# A Mishmash of Methods for Mitigating the Model Mismatch Mess

Andrew D. Ker[a] and Tomáš Pevný[b]

[a]Oxford University Department of Computer Science, Parks Road, Oxford OX1 3QD, UK.
[b]Agent Technology Center, Czech Technical University in Prague,
Karlovo naměstí 13, 121 35 Prague 2, Czech Republic.

## ABSTRACT

The model mismatch problem occurs in steganalysis when a binary classifier is trained on objects from one cover source and tested on another: an example of domain adaptation. It is highly realistic because a steganalyst would rarely have access to much or any training data from their opponent, and its consequences can be devastating to classifier accuracy. This paper presents an in-depth study of one particular instance of model mismatch, in a set of images from *Flickr* using one fixed steganography and steganalysis method, attempting to separate different effects of mismatch in feature space and find methods of mitigation where possible. We also propose new benchmarks for accuracy, which are more appropriate than mean error rates when there are multiple actors and multiple images, and consider the case of 3-valued detectors which also output 'don't know'. This pilot study demonstrates that some simple feature-centering and ensemble methods can reduce the mismatch penalty considerably, but not completely remove it.

## 1. INTRODUCTION

Treating steganalysis as a problem of binary classification has been very successful,[1,2] but such a scenario assumes that the detector has access to the steganographic embedding method and, crucially, the cover source used by the sender. In reality one cannot normally obtain the exact cover source, and in practice it is necessary to train the classifier on a different one with hopefully-similar characteristics (an example of domain adaptation[3]). This induces the *model mismatch* problem, which has been demonstrated to reduce steganalysis accuracy[4–6] by very significant, and unpredictable, amounts. The unpredictability is particularly troublesome, and the reason why model mismatch is a mess for the practitioner: it is difficult to trust the output of steganalysis if the accuracy is unpredictable.

In this paper we distinguish two degrees of mismatch: *total mismatch*, in which the steganalyst must train on a source disjoint with the testing source (known as *conservative* domain adaptation in the machine learning literature), and *partial mismatch* where a small amount of training data is available for the target but which is insufficient to train a classifier (*adaptive* domain adaptation).

This is a timely research topic if steganalysis is to be applied in real-world circumstances.[7] There has, so far, been only a small amount of literature attempting to reduce the classification errors due to model mismatch. One approach was clustering, followed by a special steganalyzer well-suited to that cluster[6] apart from computational cost and the need to train many classifiers, it also does not help in the case when a sample is far from all clusters. An early spatial-domain method was adapted for more robustness,[6] but this could not be used for modern steganalysis, and mismatch was measured but not mitigated in Ref. 4. Recently, Ref. 5 used simpler classifiers trained on large amounts of data, attempting to avoid overfitting a particular training source (this requires a large amount of training data from a number of actors, but can attack total mismatch), and Ref. 9 accepted mismatch as a fact and used a hierarchical model to share information between personalized classifiers (this only works for partial mismatch). In neither case were the misclassification errors reduced by very much.

Further author information:
A. D. Ker: E-mail: adk@cs.ox.ac.uk, Telephone: +44 1865 283530
T. Pevný: E-mail: pevnak@gmail.com, Telephone: +420 22435 7608

| | Camera model | Image resolution (Kpix) | | |
|---|---|---|---|---|
| | | minimum | mean | maximum |
| Actor 1 | Fujifilm FinePix S5Pro | 1516 | 2733 | 4159 |
| Actor 2 | Canon EOS REBEL T3 | 1707 | 2479 | 3379 |
| Actor 3 | Sony DSC-W110 | 263 | 1063 | 6697 |
| Actor 4 | Canon EOS DIGITAL REBEL XTi | 1704 | 2196 | 3908 |
| Actor 5 | PENTAX *ist DL | 286 | 2771 | 4903 |
| Actor 6 | Sony DSLR-A200 | 167 | 544 | 1097 |
| Actor 7 | Canon EOS DIGITAL REBEL XTi | 2329 | 2795 | 3320 |
| Actor 8 | Casio EX-Z100 | 693 | 1231 | 7078 |
| Actor 9 | Canon EOS 450D | 426 | 990 | 1135 |

Table 1: Camera models used by individual actors.

This paper is an exploration of the model mismatch problem. We fix on a particular set of mismatched steganographers – actors using nine different camera sources, and plain nsF5 steganography – and attempt to measure how mismatch arises in steganographic features, finding mitigation for the mismatch if possible. A number of different techniques will be attempted and benchmarked, but we do not attempt to determine the best 'solution' to the mismatch problem. Our approach is quite geometric: we will try to isolate shifts of location and changes of direction and 'speed' of change (with payload) in feature space, along with false certainty arising from making decisions in sparsely-populated regions. We do not, at this stage, attempt to determine *why* the mismatch has a particular form, or whether this is due to the design of the features or particular differences between the cameras; that is research for another day.

The paper is structured as follows. In section 2 we discuss how overall detector accuracy should be measured, when there are many actors and potentially many images each. We argue that a 'mean accuracy' measure does not properly capture the value of stability, and define two other aggregate metrics based on simple models of pooled steganalysis. In the following three sections we examine three potential sources of error in mismatched steganalysis: shift in the center of cover features between sources (sect. 3), different direction and/or rate of change of features as steganography is embedded (sect. 4), and false certainty by classifiers which make decisions in regions of particularly sparse training (sect. 5). In each case we attempt first to *measure* the effect, and then to *mitigate* it, if possible. Finally, in section 6 we summarise the results of the entire paper and draw some conclusions. This is only a pilot study and much further work is needed.

### 1.1 Experimental setup

Throughout this paper we use a database of real-world images: 9000 images, taken with the same camera, from each of nine uploaders (actors) on the popular image sharing site *Flickr*. The camera model was identified from EXIF data, and the images were downloaded in 'original' size. We cannot know the exact processing chain before the images were uploaded, and the images are not all the same size suggesting some resampling or (more likely in most cases here) cropping, and potentially double-compression; although they make steganalysis difficult, these are exactly the sorts of artifacts we expect to see in real world sources. We selected these particular nine uploaders because, although they use 8 different models from 4 manufacturers, they all produced images with the same JPEG quality factor (85). The camera models used by the actors, and some information about the image sizes, are displayed in Table 1.

Mismatch due to differences in JPEG quality factor is certainly an important (and under-researched) topic, but we wanted to isolate it for the purposes of this study. Because different quality factors imply different DCT quantization bin widths, the features extracted from images with different quality factors are effectively different features altogether (they are counting different things).

We simulated steganography using the nsF5 embedding operation without adaptive matrix embedding, with a fixed embedding efficiency of 2 bits per change. For benchmarking the accuracy of (matched, partially-matched, and totally-mismatched) detectors we always tested the fixed payload of 0.05 bits per nonzero coefficient (bpnc),

but for exploration of the mismatch phenomenon we also embedded random-length payloads. We used this embedding method because it is well understood and, being non-adaptive, the rate of embedding changes does not vary with the cover image: this is another factor we wished to isolate from our study.

For detection we used the $\mathcal{CF}^*$ features.[10] These are moderately recent and fit the 'rich model' paradigm used by the very state-of-art steganalysis (they are 7850 dimensional), but can be extracted rather faster than the latest but expensive JRM[1] or PSRM[11] features. After the results of this pilot study are understood, further research can verify its application to other feature sets and other embedding algorithms, but we expect that the conclusions will hold rather widely because steganalysis features are all doing rather similar things: counting occurrences of (possibly filtered) coefficients.

Exploratory data analysis was conducted on all 81000 images, but when we benchmark detection accuracy we always use a Fisher Linear Discriminant (except in section 5) trained on 6000 of each actors' images, and tested on the other 3000: thus the training and testing sets are disjoint. The accuracy of each classifier is measured by the minimum equal-prior error rate

$$P_{\mathrm{E}} = \tfrac{1}{2}\min(P_{\mathrm{FP}} + P_{\mathrm{FN}})$$

where $P_{\mathrm{FP}}$ and $P_{\mathrm{FN}}$ represent the false positive and false negative rate and the minimum is taken over parallel decision boundaries. We used a simple FLD (rather than the nonlinear classifiers used in most literature) because it has a simple geometric intuition, and because our experience is that linear classifiers[5] and regressors[12] can still provide good performance. This is particularly so when the size of the training data is small, relative to the number of features. For future work it would be straightforward to extend the testing to nonlinear classifiers such as the ensemble FLD used in Ref. 1.

We display the accuracy of these classifiers, and the effect of complete mismatch, in Table. 2: nine classifiers were trained (one per actor) and then tested on each actor's images. Aggregation of these numbers is explained in the following section, but the mismatch penalty is easily apparent with the off-diagonal error rates being approximately five times higher than the diagonal (the matched case), and in the worst case hardly better than random guessing (an error rate of 39%). This motivates the investigations of this paper.

It is interesting that the worst mismatched error occurs for actor three, who used a Sony camera with rather small average image size. The best mismatched error of this actor was with an actor six, who also used Sony camera, but with even smaller images. Notice that actor nine also posted images with a small resolution, similar on average to those of actor three, but they were unable to classify each others' images well (error rates 31% and 19%, depending on which was trained and which tested), suggesting that the mismatch is probably not primarily due to image size.

## 2. METRICS IN A MULTI-ACTOR, MULTI-IMAGE WORLD

We want to measure the success of methods for mitigating model mismatch. How should we do this? This requires a reconsideration of the very benchmarks we use for steganalysis. In the literature, when mismatched covers are tested at all their results are either displayed without aggregation, or the mean error rate (equivalently, mean accuracy) are displayed. The former is fine but does not allow for easy comparison. The latter, we argue, is somewhat flawed.

First, consider the following situation. As steganalysts, we are given two detectors. One is always 80% accurate and the other is 70% accurate or 90% accurate, equally likely but depending on some facets of the target that you cannot measure. Which detector do we prefer? Clearly, the detector with a stable error rate is better, but they both have the same average. Stability of error rate, between match and mismatch and between different cases of mismatch, is valuable because it implies reliability.

The problem runs deeper. We perform a second thought experiment. Suppose that the world consists of $k$ actors, and that we have trained one or more classifiers (which might be matched or mismatched, perhaps even as many as one classifier for each actor) with false positive/negative rates of $P^1_{\mathrm{FP}}/P^2_{\mathrm{FN}}$, ..., $P^k_{\mathrm{FP}}/P^k_{\mathrm{FN}}$ when applied to images from actor $1, \ldots, k$, respectively. Now suppose that we have to apply the classifier(s) to a target who is, let us say, uniformly randomly picked from the $k$ actors. What is our error rate?

| Training actor \ Testing actor | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.0029 | 0.0157 | 0.1330 | 0.2368 | 0.0169 | 0.1095 | 0.0592 | 0.0470 | 0.0823 |
| 2 | 0.0218 | 0.0052 | 0.1662 | 0.1011 | 0.0270 | 0.1077 | 0.0482 | 0.0463 | 0.0629 |
| 3 | 0.3648 | 0.3341 | 0.0273 | 0.3887 | 0.2450 | 0.2373 | 0.3692 | 0.2953 | 0.3142 |
| 4 | 0.0340 | 0.0132 | 0.1848 | 0.0043 | 0.0141 | 0.0913 | 0.0338 | 0.0415 | 0.0637 |
| 5 | 0.0414 | 0.0108 | 0.1141 | 0.0780 | 0.0031 | 0.0938 | 0.0310 | 0.0248 | 0.0629 |
| 6 | 0.0694 | 0.0298 | 0.1697 | 0.0427 | 0.0357 | 0.0663 | 0.0730 | 0.0303 | 0.0718 |
| 7 | 0.0177 | 0.0099 | 0.2149 | 0.1000 | 0.0090 | 0.1174 | 0.0101 | 0.0228 | 0.0647 |
| 8 | 0.0504 | 0.0343 | 0.1202 | 0.0944 | 0.0143 | 0.0943 | 0.0495 | 0.0108 | 0.0842 |
| 9 | 0.0357 | 0.0113 | 0.1941 | 0.1742 | 0.0179 | 0.1175 | 0.0829 | 0.0459 | 0.0540 |

| | matched cases | | | mismatched cases | | |
|---|---|---|---|---|---|---|
| | $\mu_1$ | $\mu_2$ | $\mu_\infty$ | $\mu_1$ | $\mu_2$ | $\mu_\infty$ |
| Aggregate error rates | 0.0204 | 0.0315 | 0.0663 | 0.0981 | 0.1369 | 0.3887 |

Table 2: Error rate ($P_{\text{E}}$) of FLD classifiers using $\mathcal{CF}^*$ features, tested against cover and 0.05bpnc nsF5 stego images. Each classifier was trained on 6000 of one actor's images, and tested against 3000 of another. The aggregate benchmarks are explained in section 2.

If there is just one image to test, then our false negative rate is indeed the mean false negative

$$P_{\text{FN}} = \sum_{i=1}^{k} \Pr[\text{actor } i \text{ picked}] P^i_{\text{FN}} = \overline{P_{\text{FN}}},$$

and similar for false positive. But if, as seems very likely in almost any steganalysis application, we are actually given $n > 1$ images from this actor and asked whether the actor is *guilty* (of using steganography), the situation is more complicated, because this is an example of *pooled steganalysis*,[13] which has not been solved.

First, suppose that $P^i_{\text{FP}} = 0$ for all $i$. Then there is only one sensible detector given the results on $n$ images: return a guilty verdict if *any* of the $n$ images give a positive detection. The false positive rate of this pooled detector is zero and its false negative rate is

$$P_{\text{FN}} = \sum_{i=1}^{k} \Pr[\text{actor } i \text{ picked}] \Pr[\text{no false negatives in } n \text{ images}] = \frac{1}{k} \sum_{i=1}^{k} \big(P^i_{\text{FN}}\big)^n.$$

This is not the mean false negative rate; when $n = 2$ it is the mean *squared* error rate, and as $n \to \infty$ it tends to $\frac{1}{k}(\max_i P^i_{\text{FN}})^n$ in the sense that

$$\frac{P_{\text{FN}}}{\big(\frac{1}{k} \max_i P^i_{\text{FN}}\big)^n} \to 1.$$

Thus the *worst-case* error dominates for large $n$. (Curiously, if there are $l$ actors tied for worst error rate then the asymptotic error rate is $l$ times larger.) That makes intuitive sense: if the experiment were repeated many times with different actors under suspicion, most of the mistakes would be made on the most difficult actor.

The same is true, with additional complication, for arbitrary $P^i_{\text{FP}}$ and $P^i_{\text{FN}}$. If we count how many positive detections arise from $n$ images (a strategy from Ref. 13), and assuming that innocent/guilty actors transmit $n$ cover/stego objects (no mixtures) where $n$ is large, then making the Gaussian approximation to the Binomial we approximate the distribution of this count as

$$\mathrm{N}\big(n(1 - P^i_{\text{FN}}), nP^i_{\text{FN}}(1 - P^i_{\text{FN}})\big)$$

in the case of guilty actor $i$, and

$$\mathrm{N}\big(nP^i_{\mathrm{FP}}, nP^i_{\mathrm{FP}}(1-P^i_{\mathrm{FP}})\big)$$

for innocent actor $i$. It can be shown that the decision with minimum $P_{\mathrm{E}}$ splits the deflection equally, which for actor $i$ gives equal false positive and negative rates of

$$\Pr\big[Z > \sqrt{n}d^i\big], \text{ where } d^i = \frac{1-P^i_{\mathrm{FP}}-P^i_{\mathrm{FN}}}{\sqrt{P^i_{\mathrm{FP}}(1-P^i_{\mathrm{FP}})}+\sqrt{P^i_{\mathrm{FN}}(1-P^i_{\mathrm{FN}})}} \tag{1}$$

and $Z$ is a standard Gaussian variable. (The deflection $d^i$ will be adapted to a measure of single-actor accuracy in section 5.)

Now for a uniformly random actor, because the tails of the Gaussian are exponential, with large $n$ the error rate converges to

$$P_{\mathrm{FP}} = P_{\mathrm{FN}} = \tfrac{1}{k}\Pr\big[Z > \sqrt{n}\min_i d^i\big]$$

with the worst-case actor again dominating. (Assuming no exact ties in the $d_i$).

Similar results exist for other setups as well. The result does not (unlike our first thought experiment) need the detector to be at all ignorant about the actor they investigating: it is simply because we expect, for large $n$, any detector's error rate to decay (probably exponentially in $n$), and so for large $n$ the actors with smaller error rates are dominated by the largest.

In this paper, we will use three summary measures of performance. Given $P_{\mathrm{E}}$ error rates $P^1_{\mathrm{E}}, \ldots, P^k_{\mathrm{E}}$ from $k$ different classifiers, and taking matched and mismatched cases separately, we display:

(i) the mean error rate $\mu_1 = \frac{1}{k}\sum_i P^i_{\mathrm{E}}$ (to reflect the case $n = 1$),

(ii) the root mean square (RMS) error rate $\mu_2 = \sqrt{\frac{1}{k}\sum_i \big(P^i_{\mathrm{E}}\big)^2}$ (to reflect the case $n = 2$ in the zero-false-positive thought experiment, or other small values of $n$; the square root is taken to give a similar scale to the other aggregate metrics),

(iii) maximum error rate $\mu_\infty = \max_i P^i_{\mathrm{E}}$ (to reflect the case of large $n$).

These metrics are displayed at the end of Table. 2) for the standard steganalysis classifier. By Jensen's inequality, we always have $\mu_1 \le \mu_2 \le \mu_\infty$, but the numbers will be closer for more stable classifiers.

## 3. MISMATCH DUE TO COVER FEATURE SHIFT

Our first investigation is whether model mismatch manifests as a simple shift in the cover sources, a situation shown abstractly in Figure 1. In the figure, different shapes denote different actors, and the solid shapes are covers: the hypothesis is (i) that the different actors' sources produce clusters of covers which are differently-centered, and additionally (ii) that the effect of steganography is the same for all actors. The figure shows a situation where, as larger payloads are embedded, the stego objects (shown as hollow shapes on a path from each cover) move in the same direction and at the same speed. This is only an abstraction and we would not expect the diagram to reflect the true situation, but it illustrates a particular type of model mismatch.

### 3.1 Measurement

To what extent is it true, that cover sources have different locations, and (more importantly) to what extent does this affect the accuracy of steganalysis? We might try to answer the question by drawing diagrams like Figure 1 for real steganalysis data, and indeed the authors have done so, but it is not easy to see what is going on: the features are strongly colinear, and there are thousands of dimensions. What may be apparent in some two-dimensional projections may not reflect the true situation.

Instead, we performed two experiments. First, we measured how far 'apart' were the centroids of each actors' cover images. Distance is not easy to measure in such a large-dimensional space, where the features are strongly

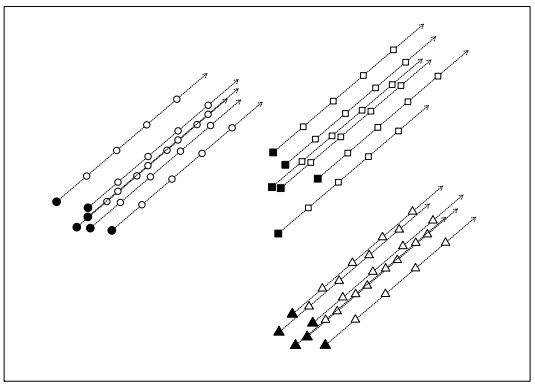

Figure 1: An abstract representation of the situation we examine in section 3. Different shapes represent features from different actors; filled shapes are cover images and hollow shapes represent the evolution of features as payload is added. In this case, we imagine that all actors' features are equally affected by payload, but the actors have differently-located cover clusters.

| | | Actor's whose centroid distance is measured... | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| ...from other actor's centroid | 1 | 0 | 0.5914 | 3.9337 | 12.5861 | 0.6679 | 1.9095 | 2.8292 | 0.7221 | 0.7949 |
| | 2 | 1.4019 | 0 | 7.2735 | 3.6786 | 1.3544 | 0.3831 | 1.0779 | 0.4848 | 1.0259 |
| | 3 | 12.8420 | 12.4698 | 0 | 19.0288 | 10.5691 | 9.2591 | 13.9270 | 12.0040 | 12.5879 |
| | 4 | 5.0152 | 0.5052 | 8.7721 | 0 | 0.7035 | 0.3730 | 2.0112 | 1.9772 | 1.0642 |
| | 5 | 0.4989 | 1.0932 | 2.5417 | 5.3951 | 0 | 2.1761 | 2.2975 | 2.5644 | 2.0004 |
| | 6 | 2.1884 | 0.8123 | 5.0924 | 1.7257 | 0.5185 | 0 | 1.0476 | 1.0968 | 2.0817 |
| | 7 | 0.2635 | 0.3514 | 7.5396 | 2.9236 | 0.5322 | 0.7672 | 0 | 0.4999 | 0.1027 |
| | 8 | 1.6527 | 1.0550 | 2.0083 | 5.0935 | 1.2849 | 1.3361 | 1.8654 | 0 | 3.6914 |
| | 9 | 0.2680 | 0.6131 | 4.8203 | 3.3132 | 0.4497 | 0.9585 | 1.2621 | 1.3073 | 0 |

Correlation with Table 2: $\rho = 0.91$ $\tau = 0.51$

Table 3: Distances of centroids of each actors' cover image features, projected on the FLD regression vectors trained on images from each actor.

colinear and of different scale. One could perform a whitening procedure, but that does not fix the different scales of the features, and fixing scaling introduces shear which can change the results. We argue that, if the location mismatch occurs, it does not matter how 'far' in the feature space the cover images of different actors are, but how far they are after being projected in the direction vector of the fisher linear discriminant (hereafter abbreviated as the regression vector, due to the connection between ordinary least-squares regression and the FLD). If the features of different actors' cover images are far apart in the full feature space, but close after being projected on the regression vector, then the mismatch is irrelevant to steganalysis accuracy.

Table 3 shows the results. The distances are not symmetric because the displacements between two actors' centroids will be different when projected onto their different regression vectors. Comparing this distance with errors $P_E$ in Table 2 we notice a strong relationship between both quantities: the linear correlation coefficient between non-diagonal elements of the matrices is $\rho = 0.91$, and Kendall's rank correlation coefficient (which is robust to outliers) is $\tau = 0.51$. These numbers suggests that location plays an important role in the accuracy penalty induced by this example of model mismatch.

### 3.2 Mitigation

How can we mitigate the effect of shifted locations, between different actors? One option, only available *in the case of partial mismatch* (recall: the detector has a small amount of known cover data from each actor, but

| Training actor \ Testing actor | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.0029 | 0.0142 | 0.1260 | 0.2332 | 0.0193 | 0.1146 | 0.0369 | 0.0418 | 0.0793 |
| 2 | 0.0214 | 0.0052 | 0.1135 | 0.0747 | 0.0198 | 0.1082 | 0.0453 | 0.0448 | 0.0612 |
| 3 | 0.0788 | 0.0758 | 0.0273 | 0.2186 | 0.0209 | 0.1185 | 0.0848 | 0.0296 | 0.1375 |
| 4 | 0.0331 | 0.0152 | 0.1220 | 0.0043 | 0.0130 | 0.0914 | 0.0361 | 0.0313 | 0.0662 |
| 5 | 0.0395 | 0.0141 | 0.1061 | 0.0624 | 0.0031 | 0.0897 | 0.0421 | 0.0184 | 0.0733 |
| 6 | 0.1160 | 0.0417 | 0.1118 | 0.0460 | 0.0347 | 0.0663 | 0.0818 | 0.0337 | 0.0844 |
| 7 | 0.0173 | 0.0086 | 0.1323 | 0.0732 | 0.0100 | 0.1191 | 0.0101 | 0.0254 | 0.0647 |
| 8 | 0.0738 | 0.0447 | 0.1104 | 0.0960 | 0.0147 | 0.0936 | 0.0695 | 0.0108 | 0.1059 |
| 9 | 0.0339 | 0.0086 | 0.1278 | 0.1496 | 0.0194 | 0.1182 | 0.0793 | 0.0537 | 0.0540 |

| | matched cases | | | mismatched cases | | |
|---|---|---|---|---|---|---|
| | $\mu_1$ | $\mu_2$ | $\mu_\infty$ | $\mu_1$ | $\mu_2$ | $\mu_\infty$ |
| Aggregate error rates | 0.0204 | 0.0315 | 0.0663 | 0.0691 | 0.0838 | 0.2332 |

Table 4: Error rate ($P_{\mathrm{E}}$) of FLD classifiers, when the centroid of the training cover data was subtracted from each actors' features.

not enough to train a classifier), is to subtract an estimated centroid of each actor's cover features, effectively centering all of their cover clusters at the origin. We do not need much training data for this, because it only requires estimating a mean, the accuracy of which will be independent of the dimension of the features (the same is not true for the covariance matrix, which is why we expect insufficient data to train a personalized FLD for each actor).

When we modify the FLD classifier to subtract the mean, estimated from the 3000 image testing set, we see an immediate improvement in mismatched accuracy, displayed in Table 4. (There is, of course, no change to matched classifier accuracy, because the FLD is invariant when the same shift is applied to both classes.) Depending on which aggregation metric is used, the accuracy of mismatched detectors reduces from approximately five to approximately three times that of matched accuracy. In this sense, about half of the mismatch accuracy has been eliminated.

This mitigation does rely on the partial mismatch case, but we will be able to work around this in the following section.

## 4. MISMATCH DUE TO DIFFERENT CHANGE IN STEGO FEATURES

With differences in cover mean center removed, are there additional mismatches between actors' sources? Our second investigation is whether model mismatch also manifests as different effects of payload. In Figure 2 we show two scenarios, where the actors' features move under payload (i) in a different direction, or (ii) at different rates. Again, we emphasise that this is only an abstract (in reality, all images move in slightly different directions, and their paths are not straight).

### 4.1 Measurement

Following similar methodology to section 3, we measure the extent of these two sources of mismatch. We calculate the FLD regression vector for each actor, denoting them $w_1, \ldots, w_9$. To test for different payload directions we computed the cosine of the angles between $w_i$ and $w_j$,

$$\cos \alpha_{i,j} = \frac{w_i \cdot w_j}{\|w_i\| \|w_j\|}.$$

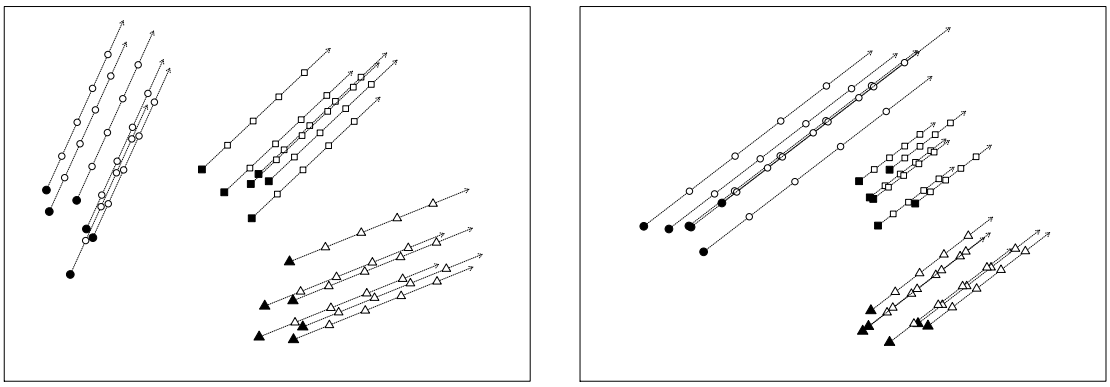

Figure 2: An abstract representation of the situations we examine in section 3. Left, each actors' features move in a different direction under embedding; Right, the features' rate of change with respect to payload is different for each actor. In both cases the cover mean of the actors may be different.

| | First actor | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Second actor | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 1.000 | 0.310 | 0.143 | 0.265 | 0.299 | 0.144 | 0.326 | 0.214 | 0.246 |
| 2 | 0.310 | 1.000 | 0.146 | 0.280 | 0.283 | 0.163 | 0.310 | 0.207 | 0.234 |
| 3 | 0.143 | 0.146 | 1.000 | 0.139 | 0.162 | 0.107 | 0.142 | 0.114 | 0.117 |
| 4 | 0.265 | 0.280 | 0.139 | 1.000 | 0.266 | 0.148 | 0.299 | 0.191 | 0.225 |
| 5 | 0.299 | 0.283 | 0.162 | 0.266 | 1.000 | 0.159 | 0.304 | 0.215 | 0.230 |
| 6 | 0.144 | 0.163 | 0.107 | 0.148 | 0.159 | 1.000 | 0.131 | 0.127 | 0.126 |
| 7 | 0.326 | 0.310 | 0.142 | 0.299 | 0.304 | 0.131 | 1.000 | 0.215 | 0.243 |
| 8 | 0.214 | 0.207 | 0.114 | 0.191 | 0.215 | 0.127 | 0.215 | 1.000 | 0.171 |
| 9 | 0.246 | 0.234 | 0.117 | 0.225 | 0.230 | 0.126 | 0.243 | 0.171 | 1.000 |

Correlation with Table 4: $\rho = -0.52$ $\tau = -0.44$

| | First actor | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 0.000 | 0.096 | 0.084 | 0.039 | 0.166 | 0.099 | 0.297 | 0.141 | 0.421 |
| 2 | 0.096 | 0.000 | 0.180 | 0.135 | 0.262 | 0.014 | 0.201 | 0.237 | 0.324 |
| 3 | 0.084 | 0.180 | 0.000 | 0.045 | 0.082 | 0.183 | 0.382 | 0.056 | 0.507 |
| 4 | 0.039 | 0.135 | 0.045 | 0.000 | 0.127 | 0.138 | 0.336 | 0.102 | 0.461 |
| 5 | 0.166 | 0.262 | 0.082 | 0.127 | 0.000 | 0.266 | 0.466 | 0.028 | 0.592 |
| 6 | 0.099 | 0.014 | 0.183 | 0.138 | 0.266 | 0.000 | 0.197 | 0.240 | 0.320 |
| 7 | 0.297 | 0.201 | 0.382 | 0.336 | 0.466 | 0.197 | 0.000 | 0.440 | 0.122 |
| 8 | 0.141 | 0.237 | 0.056 | 0.102 | 0.028 | 0.240 | 0.440 | 0.000 | 0.566 |
| 9 | 0.421 | 0.324 | 0.507 | 0.461 | 0.592 | 0.320 | 0.122 | 0.566 | 0.000 |

Correlation with Table 4: $\rho = -0.10$ $\tau = -0.09$

Table 5: Left, cosines of the angles between regression vectors of different actors, $\cos \alpha_{i,j}$. Right, relative difference, between different actors, of rate of change of stego features, $r_{i,j}$.

Values of 1 indicate perfect alignment, which happens only for $i = j$, and lower values indicate less alignment; it is difficult to interpret values between 0 and 1 directly, because in high-dimensional space most vectors are nearly orthogonal, but their relative size is an indication of relative alignment.

To test for different rates of change (in the direction of the regression), we calculated a normalized quantity

$$r_{i,j} = \frac{|\|w_i\| - \|w_j\||}{\sqrt{\|w_i\|\|w_j\|}},$$

for which higher values indicate more difference.

These values are displayed in Table 5; they are of course symmetric about the diagonal. Following the same methodology as before we correlate them with the residual error after cover location has been removed, Table 4. There is a significant correlation in the first case ($\rho = -0.52$, $\tau = -0.44$: larger angles indicate higher error), and not the second ($\rho = -0.10$, $\tau = -0.09$); we conclude that different actors' features do indeed move in significantly different directions under embedding, but not at different rates. Furthermore, we observe a significant correlation between the entries of Table 5 (left) and Table 3 ($\rho = -0.39$, $\tau = -0.25$). This suggests that detectors with a similar stego directions have similar intercept as well, and significantly different directions of travel imply large distances between the centroids. This relationship is important, since the next subsection shows that the direction of travel can (to some extent) be estimated.

### 4.2 Mitigation

In this section we realistically assume that the steganalyst has access to images from a number of cover sources (but not necessarily that of images being classified). Suppose that each cover source is used to train a detector specialized to it, but during classification, the steganalyst tries to use the detector for 'best matched' source. We will simulate this scenario by using eight detectors to classify images from the remaining ninth actor, and measure error on that actor. The result of the eight classifiers can be fused by a voting strategy, putting them into a kind of ensemble. The simplest strategy to pool outputs from multiple detectors is the majority vote; we consider this the baseline for comparison.

If the picture in Figure 2 were true, it might be possible to estimate the direction (and speed) of a particular image, and therefore the 'best matched' in some sense, by re-embedding. This will work for low embedding rates where distortion is unlikely to cancel out (when changes are placed on top of changes). Note that this works even in the totally mismatched case; in the partial mismatch scenario we may also be able to center the features.

We can estimate the direction of change by embedding a small payload (say 0.01bpnc) to the observed image. The difference this induces in the features will be denoted $\delta$. Two measures to find the best matching detector were investigated. One is the cosine of the angle between $\delta$ and the regression vector $w_i$ of the $i$-th classifier,

$$\cos\alpha_i = \frac{w_i \cdot \delta}{\|w_i\|\|\delta\|}.$$

The second measure, called *sensitivity*, is

$$s_i = |w_i \cdot \delta|,$$

which measures how sensitive is the classifier to this particular embedding effect. The rationale behind the second measure is that the more closely the regression vector is aligned with the direction $\delta$, the more sensitive it should be to the embedding. We emphasize that $w_i$ was not normalized to have unit norm: in our implementation of the FLD, the regression vector $w_i$ was calculated as

$$w_i = (X_{\text{cov}}^{\text{T}} X_{\text{cov}} + X_{\text{stg}}^{\text{T}} X_{\text{stg}})^{-1} (X_{\text{stg}}^{\text{T}}\mathbf{1} - X_{\text{cov}}^{\text{T}}\mathbf{1}),$$

where $X_{\text{cov}}$, $X_{\text{stg}}$ denote feature matrices from cover and stego images respectively, and $\mathbf{1}$ a vector of one of the appropriate length. The scale of the sensitivity measure is determined by the payload in training stego images as follows: imagine for now that covers are centered at the origin, and all payloads are equal; then a feature vector $x$ with $|w_i \cdot x| = 1$ is exactly in the middle of stego vectors after being projected on $w_i$. Thus *projections onto* $w_i$ are normalized with respect to the payload, which is aligned with our goal.

We use the measure $\cos\alpha_i$ or $s_i$ to weight the votes in the ensemble: either weighting all votes according to the metric, or picking only the output of the classifier with the best weight. We display the resulting (completely mismatched, above, and partially mismatched applying centering, below) classification accuracy in Table 6. The error rates are $P_{\text{E}}$, as well as their aggregates over the nine classifiers.

It seems that using all eight detectors is always better than using only one. In the case of no knowledge about the mean of cover images, using either sensitivity measure improves the worst-case detection error rate by about 2.5%, over the baseline of the entire ensemble. Comparing against the aggregate metrics of Tables 2 and 4, we have been able to reduce the mismatch error rates from approximately five times that of matched, to approximately two times (in the total mismatch case, when we cannot centre, where we use the ensemble of classifiers for other actors) or 1.5 times (in the case of partial mismatch, when we can both centre and deploy the ensemble in baseline voting mode).

## 5. MISMATCH DUE TO FALSE CERTAINTY

One reason that binary classifiers make mistakes in mismatched training/testing cases is that they give a false certainty: in areas of feature space that were sparsely or not covered by the training data, they still make a decision even though there is little evidence for it. This is illustrated in the top part of Figure 3, where a binary support vector machine (SVM) misclassifies, in the mismatched case, some cover objects as stego even though they are on the opposite side of the stego 'cluster' in a region where there was no training data.

| | Testing actor | | | | | | | | | Aggregate error | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Voting | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | $\mu_1$ | $\mu_2$ | $\mu_\infty$ |
| Equal weight | 0.0151 | 0.0060 | 0.1366 | 0.0921 | 0.0064 | 0.0586 | 0.0261 | 0.0147 | 0.0394 | 0.0439 | 0.0627 | 0.1366 |
| Weight by $\cos\alpha_i$ | 0.0128 | 0.0054 | 0.1160 | 0.0978 | 0.0074 | 0.0630 | 0.0259 | 0.0197 | 0.0416 | 0.0433 | 0.0593 | 0.1160 |
| Weight by $s_i$ | 0.0133 | 0.0058 | 0.1109 | 0.0990 | 0.0079 | 0.0631 | 0.0251 | 0.0208 | 0.0419 | 0.0431 | 0.0584 | 0.1109 |
| Only $\arg\max\cos\alpha_i$ | 0.0277 | 0.0140 | 0.1406 | 0.1156 | 0.0166 | 0.1041 | 0.0526 | 0.0409 | 0.0721 | 0.0649 | 0.0796 | 0.1406 |
| Only $\arg\max s_i$ | 0.0412 | 0.0115 | 0.1201 | 0.0795 | 0.0160 | 0.0989 | 0.0342 | 0.0338 | 0.0680 | 0.0559 | 0.0675 | 0.1201 |
| *after centering:* | | | | | | | | | | | | |
| Equal weight | 0.0274 | 0.0109 | 0.0776 | 0.0744 | 0.0058 | 0.0584 | 0.0344 | 0.0109 | 0.0519 | 0.0391 | 0.0479 | 0.0776 |
| Weight by $\cos\alpha_i$ | 0.0201 | 0.0075 | 0.0806 | 0.0718 | 0.0059 | 0.0609 | 0.0289 | 0.0139 | 0.0454 | 0.0372 | 0.0468 | 0.0806 |
| Weight by $s_i$ | 0.0235 | 0.0088 | 0.0790 | 0.0697 | 0.0059 | 0.0601 | 0.0285 | 0.0128 | 0.0470 | 0.0373 | 0.0463 | 0.0790 |
| Only $\arg\max\cos\alpha_i$ | 0.0275 | 0.0135 | 0.1177 | 0.1156 | 0.0173 | 0.1036 | 0.0387 | 0.0352 | 0.0719 | 0.0601 | 0.0737 | 0.1177 |
| Only $\arg\max s_i$ | 0.0403 | 0.0140 | 0.1101 | 0.0722 | 0.0169 | 0.0958 | 0.0397 | 0.0253 | 0.0754 | 0.0544 | 0.0648 | 0.1101 |

Table 6: Error, when eight detectors are used to classify images on the remaining ninth actor. The first column denotes the voting strategy of the ensemble: a baseline method of equal weight, weighting all classifiers according to $\cos\alpha_i$ or $s_i$, or using the one classifier with best $\cos\alpha_i$ or $s_i$.

We suggest that there could be considerable value in a steganalysis detector which is prepared to admit to 'don't knows' in regions of the space that were not well-covered by training data. This leads to a form of 3-valued logic. (A more general technique is that of logistic regression, where probabilities are estimated, but logistic regression has shown rather poor performance on steganalysis tasks.[15])

Instead of a single binary classifier, we propose to combine two one-class detectors, one trained on covers and the other on stego images. This construction was proposed in Ref. 16 for multi-class classification. Here, we use it for classification and confidence estimation at the same time. Accordingly, there are four cases:

(i) Cover detector returns *positive*, stego detector returns *negative*. This is a negative detection, and the probability of false negative detections is still denoted $P_{\mathrm{FN}}$.

(ii) Cover detector returns *positive*, stego detector returns *negative*. This is a positive detection, and the probability of false positive detections is still denoted $P_{\mathrm{FP}}$.

(iii) Both detectors return *negative*. This is a 'don't know' arising from a lack of data: the classified object is in a novel region for which there was not sufficient training data. We denote the probability of this class as $P_{\mathrm{DN}}$.

(iv) Both detectors return *positive*. This is also a 'don't know', in this case arising from contradictory evidence: the classified object would probably have been near the decision boundary of a 2-class classifier. We denote the probability of this class as $P_{\mathrm{DP}}$.

(For some metrics we might alternatively want to break down the 'don't know' cases into (a) cover objects classified as 'don't know', and (b) stego objects classified as 'don't know'. We will postpone this to further work.)

As a pilot study, we implemented one-class detectors as one-class support vector machines (1-SVM) with Gaussian kernel.[17] Since they are more effective in low dimensions, we reduced the $\mathcal{CF}^*$ feature space to 20 dimensions, by 10 repetitions of the CLS algorithm.[14] In subsection 5.2 we will compare their accuracy with that of binary support vector machines (2-SVM) with Gaussian kernel on the same features.

Both types of machine require hyper-parameter optimization: both have a kernel width $\gamma$, the 1-SVMs have the fraction of outliers $\nu$, and 2-SVMs have a regularization parameter $\lambda$. We used (a) $\ln 2\gamma$ taking five equal steps between $-2l_j$ and $-2l_m$, where $l_j$ is the natural log of the average distance of a sample to its nearest

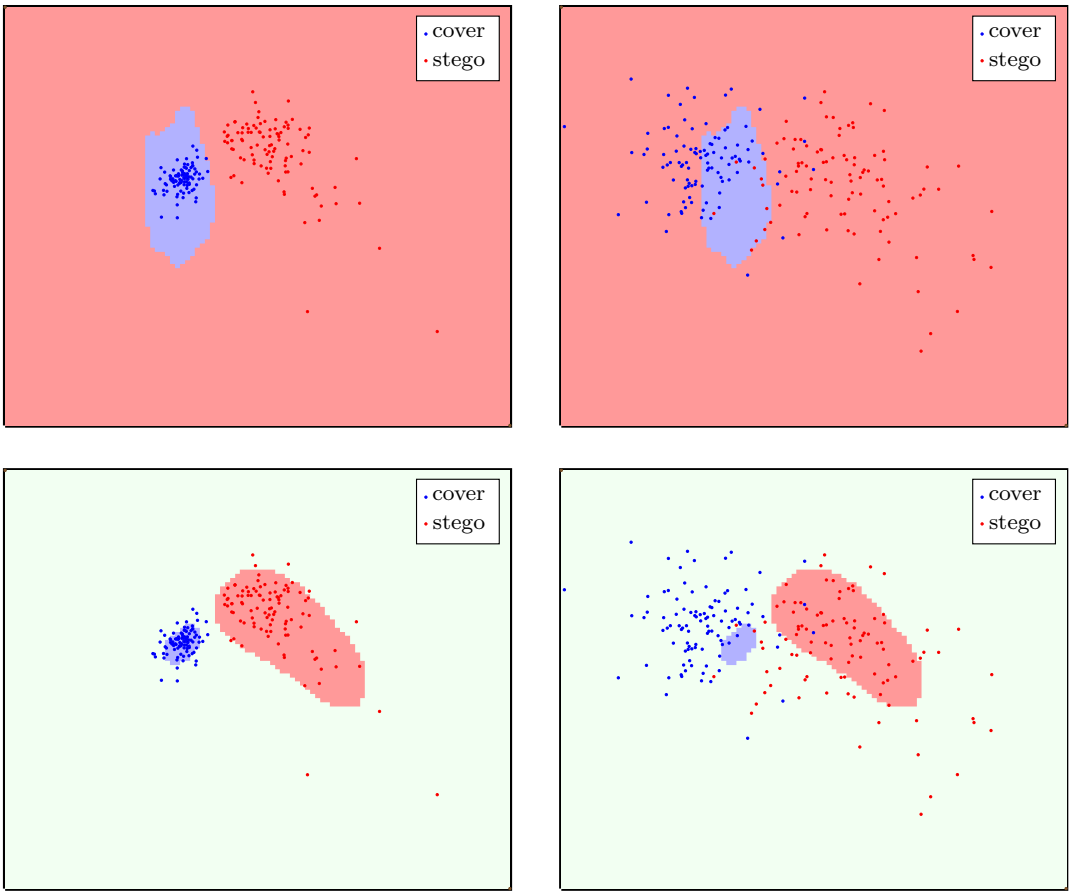


Figure 3: Depiction of binary (above) and 3-valued (below) classifiers, 2-SVMs and 1-SVMs respectively. The left figures represent the matched case, and the right figures a mismatched case. For these pictures, the features have been reduced to 2 dimensions using the CLS method.[14]

neighbour, and $l_m$ the natural logarithm of the median distance between samples, (b) $\log_{10}\lambda \in \{-4, \ldots, 4\}$, (c) $\nu \in \{0.005, 0.01, 0.02, 0.04, 0.06\}$.

But how should we optimize the parameters, in the 1-SVM case? And how should we evaluate their performance after optimization? For this, we need a metric which understands the value of 'don't know' results, which is the subject of the next section.

### 5.1 Accuracy metrics in the presence of 'don't know'

The value of a 3-valued detector depends on how you use it. If the steganalyst can have as much evidence as they want, they can afford to ignore all the 'don't know' cases and use only the positive and negative classes; then the appropriate metric is still

$$P_{\mathrm{E}} = \tfrac{1}{2}(P_{\mathrm{FP}} + P_{\mathrm{FN}}),$$

which does not penalize 'don't know' cases at all. But if we tune paired 1-SVMs using this metric, they will almost never return a positive or negative detection, because they can minimize error by conservative output. The same is true of metrics which consider the conditional probability that the object is truly cover/stego, given that the detector says so.

On the other hand, if the steganalyst simply guesses 'cover' or 'stego' whenever the detector responds with a 'don't know', the appropriate metric is

$$P_{\mathrm{E}'} = \tfrac{1}{2}(P_{\mathrm{FP}} + P_{\mathrm{FN}}) + \tfrac{1}{4}(P_{\mathrm{DP}} + P_{\mathrm{DN}}).$$

This penalizes 'don't know' cases very heavily, and 1-SVMs trained to optimize it will fall into the same errors as 2-SVMs in making decisions when they have no evidence to do so (it never hurts to guess).

We find an appropriate middle-ground by returning to a hypothetical pooled steganalysis situation. Suppose that the steganalyst has $n$ images, and wants to determine whether the actor is guilty (using steganography in all images) or innocent (using steganography in none). They ignore 'don't know' cases, which reduces their evidence base, and return a guilty verdict if more than a certain proportion of the remaining decisions are stego. On average, they will be making a decision based on $n(1-P_{\mathrm{DP}}-P_{\mathrm{DN}})$ images, instead of $n$, and the appropriate deflection metric (c.f. 1) becomes

$$d = \frac{(1-P_{\mathrm{FP}}-P_{\mathrm{FN}})\sqrt{1-P_{\mathrm{DP}}-P_{\mathrm{DN}}}}{\sqrt{P_{\mathrm{FP}}(1-P_{\mathrm{FP}})}+\sqrt{P_{\mathrm{FN}}(1-P_{\mathrm{FN}})}}. \tag{2}$$

We propose this metric for 3-valued detectors, noting that *higher* values of $d$ indicate better evidence.

We observe that all of the above formulae use only the sum of $P_{\mathrm{DP}}$ and $P_{\mathrm{DN}}$, treating 'don't know' cases equally. Perhaps further consideration will find different value between the two cases. We should also note that all the results of this section will be incomparable with those of sections 2-4, because (a) the metrics are differently-motivated, and (b) a reduced set of features is being used for detection.

We will also want to aggregate sets of deflection values arising from matched and mismatched cases. The average has no particular meaning in this case, and it is difficult to justify any particular combination other that the following two:

$$d_p = \frac{\big(1-P_{\mathrm{FP}}-P_{\mathrm{FN}}\big)\sqrt{1-P_{\mathrm{DP}}-P_{\mathrm{DN}}}}{\sqrt{P_{\mathrm{FP}}(1-P_{\mathrm{FP}})}+\sqrt{P_{\mathrm{FN}}(1-P_{\mathrm{FN}})}}$$

where the error rates $P_{\mathrm{FP}}$, etc., are *pooled* over all subcases (e.g. all matched cases). This is akin to $\mu_1$ in that it refers to average error rates. The second aggregative metric will be

$$d_\infty = \max d_i$$

where the $d_i$ are the deflection values for each of the subcases. This is akin to $\mu_\infty$ in that it represents the worst case, and is relevant for large values of $n$.

### 5.2 Results

An example result from the paired 1-SVMs is displayed in the lower part of Figure 3, in parallel situations to the 2-SVM case above. The paired one-class detectors are rather conservative, only giving decisions in regions where there was plenty of training data: this results in many 'don't knows' when tested on mismatched data, in this case particularly covers, but it makes very few mistakes.

We optimized hyperparameters $\gamma$ and $\nu$ for the 1-SVMs in two different ways:

(a) Targeting the matched case, we chose hyperparameters for each machine to minimize the deflection score on the matched (but disjoint) testing data.

(b) Targeting the mismatched case, we chose hyperparameters for each machine to minimize the worst-case deflection score when tested on the eight mismatched training sets.

Table 7 is analogous to Table 2, displaying the deflection scores when paired 1-SVMs trained on each actor are tested on each other. Optimizing for the mismatched cases seems to reduce the worst cases of mismatch a little, but not very much. Table 8 aggregates the performance of 1-SVM and 2-SVM detectors in the matched and mismatched cases, showing error rates and aggregated deflection scores according to subsection 5.1. The differently-optimized cases of 1-SVM do not in fact perform very differently from each other, but their improvement over the 2-SVM is significant: they make around one fifth of the mistakes, in both matched and mismatched case, but at the cost of reporting 'don't know' in about 25% of matched cases and 55% of mismatched cases.

According to our simple pooled steganalysis model, the 1-SVMs increase deflection by a factor of approximately two. The way to interpret deflection scores is via (1): for equivalent performance when classifying an actor based on $n$ objects, we need $n \propto 1/d^2$. Therefore increasing average- or worst-case deflection by a factor of two is equivalent to reducing the necessary number of observations by a factor of four.

| Training actor | Testing actor | | | | | | | | | Testing actor | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 15.99 | 3.45 | 1.56 | 2.90 | 4.40 | 2.17 | 2.32 | 2.13 | 1.64 | 8.90 | 3.25 | 1.53 | 2.76 | 4.27 | 3.10 | 2.34 | 2.27 | 1.57 |
| 2 | 5.15 | 13.76 | 1.75 | 3.35 | 5.79 | 2.13 | 2.66 | 3.50 | 2.41 | 5.46 | 10.60 | 1.78 | 3.43 | 4.83 | 2.01 | 1.86 | 2.68 | 2.54 |
| 3 | 2.28 | 1.57 | 6.50 | 1.19 | 2.12 | 0.99 | 0.98 | 0.93 | 0.95 | 4.06 | 2.36 | 4.54 | 1.83 | 2.74 | 1.39 | 1.12 | 1.02 | 1.35 |
| 4 | 5.70 | 5.85 | 1.47 | 11.99 | 5.36 | 1.91 | 3.34 | 2.82 | 2.43 | 4.94 | 3.70 | 1.43 | 5.80 | 4.09 | 1.96 | 2.50 | 2.33 | 2.03 |
| 5 | 4.44 | 6.87 | 2.11 | 2.41 | 35.71 | 1.94 | 3.24 | 2.88 | 2.12 | 4.30 | 5.62 | 3.82 | 3.51 | 8.19 | 1.86 | 3.48 | 2.95 | 1.96 |
| 6 | 2.56 | 6.68 | 2.31 | 2.58 | 4.21 | 6.08 | 1.83 | 3.75 | 3.56 | 3.06 | 8.10 | 1.92 | 3.28 | 4.83 | 5.02 | 2.16 | 4.03 | 3.23 |
| 7 | 7.35 | 6.77 | 1.55 | 2.86 | 7.17 | 1.66 | 8.01 | 3.69 | 2.16 | 7.69 | 5.96 | 1.58 | 2.94 | 8.38 | 1.87 | 6.13 | 3.73 | 1.78 |
| 8 | 4.21 | 4.20 | 2.38 | 2.84 | 8.22 | 1.95 | 3.51 | 22.08 | 2.15 | 3.77 | 3.14 | 2.19 | 3.37 | 6.08 | 2.47 | 2.86 | 7.99 | 2.25 |
| 9 | 5.12 | 18.48 | 1.64 | 1.89 | 7.98 | 2.95 | 5.24 | 5.23 | 4.79 | 4.97 | 14.46 | 1.50 | 1.24 | 7.64 | 2.82 | 4.70 | 4.75 | 4.56 |

Table 7: Deflection scores ($d$) of paired 1-SVM classifiers, using $\mathcal{CF}^*$ features reduced to 20 dimensions by the CLS method.[14] Left, the 1-SVM hyperparameters were optimized on matched data. Right, on mismatched data.

| | matched cases | | | | | | mismatched cases | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | pooled error rates | | | | $d_p$ | $d_\infty$ | pooled error rates | | | | $d_p$ | $d_\infty$ |
| | $P_{\text{FP}}$ | $P_{\text{FN}}$ | $P_{\text{DP}}$ | $P_{\text{DN}}$ | | | $P_{\text{FP}}$ | $P_{\text{FN}}$ | $P_{\text{DP}}$ | $P_{\text{DN}}$ | | |
| 2-SVMs | 0.020 | 0.022 | 0.000 | 0.000 | 3.345 | 1.940 | 0.165 | 0.059 | 0.000 | 0.000 | 1.279 | 0.465 |
| 1-SVMs (a) | 0.004 | 0.002 | 0.192 | 0.046 | 8.176 | 4.785 | 0.041 | 0.008 | 0.186 | 0.356 | 2.258 | 0.926 |
| 1-SVMs (b) | 0.006 | 0.006 | 0.116 | 0.082 | 5.814 | 4.542 | 0.028 | 0.010 | 0.114 | 0.425 | 2.472 | 1.017 |

(a) optimized for matched data

(b) optimized for mismatched data

Table 8: Comparison of 2-SVMs and 1-SVMs by pooled error rates and aggregated deflection metrics which appropriately value 'don't know' cases.

## 6. SUMMARY

This paper has presented some in-depth experiments to measure different types of cover mismatch in steganalysis, focusing on a particular set of nine image sources and a single type of steganography and detector. Our findings can be summarised as follows:

- The mean error rate, across different mismatched cases, is probably an incorrect benchmark (sect. 2); scores weighted more towards the worst case better value a detector's stability and in the case of large numbers of images from the same actor.
- In the case of mismatch investigated here (different uploaders to Flickr, no differences in JPEG quantization tables), model mismatch can cause an enormous penalty and error rates increase approximately fivefold (Tab. 2). This experiment mimics realistic steganalysis conditions.
- A significant amount of this mismatch penalty is due to actors' cover features being located at different centers in feature space (Tab. 3). If a small amount of matched training data is available, the features can be centered at the origin and the accuracy penalty is reduced by around a half (Tab. 4).
- A further penalty can be ascribed to stego features moving in different directions, but not at different rates (Tab. 5). An ensemble of classifiers, which weights votes according to the location direction of travel of stego features, further mitigates model mismatch without the need for any matched training data (Tab. 6). In the case of partial mismatch, the error rate in mismatch cases is reduced down to only approximately 1.5 times that of matched cases.
- It is difficult to produce a good accuracy benchmark for 3-valued detectors which can output 'don't know'; we proposed the modified deflection score (2).

· 3-valued detectors made up of paired one-class SVMs can be used to reduce false certainty; according to a simple pooled steganalysis model, they improve deflection by a factor of approximately two (Tab. 8), which is equivalent to reducing by a factor of four the number of images required for particular accuracy.

Overall, some out of this mishmash of methods should be valuable for further development of steganalysis, mitigating model mismatch.

### 6.1 Directions for further work

Some additional research can extend and confirm the results of this pilot study: we envisage larger studies of the same questions using more actors, different embedding methods, different embedding rates (though this might be a distraction), different feature sets, and nonlinear classifiers. A more interesting question is to determine the *cause* of the mismatch effects: is it difference in camera model, image content, or post-processing? Can differently-designed features be invariant (or 'less variant') to these differences?

We have not addressed the significant question of model mismatch due to different JPEG quantization tables. For the practice of steganalysis in the real world[7] we would prefer not to have to rely on a bank of different classifiers for different JPEG types.[8] It is likely to be easier to adapt for JPEG quality factors which are close rather than those which are far apart: in the former, the histogram bins driving the features substantially overlap.

More abstractly, we can treat the problem as one of domain adaptation and apply methods from the machine learning literature.[3,18] (It may be valuable to identify and remove simple large factors such as centering first.) Both conservative and adaptive methods should be considered.

Finally, we stress the importance of using good metrics for steganalysis accuracy. We have suggested some metrics for mismatched cases, and also which incorporate 'don't knows', but there may be others equally compelling. For false certainty, perhaps logistic regression holds the answer.

Perhaps one reason that steganalysis has refined the matched, homogenous, case is that typical benchmarks for steganalysis accuracy (detection error in one data set, often BOSSBase[19]) only consider it. The authors are reminded of the ubiquity of 'Lena' in image processing articles from the 1970s to 1990s, and the Editor-in-Chief who wrote 'Who knows? We may even devise image compression schemes that work well across a broader class of images, instead of being tuned to Lena.'[20] Perhaps steganalysis researchers should similarly broaden their experimental base, and include mismatched metrics.

## ACKNOWLEDGMENTS

The work on this paper was supported by European Office of Aerospace Research and Development under the research grant numbers FA8655-11-3035 and FA8655-13-1-3020. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of EOARD or the U.S. Government.

The work of T. Pevný was also supported by the Grant Agency of Czech Republic under the project P103/12/P514.

## REFERENCES

[1] Kodovsky, J., Fridrich, J., and Holub, V., "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security* **7**(2), 432–444 (2012).

[2] Holub, V., Fridrich, J., and Denemark, T., "Random projections of residuals as an alternative to co-occurrences in steganalysis," in [*Proc. SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents XIV*], **8665**, 0L01–0L11 (2013).

[3] Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W., "A theory of learning from different domains," *Machine Learning* **79**(1-2), 151–175 (2010).

[4] Pevný, T., Bas, P., and Fridrich, J., "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on Information Forensics and Security* **5**(2), 215–224 (2010).

[5] Lubenko, I. and Ker, A. D., "Steganalysis with mismatched covers: do simple classifiers help?," in [*Proc. 13th ACM Workshop on Multimedia and Security (MM&Sec 2012)*], 11–18, ACM (2012).

[6] Barni, M., Cancelli, G., and Esposito, A., "Forensics aided steganalysis of heterogeneous images," in [*Proc. IEEE Conference on Acoustics Speech and Signal Processing*], 1690–1693 (2010).

[7] Ker, A. D., Bas, P., Böhme, R., Cogranne, R., Craver, S., Filler, S., Fridrich, J., and Pevný, T., "Moving steganography and steganalysis from the laboratory into the real world," in [*Proc. 1st ACM Workshop on Information Hiding and Multimedia Security*], 45–58, ACM (2013).

[8] Pevný, T., *Kernel methods in steganalysis*, PhD thesis, Binghamton University, SUNY, NY (2008).

[9] Makelberge, J. and Ker, A. D., "Exploring multitask learning for steganalysis," in [*Proc. SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents XI*], **8665**, 0N01–0N10 (2013).

[10] Kodovský, J. and Fridrich, J., "Steganalysis of JPEG images using rich models," in [*Proc. SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents XIV*], **8303**, 0A01–0A13 (2012).

[11] Holub, V., Fridrich, J., and Denemark, T., "Random projections of residuals as an alternative to co-occurrences in steganalysis," in [*Proc. SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents XV*], **8665**, 0L01–0L11 (2013).

[12] Pevný, T., Fridrich, J., and Ker, A. D., "From blind to quantitative steganalysis," *IEEE Transactions on Information Forensics and Security* **7**(2), 445–454 (2012).

[13] Ker, A. D., "Batch steganography and pooled steganalysis," in [*Proc. 8th Information Hiding Workshop*], *LNCS* **4437**, 265–281, Springer (2006).

[14] Pevný, T. and Ker, A. D., "The challenges of rich features in universal steganalysis," in [*Proc. SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents XV*], **8665**, 0M01–0M15 (2013).

[15] Lubenko, I. and Ker, A. D., "Steganalysis using logistic regression," in [*Proc. SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents XIII*], **7880**, 0K01–0K11, SPIE (2011).

[16] Yang, X.-Y., Liu, J., Zhang, M.-Q., and Niu, K., "A new multi-class SVM algorithm based on one-class SVM," in [*Proc. 7th international conference on Computational Science, Part III: ICCS 2007*], 677–684, Springer, Berlin, Heidelberg (2007).

[17] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C., "Estimating the support of a high-dimensional distribution," *Neural computation* **13**(7), 1443–1471 (2001).

[18] Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z., "Domain adaptation under target and conditional shift," in [*Proc. 30th International Conference on Machine Learning (ICML 2013)*], (2013).

[19] Bas, P., Filler, T., and Pevný, T., "'Break Our Steganographic System': The ins and outs of organizing BOSS," in [*Proc. 13th Information Hiding Workshop*], *LNCS* **6958**, 59–70, Springer (2011).

[20] Munson, Jr., D. C., "A note on Lena," *IEEE Transactions on Image Processing* **5**(1) (1996).