\mathbf{w}} F(\mathbf{w})^{-1} + \lambda \left\|\mathbf{w}\right\|^2$, is equivalent to regularizing the between class covariance matrix $\mathbf{\Sigma} = \mathbf{\Sigma}_c + \mathbf{\Sigma}_s$ in classical FLD by adding $\lambda' \mathbf{I}_d$ to it before inversion.

To this end, we write the reciprocal Fisher ratio (2) as $1/F(\mathbf{w}) = \mathbf{w}\mathbf{\Sigma}\mathbf{w}^{\mathrm{T}}/\mathbf{w}\mathbf{\Gamma}\mathbf{w}^{\mathrm{T}}$, where we used for compactness $\mathbf{\Gamma} = (\boldsymbol{\mu}_c - \boldsymbol{\mu}_s)^{\mathrm{T}}(\boldsymbol{\mu}_c - \boldsymbol{\mu}_s)$. We note that from the definition of $\mathbf{\Gamma}$, $\mathbf{\Gamma}\mathbf{x}^{\mathrm{T}}$ is a multiple of $(\boldsymbol{\mu}_c - \boldsymbol{\mu}_s)^{\mathrm{T}}$ for any vector $\mathbf{x}$. We now differentiate the objective function w.r.t. $\mathbf{w}$ to find the minimum:

$$\frac{d}{d\mathbf{w}} \frac{F(\mathbf{w})^{-1} + \lambda \mathbf{w}\cdot\mathbf{w}^{\mathrm{T}}}{2} = \frac{\mathbf{\Sigma}\mathbf{w}^{\mathrm{T}}\mathbf{w}\mathbf{\Gamma}\mathbf{w}^{\mathrm{T}} - \mathbf{\Gamma}\mathbf{w}^{\mathrm{T}}\mathbf{w}\mathbf{\Sigma}\mathbf{w}^{\mathrm{T}}}{(\mathbf{w}\mathbf{\Gamma}\mathbf{w}^{\mathrm{T}})^2} + \lambda\mathbf{w}.$$

Setting this expression to zero and multiplying by $\mathbf{w}\mathbf{\Gamma}\mathbf{w}^{\mathrm{T}}$ gives us an equation for $\mathbf{w}$

$$(\mathbf{\Sigma} + \lambda a^2(\mathbf{w}))\mathbf{w}^{\mathrm{T}} = c(\mathbf{w})(\boldsymbol{\mu}_c - \boldsymbol{\mu}_s)^{\mathrm{T}}, \tag{9}$$

where $a(\mathbf{w}) = \mathbf{w}(\boldsymbol{\mu}_c - \boldsymbol{\mu}_s)^{\mathrm{T}}$ and $c(\mathbf{w}) = \mathbf{w}\mathbf{\Sigma}\mathbf{w}^{\mathrm{T}}/a(\mathbf{w})$ are scalars. Since $\mathbf{w}$ can be normalized so that $a(\mathbf{w}) = 1$, Eq. (9) the projection vector $\mathbf{w}$ is obtained by

$$\mathbf{w} = (\mathbf{\Sigma} + \lambda)^{-1}(\boldsymbol{\mu}_c - \boldsymbol{\mu}_s)^{\mathrm{T}},$$

which is the same as the regularization of the inverse covariance $\mathbf{\Sigma}$ in classical FLD.

APPENDIX
RELATIONSHIP BETWEEN LEAST-SQUARE AND FLD

This appendix recalls the relationship between ridge regression and FLD. The FLD projection vector $\mathbf{w}$ is given by :

$$\mathbf{w} = (\widehat{\boldsymbol{\mu}}_s - \widehat{\boldsymbol{\mu}}_c)\,(\widehat{\mathbf{\Sigma}}_c + \widehat{\mathbf{\Sigma}}_s)^{-1}.$$

The mean of cover features may be easily obtained by $\widehat{\boldsymbol{\mu}}_c = \frac{1}{N}\mathbf{C}^{\mathrm{trn}^T}\cdot\mathbf{1}_N$ with $\mathbf{1}_N$ a (column) vector of ones. Hence, recalling that the matrix of all training data is: $\mathbf{X} = \begin{pmatrix}\mathbf{C}^{\mathrm{trn}} \\ \mathbf{S}^{\mathrm{trn}}\end{pmatrix}$, and recalling that it is assumed, without loss of generality, that the label for cover is 1 and -1 for stego, it is straightforward that:

$$(\widehat{\boldsymbol{\mu}}_c - \widehat{\boldsymbol{\mu}}_s)^T = \frac{1}{N}\mathbf{X}^T\mathbf{Y}.$$

Additionally, assuming, without loss of generality, that the features are normalized so that the columns of $\mathbf{X}$ have a zero mean, or equivalently $\widehat{\boldsymbol{\mu}}_c = -\widehat{\boldsymbol{\mu}}_s$, it is then obvious (by a block-wise matrix product) that $\mathbf{X}^T\mathbf{X} = \left(\mathbf{C}^{\mathrm{trn}^T}\mathbf{C}^{\mathrm{trn}} + \mathbf{S}^{\mathrm{trn}^T}\mathbf{S}^{\mathrm{trn}}\right) = (N-1)\left(\widehat{\mathbf{\Sigma}}_c + \widehat{\mathbf{\Sigma}}_s\right)$

Thus the linear least square regression $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ can be rewritten, up to a scaling factor, as: $(\widehat{\boldsymbol{\mu}}_c - \widehat{\boldsymbol{\mu}}_s)\left(\widehat{\mathbf{\Sigma}}_c + \widehat{\mathbf{\Sigma}}_s\right)^{-1}$.

REFERENCES

[1] A. D. Ker, P. Bas, R. Böhme, R. Cogranne, S. Craver, T. Filler, J. Fridrich, and T. Pevný, "Moving steganography and steganalysis from the laboratory into the real world," in *ACM Information hiding and multimedia security, IH&MMSec'13*, 2013, pp. 45–58.

[2] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, pp. 1–13, 2014.

[3] T. Pevný, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Trans. Inform. Forensics and Security*, vol. 5, no. 2, pp. 215–224, 2010.

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

[5] J. Kodovskỳ and J. Fridrich, "Steganalysis of JPEG images using rich models,"in *IS&T/SPIE Electronic Imaging conf.*, vol. 8303, 2012, pp. 83 030A–13.

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

[7] R. Cogranne, T. Denemark, and J. Fridrich, "Theoretical model of the FLD ensemble classifier based on hypothesis testing theory," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2014, pp. 167–172.

[8] R. Cogranne and J. Fridrich, "Modeling and extending the ensemble classifier for steganalysis of digital images using hypothesis testing theory," *(accepted for publication in) Information Forensics and Security, IEEE Transactions on*, 2015.

[9] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, 2nd ed. John Wiley & Sons, 2012.

[10] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, *The elements of statistical learning: data mining, inference and prediction*, 2nd ed., Springer, 2009.

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

[12] P. Bas, T. Filler, and T. Pevný, "Break our steganographic system — the ins and outs of organizing boss," in *Information Hiding*, ser. LNCS vol.6958, 2011, pp. 59–70.

[13] T. Pevnỳ, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Information Hiding*, ser. LNCS vol. 6387, 2010, pp. 161–177.

[14] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2012, pp. 234–239.

[15] V. Sedighi, J. Fridrich, and R. Cogranne, "Content-adaptive pentary steganography using the multivariate generalized gaussian cover model," *IS&T/SPIE Electronic Imaging conf.*, vol. 9409, 2015, pp. 94 090H.

[16] T. Pevnỳ, A. D. Ker', "Towards dependable steganalysis," *IS&T/SPIE Electronic Imaging conf.*, vol. 9409, 2015, pp. 94 090I.

[17] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-Adaptive Steganography by Minimizing Statistical Detectability," *Information Forensics and Security, IEEE Transactions on* (submitted).

[18] T. Denemark & *al.*, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich, "Selection-channel-aware rich model for steganalysis of digital images," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2014, pp. 48–53.

[19] J. Fridrich, T. Pevný, and J. Kodovský, "Statistically undetectable JPEG steganography: Dead ends challenges, and opportunities," in *ACM 9th workshop on Multimedia & security, MMSec'07*, 2007, pp. 3–14.

[20] C. Wang and J. Ni, "An efficient JPEG steganographic scheme based on the block entropy of DCT coefficients," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 1785–1788.

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

[22] V. Holub and J. Fridrich, "Low-complexity features for JPEG steganalysis using undecimated DCT," *Information Forensics and Security, IEEE Transactions on*, vol. 10, no. 2, pp. 219–228, Feb 2015.

[23] ——, "Phase-aware projection model for steganalysis of JPEG images," *IS&T/SPIE Electronic Imaging conf.*, vol. 9409, 2015, pp. 94 090T.

[24] T. K. Ho, "Random decision forests" in *Proc. of International Conf. on Document Analysis and Recognition*, pp. 278–282, 1995.

[25] R. Tibshirani, " Regression shrinkage and selection via the lasso", *Journal of the Royal Statistical Society*, Series B, Vol 58, No. 1, pp. 267–288, 1996.