Extracting Relevant Features of Steganographic Schemes by Feature Selection Techniques

Yoan Miche[1,2], Patrick Bas[1,2], Amaury Lendasse[1], Christian Jutten[2], and Olli Simula[1]

[1] Helsinki University of Technology - Laboratory of Computer and Information Science
P.O. Box 5400, FI-02015 HUT, FINLAND
[2] INPG - Gipsa-Lab,
INPG, 46 avenue Félix Viallet, 38031 Grenoble cedex, FRANCE

Abstract. This paper presents a methodology for steganalysis based on a set of 193 features with two main goals. The first goal is to determine a sufficient number of images for effective training of a classifier in the obtained high-dimensional space. Second goal is to use feature selection to select most relevant features for the desired classification. Dimensionality reduction is performed using a forward selection and reduces the original 193 features set by a factor of 13, with overall same performance. Additionally, two new tools are proposed for feature selection in order to validate and possibly analyze further the current results.

1 Introduction

Steganalysis' main goal is to automatically detect, with the possibly highest accuracy, the presence of a secretly embedded content in another document. This can be seen as a typical classification problem since an optimal separation between stego images and genuine ones is seeked.

For this matter, various features are extracted from the considered images. This number of features is nowadays growing consequently (as seen on Fig 1) and might be too important for the classifiers typically employed, as well as for an interpretation of the results.

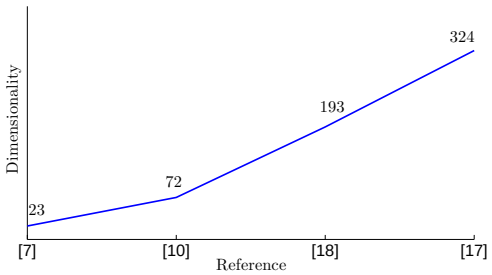


Fig. 1. Dimensionality of features sets

The very first examples of LSB-based steganalysis made use of less than ten features, with an adapted and specific methodology for each stego algorithm. The idea of "universal steganalyzers" then became popular. In 1999, Westfeld proposes a χ^2-based method, on the LSB of DCT coefficients [19].

Five years after, Fridrich in [7] uses a set of 23 features obtained by normalizations of a much larger set, whilst Farid *et al.* already proposed in 2002 a

set of 72 features [10]. Since then, an increasing number of research works are using supervised learning based classifiers in very high dimensional spaces. The recent work of Y. Q. Shi *et al.* [17] is an example of an efficient result but using 324 features based on JPEG blocks differences modeled by Markov processes.

All these results are achieving better and better performance in terms of good classification. Meanwhile, there are some clear side-effects to this growing feature spaces. First of, as long as performances seem to increase, people tend to keep on adding new features to their sets, regardless of the resulting dimensionality. It has been shown [4, 11] that the feature space dimensionality in which the considered classifier is trained in, can have a crucial impact on its performances. Second, the computational complexity of the classifier's training is another important aspect, since most of the widely used classifiers have at least a linear relationship to the dimensionality of the space. Third, a high dimensional space requires an important number of samples (images in this case) for a correct training of the classifier and accurate model parameters. The relationship between dimensionality and number of samples is not trivial and a sufficient number of samples has to be determined for the considered problem.

Another matter is about the interpretability of such results on high dimensional spaces. A correct analysis of the results and of the possible weaknesses of the stego algorithms seems to be hard and long to perform, with so many features.

We propose a general methodology to address some high dimensionality issues. The next section proposes to illustrate the main problems related to this dimensionality. Section 3 presents the main tools and classifiers used in this paper which are part of the methodology. Results using the stego algorithm Outguess [13], and a set of 193 features from Fridrich [18] follow, with an overall analysis in section 5. Some new tools to investigate the current results and possibly extend them are finally presented in section 6.

2 Side-effects of the growing number of features

The common term "curse of dimensionality" [3] refers to a wide range of problems related to a high number of features. Our concern is focused on four main aspects of this high-dimensionality, namely the lack of interpretability, the complexity increase, the possible sensitivity to irrelevant features and the "need" for more data points, related to the empty space phenomenon as presented in [4, 12, 11].

2.1 The need for data points

As an example, when considering as few as five points in a three dimensional space, the underlying structure is impossible to infer, and so is the creation of a model for it. On the contrary, with hundreds to thousands of points it becomes possible to see clusters, relationships between dimensions and such.

More generally, in order for any tool to be able to analyze and find a structure within the data, the number of needed points is growing exponentially with the dimension. Indeed, consider a d-dimensional unit side hypercube, the number of points needed to fill the Cartesian grid of step ϵ inside of it, is growing as $O((1/\epsilon)^d)$. Thus, using a common grid of step 1/10 and a dimension of 10, it requires 10^{10} points to fill the grid.

In practice, most of data analysis use at least 10 to 20 dimensions, implying a "needed" number of points impossible to achieve. As a consequence, the feature space may be not correctly filled with data points, which can give wrong models when building classifiers, having to extrapolate for the missing points.

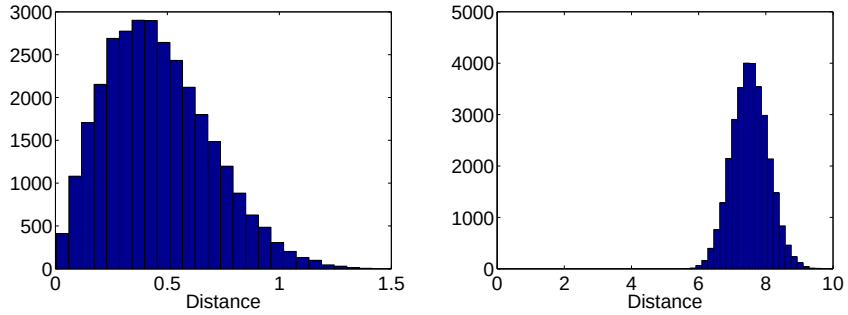
The situation is globally not limited to this lack of data points because of the dimensionality. The reason why so many points are needed when dimension increases is the empty space phenomenon, related to the positioning of randomly distributed points in high dimensions.

2.2 The empty space phenomenon and distance concentration

On a theoretical point of view: draw points from a normal distribution and consider the probability to have a point at distance r from the center of the distribution. It is given by the probability density function:

$$f(r,d) = \frac{r^{d-1}}{2^{d/2-1}} \cdot \frac{e^{-r^2/2}}{\Gamma(d/2)} \tag{1}$$

having its maximum at $r = \sqrt{d-1}$. Thus, when dimension increases, points are getting further from the center of the distribution. The phenomenon is the same for the uniform distribution as Fig 2 illustrates.



(a) Dimension 2, uniform dist. (b) Dimension 100, uniform dist.

Fig. 2. Overview of pairwise distances for uniformally distributed points in 2 and 100 dimensions: in dimension 2, distances are varying among the full possible amplitude (0 to $\sqrt{2}$); in dimension 100, all distances get very far from zero and tend to be much more concentrated close to the highest possible values.

In practical cases it is also observed [4, 12, 11] that high-dimensionality has this effect on the data distribution: average distances distribution between two points increases with dimension while the distances variance distribution tends to decrease, as in the theoretical cases above.

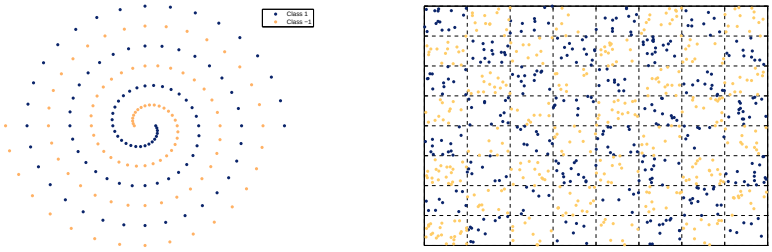
This distance concentration tends to decrease the relevance of distance-based classification since the common notions on which these classifiers are based are no longer fully reliable.

This fact and the possible disturbance of the classifier by unuseful dimensions especially motivates the use of feature selection techniques for dimensionality reduction: the forward technique presented in section 3 realizes such reduction and makes possible analysis and work in resulting lower dimensionality spaces.

2.3 Problems regarding irrelevant features

Here is presented the problem of the irrelevant and "parasite" dimensions: these dimensions bringing no useful information to the considered classification problem. An overall view of the effects of such sort of data on a SVM can be seen in [6].

The two points sets called *Whirls* and *Chessboard* are depicted in Fig 3. These two examples were made to be not trivial to classify.



(a) The double-whirl example (b) The Chessboard example

Fig. 3. Two classes on two different examples: the double-whirl and the chessboard one, not trivial to classify. Red points stand for one class and blue ones for another.

Table 1 then presents results of three widely used classifiers on these two special examples to which have been added Gaussian distributed dimensions, with no information relevant for the classification process.

As can be seen from Table 1, the classification rate drops significantly when adding these unuseful dimensions. The sensitivity of the classifier to these seems to be more noticeable for the SVM classifier than for the KNN. This robustness of the KNN to high dimensionality as well as to noisy or irrelevant data is well known.

	Whirls				Chessboard			
Noise	Without	1 dim.	2 dim.	3 dim.	Without	1 dim.	2 dim.	3 dim.
SVM	90%	61%	53%	52%	89%	63%	53%	51%
KNN	89%	74%	61%	60%	89%	75%	64%	62%
LVQ	62%	55%	51%	51%	64%	65%	52%	51%

Table 1. Effects of adding irrelevant Gaussian distributed dimensions to the classification problem.

2.4 Lack of interpretability

Eventhough the nearest neighbours classifiers keep good performances in high dimensions [4, 12, 11], other obvious problems of high dimensionality motivate the idea of feature selection. The interpretability is an important one: high performances can indeed be reached using the whole 193 features set proposed by Fridrich [18] for classification. Meanwhile, if looking for the weaknesses and reasons why these features react vividly to a specific algorithm, it seems rather impossible on this important set.

Reducing the required number of features to a small amount through feature selection enables to understand better why a steganographic model is weak on these particular details, highlighted by the selected features.

2.5 Increase of complexity

Computational time is another main reason. Nearest neighbours methods are usually implemented with a $O(d)$ dimension relationship, as for SVMs. Clearly, reducing the dimensionality by a significant order of magnitude gives much more achievable computational times. As a consequence, one can use more data points and lower this "missing points" effect.

The next section presents the proposed methodology for dimensionality reduction by feature selection. The classifiers used as well as the chosen selection techniques are detailed.

3 Methodology and techniques used

3.1 Classifiers and appropriate number of images

The "appropriate" number of images (or at least the minimum required for the dimensionality of our data) should be determined. For this matter, a KNN classifier is used with a Monte-Carlo technique [16]. This enables to estimate the noise and give a confidence interval for our results. We randomly draw (without repetitions) a subset of the whole data set, and use the obtained classifier on it.

For our experiments, two different types of classifiers have mainly been used: the first one, KNN, for its overall good performance even in high dimensional spaces, but most of all, because it is computationally very fast. SVM was also

chosen because it is among the classifiers giving the best results. Major drawback is of course the computational time. The main principles of these two major classifiers are briefly explained in the following.

K-Nearest Neighbours (KNN) KNN classifier is a supervised distance-based classifier, proposed by Devijver and Kittler in [1], usually using the Euclidean metric. It uses a majority vote among the k nearest neighbours classes to assign the class of the new considered point. The basic algorithm follows these steps:

Algorithm 1 KNN

```
for a fixed K value
for each point do
    Compute pairwise distances with all others
    for k = 1 : 2 : K do
        Take majority class of k nearest neighbours
    end for
end for
Compare obtained classes with ground truth for each k
Keep k value giving the best good classification rate
```

In practice, this algorithm is quite efficient, fast and only needs one parameter to be determined, which makes model structure selection easier. KNN algorithms usually run with a $O(N^2 d)$ dependency (with N the number of data points and d the dimension).

Cross-validation of the KNN model is achieved in this paper through Leave-One-Out [2].

Support Vector Machine (SVM) SVM have been created by Vapnik [14] in 1963 and then improved more recently (1992, 1995) by Boser, Guyon and Vapnik [5]. The original idea was to separate data using a hyperplane: this was a linear classifier. The extension of this method adds a non-linear part by the use of kernel functions.

Taking a set of d-dimensional data $\{\mathbf{x}_i\} \in \mathbb{R}^d, i \in [\![1, N]\!]$ labeled with classes $y_i \in \{-1, 1\}$ (restricted to two classes), the problem is to solve the optimization problem

$$\min_{\mathbf{w},b,\xi} ||\mathbf{w}||^2 + C \sum_{i=1}^{N} \xi_i, \tag{2}$$

under the constraint

$$(\mathbf{w}^T \psi(\mathbf{x}_i) + b) y_i \geq 1 - \xi_i, \tag{3}$$

where $\mathbf{w}$ is the separating hyperplane normal vector and ξ_i are slack variables enabling clean separation even if data is not linearly separable in the mapped

feature space. The training data $\mathbf{x}_i$ is indeed mapped into a higher dimensional space by the function ψ, space into which it is hoped to be more linearly separable.

This ψ function is related to the kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ by

$$K(\mathbf{x}_i, \mathbf{x}_j) = \psi(\mathbf{x}_i)^T \psi(\mathbf{x}_j). \tag{4}$$

In the study, Radial Basis Function kernels have been used, defined as

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma ||\mathbf{x}_i - \mathbf{x}_j||^2\right), \tag{5}$$

since they are the most commonly used kernels because of their good efficiency [15, 8].

As said before, the main issue with SVM is computational time, since equation 2 is a NP hard optimization problem. Most SVM algorithms (such as SMO [9]) are running with a dependency of order $O(N^2 d)$ up to $O(N^3 d)$, depending if $K(\mathbf{x}_i, \mathbf{x}_j)$ (eq. 5) can be stored in memory. With the choice of this particular kernel, we have two hyper-parameters – C and γ – to determine, expanding the previous line search for the k of KNN, to a grid search. Note that the determination of a good model for classification is then even more time-consuming. The validation for SVM is finally performed with a 10-fold cross-validation [2].

This again motivates the proposed feature selection step; the following is about the technique used to perform it, called forward selection.

3.2 Feature selection technique: Forward selection

The forward selection algorithm is a greedy algorithm proposed in [20]; the algorithm selects one by one the dimensions, trying to find the one that combines best with the already selected ones, as shown in the following algorithm 2 (with x^i denoting the i-th dimension of the data set):

Algorithm 2 Forward

$\mathbf{R} = \{x^i, i \in [\![1, d]\!]\}$
$\mathbf{S} = \emptyset$
while $\mathbf{R} \neq \emptyset$ **do**
  for $x^j \in \mathbf{R}$ **do**
    Evaluate performance with $\mathbf{S} \cup x^j$
  end for
  Set $\mathbf{S} = \mathbf{S} \cup \{x^k\}, \mathbf{R} = \mathbf{R} - x^k$ with x^k the dimension giving the best result in the loop
end while

Even if its capacity to isolate efficient features is obvious, the forward technique has some drawbacks: in the case where two or more features would have a high dependency and be "inefficient" when alone but very good when put together, forward might not take these into account soon enough in the selection

process. Nevertheless, the feature selection using forward has been showing very good results and seems to perform well on our feature set; this is presented in the next section.

4 Results in steganalysis

4.1 Our methodology

The three main points of the proposed methodology are detailed in the following. Fig. 4 illustrates the process. First is seeked a possibly good candidate for the

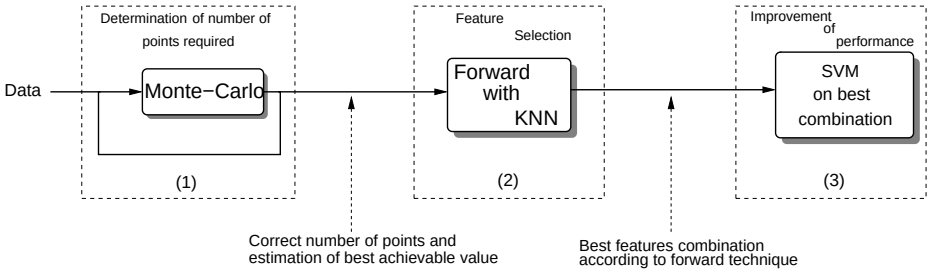


Fig. 4. Schematic view of the proposed methodology: (1) An appropriate number of data points to work with is determined using a Monte-Carlo method for statistical stability; (2) The forward selection is performed using a KNN classifier; (3) A good feature set is selected and performance is improved using SVM.

number of images to use for training with the prepared database. Using a Monte-Carlo method on low numbers of images with both SVM and KNN, averaged plots are obtained. From it, a correct idea of a sufficient number of images for the later study can be obtained, as shown in the following experiments.

Since KNN is the fastest classifier between the two presented, it is used for the next step with forward technique. This produces a ranking of the features showing how much each new feature contributes to the correct classification rate. The best features combination is selected. A SVM is finally used on this combination, to improve the performance and obtain the final best classification rate achieved.

4.2 Experimental setup

Our image base was constituted of 13 000 images of natural scenes, coming from 5 different digital cameras. Images are then all reduced to a size of 800×600 (multiples of 8) to avoid some possible block effects and artifacts due to JPEG re-compression on another grid. At the same time, they are changed from their original colorspace to grayscale colorspace (256 gray levels).

A cropping operation to 512×512 follows, since our implementation of the extractor of Fridrich's 193 features works on 512×512 image blocks (powers of 2). In the end, the whole set of images is separated into two equal parts: one is kept genuine while the other one is steganographied with the Outguess algorithm at an embedding rate of 25%.

This choice of half steganographied and half genuine can be discussed as it does not reflects a real world situation. Meanwhile, the whole steganalysis process presented is designed to be used on one image at a time, determining whether it is genuine or not. Furthermore, this choice has been done to be able to compare performances with the steganalysis community current research.

For classification and test purposes, the training set has been made with at most 8000 images. Test set is composed of the remaining, that is 5000 images. The 193 features proposed by Fridrich are used as in [18].

4.3 Determination of sufficient number of images

Presented first is the result of the evaluation of a sufficient number of images, as explained in the methodology. The Monte-Carlo is used on randomly taken subsets of 200 up to 2000 images with 10 iterations. Each model built – using KNN and SVM – is also evaluated on the test set of 5000 images.

A single point is evaluated with a randomly chosen set of 4000 images, since computational time becomes very important with such number of images. In practice, on Fig. 5 presenting the results of this study, the two cross-validation results (SVM and KNN) should not be strictly compared since they do not use the same number of images to validate the model: SVM uses a 10-fold cross-validation, while a Leave-One-Out (LOO) method is used for KNN. Test results are, on the other hand, comparable.

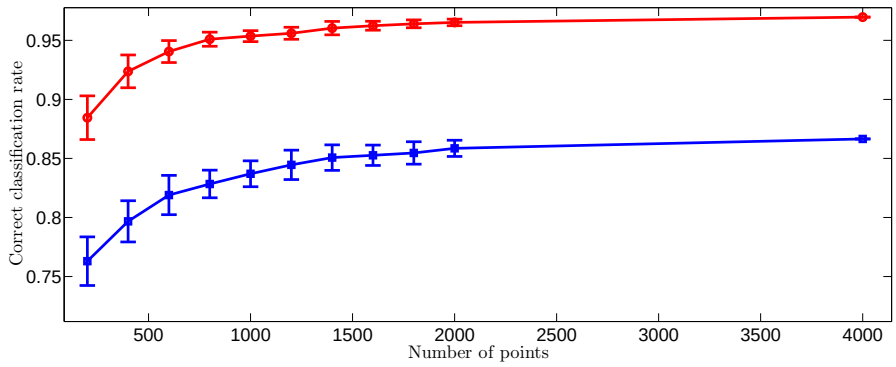


Fig. 5. Correct classification rate for SVM (circles, top curve) and KNN (squares, bottom curve) with associated variance.

One can really see on these plots that an apparently sufficient number of images is over 2000 for both classifiers, since the classification rate seems to increase very slowly over this value. For the experiments, a bigger set of 4000 images has been chosen.

4.4 Forward selection and optimization by SVM

Here, the results of the forward selection are presented shortly. As can be seen from Fig. 6, the whole process of forward selection is not fully shown since we do not go over 21 features. Meanwhile, as will be more detailed in the analysis part of these results, good performance is already performed before this value. Since the goal is to have the smallest possible feature set, while keeping average same performances, the forward selection could be stopped at this point. The rest of the features selected by the forward algorithm only make the classification rate oscillate slightly around 95%.

Test and 10-fold cross-validation remain in a much thinner interval than for our only-KNN tryouts. Moreover, the performance gain with SVM is significative as expected and reaches up to 2% in the frame of these plots.

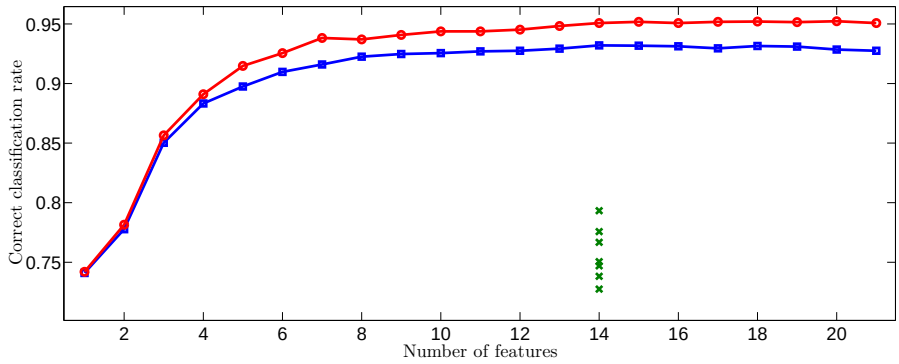


Fig. 6. Plot of the correct classification rate for SVM: 10-fold cross-validation (circles, top curve) and test (squares, bottom curve). Crosses are for performance for random sets of 14 features with a KNN classifier.

Table 2 presents the main values obtained using 193 features set. Our feature selection gives interesting results on this set. Indeed, using as few as 14 features, we are less than 1.9% behind the value obtained with all features for 10-fold cross-validation. Test values are following the exact same pattern.

	LOO / 10-fold	Test	Comp. time
KNN 193	86.65%	85.89%	4.5min
KNN 193→14	93.20%	89.02%	60s
SVM 193	96.92%	96.76%	49h
SVM 193→14	95.08%	94.86%	4.5h

Table 2. Results of the different classifiers for cross-validation and test.

5 Analysis

Some details from the previous results can be interpreted in both fields as follows.

From machine learning point of view: through the previous results, a major achievement was obtained: reducing the dimensionality by more than 13 and keeping roughly the same performance, in the variance interval.

This result is interesting for different reasons:

- Computational time is drastically reduced, since the classifiers complexity relationships to dimensionality are linear (see Tab. 2).
- By this result, the number of features is reduced by 13 and it allows future new analysis and experiments previously not possible.
- This specific 14 features set has proven to be relevant for the classification problem, leading to the next point.
- The comprehension and interpretability is kept through our method and even improves it in an important way, by highlighting Outguess' weaknesses.

From steganalysis point of view: for the steganalysis field, the obtained results are behind the actual best values, obtained for the Outguess algorithm by Fridrich. Nevertheless, the two advantages coming out of these results – namely the decrease of computational time and the gain in interpretability – can counterbalance this opinion.

The 11 features selected by forward technique from the previous 23 features set were already showing some details about Outguess' weaknesses [21]. The major advance using the extended 193 features set is that it gives an even more detailed view on them. Since all vectors are no longer normed using the L_1 norm, informations about the precise weaknesses can be inferred.

The Table 3 lists the set of obtained 14 features out of the full 193 set.

g^{-2}	$C_{-1,-1}$	$C_{-2,-1}$	$h^{21}[4]$	$h^{12}[5]$	$C_{-1,-2}$	$h^{21}[5]$
$g^{0}[6]$	$C_{-2,-2}$	$g^{-1}[9]$	$C_{0,1}$	$g^{-2}[4]$	$C_{2,-2}$	$C_{0,-2}$

Table 3. Names of all 14 selected features from the 193 features set.

From its algorithm, it is known that Outguess does not embed the information into coefficients with values 0 and 1. For this reason, and from the 7 first selected features, it is likely that it changes the coefficients with values -2 and -1, having several sensitive features to such changes react: dual histograms of coefficients -1 and -2, co-occurrence matrix coefficients related to -1 and -2 values changes...

In the end, this set of features describes in a more precise way the functioning and problems of the Outguess algorithm. Taking these into account might help improve the scheme and become less detectable.

6 New algorithms and ideas for further investigations

Two algorithms thought to investigate the current selection results achieved are presented here. These algorithms are derivations of the forward algorithm. They have been thought to try to address the drawbacks of the forward algorithm, that is, the one-by-one selection process by incremental adding of features.

6.1 "Anti-forward"

The "anti-forward" algorithm is an inverse forward in this sense that it selects the worst features first, instead of the best ones. The result on the very same feature set as before can be seen on Fig 7.

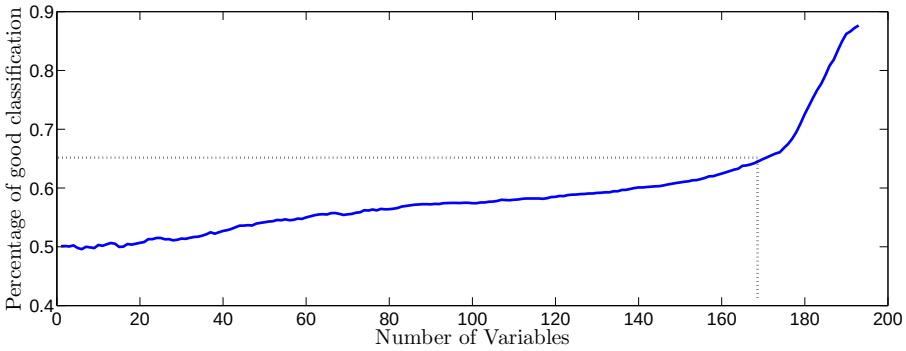


Fig. 7. Anti-forward

A line has been drawn at 65% of correct classification to illustrate how many variables are below this limit. Among the last selected features – that is, the most relevant ones, in a sense –, many belong to the set of 14 features that has been retained for the previous results and analysis. As can be seen, the "linear" relation for the beginning of the graph is broken around 174 features, to increase a lot more.

Eventhough there is a clear correlation between the forward selected set of features and the $193 - 174 = 19$ remaining features, some new features are present and their importance and influence can be investigated.

6.2 "Grouped forward"

The idea behind the "grouped" forward was to try to solve the forward problem when considering groups of relevant features when the standalone features are not as relevant. For this matter, a size S for the groups is set. The classical forward algorithm is then used on the whole data set $\mathbf{R}$, until the number of selected features reaches S. These S features are removed and a forward is launched again on the $\text{Card}(\mathbf{R}) - S$ remaining features of the data set. Until none remains in the data set.

A plot of the correct classification rate is obtained for each group: 1 to 10 in our case since we used a group size of 10 on the 193 features set. This plot is shown on Fig 8.

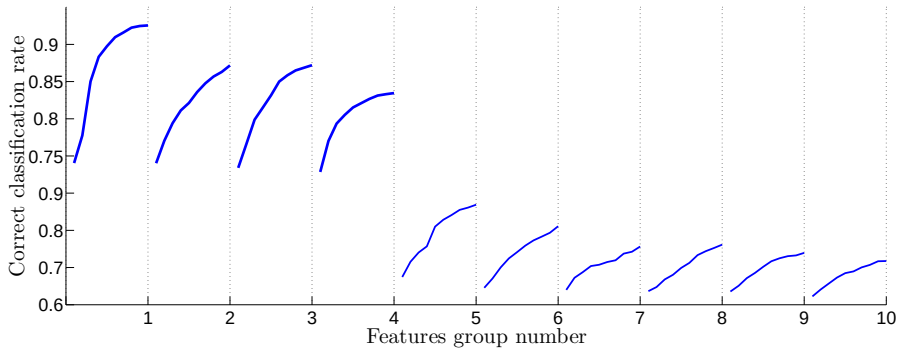


Fig. 8. Grouped Froward

This plot clearly separates the whole set of features into two: the first four groups, achieving each at least 73% of correct classification with only one feature, and the other six ones which seem to be not relevant enough for classification.

A research on the appropriate size S of the groups depending on the goals, has to be done.

7 Conclusion

This paper has presented a new methodology for dimensionality reduction by feature selection in the framework of steganalysis.

The issues of dimensionality have been addressed and the first step of our methodology proves that the theoretically required number of images for correct training is far from being needed. By the use of a Monte-Carlo technique on up to 4000 images, it has been shown that such numbers of images are sufficient for stable results. A set of 193 features extracted from all images serves the classification process, preceded by the dimensionality reduction step. This part of our methodology is achieved using a forward selection with a KNN classifier. It enables to reduce the number of required features to 14, while keeping roughly the same classification results. Computational time is thus greatly improved, divided by about 11. Further analysis becomes again possible with this low number of features: conclusions and precisions about the steganographic scheme can be inferred from the obtained feature set. The last step using SVM for improvement over the previous KNN results achieves high classification results for so small a feature set, proving that many features among the full 193 set might not be relevant enough to be kept for classification purposes.

A comparison between the obtained reduced sets of features for various steganographic algorithms might reveal some common sensitive features. An analysis of these common points could help design a more generic steganalysis method using a "low" number of features.

Meanwhile, the new tools proposed might be of great help when analyzing the retained features and trying to improve the results. Crossing these results will likely lead to validation of the current set, with possible improvements when combining with the new discovered features. The "group forward" might also lead to developments on the selection of groups of features instead of one-by-one processing, thus taking into account possible relationships and dependencies between features.

References

1. Devijver P A and Kittler J. *Pattern recognition : a statistical approach.* Prentice Hall, New York, 1982.
2. Efron B and Tibshirani R J. *An Introduction to the Bootstrap.* Chapman et al., Londres, 1994.
3. R. Bellman. *Adaptive control processes: a guided tour.* Princeton University Press, 1961.
4. François D. *High-dimensional data analysis: optimal metrics and feature selection.* PhD thesis, Universit catholique de Louvain, September 2006.
5. Boser B E, Guyon I M, and Vapnik V N. A training algorithm for optimal margin classifiers. In *Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, 27-29 Juillet 1992.
6. Lothar Hermes and Joachim M. Buhmann. Feature selection for support vector machines. In *15th International Conference on Pattern Recognition*, volume 2, pages 712–715, 3-8 September 2000.
7. Fridrich J. Feature-based steganalysis for jpeg images and its implications for future design of steganographic schemes. In *Information Hiding: 6th International*

Workshop, volume 3200 of *Lecture Notes in Computer Science*, pages 67–81, May 23-25 2004.
8. Park J and Sandberg J W. Universal approximation using radial basis functions network. *Neural Computation*, 3:246–257, 1991.
9. Platt J. *Fast Training of Support Vector Machines using Sequential Minimal Optimization*, chapter 12. MIT Press, 1998.
10. S. Lyu and H. Farid. Detecting hidden messages using higher-order statistics and support vector machines. In *5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, 2002.
11. Verleysen M and François D. The curse of dimensionality in data mining and time series prediction. In *IWANN'05 : 8th International Work-Conference on Artificial Neural Network*, volume 3512 of *Lecture Notes in Computer Science*, pages 758–770, 8-10 Juin 2005.
12. Verleysen M., François D., Simon G., and Wertz V. On the effects of dimensionality on data analysis with neural networks. In Alvarez J.R. eds. Mira J., editor, *Artificial Neural Nets Problem solving methods*, Lecture Notes in Computer Science 2687, pages 105–112. Springer-Verlag, 2003.
13. Provos N. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, pages 323–335, 13-17 April 2001.
14. Vapnik V N. *Statistical Learning Theory*. Wiley-Interscience, 1998.
15. Wand M P and Schucany W R. Gaussian-based kernels. *Canadian Journal of Statistics*, 18(3):197–204, September 1990. doi:10.2307/3315450.
16. Christian P R and Casella G. *Monte Carlo statistical methods*. Springer, 1999. ISBN:038798707X.
17. Yun Qing Shi, Chunhua Chen, and Wen Chen. A markov process based approach to effective attacking jpeg steganography. In *ICME'06 : Internation Conference on Multimedia and Expo*, Lecture Notes in Computer Science, 9-12 July 2006.
18. Pevny T and Fridrich J. Merging markov and dct features for multi-class jpeg steganalysis. In *IS&T/SPIE EI 2007*, volume 6505 of *Lecture Notes in Computer Science*, January 29th - February 1st 2007.
19. Andreas Westfeld and Andreas Pfitzmann. Attacks on steganographic systems. In *IH '99: Proceedings of the Third International Workshop on Information Hiding*, pages 61–76, London, UK, 2000. Springer-Verlag.
20. A W Whitney. A direct method of nonparametric measurement selection. In *IEEE Transactions on Computers*, volume C-20, pages 1100–1103, September 1971.
21. Miche Y, Roue B, Lendasse A, and Bas P. A feature selection methodology for steganalysis. In Bilge Günsel, Anil K. Jain, A. Murat Tekalp, and Bülent Sankur, editors, *Multimedia Content Representation, Classification and Security, International Workshop, MRCS 2006, Istanbul, Turkey*, volume 4105 of *Lecture Notes in Computer Science*, pages 49–56. Springer, September 2006.