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

1. Preamble
2. The Kodovsky's Ensemble Classifiers
3. Boosting by regression
4. Post-selection of features
5. Experiments
6. Conclusion
Steganography vs Steganalysis

Alice

Cover image C

Emb

Message M

Stego image S

Stego ≈ Cover

Eve

Ext

Message M

Bob

Secret key K
An Improvement of a state-of-the-art steganalyzer

$P_E \downarrow$ of the steganalyzer THANKS TO

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

1. Preamble
2. The Kodovsky’s Ensemble Classifiers
3. Boosting by regression
4. Post-selection of features
5. Experiments
6. Conclusion
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,”
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:

$$h_l : \mathbb{R}^d \rightarrow \{0, 1\}$$

$$\mathbf{x} \rightarrow h_l(\mathbf{x})$$
How does classification work?

1. Take an image to analyse (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):

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

1. Preamble
2. The Kodovsky’s Ensemble Classifiers
3. Boosting by regression
4. Post-selection of features
5. Experiments
6. Conclusion
Weighting the weak classifiers

The classification (steganalysis) process was:

1. Take an image to analyze (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):

$$C(\mathbf{x}) = \begin{cases} 
0 & \text{if } \sum_{i=1}^{L} h_i(\mathbf{x}) \leq L/2, \\
1 & \text{otherwise.}
\end{cases}$$
Weighting the weak classifiers

The classification (steganalysis) process was:

1. Take an image to analyse (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):

\[
C(\mathbf{x}) = \begin{cases} 
0 & \text{if } \sum_{i=1}^{L} h_i(\mathbf{x}) \leq L/2, \\
1 & \text{otherwise}.
\end{cases}
\]

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 analyse (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):

$$C(\mathbf{x}) = \begin{cases} 
0 & \text{if } \sum_{i=1}^{L} \alpha_i h_i(\mathbf{x}) \leq \frac{\sum_{i=1}^{L} \alpha_i}{2}, \\
1 & \text{otherwise}.
\end{cases}$$
Weighting the weak classifiers

The classification (steganalysis) process is now:

1. Take an image to analyse (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):

$$C(\mathbf{x}) = \begin{cases} 0 & \text{if } \sum_{i=1}^{L} \alpha_i h_i(\mathbf{x}) \leq \frac{\sum_{i=1}^{L} \alpha_i}{2} , \\ 1 & \text{otherwise.} \end{cases}$$

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
  \[ \Rightarrow \text{linear system } A.X = B \text{ with } X \text{ the weights} : \]

\[
A_{i,j} = \sum_{n=1}^{N} h_i(x_n) h_j(x_n), \quad B_i = \sum_{n=1}^{N} h_i(x_n) y_n.
\]

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

Order of complexity unchanged.
Outline

1. Preamble
2. The Kodovsky’s Ensemble Classifiers
3. Boosting by regression
4. Post-selection of features
5. Experiments
6. Conclusion
Reducing the dimension with few computations

Remember: The classification (steganalysis) process is now:

1. Take an image to analyse (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):

$$C(\mathbf{x}) = \begin{cases} 
0 & \text{if } \sum_{i=1}^{L} \alpha_i h_i(\mathbf{x}) \leq \frac{\sum_{i=1}^{L} \alpha_i}{2}, \\
1 & \text{otherwise.} 
\end{cases}$$

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

1. Preamble
2. The Kodovsky’s Ensemble Classifiers
3. Boosting by regression
4. Post-selection of features
5. Experiments
6. Conclusion
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”

HOLMES: “Steganalysis of Content-Adaptive Steganography in Spatial Domain”
Steganalysis by Ensemble Classifiers - Marc Chaumont - ICIP’2012

Experiments

Steganalysis results

Recall as a function of different settings with L=71 and d_{red} = 350
Recall increase = 1.7%
Same computational complexity order
Outline

1. Preamble
2. The Kodovsky’s Ensemble Classifiers
3. Boosting by regression
4. Post-selection of features
5. Experiments
6. Conclusion
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:

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

with:

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

\[ c_3^{(l)}[j] = \sum_{i=1}^{i=N} \frac{count(x_i^{(l)}[j], w^{(l)}[j], y_i)}{\sum_{k=d_{\text{red}}}^{k=1} count(x_i^{(l)}[k], w^{(l)}[k], y_i)}. \]
Feature correlation with the class:

\[ c_4^{(l)}[j] = corr(x^{(l)}[j], y) = \frac{\sum_{i=1}^{N} (x_i^{(l)}[j] - \overline{x}^{(l)}[j]) (y_i - \overline{y})}{\sqrt{\sum_{i=1}^{N} (x_i^{(l)}[j] - \overline{x}^{(l)}[j])^2} \sqrt{\sum_{i=1}^{N} (y_i - \overline{y})^2}}. \]

Feature correlation with the weak classifier:

\[ c_5^{(l)}[j] = corr(x^{(l)}[j].w^{(l)}[j], y). \]
Annex: $P_E$ in the Boosting by Regression

During learning step:

\[
\{\alpha_l\} = \arg\min_{\{\alpha_i\}} P_E.
\]

\[
P_E = \frac{1}{N} \sum_{i=1}^{i=N} \left( f \left( \sum_{l=1}^{l=L} \alpha_l h_l(x_i) \right) - y_i \right).
\]

with $f$ a thresholding function defined by:

\[
f: \mathbb{R} \rightarrow \{0, 1\}
\]

\[
x \rightarrow f(x) = \begin{cases} 
0 & \text{if } x \leq \frac{\sum_{l=1}^{l=L} \alpha_l}{2}, \\
1 & \text{otherwise}.
\end{cases}
\]

Let’s simplify, $P_E$:

\[
P_E \approx \frac{1}{N} \sum_{i=1}^{i=N} \left( \sum_{l=1}^{l=L} \alpha_l h_l(x_i) - y_i \right)^2.
\]

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