# Rethinking Optimal Embedding

**Andrew Ker**
adk@cs.ox.ac.uk

**Tomáš Pevný**
pevnak@gmail.com

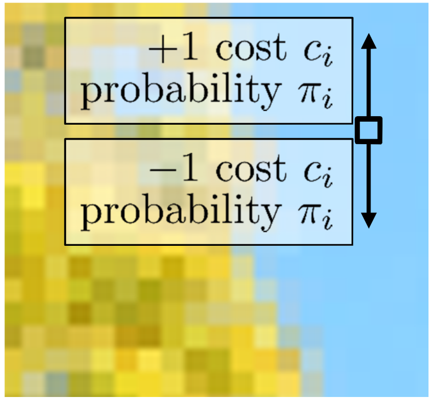
**Patrick Bas**
Patrick.Bas@ec-lille.fr

4th ACM Workshop on Information Hiding & Multimedia Security

# Adaptive steganography

*Some embedding changes are more detectable than others.*

- Assign each possible change a cost $c_i$.



# Adaptive steganography

*Some embedding changes are more detectable than others.*

- Assign each possible change a cost $c_i$.
- Use coding (STCs) to minimize average cost $\sum_i \pi_i c_i$.

e.g. HUGO [2010], WOW [2012], UNIWARD [2013-4], HILL [2014], ...

# Adaptive steganography

*Some embedding changes are more detectable than others.*

- Assign each possible change a cost $c_i$.
- Use coding (STCs) to minimize average cost $\sum_i \pi_i c_i$.

*What if the enemy is aware of your adaptivity?*

e.g. 'tSRM' attack on WOW [Tang et al., 2014]
'CSR' on $1^{\text{st}}$ version of UNIWARD [Denemark et al., 2014]
'maxSRM' on $2^{\text{nd}}$ version of UNIWARD [Denemark et al., 2014]

# Adaptive steganography

*Some embedding changes are more detectable than others.*

- Assign each possible change a cost $c_i$.
- Use coding (STCs) to
  minimize average cost $\sum_i \pi_i c_i$.

*What if the enemy is aware of your adaptivity?*

- **Use coding (STCs) to minimize $\boldsymbol{\sum_i \pi_i^2 c_i}$.**

# Two-player, zero-sum game

**Embedder**
chooses probability of changing each location $\pi_i$ ('p-map').

**Detector**
chooses weights for each observation $\omega_i$.

Embedder's payoff = – (Detector's payoff) = **FP-50:**

**false positive rate @ 50% true positives**

- Used in game theory of embedding since at least 2007.
- Slightly simplifies the analysis.

# Two-player, zero-sum game

**Embedder**
chooses probability of changing each location $\pi_i$ ('p-map').

**Detector**
chooses weights for each observation $\omega_i$.

Embedder's payoff = − (Detector's payoff) = **FP-50:**

**false positive rate @ 50% true positives**

If based on some detection value $\ell$, inverse in the *deflection*:

$$\delta = \frac{\mathrm{E}_{Stego}[\ell] - \mathrm{E}_{Cover}[\ell]}{\sqrt{\mathrm{Var}_{Cover}[\ell]}}.$$

(assuming $\ell$ asymptotically Gaussian).

# Two-player, zero-sum game

**Embedder**

chooses probability of changing each location $\pi_i$ ('p-map').

**Detector**

chooses weights for each observation $\omega_i$.

Embedder's payoff = – (Detector's payoff) = **FP-50:**

**false positive rate @ 50% true positives**

If based on some detection value $\ell$, inverse in the *deflection*:

$$\delta = \frac{\mathrm{E}_{Stego}[\ell] - \mathrm{E}_{Cover}[\ell]}{\sqrt{\mathrm{Var}_{Cover}[\ell]}}.$$

- Also used in game theory of embedding since at least 2007, and recently.
- Monotone relationship with other popular metrics.

# Binary covers

Independent pixels taking binary values $(X_1, \ldots, X_n)$.

Embedder flips pixels.

*In cover:*

$\mathrm{P}[X_i = 1] = p_i$

*In stego:*

$\mathrm{P}[X_i = 1] = p_i + \pi_i(1 - 2p_i)$

Embedder's strategy (change probabilities)

# Binary covers

*We may assume the detector is based on log likelihood ratio:*

$$\ell = \sum_i X_i \omega_i$$

Detector's strategy (weights)

*In cover:*

$$\mathrm{P}[X_i = 1] = p_i$$

*In stego:*

$$\mathrm{P}[X_i = 1] = p_i + \pi_i (1 - 2p_i)$$

# Binary covers

*We may assume the detector is based on log likelihood ratio:*

$$\ell = \sum_i X_i \omega_i$$

*In cover:*

$P[X_i = 1] = p_i$ $\quad$ $E[\ell] = \sum_i p_i \omega_i$ $\quad$ $\mathrm{Var}[\ell] = \sum_i \overbrace{p_i(1-p_i)}^{e_i} \omega_i^2$

*In stego:*

$P[X_i = 1] = p_i + \pi_i(1-2p_i)$ $\quad$ $E[\ell] = \sum_i p_i \omega_i + \sum_i \pi_i \overbrace{(1-2p_i)}^{d_i} \omega_i$

*Embedder wants to minimize*
*Detector wants to maximize* } *Deflection:* $\delta = \dfrac{\sum_i \pi_i d_i \omega_i}{\sqrt{\sum_i e_i \omega_i^2}}$

# Binary covers

*Detector maximizes $\delta$*

*Embedder minimizes $\delta$*

$$\delta = \frac{\sum_i \pi_i d_i \omega_i}{\sqrt{\sum_i e_i \omega_i^2}}$$

*knowing*

$$\arg\max_{\omega_i} \delta \propto \pi_i \frac{d_i}{e_i}$$

$$\delta = \frac{\sum_i \pi_i^2 d_i^2 / e_i}{\sqrt{\sum_i \pi_i^2 d_i^2 / e_i}} \quad \text{i.e.} \quad \min_{\pi_i} \sum_i \pi_i^2 c_i$$

Minimax in two-player, zero-sum game, hence equilibrium.

*ignorant*

$$\arg\max_{\omega_i} \delta \propto \frac{d_i}{e_i}$$

$$\delta = \frac{\sum_i \pi_i d_i^2 / e_i}{\sqrt{\sum_i d_i^2 / e_i}} \quad \text{i.e.} \quad \min_{\pi_i} \sum_i \pi_i c_i$$

# Binary covers

Detector's optimal behaviour is to weight each pixel according to its 'p-map'.

*ıbedder minimizes $\delta$*

$$\delta = \frac{\sum_i \pi_i d_i \omega_i}{\sqrt{\sum_i e_i \omega_i^2}}$$

*knowing*: $\arg\max_{\omega_i} \delta \propto \pi_i \frac{d_i}{e_i}$

$$\delta = \frac{\sum_i \pi_i^2 d_i^2 / e_i}{\sqrt{\sum_i \pi_i^2 d_i^2 / e_i}} \quad \text{i.e.} \quad \min_{\pi_i} \sum_i \pi_i^2 c_i$$

Minimax in two-player, zero-sum game, hence equilibrium.

*ignorant*: $\arg\max_{\omega_i} \delta \propto \frac{d_i}{e_i}$

$$\delta = \frac{\sum_i \pi_i d_i^2 / e_i}{\sqrt{\sum_i d_i^2 / e_i}} \quad \text{i.e.} \quad \min_{\pi_i} \sum_i \pi_i c_i$$

# Arbitrary covers

Independent pixels taking *k*-ary values, with a different distribution at each pixel.

- Fixed embedding operation, at pixel $i$ with probability $\pi_i$,

vs. ignorant: $\min_{\pi_i} \sum_i \pi_i c_i$ vs. knowing: $\min_{\pi_i} \sum_i \pi_i^2 c_i$

- Arbitrarily changing embedding operations,

vs. ignorant: $\min_{\boldsymbol{\pi_i}} \sum_i \boldsymbol{\pi_i^T e_i}$ vs. knowing: $\min_{\boldsymbol{\pi_i}} \sum_i \boldsymbol{\pi_i^T} E_i^{-1} \boldsymbol{\pi_i}$

# Connections with other work

- Optimal detectors weight the evidence.

e.g. maxSRM [Denemark et al., 2014] and tSRM [Tang et al., 2014].

- Squared probabilities.

Intuitive. Appear as far back as [Ker, 2007].

- Generalizes recent work of Sedighi, Cogranne & Fridrich:

  - independent discretized Gaussian pixels, varying variance,
  - symmetric ternary coding: $\min_{\pi_i} \sum_i \pi_i^2 / \sigma_i^4$
  - pentary coding: $\min_{\boldsymbol{\pi_i}} \sum_i \boldsymbol{\pi_i^T} E_i^{-1} \boldsymbol{\pi_i}$

# Payload constraint

$$\text{subject to } \sum_i H(\pi_i) \geq \text{payload length},$$

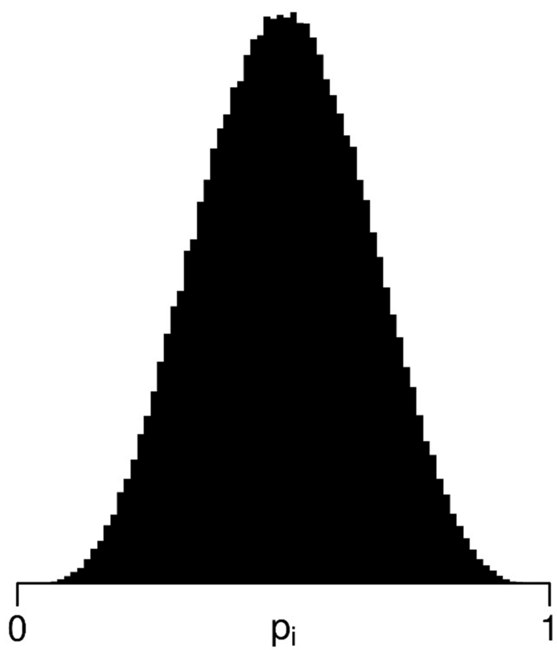
- **Naive:** $\min_{\pi_i} \sum_i \pi_i c_i \longrightarrow H'(\pi_i) = \lambda c_i$
- **Equilibrium:** $\min_{\pi_i} \sum_i \pi_i^2 c_i \longrightarrow H'(\pi_i)/\pi_i = \lambda c_i$
- **Equilibrium:** $\min_{\boldsymbol{\pi_i}} \sum_i \boldsymbol{\pi_i^T} E_i^{-1} \boldsymbol{\pi_i} \longrightarrow$ convex set of equations

  (arbitrary embedding)

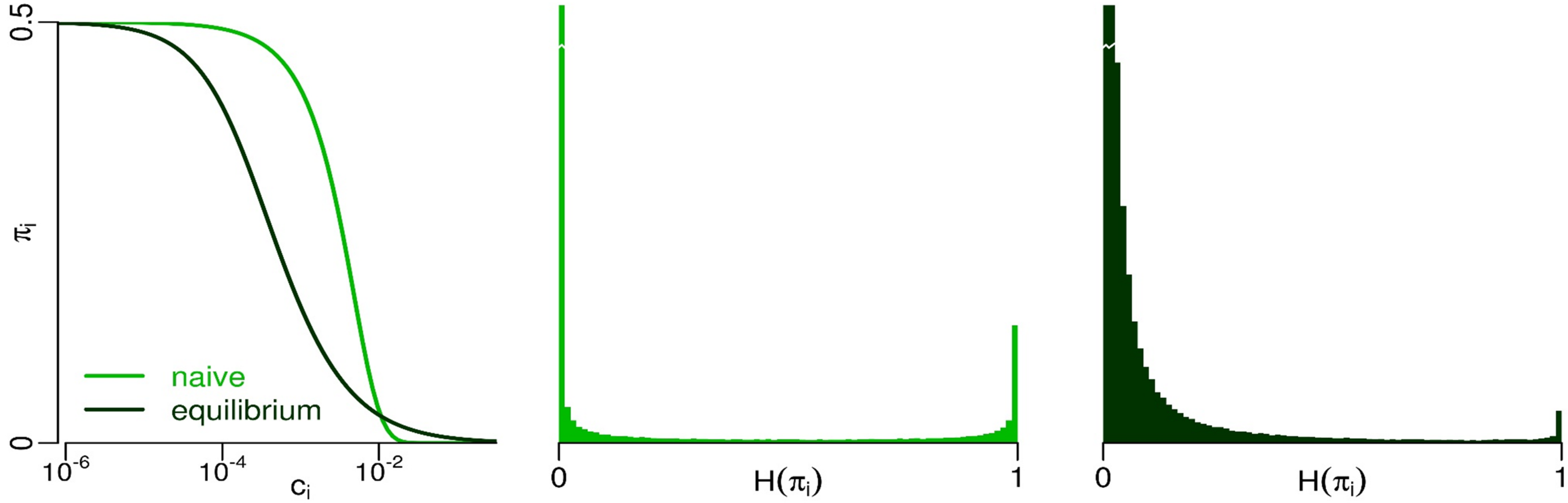
# Experiments on binary covers

We generated artificial binary covers:

- $n = 2^{18}$ pixels (à la BOSSBase),
- $p_i$ drawn from $\mathrm{Beta}(5, 5)$,



# Experiments on binary covers




- simulated payload of 0.1 bits per pixel with optimal coding:
  - constant $\pi_i$,
  - naive adaptivity: $\min_{\pi_i} \sum_i \pi_i c_i$,
  - equilibrium adaptivity: $\min_{\pi_i} \sum_i \pi_i^2 c_i$.
- Used likelihood ratio tests on 10 000 covers & stego objects.

# Experiments on binary covers

| LRT detector for... | Embedding probabilities<br>Constant $\pi_i$ | Naive $\pi_i$ | Equilibrium $\pi_i$ |
|---|---|---|---|
| Constant $\pi_i$ | 0.000 | 0.492 | 0.335 |
| Naive $\pi_i$ | 0.443 | 0.023 | 0.225 |
| Equilibrium $\pi_i$ | 0.038 | 0.081 | **0.145** (equilibrium) |

FP-50 (false positive rate at 50% true positive)

- simulated payload of 0.1 bits per pixel with optimal coding:
  - constant $\pi_i$,
  - naive adaptivity: $\min_{\pi_i} \sum_i \pi_i c_i$,
  - equilibrium adaptivity: $\min_{\pi_i} \sum_i \pi_i^2 c_i$.
- Used likelihood ratio tests on 10 000 covers & stego objects.

# Experiments on S-UNIWARD

Computes cost in a wavelet domain: [Holub et al., 2014]

$$c_i = \sum_j \frac{|W_j(\text{Cover}) - W_j(\text{Cover} + \text{change } i)|}{\sigma + |W_j(\text{Cover})|}$$

wavelet coefficient

# Experiments on S-UNIWARD

Computes cost in a wavelet domain: [Holub et al., 2014]

$$c_i = \sum_j \frac{|W_j(\text{Cover}) - W_j(\text{Cover} + \text{change } i)|}{\sigma + |W_j(\text{Cover})|}$$

stabilization value

In the original definition, $\sigma = 10^{-15}$

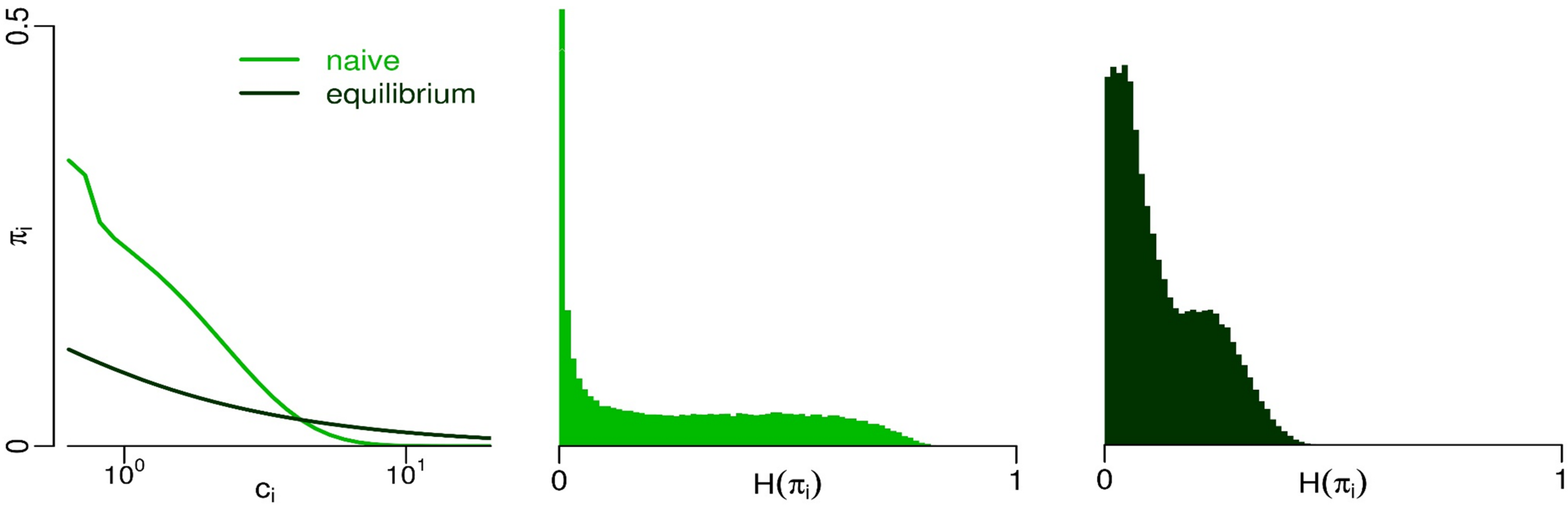
... exploited by 'CSR features' [Denemark et al., 2014]

# Experiments on S-UNIWARD

Computes cost in a wavelet domain: [Holub et al., 2014]

$$c_i = \sum_j \frac{|W_j(\text{Cover}) - W_j(\text{Cover} + \text{change } i)|}{\sigma + |W_j(\text{Cover})|}$$

stabilization value



# Experiments on S-UNIWARD

| Detector trained on ... | Embedding probabilities: Naive $\pi_i$ | Embedding probabilities: Equilibrium $\pi_i$ |
|---|---|---|
| Naive $\pi_i$ | 0.007 | 0.500 |
| Equilibrium $\pi_i$ | 0.502 | **0.130** **(NOT equilibrium)** |

$$P_{\text{err}} = 0.5(P_{\text{fp}} + P_{\text{fn}})$$

- BOSSBase images (8000 training, 2000 testing, 10 iterations),
- simulated payload of 0.3 bits per pixel,
- CSR features, ensemble of FLDs detector.

# Conclusions

- $\min_{\pi_i} \sum_i \pi_i c_i \quad \rightarrow \quad \min_{\pi_i} \sum_i \pi_i^2 c_i$ is not a panacea!
  - *Need to start with statistically correct costs.*

- Very general, but completely theoretical, results.
  - *Assumes both players know cover source exactly.*
  - *Unlike MiPOD, does not give a new embedding method.*

- (Recent work) the square root law still holds…
  - *with some interesting wrinkles.*

- (Further work) for non-independent pixels/changes/costs?