Steganalysis by Ensemble Classifiers with Boosting by Regression, and Post-Selection of Features

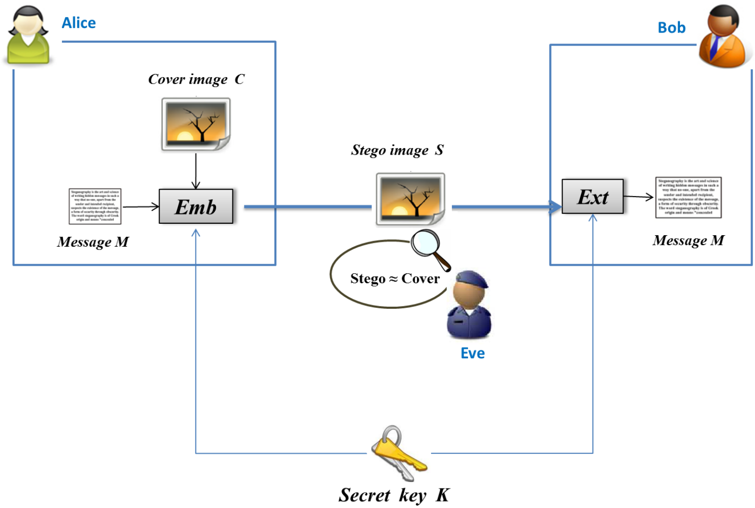
Marc Chaumont, Sarra Kouider
LIRMM, Montpellier, France

October 2, 2012

IEEE International Conference on Image Processing 2012,
Sept. 30 - Oct. 3 2012, Orlando, USA.

Outline

Steganography vs Steganalysis



The proposition

An Improvement of a state-of-the-art steganalyzer

$P_E \searrow$ of the steganalyzer THANKS TO

- boosting by regression of low complexity,
- post-selection of features of low complexity.

Outline

Notable properties

- Appeared during BOSS challenge (sept. 2010 - jan. 2011),
- Performances $\equiv$ to SVM,
- Scalable regarding the dimension of the features vector,
- Low computational complexity,
- Low memory complexity,
- Easily parallelizable.

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

Definition of a weak classifier

Ensemble Classifiers is made of L weak classifiers

- Let $\mathbf{x} \in \mathbb{R}^d$ a feature vector,
- A weak classifier, h_l, returns 0 for cover, 1 for stego :

$$\begin{array}{rcl} h_l : \mathbb{R}^d & \rightarrow & \{0, 1\} \\ \mathbf{x} & \rightarrow & h_l(\mathbf{x}) \end{array}$$

How does classification work?

1. Take an image to analys (i.e. classify in cover or stego),
2. Extract the features vector $\mathbf{x} \in \mathbb{R}^d$,
3. Decide to classify cover or stego (majority vote):

$$C(\mathbf{x}) = \begin{cases} 0 \text{ if } \sum_{l=1}^{l=L} h_l(\mathbf{x}) \leq L/2, \\ 1 \text{ otherwise.} \end{cases}$$

Outline

Weighting the weak classifiers

The classification (steganalysis) process was:

1. Take an image to analys (i.e. classify in cover or stego),
2. Extract the features vector $\mathbf{x} \in \mathbb{R}^d$,
3. Decide to classify cover or stego (majority vote):

$$C(\mathbf{x}) = \begin{cases} 0 \text{ if } \sum_{l=1}^{l=L} h_l(\mathbf{x}) \leq L/2, \\ 1 \text{ otherwise.} \end{cases}$$

Weighting the weak classifiers

The classification (steganalysis) process was:

1. Take an image to analys (i.e. classify in cover or stego),
2. Extract the features vector $\mathbf{x} \in \mathbb{R}^d$,
3. Decide to classify cover or stego (majority vote):

$$C(\mathbf{x}) = \begin{cases} 0 \text{ if } \sum_{l=1}^{l=L} h_l(\mathbf{x}) \leq L/2, \\ 1 \text{ otherwise.} \end{cases}$$

BUT : some weak classifiers are less efficient than others.
THEN : introduce weights !

Weighting the weak classifiers

The classification (steganalysis) process is now:

1. Take an image to analys (i.e. classify in cover or stego),
2. Extract the features vector $\mathbf{x} \in \mathbb{R}^d$,
3. Decide to classify cover or stego (weighted vote):

$$C(\mathbf{x}) = \begin{cases} 0 \text{ if } \sum_{l=1}^{l=L} \alpha_l h_l(\mathbf{x}) \leq \frac{\sum_{l=1}^{l=L} \alpha_l}{2}, \\ 1 \text{ otherwise.} \end{cases}$$

Weighting the weak classifiers

The classification (steganalysis) process is now:

1. Take an image to analys (i.e. classify in cover or stego),
2. Extract the features vector $\mathbf{x} \in \mathbb{R}^d$,
3. Decide to classify cover or stego (weighted vote):

$$C(\mathbf{x}) = \begin{cases} 0 \text{ if } \sum_{l=1}^{l=L} \alpha_l h_l(\mathbf{x}) \leq \frac{\sum_{l=1}^{l=L} \alpha_l}{2}, \\ 1 \text{ otherwise.} \end{cases}$$

How to calculate those weights with a small computational complexity ?

Analytic expression of the weights

During learning step:

$$\{\alpha_l\} = \arg\min_{\{\alpha_l\}} P_E.$$

- simplify P_E expression,
- least squares problem
  $\Rightarrow$ linear system $A.X = B$ with X the weights :

$$A_{i,j} = \sum_{n=1}^{n=N} h_i(\mathbf{x}_n) h_j(\mathbf{x}_n), \qquad B_i = \sum_{n=1}^{n=N} h_i(\mathbf{x}_n) y_n.$$

... solved thanks to a library of linear algebra.

Order of complexity unchanged.

Outline

Reducing the dimension with few computations

Remember: The classification (steganalysis) process is now:

1. Take an image to analys (i.e. classify in cover or stego),
2. Extract the features vector $\mathbf{x} \in \mathbb{R}^d$,
3. Decide to classify cover or stego (weighted vote):

$$C(\mathbf{x}) = \begin{cases} 0 \text{ if } \sum_{l=1}^{l=L} \alpha_l h_l(\mathbf{x}) \leq \frac{\sum_{l=1}^{l=L} \alpha_l}{2}, \\ 1 \text{ otherwise.} \end{cases}$$

Selection of features:
Pre-selection may cost a lot.
What about post-selection?

Once a weak classifier learned : suppress the features reducing P_E :

Algorithm :

1. Compute a **score** for each feature; first database reading,
2. Define an order of selection of the features,
3. Find the best subset (lowest P_E); second database reading.

Order of complexity unchanged.

Outline

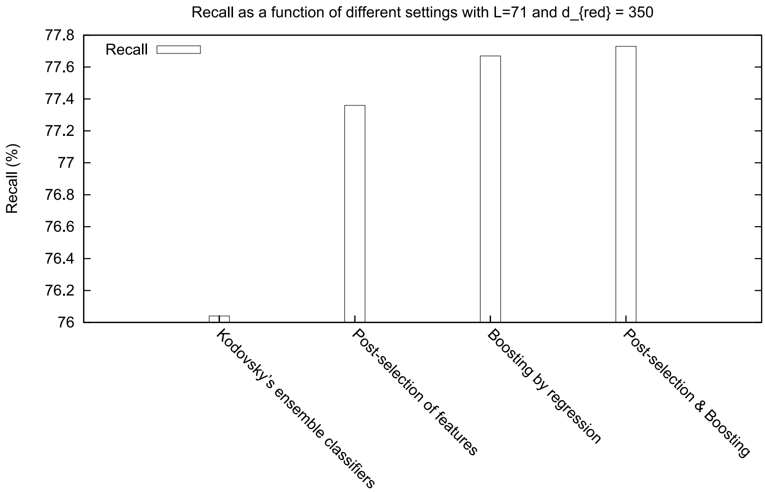
Experimental conditions

- 10 000 greyscale images (512×512, BOSS database),
- The same 10 000 embedded at 0.4 bpp with HUGO,
- Feature vector dimension $d = 5330$ features (HOLMES subset),
- 5 different splits, 5 different seeds,

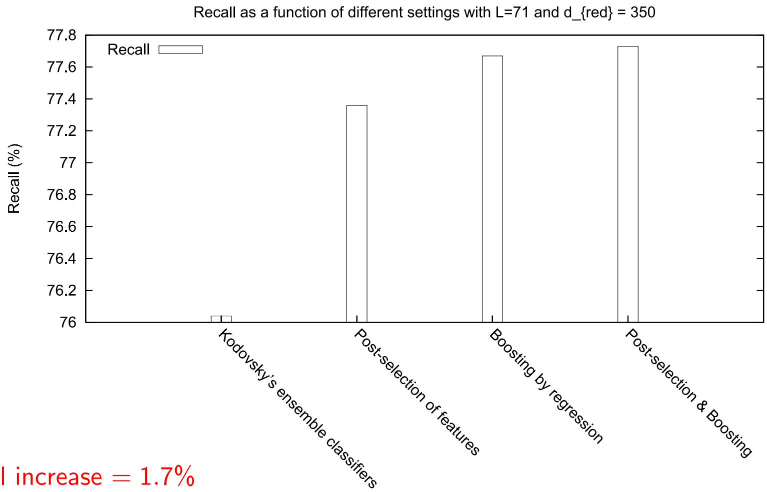
HUGO: "Using High-Dimensional Image Models to Perform Highly Undetectable Steganography"
T. Pevný, T. Filler, and P. Bas, in Information Hiding, IH'2010.

HOLMES: "Steganalysis of Content-Adaptive Steganography in Spatial Domain"
J. Fridrich, J. Kodovský, V. Holub, and M. Goljan, in Information Hiding, IH'2011.

Steganalysis results



Steganalysis results



Recall increase = 1.7%

Same computational complexity order

Outline

Summary

- Two propositions for the Kodovský steganalyzer:
  - boosting by regression,
  - post-selection of features.
- Significant recall increase (1.7%)
- No change in computational complexity order

Annex: Metrics (1)

- Distance between the two classes:

$$c_1^{(l)}[j] = \frac{|\mu_1[j] - \mu_0[j]|}{\sqrt{\sigma_1^2[j] + \sigma_0^2[j]}}.$$

- Influence of a feature on the final correlation/decision (= dot product) used to classify:

$$c_2^{(l)}[j] = \sum_{i=1}^{i=N} count(\mathbf{x}_i^{(l)}[j], \mathbf{w}^{(l)}[j], y_i),$$

with:

$$count(x, w, y) = \begin{cases} 1 \text{ if } [(x.w > 0 \text{ and } y = 1) \\ \qquad \text{or } (x.w < 0 \text{ and } y = 0)], \\ 0 \text{ otherwise.} \end{cases}$$

$$c_3^{(l)}[j] = \sum_{i=1}^{i=N} \frac{count(\mathbf{x}_i^{(l)}[j], \mathbf{w}^{(l)}[j], y_i)}{\sum_{k=1}^{k=d_{red}} count(\mathbf{x}_i^{(l)}[k], \mathbf{w}^{(l)}[k], y_i)}.$$

Annex: Metrics (2)

- Feature correlation with the class:

$$
\begin{aligned}
c_4^{(l)}[j] &= corr(\mathbf{x}^{(l)}[j], y) \\
&= \frac{\sum_{i=1}^{i=N}\left(\mathbf{x}_i^{(l)}[j] - \overline{\mathbf{x}^{(l)}[j]}\right)(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{i=N}\left(\mathbf{x}_i^{(l)}[j] - \overline{\mathbf{x}^{(l)}[j]}\right)^2}\sqrt{\sum_{i=1}^{i=N}(y_i - \overline{y})^2}}.
\end{aligned}
$$

- Feature correlation with the weak classifier:

$$
c_5^{(l)}[j] = corr(\mathbf{x}^{(l)}[j].\mathbf{w}^{(l)}[j], y).
$$

Annex: P_E in the Boosting by Regression

During learning step:

$$\{\alpha_l\} = \underset{\{\alpha_l\}}{\arg\min}\, P_E.$$

$$P_E = \frac{1}{N}\sum_{i=1}^{i=N}\left(f\left(\sum_{l=1}^{l=L}\alpha_l h_l(\mathbf{x}_i)\right) - y_i\right).$$

with f a thresholding function defined by:

$$\begin{array}{rcl} f:\mathbb{R} & \rightarrow & \{0,1\} \\ x & \rightarrow & f(x) = \left\{ \begin{array}{l} 0 \text{ if } x \leq \frac{\sum_{l=1}^{l=L}\alpha_l}{2}, \\ 1 \text{ otherwise.} \end{array} \right. \end{array}$$

Let's simplify, P_E:

$$P_E \approx \frac{1}{N}\sum_{i=1}^{i=N}\left(\sum_{l=1}^{l=L}\alpha_l h_l(\mathbf{x}_i) - y_i\right)^2.$$

$\Rightarrow$ least squares problem ... solved thanks to a library of linear algebra.