A Feature Selection Methodology for Steganalysis

Yoan Miche, Benoit Roue, Amaury Lendasse, Patrick Bas

A Feature Selection Methodology for Steganalysis

Yoan Miche[1], Benoit Roue[2], Amaury Lendasse[1], Patrick Bas[1 2]

[1] Laboratory of Computer and Information Science
Helsinki University of Technology
P.O. Box 5400 FI-02015 HUT FINLAND
[2] Laboratoire des Images et des Signaux de Grenoble
961 rue de la Houille Blanche Domaine universitaire
B.P. 46 38402 Saint Martin d'Hères cedex FRANCE

Abstract. This paper presents a methodology to select features before training a classifier based on Support Vector Machines (SVM). In this study 23 features presented in [1] are analysed. A feature ranking is performed using a fast classifier called K-Nearest-Neighbours combined with a forward selection. The result of the feature selection is afterward tested on SVM to select the optimal number of features. This method is tested with the Outguess steganographic software and 14 features are selected while keeping the same classification performances. Results confirm that the selected features are efficient for a wide variety of embedding rates. The same methodology is also applied for Steghide and F5 to see if feature selection is possible on these schemes.

1 Introduction

The goal of steganographic analysis, also called steganalysis, is to bring out drawbacks of steganographic schemes by proving that an hidden information is embedded in a content. A lot of steganographic techniques have been developed in the past years, they can be divided in two classes: *ad hoc* schemes (schemes that are devoted to a specific steganographic scheme) [1–3] and schemes that are generic and that use classifiers to differentiate original and stego images[4, 5]. The last ones work in two steps, generic feature vectors (high pass components, prediction of error...) are extracted and then a classifier is trained to separate stego images from original images. Classifier based schemes have been more studied recently, and lead to efficient steganalysis. Thus we focus on this class in this paper.

1.1 Advantages of feature selection for steganalysis

Performing feature selection in the context of steganalysis offers several advantages.

- it enables to have a more rational approach for classifier-based steganalysis: feature selection prunes features that are meaningless for the classifier;

– feature selection may also be used to improve the classification performance of a classifier (in [6] it is shown that the addition of meaningless features decreases the performance of a SVM-based classifier);
– another advantage of performing feature selection while training a classifier is that the selected features can help to point out the features that are sensitive to a given steganographic scheme and consequently to bring a highlight on its weaknesses.
– The last advantage of performing feature selection is the reduction of complexity for both generating the features and training the classifier. If we select a set of N features from a set of M, the training time will be divided by M/N (this is due to the linear complexity of classifiers regarding the dimension). The same complexity reduction can also be obtained for feature generation if we assume that the complexity to generate each feature is equivalent.

2 Fridrich's Features

The features used in this study were proposed by Fridrich *et al* [1]. All features are computed in the same way: a vector functional F is applied to the *stego JPEG image* J_1 and to the virtual *clean JPEG image* J_2 obtained by cropping J_1 with a translation of 4×4 pixels. The feature is finally computed taking the $L1$ of the difference of the two functionals :

$$f = ||F(J_1) - F(J_2)||_{L1}. \tag{1}$$

The functionals used in this paper are described in the Table 1.

Functional/Feature name	Functional **F**
Global histogram	$\mathbf{H}/\|\|\mathbf{H}\|\|$
Individual histogram for 5 DCT Modes	$\mathbf{h}^{21}/\|\|\mathbf{h}^{21}\|\|$,$\mathbf{h}^{12}/\|\|\mathbf{h}^{12}\|\|$,$\mathbf{h}^{13}/\|\|\mathbf{h}^{13}\|\|$, $\mathbf{h}^{22}/\|\|\mathbf{h}^{22}\|\|$,$\mathbf{h}^{31}/\|\|\mathbf{h}^{31}\|\|$
Dual histogram for 11 DCT values $(-5,\ldots,5)$	$\mathbf{g}^{-5}/\|\|\mathbf{g}^{-5}\|\|$,$\mathbf{g}^{-4}/\|\|\mathbf{g}^{-4}\|\|$,$\mathbf{g}^{-3}/\|\|\mathbf{g}^{-3}\|\|$,$\mathbf{g}^{-2}/\|\|\mathbf{g}^{-2}\|\|$,$\mathbf{g}^{-1}/\|\|\mathbf{g}^{-1}\|\|$, $\mathbf{g}^{0}/\|\|\mathbf{g}^{0}\|\|$,$\mathbf{g}^{1}/\|\|\mathbf{g}^{1}\|\|$,$\mathbf{g}^{2}/\|\|\mathbf{g}^{2}\|\|$,$\mathbf{g}^{3}/\|\|\mathbf{g}^{3}\|\|$,$\mathbf{g}^{4}/\|\|\mathbf{g}^{4}\|\|$,$\mathbf{g}^{5}/\|\|\mathbf{g}^{5}\|\|$
Variation	$\mathbf{V}$
$L1$ and $L2$ blockiness	$\mathbf{B}_1$, $\mathbf{B}_2$
Co-occurrence	N_{00}, N_{01}, N_{11}

Table 1. List of the 23 used features.

3 Classifiers for steganalysis

This section presents two classifiers that differ in term of complexity and a method to estimate the mean and variance of the classification accuracy obtained by any classifier.

- K-Nearest Neighbours: the K-NN classifiers use an algorithm based on a majority vote: using a norm (usually Euclidean), the K nearest points from the

one to classify are determined. The classification is then based on the class that belongs to the most numerous closest points, as shown on the figure (Fig 1). The choice of the K value is dependent on the data, and the best value is found using using a leave-one-out cross-validation procedure [7]. Note that if K-NN classifiers are usually less accurate than SVM classifiers, nevertheless, the computational time for training a K-NN is around 10 times smaller than for training a SVM.

- Support Vector Machines: SVM classification uses supervised learning systems to map in a non-linear way the features space into a higher dimensional feature space [8]. A hyper-plane can then be found in this high-dimensional space, which is at the maximum distance from the nearest data points of the two classes so that points to be classified can benefit from this optimal separation.

- Bootstrapping for noise estimation: the bootstrap algorithm enables to have a confidence interval for the performances [7]. A random mix with repetitions of the test set is created, and then used with the SVM model computed before with a fixed train set. This process is repeated R times and thus gives by averaging a correct noise estimation when N is large enough.

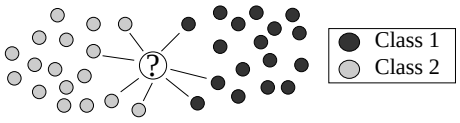

Fig. 1. Illustration of the K-NN algorithm. Here, $K = 7$: The Euclidean distance between the new point (?) and the 7 nearest neighbours is depicted by a line. In this case we have the majority for the light grey (4 nearest neighbours): the new point is said to be of class 2.

4 Feature selection methods

This section presents two different feature selection methods.

- Exhaustive search: in this case, we use a full scan of all possible features combinations and keep the one giving the best result. If you consider N features, the computational time to perform the exhaustive search equals the time to train/test one classifier multiplied by $2^N - 1$. Consequently this method can only be used with fast classification algorithms.

- The "forward" selection algorithm: The forward approach proposes a suboptimal but efficient way to incrementally select the best features [9]. The following steps illustrate this algorithm :

1. try the $\alpha_{i,i\in[\![1,N]\!]}$ features one by one;
2. keep the feature α_{i1} with the best results;
3. try all couples with α_{i1} and one feature among the remaining $N - 1$;

4. keep the couple $(\alpha_{i1}, \alpha_{i2})$ giving the best results;
5. try all triplets with $(\alpha_{i1}, \alpha_{i2})$ and one feature among the remaining $N-2$;
6. ... iterate until none remains.

The result is an array containing the N the features ranked by minimum error. The computational time is equal to $N \times (N+1)/2$ multiplied by the time spent to train/test one classifier.

4.1 Applying feature selection to SVMs

Using the forward algorithm directly on SVM is too time-consuming. Consequently we propose to perform the feature selection for SVMs in three steps depicted on Figure 2.

1. Forward using K-NN: in this step, we use the explained forward algorithm with a K-NN classification method to rank features vectors. Since the K-NN is fast enough, it is possible to run this step in a reasonable time.
2. SVM and Bootstrapping: using the ranked features list found by the K-NN forward algorithm, we run 23 SVMs using the 23 different feature vectors, and a bootstrap on the test set, with approximately 5000 iterations.
3. Features selection: in the end, the curve from the bootstrap data shows that within the noise estimation, we can reduce the number of features, based on the fact that the addition of some features degrades the classification result. Within the noise range, the first $L < N$ selected features present the best compromise for a same classification performance.

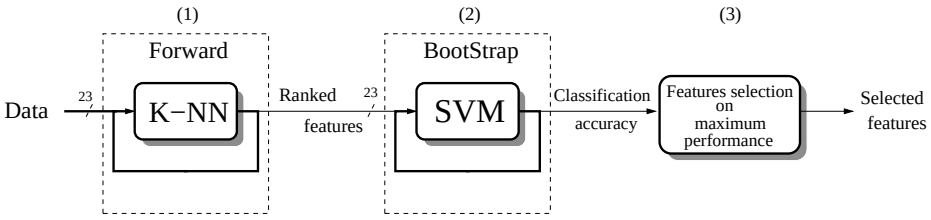

Fig. 2. Feature selection steps : features are first ranked by importance by the K-NN forward algorithm (1), SVMs give then improvement and an accuracy estimation thanks to a bootstrap (2). Features are in the end taken from the best SVM result (3).

5 Experimental Results

The experiments have been performed using a set of 5075 images from 5 different digital cameras (all over 4 megapixels). A mix of these images has then been made, and half of them have been watermarked using Outguess 0.2 [10], with and embedding rate of 10% of non zero quantised DCT coefficients. Each image has been scaled and cropped to 512×512, converted in grey levels and compressed using a JPEG quality factor of 80%. The extracted features from the 5075 images have then been divided in a training (1500 samples) and test set (3575 samples). The SVM library used is the libSVMtl [11].

5.1 Accuracy of KNN with feature selection

We present here (Fig 3) the classification accuracy of the forward algorithm using the K-NN method. In our case, the decision on whether to keep or leave out a feature has been made only on the results of the leave-one-out (i.e. using only the training set). As one can see from the curves, it finds the best set of features with only 6 of them (Leave-one-out classification rate around 0.705). Adding more features only results here in a degradation of the classification result.

But tryouts using only those 6 features have proven that it is not the best solution for SVM. Consequently, we choose to use this step of the process only to obtain a ranking of the features.

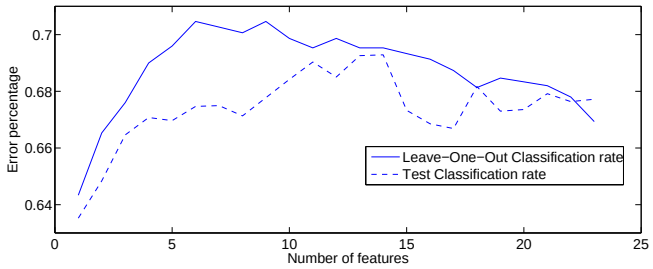

Fig. 3. The K-NN accuracy using the forward algorithm.

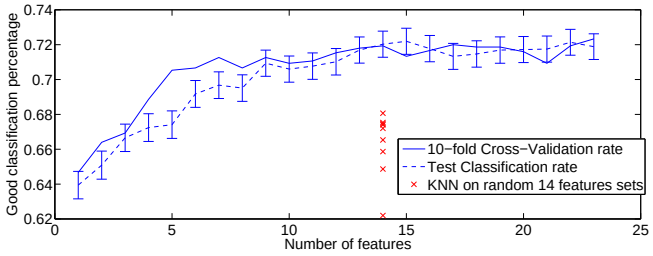

Fig. 4. The SVM accuracy using the result of the K-NN forward. The vertical segments show the noise estimation obtained using the bootstrap technique. Crosses present the results of K-NN on 10 sets of 14 features randomly selected.

5.2 Accuracy of SVM with feature selection

Since the 6 forward K-NN selected features are not enough, this process step uses all features, but according to the ranking order given by the forward K-NN. The SVM is thus used (RBF-type kernel), with the same training and test sets. As mentioned before, we use here a bootstrap technique to have a more robust result and an estimation of the noise. As it can be seen (cf Figure 4), the best accuracy is obtained using 14 features, achieving 72% of correct classification

(10-fold cross-validation). In this case, the test error curve stays close to the 10-fold one. For comparison purposes we have also plotted the performance of the K-NN on sets of 14 features taken randomly from the original ones. As illustrated on figure 3, it never achieves more than 68% in correct classification (training). This proves that the selected features using the forward technique are relevant enough.

5.3 Selected features

Table 2 presents the set of features that have been selected. For sake of simplicity the cardinal part for each feature has been skipped. Table 3 presents the final results from the explained method. It can be seen that the selected 14 features set is giving better results (within the noise estimation) than with all 23 features. Note that even-though the result is always superior using only 14 features, the noise is still to take into account (Fig 4).

N_{11}	$\mathbf{g}^{-1}$	$\mathbf{g}^{-2}$	$\mathbf{g}^{-3}$	$\mathbf{g}^{1}$	$\mathbf{H}$	$\mathbf{g}^{4}$	$\mathbf{g}^{0}$	$\mathbf{h}^{21}$	$\mathbf{g}^{-4}$	N_{01}	$\mathbf{B}_2$	$\mathbf{h}^{13}$	$\mathbf{h}^{12}$

Table 2. List of the selected features done by the forward algorithm using K-NN. Feature are ordered according to the forward algorithm.

Embedding rate	14 features	23 features
10%	72.0% (71.9%)	71.9% (72.3%)
25%	88.0% (92.9%)	87.2% (93.1%)
50%	97.8% (99.3%)	97.0% (99.2%)
75%	99.2% (99.7%)	98.0% (99.8%)

Table 3. The test error (in plain) and 10-fold cross-validation error (bracketed) for 14 and 23 features at different embedding rates.

5.4 Weaknesses of Outguess

Feature selection enables to link the nature of the selected features with Outguess v0.2, the steganographic software that has been used [10] and then to outline its weaknesses. We recall that Outguess embeds information by modifying the least significant bits of the quantised DCT coefficients of a JPEG coded image. In order to prevent easy detection, the algorithm does not embed information into coefficients equal to 0 and 1. Outguess also preserves the global histogram of the DCT coefficients between the original and stego image by correcting statistical deviations.

The selected features presented in Table 2 present strong links with the way the embedding scheme performs:

- The feature N_{11} is the first feature selected by the forward algorithm and describes the difference between co-occurrence values for coefficients equal to 1

or -1 on neighbouring blocks. This feature seems to react mainly to the flipping between coefficients -1 and -2 during the embedding. Note also that coefficients -2 and 2 are, after 0 and 1, the most probable DCT coefficients in a given image.

- The second and third selected features are $\mathbf{g}^{-1}$ and $\mathbf{g}^{-2}$. They represent the dual histogram of coefficients respectively equal to -1 and -2 with respect to their coordinates. Once again, these features concern the same coefficients than previously but only on the first order (histogram).

- We can notice that nearly half of features related to the dual histogram have been selected. Due to symmetry one might think that features $\mathbf{g}^{-5}$, $\mathbf{g}^{-4}$, $\mathbf{g}^{-3}$, $\mathbf{g}^{-2}$ carry respectively the same information than $\mathbf{g}^{5}$, $\mathbf{g}^{4}$, $\mathbf{g}^{3}$, $\mathbf{g}^{2}$, consequently it is not surprising that only one in each set has been chosen (with the exception of $\mathbf{g}^{-4}$ and $\mathbf{g}^{4}$).

- Note that it can seem first curious that features $\mathbf{g}^{0}$ and $\mathbf{g}^{1}$ have been selected as meaningful features for the classifier because they are not modified by the embedding algorithm. However, these features can have been affected on the stego and cropped image: coefficients equal to 2 or 3 on the stego image can be reduced to 1 or 2 on the cropped image. Another reason can be that feature $\mathbf{g}^{1}$ can be selected in association with feature $\mathbf{g}^{-1}$ because it has a different behaviour for watermarked images but a similar behaviour for original images.

5.5 Obtained results for other steganographic schemes

This feature selection method has also been tested for two other popular steganographic schemes called F5 and Steghide. Our test confirms that it is also possible to use K-NN-based feature selection on Steghide and to select 13 features which provide similar performances. The list of the 13 selected features is given on table 4 and the performances for different embedding rates is given on table 5. However, we have noticed that for the F5 algorithm performing feature selection is not efficient if the ratio of selected features is below 80%. Forward feature selection for F5 selects still 15 features and backward feature selection selects 22 features. The high number of selected features means that nearly each of the initial feature for F5 is significant for the detection process. Such a consideration is not surprising because F5 is the most undetectable of the three analysed steganographic schemes.

N_{00}	$\mathbf{g}^{2}$	$\mathbf{h}^{22}$	$\mathbf{H}$	$\mathbf{g}^{5}$	N_{01}	$\mathbf{g}^{-2}$	$\mathbf{g}^{-1}$	$\mathbf{h}^{13}$	$\mathbf{g}^{-5}$	$\mathbf{g}^{1}$	$\mathbf{g}^{5}$	$\mathbf{V}$

Table 4. List of the 13 selected features done by the forward algorithm using K-NN for Steghide. Features are ordered according to the forward algorithm.

Embedding rate	13 features	23 features
10%	67.28% (69.39%)	68.73% (68.79%)
25%	75.21% (77.90%)	77.81% (81.03%)
50%	91.66% (90.77%)	93.25% (93.79%)
75%	97.84% (97.93%)	98.37% (98.88%)

Table 5. The test error (in plain) and 10-fold cross-validation error (bracketed) for 13 and 23 features at different embedding rates for Steghide algorithm.

6 Conclusions and Future Works

This paper proposes a methodology to select meaningful features for a given steganographic scheme. Such a selection enables both to increase the knowledge on the weakness of a steganographic algorithm and to reduce its complexity while keeping the classification performances. Our future works will consist in combining input selection techniques with feature scaling in order to increase the performance of the classifiers.

References

1. J.Fridrich. (In: 6th Information Hiding Workshop, LNCS, vol. 3200)
2. S.Dumitrescu, X.Wu, Z.Wang: Detection of LSB steganography via sample pair analysis. In: IEEE transactions on Signal Processing. (2003) 1995–2007
3. B.Roue, P.Bas, J-M.Chassery: Improving lsb steganalysis using marginal and joint probabilistic distributions. In: Multimedia and Security Workshop, Magdeburg (2004)
4. S.Lyu, H.Farid: Detecting hidden message using higher-order statistics and support vector machine. In: 5^{th} International Workshop on Information Hiding, Netherlands (2002)
5. T.Pevny, J.Fridrich: Toward multi-class blind steganalyser for jpeg images. In: International Workshop on Digital Watermarking, LNCS vol. 3710. (2005) 39–53
6. Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., Vapnik, V.: Feature selection for SVMs. In Leen, T.K., Dietterich, T.G., Tresp, V., eds.: NIPS, MIT Press (2000) 668–674
7. Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. Chapman and Hall, London (1993)
8. Zhang, T.: An introduction to support vector machines and other kernel-based learning methods. AI Magazine (2001) 103–104
9. Rossi, F., Lendasse, A., François, D., Wertz, V., Verleysen, M.: Mutual information for the selection of relevant variables in spectrometric nonlinear modelling. Chemometrics and Intelligent Laboratory Systems, vol 80 (2006) 215–226
10. Provos, N.: Defending against statistical steganalysis. In USENIX, ed.: Proceedings of the Tenth USENIX Security Symposium, August 13–17, 2001, Washington, DC, USA, USENIX (2001)
11. Ronneberger, O.: Libsvmtl extensions to libsvm. http://lmb.informatik.uni-freiburg.de/lmbsoft/libsvmtl/ (2004)