From Blind to Quantitative Steganalysis

Tomáš Pevný[a], Jessica Fridrich[b*], Andrew D. Ker[c]

[a]INPG - Gipsa-Lab, 961 rue de la Houille Blanche, 38402, Grenoble, France
[b]Department of Electrical and Computer Engineering, Binghamton University,
State University of New York
[c]Oxford University Computing Laboratory, Parks Road, Oxford OX1 3QD,
England

ABSTRACT

Quantitative steganalyzers are important in forensic steganalysis as they can estimate the payload, or, more precisely, the number of embedding changes in the stego image. This paper proposes a general method for constructing quantitative steganalyzers from features used in blind detectors. The method is based on support vector regression, which is used to learn the mapping between a feature vector extracted from the image and the relative embedding change rate. The performance is evaluated by constructing quantitative steganalyzers for eight steganographic methods for JPEG files, using a 275-dimensional feature set. Error distributions of within- and between-image errors are empirically estimated for Jsteg and nsF5. For Jsteg, the accuracy is compared to state-of-the-art quantitative steganalyzers.

1. INTRODUCTION

The objective of steganalysis is to detect steganographic channels. Technically, steganography is considered broken when the mere presence of the secret message can be established. In practice, however, the investigation is not likely to stop when the use of steganography is discovered. The analyst may want to uncover more details about the covert communication, such as the number of modifications due to steganographic embedding. Because the number of embedding changes is in general strongly correlated with the message length, one can obtain valuable forensic information about the type of hidden data or the fact that the message is encrypted (if the message length estimates are clustered around multiples of some typical cipher block lengths).

Steganalyzers that can estimate the relative number of embedding changes (the change rate) are called quantitative. They are typically built from heuristic principles and always rely on full knowledge of the embedding algorithm (see, e.g., Refs. 2, 4–6, 10, 13, 25, 27). Even though it is possible to identify within these attacks some general principles for constructing quantitative steganalyzers, their design is still more art than a well-developed methodology. This is confirmed by the fact that the vast majority of current quantitative attacks only cover LSB embedding. Although there exist a few quantitative steganalyzers for other embedding operations, such as ±1 embedding in the spatial domain[23] or the embedding operation of F5,[6] quantitative steganalyzers are missing for most steganographic algorithms despite the fact that essentially all of them can be reliably detected by blind steganalyzers.

This paper proposes a novel approach to quantitative steganalysis that is quite general and does not need detailed knowledge of the embedding mechanism. The basic idea is to turn a blind

[*] Jessica Fridrich: E-mail: fridrich@binghamton.edu, Telephone: +1 607 777 6177, Fax: +1 607 777 4464

steganalyzer into an estimator of the change rate by learning the relationship between the position of stego image features and the change rate. In blind steganalysis, images are modeled using features designed to be sensitive to steganographic embedding. It works when the clusters of cover and stego image features can be separated.[14, 16, 20, 26] Because the clusters' separation is a deterministic function of the change rate, one could conceivably train a multi-classifier to detect a small number of different payloads. Pursuing this idea further, it should be possible to build an estimator of the change rate using regression by mathematically describing the relationship between the feature vector and its position in the feature space. This idea should work for any continuous-valued feature space within which a given steganographic system is detectable.

In our work, we explore ordinary linear least square regression (OLS) and its kernelized version called support vector regression (SVR), essentially a data-driven method similar in spirit to a support vector machine. An important design element of every SVR is the penalization of the regression error: more stable results are typically obtained using non-quadratic penalization, such as the ϵ-insensitive loss or the Huber loss.

This approach to quantitative steganalysis has a very important advantage over previous art: we can design a quantitative steganalyzer without any knowledge of the embedding mechanism. All that is required is access to a database of images embedded with a range of known payloads. Such images can be generated if the steganalyst has access to the embedding algorithm but not necessarily to its inner workings (e.g., if only an executable file is available). There does have to exist a feature set and a blind steganalyzer that can reliably detect the embedding, and the accuracy of the resulting quantitative steganalyzer depends on the sensitivity of the feature set to the attacked steganographic scheme.

This paper is organized as follows. Section 2 presents the methodology for constructing quantitative steganalyzers from features. The methodology is evaluated experimentally in Section 3, where we report the accuracy of quantitative steganalyzers for eight steganographic schemes. Section 4 contains detailed analysis of the estimator error for Jsteg and nsF5 by decomposing it into within-image and between-image components. For Jsteg, the results are compared with a state-of-the-art quantitative steganalyzer. The paper is concluded in Section 5.

2. APPROACH

In this section, we describe the basic method for constructing change rate estimators by learning the relationship between features' location and the change rate, using regression on some training set of stego features and their corresponding change rates. By *change rate* we mean the ratio between the total number of embedding changes and the number of cover elements that can be used for embedding. We favor estimating the change rate as opposed to the relative message length, because the features are sensitive to the number of embedding changes and not to the length of the message. The relationship between both quantities is stochastic and can be further shaped by matrix embedding and source coding.

The process of extracting steganographic features from an image is a mapping $\mathbf{f} : \mathcal{C} \mapsto \mathbb{R}^d$ from the space of all covers, $\mathcal{C}$, to a d-dimensional feature space. In blind steganalysis, machine learning tools are used to find a distinguishing statistic $S : \mathbb{R}^d \mapsto \mathbb{R}$, on which a threshold is set to classify images to the classes of cover and stego.[15] In contrast, in our current problem we seek a function $\psi : \mathbb{R}^d \mapsto [0, 1]$ revealing the relationship between the location of the features and the change rate.

To formalize the problem, let $\mathbf{X} = \left\{ (\mathbf{x}_i, y_i) | \, \mathbf{x}_i \in \mathbb{R}^d, y_i \in [0,1], i \in \{1, \ldots, l\} \right\}$ denote l samples consisting of feature vectors $\mathbf{x}_i = \mathbf{f}(c_i)$ computed from l images c_i embedded with relative number of embedding changes $y_i \in [0,1]$. Our goal is to construct a quantitative steganalyzer by finding a function $\hat{\psi} : \mathbb{R}^d \mapsto [0,1]$ that minimizes the error on $\mathbf{X}$, or

$$\hat{\psi} = \arg \min_{\psi \in \mathcal{F}} \frac{1}{l} \sum_{i=1}^{l} e\left(\psi(\mathbf{x}_i), y_i\right), \tag{1}$$

where $e : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}_0^+$ is an error function (also called a loss function) and $\mathcal{F}$ is an appropriately chosen class of functions $\psi : \mathbb{R}^d \mapsto [0,1]$. For example, in linear ordinary least square (OLS) regression, $e(\hat{y}, y) = (\hat{y} - y)^2$ and $\mathcal{F} = \left\{ \psi(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - b | \mathbf{w} \in \mathbb{R}^d, \; b \in \mathbb{R} \right\}$.

The error function $e(x, x')$ and the class of functions $\mathcal{F}$ influence the accuracy of the resulting regressors $\hat{\psi}$. It is possible that the desired accuracy is not achieved for a given feature set simply because of a wrong combination of e and $\mathcal{F}$. It will be shown later that the computational complexity of solving the optimization problem (1) also needs to be taken into consideration.

In this paper, we solve the regression problem (1) by linear ordinary least-square regression (OLS) and by support vector regression (SVR).[22] While the former is very simple, intuitive, and has a low computational complexity, the latter can reveal more complicated non-linear dependencies at the cost of increased implementation complexity. Assuming the reader is familiar with OLS, the rest of this section describes the main ideas behind SVR. More details can be found in a tutorial on SVR.[22]

2.1. Support Vector Regression

The main idea behind SVR is to map the model space $\mathbb{R}^d$ through a possibly non-linear data-driven mapping $\phi : \mathbb{R}^d \mapsto \mathcal{H}$ into a high-dimensional vector space $\mathcal{H}$, where a linear regression is performed. Thus, for the set of functions $\mathcal{F}$ over which the optimization is carried, we have $\mathcal{F} = \{\psi(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}) - b | \mathbf{w} \in \mathcal{H}, b \in \mathbb{R}\}$. The space $\mathcal{H}$ and function ϕ must be chosen such that there exists a positive definite function (called the kernel) $k : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ satisfying $\left(\forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d\right)$ $\left(k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}}\right)$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is a dot product in $\mathcal{H}$. The function ϕ and the space $\mathcal{H}$ are in practice defined implicitly by the kernel k. The most popular kernels are the Gaussian kernel

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right) \tag{2}$$

and the polynomial kernel $k(\mathbf{x}, \mathbf{x}') = \left(\langle \mathbf{x}, \mathbf{x}' \rangle_{\mathbb{R}^d} + 1\right)^d$.

Depending on the richness of the function class $\mathcal{F}$, the problem (1) can be ill-posed. In order to stabilize it, SVR introduces an additional term $\|\mathbf{w}\|_{\mathcal{H}}^2$ that penalizes complex solutions. Hence, the optimization problem solved by SVR attains the following form

$$\min_{\mathbf{w} \in \mathcal{H}, b \in \mathbb{R}} \|\mathbf{w}\|_{\mathcal{H}}^2 + C \frac{1}{l} \sum_{i=1}^{l} e\left(\mathbf{w} \cdot \phi(\mathbf{x}_i) - b, y_i\right), \tag{3}$$

where C is a parameter describing the trade-off between complexity of the solution and error on the training set.

Ideally, the error function e should be determined from the statistical properties of the noise in features. In our case, however, the noise properties are hard to estimate due to the high dimensionality and because the noise is a complex superposition of cover work irregularities and the random selection of cover elements used for embedding. In order to make the optimization problem (3) computationally tractable, the error function e should be convex. The most popular error functions in SVR are the ϵ-insensitive loss

$$e_\epsilon(\hat{y}, y) = \begin{cases} |\hat{y} - y| - \epsilon & \text{if } |\hat{y} - y| > \epsilon \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

and the Huber loss (Ref. 22 lists other examples). We tested both error functions and eventually decided to use the ϵ-insensitive loss because it gave very similar results as the Huber loss. Note that the free parameter ϵ determines the width of the tube where errors are not penalized (i.e., the estimates within this tube are treated as estimated perfectly). In theory, ϵ should be set to the variance of noise in features.[21] However, as already explained above because the statistical properties of the noise are unknown in our case, the parameter ϵ, together with the kernel parameters were determined using exhaustive search.

A support vector regressor with the Gaussian kernel (2) and ϵ-insensitive loss (4) has three hyper-parameters that need to be set prior to training (solving (3)). They are: the penalization parameter C, the width of the Gaussian kernel γ, and ϵ. The choice of the hyper-parameters has a significant influence on the ability of the regressor to generalize (to accurately estimate the change rate on samples not in the training set). Since there is no optimal method to set them, in our experiments we used a search on a predefined set of triplets (C, γ, ϵ), on which the generalization was estimated by cross-validation. The implementation details of the search are described in Subsection 3.2.

3. EXPERIMENTS ON EIGHT JPEG STEGOSYSTEMS

This section presents a practical evaluation of the proposed method by testing quantitative steganalyzers for eight steganographic algorithms with diverse embedding mechanisms: JP Hide&Seek,[12] Jsteg,[24] Model Based Steganography without deblocking (MBS1),[19] MMx,[11] F5 with shrinkage removed by wet paper codes with matrix embedding turned off (nsF5),[8] OutGuess,[17] Perturbed Quantization,[7] and Steghide.[9] The estimators' accuracy is evaluated on images with relative payloads uniformly distributed on $[0, 1]$, meaning that the length of the message was chosen randomly between zero and the maximum embedding capacity for each algorithm and each image.

All experiments were performed on single-compressed grayscale JPEG images with quality factor 80 created from a database of 9163 raw images taken by digital cameras spanning 23 different models. Prior to any processing, the images were divided into two sets of equal size (≈ 4600 images per set). One set was used exclusively for training the regressor, while the other set was used exclusively for testing its performance.

3.1. Regressor training

As a feature set $\mathbf{f}$, we used the 274 "calibrated Merged features" from Ref. 16, augmented with the number of non-zero DCT coefficients, n_0, as an additional 275th feature. All 275 features were normalized to have zero mean and unit variance. The normalization coefficients were always calculated on the training set.

The hyper-parameters (C, γ, ϵ) were determined by n-fold cross-validation using the following two-phase algorithm to decrease the computational complexity. In the first phase, the parameters were estimated by 5-fold cross-validation on the following grid

$$\begin{aligned}(C, \gamma, \epsilon) \quad \in \mathcal{S}_1 = \quad & \left\{\left(10^i, 2^j, 0.005 \cdot k\right) \mid i \in \{-3, \ldots 4\},\right. \\ & \left. j \in \{-11, \ldots, -5\}, k \in \{1,2,3,4\}\right\}.\end{aligned}$$

The triplet $(C_1, \gamma_1, \epsilon_1)$ with the least estimated generalization error on $\mathcal{S}_1$ was used to seed the search in the second phase. The search in the second phase was performed on the grid

$$\mathcal{S}_2 = \left\{\left(10^i, 2^j, 0.005 \cdot k\right) \mid i, j \in \mathbb{Z}, k \in \mathbb{N}\right\}.$$

In each iteration, the point with least generalization error (again estimated by 5-fold cross-validation) was checked to see whether it lay on the grid boundary. If so, the error was estimated on the neighboring points from the set $\mathcal{S}_2$ and the check was repeated. If not, the search was stopped and the triplet (C, γ, ϵ) with the least estimated generalization error was used for training.

The idea behind the two-phase search is to ensure that the point with least estimated generalization error is not the boundary point of the explored set. Under the assumption that the generalization error surface is convex, which is very reasonable, this algorithm keeps the number of explored points relatively low, while returning a suitable set of hyper-parameters.

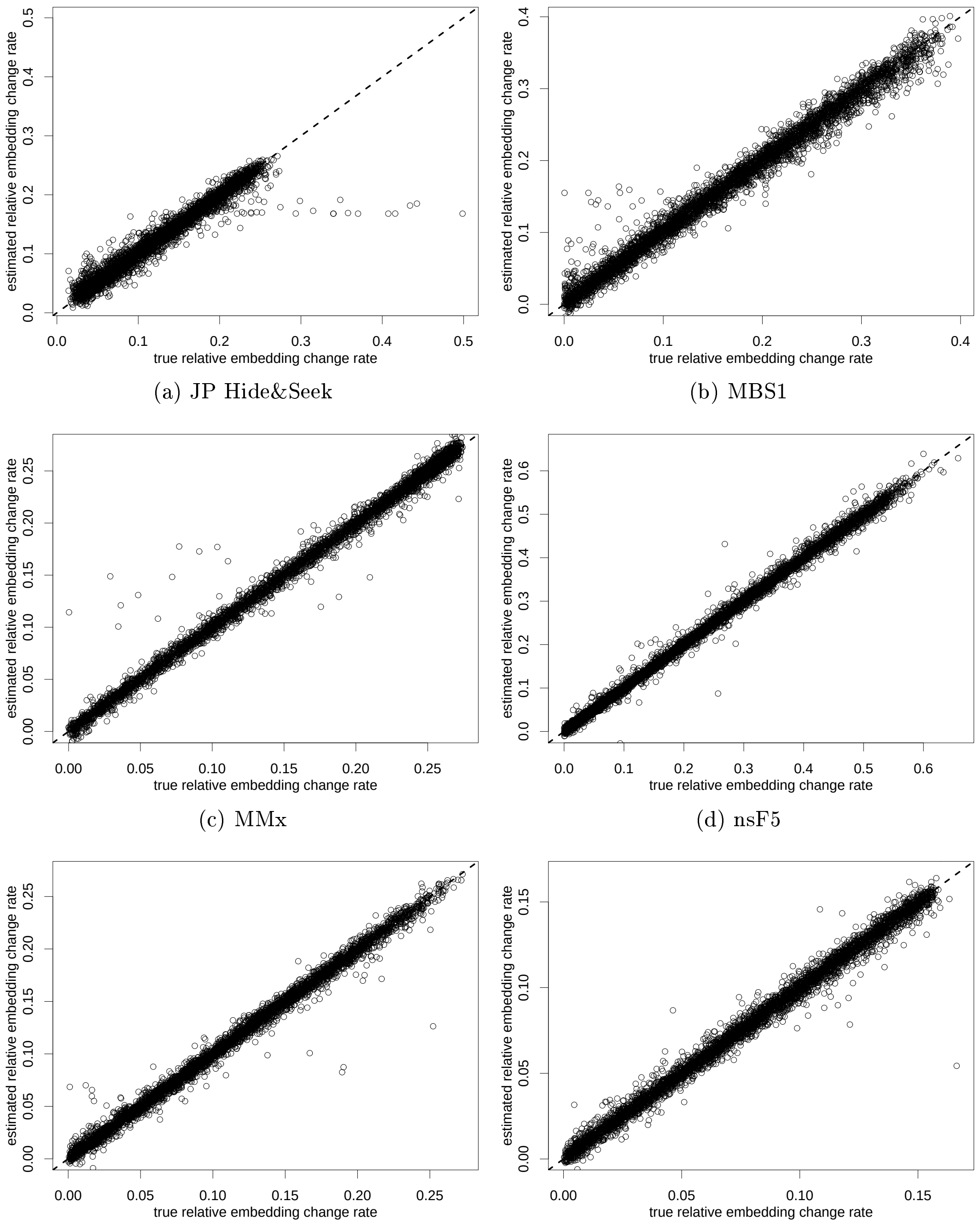
3.2. General results

We prepared two quantitative steganalyzers for each algorithm. One used plain OLS regression, while the other used SVR as outlined above. Because the error distribution of quantitative steganalyzers has typically heavy tails[3] (we will see in Subsection 4.2 that this is so for our steganalyzer too), we evaluate the performance of the regressors using robust statistics rather than the variance. Figure 1 shows a scatter plot of the estimated change rates against the true values. Table 1 displays the sample Median Absolute Error (MAE) and bias computed from all estimates. We can see that all quantitative steganalyzers except steganalyzer for PQ have the MAE of the estimation of relative change-rate of the order of 10^{-3}, with an order of magnitude lower bias. The OLS regressor has a slightly higher MAE but exhibits a lower bias for several embedding algorithms. The fact that the median absolute error of the OLS regression is of the same order as the error of SVR suggests that the features change almost linearly with the number of embedding changes. Despite the slightly higher MAE of the OLS regression, it offers an attractive choice because of its low computational complexity: the search for hyper-parameters and subsequent training of the SVR took about 1 day on a 64bit AMD Opteron 2.4GHz, while the time to train the OLS regression on the same machine and training set was less than 1 minute.

4. DETAILED RESULTS FOR JSTEG AND NSF5

This section presents analysis of the error distributions of quantitative steganalyzers for nsF5 and Jsteg. We chose these two algorithms as representatives of the least and most detectable stego algorithms for JPEG, respectively, and because their simple embedding mechanism allows the precise control of the change rate required for our experiments.

We first consider how the estimation error is influenced by payload size. Then we decompose the error into two factors: that due to the cover, and that due to location of the payload. This allows



(a) JP Hide&Seek

(b) MBS1

(c) MMx

(d) nsF5

(e) Outguess

(f) Steghide

Figure 1. Scatter plot showing the change rate estimated by SV regressors with respect to the true change rate for JP Hide&Seek, Model Based Steganography without deblocking (MBS1), MMx, no-shrinkage F5 (nsF5), OutGuess and Steghide; Jsteg and Perturbed Quantization were omitted for space reasons. All estimates were made on images from the testing set. The dashed line corresponds to perfect estimation.

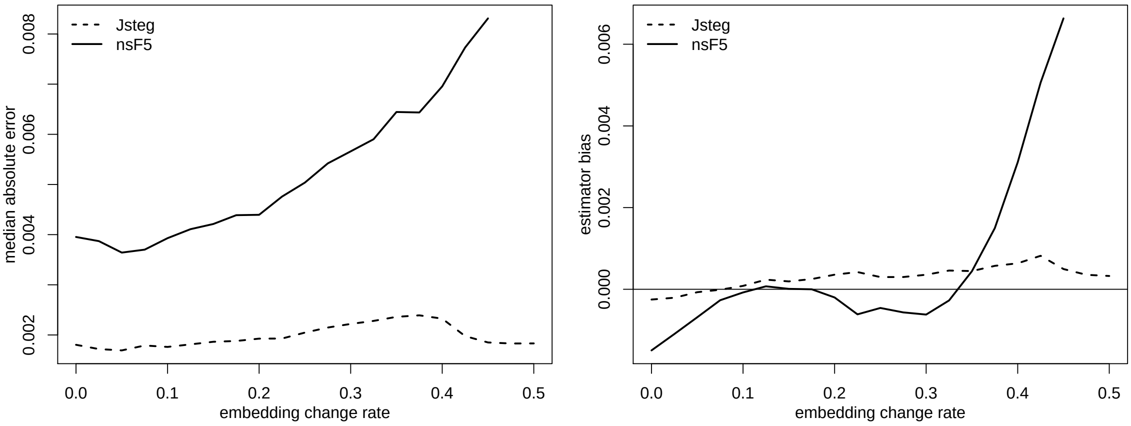
	SVR		OLS	
Algorithm	MAE	Bias	MAE	Bias
JP Hide&Seek	$5.24{\cdot}10^{-03}$	$2.41{\cdot}10^{-04}$	$7.91{\cdot}10^{-03}$	$-1.70{\cdot}10^{-04}$
Jsteg	$1.90{\cdot}10^{-03}$	$2.50{\cdot}10^{-04}$	$8.38{\cdot}10^{-03}$	$-5.29{\cdot}10^{-04}$
MB1	$6.63{\cdot}10^{-03}$	$-1.63{\cdot}10^{-04}$	$9.07{\cdot}10^{-03}$	$3.86{\cdot}10^{-05}$
MMx	$2.70{\cdot}10^{-03}$	$1.08{\cdot}10^{-04}$	$3.25{\cdot}10^{-03}$	$1.58{\cdot}10^{-04}$
nsF5	$4.82{\cdot}10^{-03}$	$-2.51{\cdot}10^{-04}$	$8.39{\cdot}10^{-03}$	$-5.29{\cdot}10^{-04}$
OutGuess	$2.48{\cdot}10^{-03}$	$3.67{\cdot}10^{-04}$	$2.53{\cdot}10^{-03}$	$1.51{\cdot}10^{-04}$
PQ	$4.83{\cdot}10^{-02}$	$-3.78{\cdot}10^{-02}$	$5.69{\cdot}10^{-02}$	$-2.89{\cdot}10^{-03}$
Steghide	$2.04{\cdot}10^{-03}$	$1.80{\cdot}10^{-04}$	$3.23{\cdot}10^{-03}$	$2.60{\cdot}10^{-04}$

Table 1. Median absolute error (MAE) and bias for OLS regressor and SVR with Gaussian kernel and ϵ-insensitive loss, on eight steganographic algorithms. The accuracy is measured on the testing set.

us to compare the nature of our estimation error against quantitative estimators for spatial-domain LSB replacement. Finally, we compare the accuracy of our quantitative estimator for Jsteg with prior art.[25]

4.1. Compound error

The accuracy of steganalyzers presented in Subsection 3.2 was estimated on images from the training set embedded with messages of random length. To find out how the errors depend on the number of embedding changes, nsF5 and Jsteg were forced to produce a predefined set of 21 change rates $\beta \in \mathcal{B} \triangleq \{0, 0.025, 0.05, \ldots, 0.475, 0.5\}$. To be absolutely precise, $\beta = \frac{c}{n_0}$, where c is the number of changed DCT coefficients and n_0 is the total number of all non-zero DCT coefficients in the cover.



(a) median absolute error

(b) bias

Figure 2. Median absolute error and bias of quantitative steganalyzers for Jsteg and nsF5, with respect to change rate. The graph for nsF5 is shown only up to change rate $\beta = 0.45$, because at higher rates the algorithm frequently fails to embed the message.

Figure 2 shows the accuracy of the steganalyzers from Subsection 3.2 for each change rate from $\mathcal{B}$. From 2(a) we can see that the MAE of the quantitative steganalyzer for nsF5 increases with the payload, but remains in the same order of 10^{-3}. Figure 2(b) reveals that the increase in MAE is due to increased bias of the estimator on images with higher payload. The accuracy of the estimator for Jsteg remains stable with respect to the image payload.

It is interesting to observe that the accuracy of estimators on cover images does not deviate, even though cover images were not included in the training set (the probability that the embedding rate will be exactly zero is almost zero).

4.2. Within- and between-image error

In general, payload size estimation error can be decomposed into three parts, as first described in Ref. 3 and extended in Ref. 1. When a payload is embedded, the number of embedding changes depends on random correlations with the cover, and so this does not indicate exactly the size of payload (in our experiments we have eliminated this deviation by measuring embedding changes directly, but it occurs when the estimator is applied to genuine stego images). Then the remaining error can be partly attributed to random placement of the payload in the cover, called *within-image error*, and the rest to the cover itself, called *between-image error*. In Ref. 1 a payload-size estimator $\hat{p}$ is explained in terms of the true payload size p and three error terms

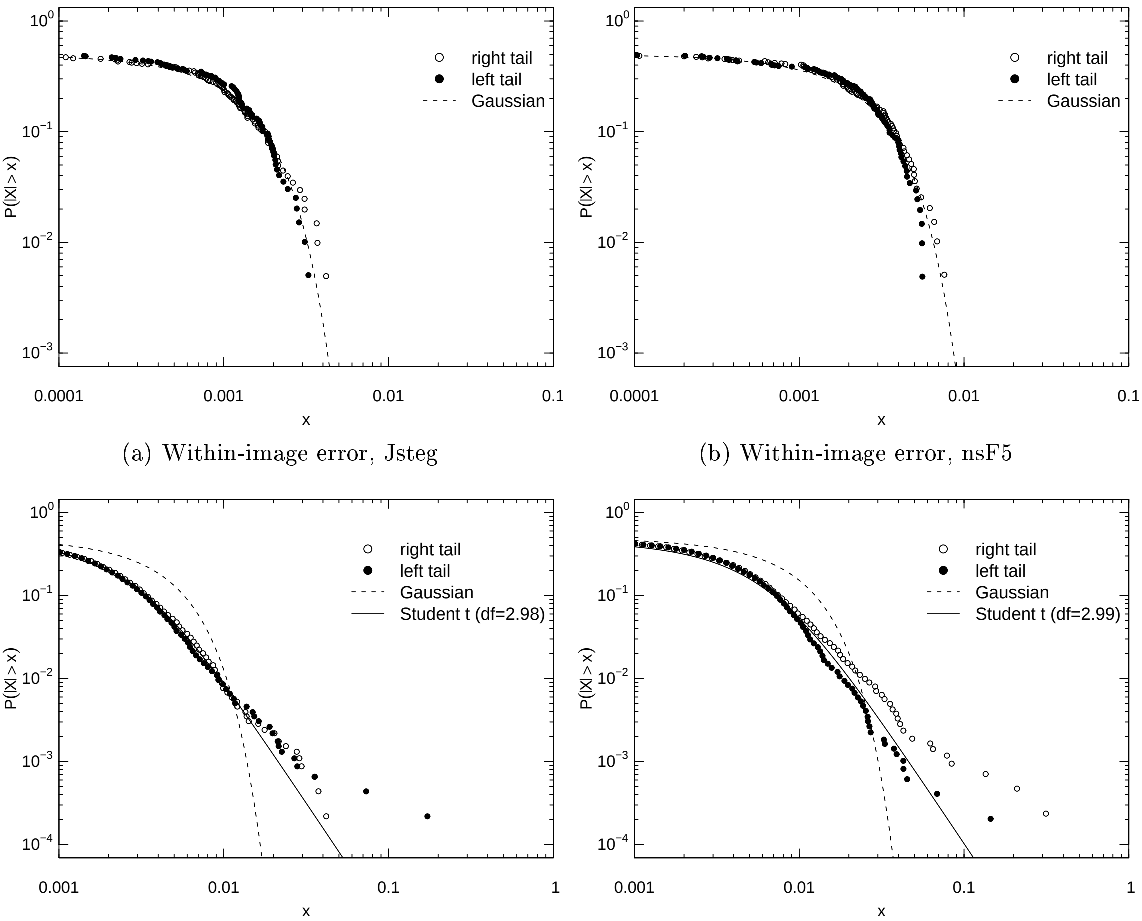
$$\hat{p} = p + Z_{\text{cov}} + Z_{\text{pos}} + Z_{\text{flip}}$$

where Z_{cov} is the between-image error, Z_{pos} the within-image error due to payload position, and Z_{flip} the uncertainty in the embedding change rate. These errors are not truly independent, but can be approximately separated and compared by repeatedly embedding different payloads in each cover.

We picked six embedding change rates, $\beta \in \{0, 0.025, 0.05, 0.125, 0.25, 0.375\}$, and embedded 200 random payloads into each of the 4567 images in the training set, using both Jsteg and nsF5. We term each combination of embedding algorithm, change rate, and cover image, a *cell*, so that each cell contains estimates of 200 equally-sized but differently-located payloads (except for cells with no payload, for which there is only one possible object per cover). The total experimental base comprises 9.1M attacks.

First, we measure the shape of the within- and between-image errors, without regard to their magnitude. A good way to examine the tails of a distribution is with a log-log empirical cdf plot: picking a single cell for each of Jsteg and nsF5, we display such plots for the 200 estimates, in the upper part of Fig. 3 (the data for all such plots has the mean subtracted, to center the distribution, and the Gaussian fit is selected to match the sample variance). There appears to be an excellent fit with the Gaussian distribution, and we see similar results across all images and all embedding rates. A summary of these fits is found in two columns of Tab. 2: we computed Shapiro-Wilk tests[18] for normality in every cell, and display the proportion of cells with p values above 0.1. If the cells are truly Gaussian, we would expect that 90% of cells would pass this test, and the displayed data are in accordance with this prediction. More precisely, we can say that any deviation from normality is small enough to be undetectable with 200 samples per cell.

Once we know that the estimates within each cell are Gaussian, we can safely average them to (almost entirely) remove the within-image error. Then the cell means describe the between-image error, and we plot log-log empirical cdfs for one particular embedding rate in the lower part of



(a) Within-image error, Jsteg

(b) Within-image error, nsF5

(c) Between-image error, Jsteg

(d) Between-image error, nsF5

Figure 3. Log-log tail plots of empirical distributions of within- (above) and between- (below) image errors, for Jsteg (left) and nsF5 (right) embedding. Gaussian and Student t fits are shown.

Fig. 3. These data are clearly not Gaussian, but there is a good fit with the Student t-distribution. The number of degrees of freedom in the t-distribution is estimated around 3 (this is the case for all embedding rates), but we see in the tail plots that the distribution tails seem slightly heavier and in fact they fit better with around 2 df. This accords closely with what was observed for LSB replacement estimators in Refs. 1 and 3. It is somewhat surprising that quantitative steganalysis of JPEG embedding via SVR displays the same characteristics as quantitative steganalysis of spatial-domain embedding via structural steganalysis, particularly since their modes of operation are so different: there was no particular reason to believe that their tails should decay at the same rate, but this does appear to happen. Incidentally, the heavy between-image tails mean that it would have been unsound to measure sample variation or standard deviation (or mean square error) for our estimators: the population variance may well be infinite, but even if finite the sample variance will converge only very slowly to the true value.

	Jsteg				nsF5			
β	Shapiro-Wilk $p > 0.1$	Between IQR $\Delta Q(Z_{\text{cov}})$	Within IQR $\Delta Q(Z_{\text{pos}})$	Flips IQR $\Delta Q(Z_{\text{flip}})$	Shapiro-Wilk $p > 0.1$	Between IQR $\Delta Q(Z_{\text{cov}})$	Within IQR $\Delta Q(Z_{\text{pos}})$	Flips IQR $\Delta Q(Z_{\text{flip}})$
0	–	3.63	0.00	0.00	–	7.74	0.00	0.00
0.025	90.2%	3.23	1.52	0.28	93.9%	6.99	2.81	0.29
0.05	89.9%	3.02	1.91	0.39	93.9%	6.79	3.52	0.41
0.125	90.2%	2.79	2.57	0.59	93.7%	6.93	4.78	0.62
0.25	89.8%	2.87	3.25	0.78	94.2%	8.31	6.77	0.81
0.375	90.3%	3.69	3.56	0.87	94.2%	10.63	8.47	0.91
		$\times 10^{-3}$	$\times 10^{-3}$	$\times 10^{-3}$		$\times 10^{-3}$	$\times 10^{-3}$	$\times 10^{-3}$

Table 2. Comparison of magnitudes of between- and within-image errors, and embedding change uncertainty, measured by inter-quartile range (IQR) for six embedding change rates. Also shown is the number of cells passing the Shapiro-Wilk test for normality of within-image error, at 10% significance.

Finally, we compare the magnitudes of the within- and between-image errors, also including the theoretical predictions for embedding change rate variation (if there are n locations and 2β locations are used for payload, without matrix or source coding, under mild assumptions about random embedding the number of embedding changes will follow a $\text{Bi}(n, \beta)$ distribution; n is not equal for Jsteg and nsF5 because Jsteg does not use coefficients equal to 1 and F5 does not use DC coefficients). Bias is assigned to between-image error and, for this analysis, discounted. Because of the heavy tails in the between-image error, we use inter-quartile range (IQR) as a measure of spread. For 6 embedding change rates, the IQRs of these three error factors are displayed in Tab. 2. Because the errors Z_{flip} and Z_{pos} depend (somewhat) on the covers, the table displays the average values for these IQRs.

Similarly to the results for LSB replacement estimators in Refs. 1 and 3, the magnitude of Z_{flip} is generally negligible. Here, the IQRs of within-image error Z_{pos} are not negligible, even for fairly small embedding rates: this is in contrast to the spatial-domain estimators. Also, the between-image error Z_{cov} remains stable or increases at larger embedding rates, whereas the opposite was observed to hold for spatial-domain estimators.

4.3. Comparison with prior art — Jsteg

In this experimental section, we compare the accuracy of our quantitative steganalyzer for Jsteg with the most accurate steganalyzers known today. We chose Jsteg because it is currently the best studied JPEG steganography algorithm with many known accurate quantitative steganalyzers. As shown in Ref. 25, it is possible to construct quantitative steganalyzers for Jsteg by adapting steganalysis methods developed for LSB steganography in the spatial domain.

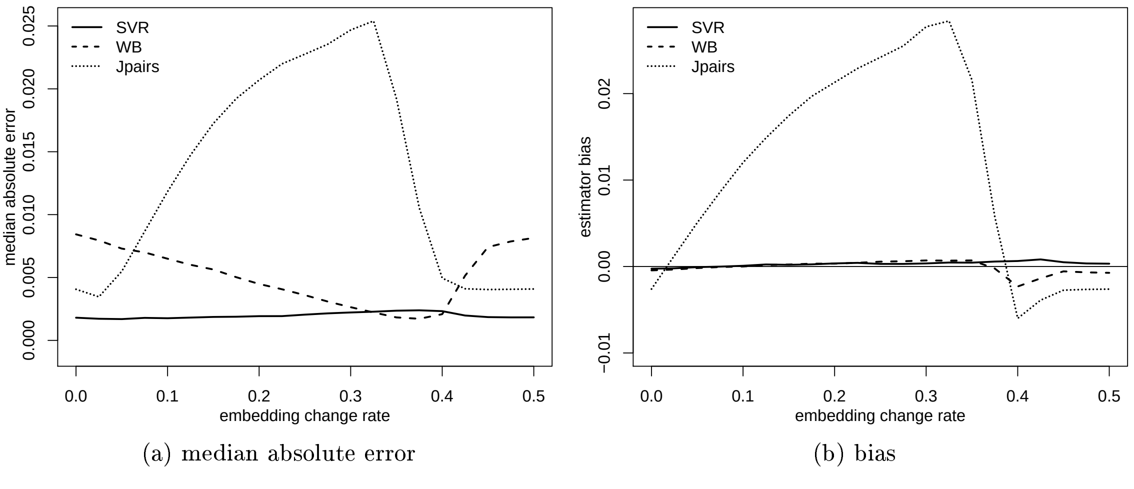
Among the multitude of methods described in Ref. 25, we selected Jpairs and Weighted Non-steganographic Borders Attack (WB) and compared their performance to the quantitative steganalyzer developed in Subsection 3.2. According to Ref. 25, the Jpairs quantitative steganalyzer was one of the most accurate quantitative steganalyzers for Jsteg. The algorithms were compared on the 4567 images in the testing set, at 21 embedding change rates from the set

$\beta \in \mathcal{B} \triangleq \{0, 0.025, 0.05, \ldots, 0.475, 0.5\}$ (the images were the same images used in the previous two subsections.)

Figure 4 shows that the quantitative steganalyzer with SVR has almost always better performance than both Jpairs and WB attacks. Moreover, its performance is more stable with respect to the change rate. Contrary to the conclusion reached in Ref. 25, we found that the WB attack was more precise than Jpairs attack; this discrepancy could be caused by us using a different database of images. Note, though, that Fig. 4 over-states the accuracy of JPairs, because the JPairs method sometimes fails to produce an estimate at all. This happens most often for large embedding rates: for $\beta = 0.375$, as many as one third of estimates fail. The SVR and WB methods never fail to produce an estimate.

5. CONCLUSION

Up until now, quantitative steganalysis was a collection of clever tricks developed for a rather small set of specific embedding methods. Moreover, there existed an alarming absence of a solid foundation that would enable easy construction of quantitative attacks for arbitrary steganographic systems. In this paper, we proposed a new approach to quantitative steganalysis that is general and widely-applicable to essentially any embedding method, without necessarily knowing the exact details of the embedding algorithm. The idea is to use the features from blind steganalysis and model the relationship between the features' location and the change rate (relative number of embedding changes) using regression. Indeed, it is quite feasible to expect that when a steganographic method is detectable using blind steganalysis, we should be able to extract some quantitative information from the feature vector rather than just the binary membership of the set of cover or stego images. On the example of eight steganographic algorithms in the JPEG domain, we have demonstrated the power of the proposed approach and showed that quantitative steganalyzers can be constructed for stegosystems for which no quantitative attacks were constructed so far. Moreover, the accuracy appears to be at least as good as the accuracy of known quantitative steganalyzers (this was shown



(a) median absolute error (b) bias

Figure 4. Comparison of the accuracy of Jpairs and Weighted Nonsteganographic Borders Attack (WB) with the support vector regression (SVR) quantitative steganalyzer, at 21 embedding change rates.

only for Jsteg). We also showed that the within-image error was significant in magnitude, and the between-image error has heavy tails so that authors must take care to use robust measures of accuracy: variance and mean square error are unsound in such a circumstance.

We believe that in this paper we have only scratched the surface of possibilities that we view as quite enormous. An intriguing possibility is to combine quantitative LSB estimators, such as triples, couples, SPA, and WS estimators and use them in the proposed framework to construct a new quantitative steganalyzer from them. Another direction possibly worth exploring is improving the control of the false positive rate in targeted blind steganalysis (blind steganalyzer trained as targeted) due to the fact that the estimated change rate is a scalar quantity. We also plan to investigate this approach for blind steganalyzers in the spatial domain.

6. ACKNOWLEDGMENTS

Tomáš Pevný is supported by the National French projects Nebbiano ANR-06-SETIN-009, ANR-RIAM Estivale, and ANR-ARA TSAR. The work of Jessica Fridrich was supported by Air Force Office of Scientific Research under the research grant number FA9550-08-1-0084. 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. Andrew Ker is a Royal Society University Research Fellow. We would like to thank Andreas Westfeld for providing code and support for WS and Jpairs attacks on Jsteg and Rainer Böhme for the original code used to produce log-log empirical cdf plots.

REFERENCES

1. R. Böhme. *Improved Statistical Steganalysis using Models of Heterogeneous Cover Signals.* PhD thesis, Technische Universität Dresden, September 2008.
2. R. Böhme. Weighted stego-image steganalysis for JPEG covers. In K. Solanki, editor, *Information Hiding, 10th International Workshop*, Lecture Notes in Computer Science, pages 178–194, Santa Barbara, CA, June 19–21, 2008. Springer-Verlag, New York.
3. R. Böhme and A. D. Ker. A two-factor error model for quantitative steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 59–74, San Jose, CA, January 16–19, 2006.
4. S. Dumitrescu, X. Wu, and Z. Wang. Detection of LSB steganography via sample pair analysis. In F. A. P. Petitcolas, editor, *Information Hiding, 5th International Workshop*, volume 2578 of *Lecture Notes in Computer Science*, pages 355–372, Noordwijkerhout, The Netherlands, October 7–9, 2002. Springer-Verlag, New York.
5. J. Fridrich and M. Goljan. On estimation of secret message length in LSB steganography in spatial domain. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306, pages 23–34, San Jose, CA, January 19–22, 2004.
6. J. Fridrich, M. Goljan, D. Hogea, and D. Soukal. Quantitative steganalysis of digital images: Estimating the secret message length. *ACM Multimedia Systems Journal*, 9(3):288–302, 2003.
7. J. Fridrich, M. Goljan, and D. Soukal. Perturbed quantization steganography. *ACM Multimedia System Journal*, 11(2):98–107, 2005.

8. J. Fridrich, T. Pevný, and J. Kodovský. Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities. In J. Dittmann and J. Fridrich, editors, *Proceedings of the 9th ACM Multimedia & Security Workshop*, pages 3–14, Dallas, TX, September 20–21, 2007.
9. S. Hetzl and P. Mutzel. A graph–theoretic approach to steganography. In J. Dittmann, S. Katzenbeisser, and A. Uhl, editors, *Communications and Multimedia Security, 9th IFIP TC-6 TC-11 International Conference, CMS 2005*, volume 3677 of *Lecture Notes in Computer Science*, pages 119–128, Salzburg, Austria, September 19–21, 2005.
10. A. D. Ker. A general framework for structural analysis of LSB replacement. In M. Barni, J. Herrera, S. Katzenbeisser, and F. Pérez-González, editors, *Information Hiding, 7th International Workshop*, volume 3727 of *Lecture Notes in Computer Science*, pages 296–311, Barcelona, Spain, June 6–8, 2005. Springer-Verlag, Berlin.
11. Y. Kim, Z. Duric, and D. Richards. Modified matrix encoding technique for minimal distortion steganography. In J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, editors, *Information Hiding, 8th International Workshop*, volume 4437 of *Lecture Notes in Computer Science*, pages 314–327, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.
12. A. Latham. `http://linux01.gwdg.de/~alatham/stego.html`.
13. K. Lee and A. Westfeld. Generalized category attack—improving histogram-based attack on JPEG LSB embedding. In T. Furon, F. Cayre, G. Doërr, and P. Bas, editors, *Information Hiding, 9th International Workshop*, Lecture Notes in Computer Science, pages 378–392, Saint Malo, France, June 11–13, 2007. Springer-Verlag.
14. S. Lyu and H. Farid. Steganalysis using higher-order image statistics. *IEEE Transactions on Information Forensics and Security*, 1(1):111–119, 2006.
15. T. Pevný. *Kernel Methods in Steganalysis.* PhD thesis, Binghamton University, SUNY, May 2008.
16. T. Pevný and J. Fridrich. Merging Markov and DCT features for multi-class JPEG steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 3 1–3 14, San Jose, CA, January 29 – February 1, 2007.
17. N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, Proceedings of the ACM Symposium on Applied Computing, August 13–17, 2001.
18. P. Royston. An extension of Shapiro and Wilk's test for normality to large samples. *Applied Statistics*, 31:115–124, 1982.
19. P. Sallee. Model-based steganography. In T. Kalker, I. J. Cox, and Y. Man Ro, editors, *Digital Watermarking, 2nd International Workshop*, volume 2939 of *Lecture Notes in Computer Science*, pages 154–167, Seoul, Korea, October 20–22, 2003. Springer-Verlag, New York.
20. Y. Q. Shi, C. Chen, and W. Chen. A Markov process based approach to effective attacking JPEG steganography. In J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, editors, *Information Hiding, 8th International Workshop*, volume 4437 of *Lecture Notes in Computer Science*, pages 249–264, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.
21. A. J. Smola, N. Murata, B. Scholkopf, and K. r. Muller. Asymptotically optimal choice of ϵ-loss for support vector machines. In *Proceedings of the 8th International Conference on Artificial Neural Networks, Perspectives in Neural Computing, pages 105 – 110*, pages 105–110. Springer Verlag, 1998.
22. A. J. Smola and B. Schöelkopf. A tutorial on support vector regression. Technical report, NeuroCOLT2 Technical Report NC2-TR-1998-030, 1998.

23. D. Soukal, J. Fridrich, and M. Goljan. Maximum likelihood estimation of secret message length embedded using $\pm k$ steganography in spatial domain. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 595–606, San Jose, CA, January 16–20, 2005.
24. D. Upham. `http://zooid.org/~paul/crypto/jsteg/`.
25. A. Westfeld. Generic adoption of spatial steganalysis to transformed domain. In K. Solanki, editor, *Information Hiding, 10th International Workshop*, Lecture Notes in Computer Science, pages 161–177, Santa Barbara, CA, June 19–21, 2008. Springer-Verlag, New York.
26. G. Xuan, Y. Q. Shi, J. Gao, D. Zou, C. Yang, Z. Z. P. Chai, C. Chen, and W. Chen. Steganalysis based on multiple features formed by statistical moments of wavelet characteristic functions. In M. Barni, J. Herrera, S. Katzenbeisser, and F. Pérez-González, editors, *Information Hiding, 7th International Workshop*, volume 3727 of *Lecture Notes in Computer Science*, pages 262–277, Barcelona, Spain, June 6–8, 2005. Springer-Verlag, Berlin.
27. T. Zhang and X. Ping. A fast and effective steganalytic technique against JSteg-like algorithms. In *Proceedings of the ACM Symposium on Applied Computing*, pages 307–311, Melbourne, FL, March 9–12, 2003.