A New Paradigm for Steganalysis via Clustering

Andrew D. Ker[a] and Tomáš Pevný[b]

[a]Oxford University Computing Laboratory, Parks Road, Oxford OX1 3QD, England;
[b]Agent Technology Center, Department of Cybernetics, Czech Technical University in Prague,
Karlovo namesti 13, 121 35 Prague 2, Czech Republic.

ABSTRACT

We propose a new paradigm for blind, universal, steganalysis in the case when multiple actors transmit multiple objects, with guilty actors including some stego objects in their transmissions. The method is based on clustering rather than classification, and it is the actors which are clustered rather than their individual transmitted objects. This removes the need for training a classifier, and the danger of training model mismatch. It effectively judges the behaviour of actors by assuming that most of them are innocent: after performing agglomerative hierarchical clustering, the guilty actor(s) are clustered separately from the innocent majority. A case study shows that this works in the case of JPEG images. Although it is less sensitive than steganalysis based on specifically-trained classifiers, it requires no training, no knowledge of the embedding algorithm, and attacks the pooled steganalysis problem.

Keywords: Steganalysis, Clustering, Hierarchical Clustering, MMD, Batch Steganography, Pooled Steganalysis

1. INTRODUCTION

In steganalysis, the problem of detecting hidden data is usually restricted in two ways: only a single actor is considered, and they send only a single object. Such a restriction is implicit in the vast majority of the literature which optimizes detection of hidden payload in single objects. But such a scenario is unrealistic: in practice, any steganalyst will surely have to consider multiple actors (for example, if they are monitoring a network for leakage of confidential information, there will be many users transmitting objects which must be scrutinised for payload), and each actor will transmit multiple objects. To make matters even more complicated, an actor who is guilty of performing steganography will probably behave innocently some of the time, mixing their stego objects with genuine covers. This is known as *batch steganography*,[1] and detecting it is known as *pooled steganalysis*, a problem presented in 2006 which has still not been addressed successfully.

Another limitation is that most traditional steganalysis involves a classification algorithm, which has to be trained on large sets of innocent covers and stego objects.[2–4] (A few steganography schemes admit efficient detectors which do not require such training,[5–8] but these exceptions generally involve particularly flawed embedding operations.) In practice, the cover source of the actors will not be identical to that used for training by the detector, so as well as being computationally demanding the training itself becomes a possible source of error if an innocent actor is falsely accused because of training model mismatch.

We propose a new paradigm for steganalysis, which uses traditional steganalysis features but *clustering* rather than *classification* algorithms; furthermore, it is the actors which are clustered, based on their aggregate transmissions, rather than their individual transmitted objects. In this way we remove the need for training, and effectively judge the behaviour of actors by assuming that most of them are innocent. After performing agglomerative hierarchical clustering, the guilty actor(s) should be clustered separately from the innocent ones. We expect this to be less sensitive than steganalysis based on specifically-trained classifiers, but more robust when homogeneity of covers is violated, and hence more practically applicable. It also attacks the pooled steganalysis

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

problem and is *universal*, in the sense that an unknown or new embedding algorithm may also be detected, as long as the underlying steganalysis features are sensitive to it.

The general technique of using clustering to identify guilty behaviour is not new, being found for example in intrusion detection.[9] Clustering has also been used in watermarking, to obtain information on the embedding key when a flawed watermarking algorithm creates clusters.[10] To our knowledge, however, this application to steganalysis is entirely new.

The structure of the paper is as follows. We discuss the general techniques of agglomerative clustering, and MMD distance, in Sects. 2 and 3. In Sect. 4 we combine them into a proposed method for steganalysis in the multiple-actor, multiple-object setting, which clusters the actors. We test the accuracy of the proposed method, for three particular scenarios including up to 13 actors, in Sect. 5, and discuss directions for further research in Sect. 6.

2. AGGLOMERATIVE CLUSTERING

The aim of a clustering algorithm is to partition a set of objects, such that "similar" objects are in the same partition. The partitions are called clusters. Similarity is problem-specific and measured using some sort of distance, and various methods of clustering produce different numbers and types of partition. Hierarchical clustering refines this problem, producing a tree of nested clusters: each cluster contains either a singleton object or a set (usually pair) of clusters. It has a number of advantages over simple (flat) clustering, requiring no advance information about the number of clusters to search for, and being more informative in its structured output, but usually at the cost of higher time complexity.

Agglomerative clustering is one type of hierarchical clustering. Initially, all objects are placed into singleton clusters, the nearest two clusters are combined, and this is repeated until all clusters have been combined and a complete binary tree has been constructed. All that is needed is a method to compute a distance between two clusters, and there are a number of options.

Let us write $d(x, y)$ for the distance between two objects x and y, and $D(X, Y)$ for the distance between two clusters X and Y. The *single linkage* algorithm uses the distance between the nearest points in the two clusters:

$$D_{\mathrm{SL}}(X, Y) = \min_{\substack{x \in X \\ y \in Y}} d(x, y),$$

while *complete linkage* uses the furthest points:

$$D_{\mathrm{CL}}(X, Y) = \max_{\substack{x \in X \\ y \in Y}} d(x, y).$$

Single linkage can cause long chains of clusters, whereas complete linkage prefers compact clusters; other agglomerative clustering algorithms are intermediate, including *centroid* clustering

$$D_{\mathrm{CEN}}(X, Y) = \frac{1}{|X| \cdot |Y|} \sum_{x \in X,\, y \in Y}\!\!\sum d(x, y),$$

average linkage

$$D_{\mathrm{AVG}}(X, Y) = \frac{1}{(|X \cup Y|)(|X \cup Y| - 1)} \sum_{u,v \in X \cup Y,\; u \neq v}\!\!\sum d(u, v),$$

and similarly *median linkage*. Other measures of distance include *McQuitty's*[11] and *Ward's*,[12] the latter related to analysis of variance. The best choice of distance depends on the nature of the data being clustered. Almost all agglomerative clustering algorithms require only to know the distance between all pairs of objects being clustered, and the objects themselves do not necessarily have to be represented in a vector space (though it can be helpful to embed the objects into an inner product space, for reasons of efficiency, this need not concern us here).

Thus the input to an agglomerative clustering algorithm is a distance (or "dissimilarity", since it need not always satisfy the triangle inequality) matrix between all pairs. The output can be displayed in a *dendrogram*, a tree of the successive clusters agglomerations which uses "height" to indicate the distance between the clusters being merged.

3. THE MMD MEASURE

To apply agglomerative clustering to actors in multiple-actor, multiple-object steganalysis, we need a measure of distance between two actors. Supposing that each object has been reduced to a feature vector (of which more later), we can consider the feature vectors they transmit to arise from a probability distribution characterising the actor's object source and their use, or not, of steganography. Thus, we are given samples from two probability distributions, and require to estimate some sort of distance between the two distributions.

Maximum Mean Discrepancy (MMD)[13] is ideal for this purpose. Given two distributions $\mathcal{P}$ and $\mathcal{Q}$ with domain D, it is defined as

$$\mathrm{MMD}(\mathcal{P}, \mathcal{Q}) = \max_f \left|\mathbf{E}_{X\sim\mathcal{P}}[f(X)] - \mathbf{E}_{X\sim\mathcal{Q}}[f(X)]\right|,$$

where the maximum is over mappings $f : D \mapsto \mathbb{R}$ from a unit ball $\mathcal{F}$ in a Reproducing Kernel Hilbert Space (RKHS). MMD is nonnegative, symmetric, and satisfies the triangle inequality. It is positive definite, thus becoming a true metric, if the kernel is universal.

MMD has been used for measuring the security of steganography schemes,[14,15] but here it is simply a measure for comparing two actors' sets of feature vectors. It is ideal for our purposes because it can be estimated, even for very high dimensional vectors, from relatively few empirical samples: given n observations $\mathbf{x} = (x_1, \ldots, x_n)$ of $X \sim \mathcal{P}$ and n observations $\mathbf{y} = (y_1, \ldots, y_n)$ of $Y \sim \mathcal{Q}$, an unbiased estimate for the *squared* MMD is

$$\frac{1}{n(n-1)} \sum\sum_{i \neq j} k(x_i, x_j) + k(y_i, y_j) - k(x_i, y_j) - k(x_j, y_i), \tag{1}$$

where k is a bounded universal kernel $k : D \times D \mapsto \mathbb{R}$ that defines the dot product in the RKHS.[16] The estimate for $\mathrm{MMD}(\mathcal{P}, \mathcal{Q})$ is recovered by taking the square root of (1). (An equivalent formula holds when the sample sizes are unequal.)

There are choices for k, and we consider two possibilities: the *linear kernel* which is simply a scalar product

$$k(x, y) = x \cdot y,$$

and the *Gaussian kernel*

$$k(x, y) = \exp(-\gamma\|x - y\|^2),$$

where γ is a parameter. Typically, γ is set to η^{-2}, where η is the median of the L_2-distances between features in the set of images being considered: this means that the exponents are, on average, close to -1.

For the linear kernel, the squared MMD estimator can be simplified to

$$\frac{1}{n(n-1)} \sum\sum_{i,j}(x_i - y_i)\cdot(x_j - y_j) - \frac{1}{n(n-1)} \sum_i (x_i - y_i)\cdot(x_i - y_i) = \frac{n}{(n-1)}\|\bar{\mathbf{x}} - \bar{\mathbf{y}}\|^2 - \frac{1}{n(n-1)} \sum_i \|x_i - y_i\|^2,$$

where $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ are the centroids of the samples $\mathbf{x}$ and $\mathbf{y}$. Thus the MMD is almost equivalent, apart from a negligible term for large n, to the Euclidean distance between these centroids. For nonlinear kernels, however, the MMD measure takes account of differences in distributional shape as well as location, which may be valuable for our analysis if, hypothetically, the output features from a steganographer are not on average shifted from, but are more varied than, those from an innocent actor.

Rather than use raw feature vectors, it is sensible to compute MMD between *normalized* feature vectors, where each component of the vector has been scaled so that some global mean is (approximately) zero and global variance is (approximately) one. This can be performed by a linear transformation, and ensures that one component of the feature vector does not dominate the calculations. As with the kernel parameter γ, the precise linear transformation used is not important as long as the features turn out roughly equal in mean and variance, and as long as the scaling does not vary for each MMD calculation. In the next section we will discuss how scaling factors, and γ where needed, will be computed for our application.

4. PROPOSED METHOD

Suppose multiple actors each transmit multiple objects, all of which have been intercepted; we assume that each actor has just one source of objects, but that these sources are different. We do not anticipate that the same sources are available to us. Furthermore, suppose that we know which actor sent which object. For every object, we compute a feature vector which aims to distinguish innocent covers from stego objects. We can think of this as many clouds of points in the feature space, one cloud for each actor. A *guilty* actor is one who using steganography in (some of) their transmitted objects, and we aim to identify guilty actors if their clouds of feature vectors stand out, in some way, from the innocent actors'.

Performing steganalysis on each object individually might be valuable, but it is likely that any guilty party will be lost in a crowd of false positives. If we use a training model for cover images, it will not match any of the actors' sources exactly: the more objects each actor sends, the more certain we will become that their source does not match our training model, and the less information we gain.

It would be possible to perform clustering on the individual objects, but we expect that this will provide little useful information. It is quite likely that this will simply tell us what we already know: each actor's objects will end up in a cluster characterising their source. Instead, we cluster the *actors*, hoping to separate an innocent majority from a guilty minority.

Thus we combine the technique of hierarchical clustering, from Sect. 2, with the MMD distance measure from Sect. 3: distances between actors are defined as the MMD distance between the sets of features which represent all the objects they have transmitted. One can think of this as the dissimilarity between their clouds of points in feature space. We emphasise that the actors are *not* themselves necessarily represented by a point in the vector space of features (for example the centroid of their cloud): the MMD measure allows us to use clustering algorithms on the actors without expressing them in any space[*].

We must decide how to compute a sensible normalization scaling, and where necessary the kernel parameter γ. It is important that, for measuring MMDs between different pairs of actors, these parameters are fixed, otherwise the MMDs are not comparable. Our procedure is, for each set of actors to be clustered, to calculate all the sets of features and then calculate scaling so that the pooled mean of each feature is zero, and variance one, and for kernel MMD to take $\gamma = \eta^{-2}$ where η is the global median of distances between features.

If the features have been well-chosen, so that the difference between actors' sources is less than the difference between guilty and innocent actors, the final agglomeration will be between two clusters, the innocent and the guilty. Thus we can extract a list of suspected guilty actors from the cluster dendrogram. As well as depending on the features, the accuracy of this method will also depend on the proportion of objects which guilty actors embed in, and the amount they embed. The detector assumes that the majority of actors is innocent, and tries to identify guilty actors as an outlier cluster.

It would be possible for the detector to seed a "guilty" cluster by inserting their own stego objects, making it easier to spot the guilty actors, though we will not pursue this idea in this work. That aside, there is *no training* in this scenario: the majority of innocent actors supply a sort of training data in themselves. Furthermore, the variation between innocent actors supplies information about *how much* of an outlier a suspect actor is: information on the variation between different sources is simply not available in the traditional classification paradigm. In this sense we turn the difficulties of batch steganography to our advantage.

In some scenarios it may be known that a guilty actor exists, in others there may be uncertainty. So it is useful to consider ways to measure just how much of an outlier the "guilty" cluster is. This is mainly something we leave for further work, but we will perform some experiments which use the dendrogram "height" at the final agglomeration, which is the difference caused to the objective function during the minimization process of the agglomerative clustering.

[*] Nonetheless, with linear MMD it is practically equivalent to identifying each actor with the centroid of their feature cloud.

5. EXPERIMENTAL RESULTS

We apply this technique to two particular versions of the pooled steganalysis problem. Suppose that there are A actors, each of whom has transmitted N digital images, and we want to identify a single guilty actor from amongst them. In these experiments, the actors will be simulated by images taken from A different digital cameras, all the images will be JPEG compressed with the same quality factor, and the guilty actor will embed (a pseudorandom message) using the nsF5 embedding algorithm[17] in some – but not necessarily all – of the images they transmit. Each of the actors' images will be reduced to a 274-dimensional feature vector called PF-274,[4] designed to detect steganography in the JPEG domain and previously shown to be effective against nsF5. We will use the hierarchical clustering technique, with MMD distance between actors; our expectation is that the final agglomeration should be between one guilty actor and a cluster of $A - 1$ actors users. We stress that there is no training in this paradigm, and we do not assume any knowledge of the actors' cameras or the embedding algorithm.

5.1 Experiments with Seven Actors, One Guilty

We began with experiments on seven actors sending JPEG images:

- **Actor A** uses an Olympus c765, which has native resolution 2288 × 1712;
- **Actor B** uses a Nikon D100, which has native resolution 3008 × 2000;
- **Actor C** uses a Sigma SD9, which has native resolution 2268 × 1512;
- **Actor D** uses a Minolta DiMage A1, images slightly cropped to 2000 × 1500;
- **Actor E** uses a Canon Powershot G2, which has native resolution 2272 × 1704;
- **Actor F** uses a Canon Powershot S40, which also has native resolution 2272 × 1704;
- **Actor G** uses a Kodak DC290, which has native resolution 1792 × 1200.

Each actor takes N RAW photos and converts them to JPEG quality factor 80[†] before transmitting them. For simplicity, each actor uses images of a constant size (the default resolution of the camera), but the image size does vary from actor to actor (because the different cameras have different resolutions). Our database consists of 300 images from each actor.

To begin with, we randomly sampled 50 images from each actor, and performed no embedding. We then computed linear MMDs between the image features, for each pair of actors. One way to represent the "shape" of the innocent actors is the multidimensional scaling (MDS) technique,[18] which attempts to locate each actor as a point in a low dimensional space, such that the Euclidean distances between the points are approximately equal to the computed MMD distances. This is an imperfect representation useful, when reducing to two dimensions, for visualising the data. A MDS representation of seven innocent actors appears in the left of Fig. 1, with the results of hierarchical clustering (using single linkage agglomeration) beneath. There is no particularly strong outlier.

Then we embedded a payload of size 0.7 bits per nonzero DCT coefficient (bpnc)[‡] into a randomly chosen 35 (70%) of actor A's images, using the nsF5 embedding algorithm[17] and leaving the other 15 images untouched, and repeated the MMD calculation and clustering. The MDS plot is shown in the top middle of Fig. 1, and below is the cluster dendrogram, which clearly identifies two clusters consisting of actor A alone, and the innocent users. The "height" of the agglomeration of A with the others indicates that to merge them would require

[†] We fixed the quality factor in the knowledge that the PF-274 feature set is very sensitive to changes in it; we would not expect the results to be good if the actors choose different quality factors, but this is a drawback of the particular feature set rather than the clustering methodology.

[‡] This is something like a 70% payload, since each nonzero coefficient carries an absolute maximum of 1 bit. But payloads larger than 0.7 bpnc cannot be embedded in all images, due to the wet paper code needed for the "no shrinkage" part of nsF5. Hence all our experiments are capped at 0.7 bpnc.

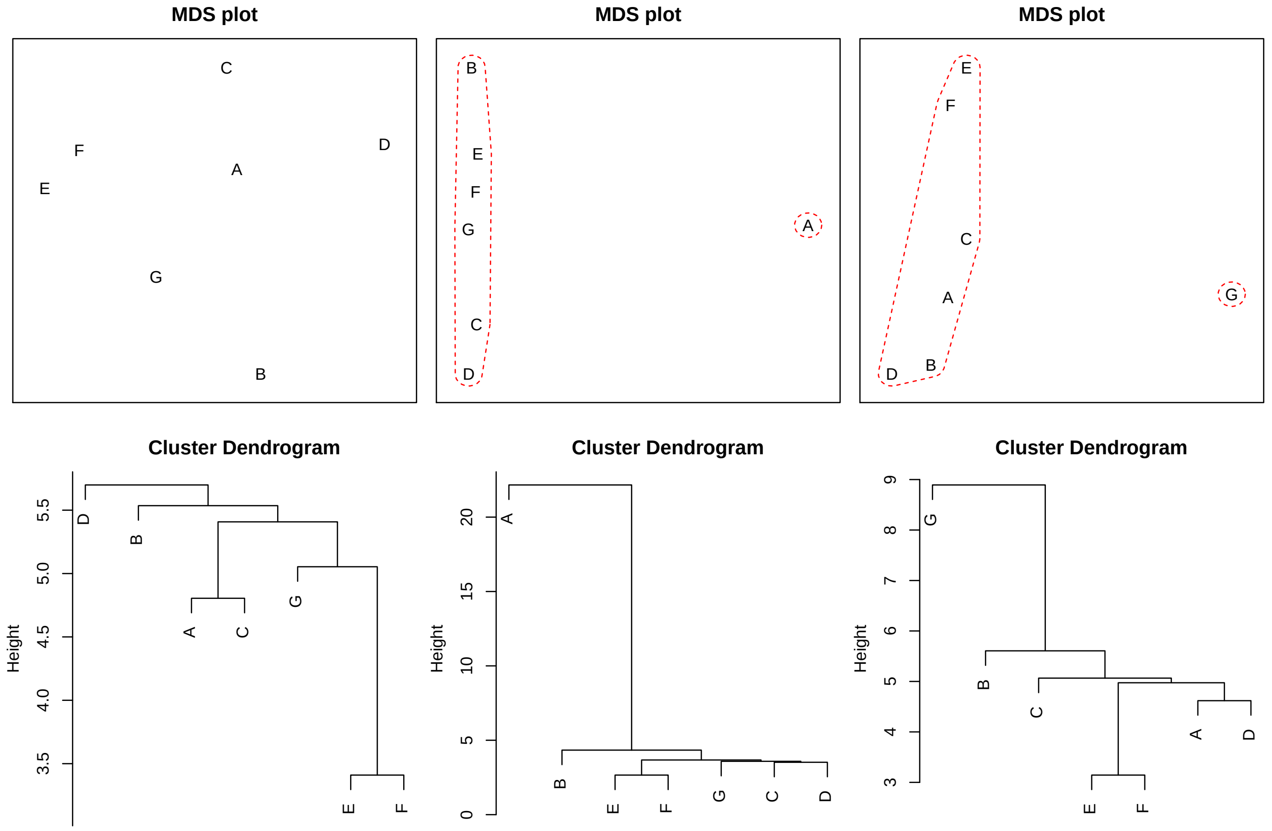

Figure 1. Above, MDS representation of actors; below, cluster dendrograms, from which the final two clusters are encircled in the MDS plot. Left, 7 innocent actors. Middle, actor A embeds payload of size 0.7 bits per nonzero coefficient (bpnc) in 70% of their images; Right, actor G embeds at 0.3 bpnc in 30% of their images.

spanning a long distance (the MDS representation substantially under-represents the distance between A and the rest, compared with the distances between the other actors). A has been identified as the guilty actor, despite the detector having information on neither the cover images nor the embedding algorithm. We repeated the experiments with actor B, . . . , G as the guilty party, and each was clearly identified.

Table 1. Confusion matrics for the identification of one guilty actor, with each experiment repeated 100 times per guilty actor. Every actor transmits 50 images: on the left, the guilty actor embeds 0.25 bpnc in 25% of their images (overall accuracy of identification 90.3%); on the right, 0.3 bpnc in 30% (overall accuracy 99.9%). Clustering using single linkage and linear MMD.

Guilty	Identified actor						
actor	**A**	**B**	**C**	**D**	**E**	**F**	**G**
A	100	0	0	0	0	0	0
B	0	100	0	0	0	0	0
C	0	0	100	0	0	0	0
D	0	1	5	94	0	0	0
E	0	9	7	0	84	0	0
F	0	9	14	0	0	77	0
G	0	16	7	0	0	0	77

Guilty	Identified actor						
actor	**A**	**B**	**C**	**D**	**E**	**F**	**G**
A	100	0	0	0	0	0	0
B	0	100	0	0	0	0	0
C	0	0	100	0	0	0	0
D	0	0	0	100	0	0	0
E	0	0	0	0	100	0	0
F	0	0	1	0	0	99	0
G	0	0	0	0	0	0	100

Table 2. Overall accuracy, after 100 repetitions with each of 7 guilty actors, of the proposed method. Each actor transmits N images, and a single guilty actor embeds a certain size random payload (different for each experiment) in a certain proportion of their images. Some different clustering algorithms are compared.

MMD kernel	Cluster linkage	N	Guilty actor's proportion of stego images @ payload per image							
			10% @ 0.1bpnc	20% @ 0.2bpnc	25% @ 0.25bpnc	30% @ 0.3bpnc	40% @ 0.4bpnc	50% @ 0.5bpnc	60% @ 0.6bpnc	70% @ 0.7bpnc
Linear	Single	200	21.3%	58.9%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Linear	Single	100	18.0%	54.9%	98.9%	100.0%	100.0%	100.0%	100.0%	100.0%
Linear	Single	50	17.7%	51.7%	90.3%	99.9%	100.0%	100.0%	100.0%	100.0%
Linear	Single	20	14.1%	44.1%	79.6%	96.7%	100.0%	100.0%	100.0%	100.0%
Linear	Single	10	15.9%	41.6%	54.3%	86.4%	98.6%	99.9%	100.0%	100.0%
Linear	Centroid	200	17.0%	76.7%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Linear	Centroid	100	16.6%	63.1%	98.7%	100.0%	100.0%	100.0%	100.0%	100.0%
Linear	Centroid	50	16.3%	56.7%	93.6%	100.0%	100.0%	100.0%	100.0%	100.0%
Linear	Centroid	20	13.4%	46.6%	83.7%	97.0%	100.0%	100.0%	100.0%	100.0%
Linear	Centroid	10	15.4%	40.4%	56.4%	87.1%	98.9%	99.9%	100.0%	100.0%
Linear	Average	200	17.0%	74.7%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Linear	Average	100	16.4%	61.1%	98.7%	100.0%	100.0%	100.0%	100.0%	100.0%
Linear	Average	50	15.7%	55.1%	93.4%	100.0%	100.0%	100.0%	100.0%	100.0%
Linear	Average	20	14.3%	45.0%	83.0%	97.0%	100.0%	100.0%	100.0%	100.0%
Linear	Average	10	16.6%	40.0%	54.3%	87.1%	98.9%	99.9%	100.0%	100.0%
Linear	Complete	200	15.4%	45.3%	91.4%	100.0%	100.0%	100.0%	100.0%	100.0%
Linear	Complete	100	14.4%	46.7%	91.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Linear	Complete	50	13.9%	46.1%	83.1%	99.3%	100.0%	100.0%	100.0%	100.0%
Linear	Complete	20	15.6%	40.3%	75.7%	94.9%	100.0%	100.0%	100.0%	100.0%
Linear	Complete	10	16.6%	35.4%	48.3%	84.0%	98.6%	99.9%	100.0%	100.0%
Gaussian	Centroid	200	14.0%	23.0%	41.0%	92.0%	100.0%	100.0%	100.0%	100.0%
Gaussian	Centroid	100	13.7%	23.7%	44.0%	87.4%	100.0%	100.0%	100.0%	100.0%
Gaussian	Centroid	50	12.6%	23.3%	40.3%	76.1%	100.0%	100.0%	100.0%	100.0%
Gaussian	Centroid	20	12.9%	18.4%	37.9%	60.1%	94.9%	99.7%	100.0%	100.0%
Gaussian	Centroid	10	15.3%	18.9%	20.7%	40.9%	79.9%	98.4%	99.9%	100.0%

To simulate more subtle embedding, we reduced the payload, with the guilty actor using only 60% of images to contain 0.6 bpnc, 50% to contain 0.5 bpnc, and so on down to 10% of images containing 0.1 bpnc. (We included 0.25 bpnc in 25% of images too, in view of the sudden change, which the reader will shortly observe, between the 20% and 30% cases.) The case of 30% is displayed on the right of Fig. 1, in this case with actor G as the guilty party, and still they are identified, although the dendrogram indicates that the distance from G to the rest of the users is not very large compared with the distance between innocent users.

We repeated these experiments 100 times for each guilty party, with different images selected from the database of sources, and different random payloads. The identification of the guilty party is performed by cutting the dendrogram at the final agglomeration: when this consists of a singleton actor merged into a cluster of $A-1$, we accuse that actor; when it consists of the merger of a small cluster $\mathcal{C}$ with a large complement, we accuse a member of $\mathcal{C}$ at random. At payloads of 0.4 bpnc (in 40% of images) or greater, the result is perfect detection. We display confusion matrices for the payloads 0.3 bpnc (in 30% of images) and 0.25 bpnc (in 25% of images) in Tab. 1, where we observe overall accuracy of 99.9% and 90.3% respectively. Performance falls off sharply for smaller payloads as the guilty actors fade into the innocent cluster and they cannot be identified accurately.

We performed these experiments using both linear and Gaussian MMD, and all seven of the linkage measures from Sect. 2, and with different numbers of images transmitted by each actor: $N = 10, 20, 50, 100, 200$. This was in order to compare the efficacy of the different combinations. Some of the results are displayed in Tab. 2. We found that, consistently, the clustering based on linear MMD worked better than that based on Gaussian MMD. We also found that the choice of agglomeration algorithm did not make a substantial difference, but that the worst performance arose from tight linkage such as complete linkage, quite good performance from single linkage, and the best performance from centroid linkage. Some of these can be seen in Tab. 2. As expected, there is a dependency on N: more evidence allows the detector to make more accurate detection from smaller payloads. When $N = 200$, linear MMD and centroid linkage allowed perfectly accurate detection of the guilty party with payloads as small as 0.25 bpnc embedded in only 25% of the images: this is an overall relative payload of 0.0625. Even when N is only 20, perfect detection occurs at a relative payload of 0.16.

For comparison, the detector of Ref. 4, trained for a single specific cover source, typically identifies payloads down to about 0.05 bpnc (for typical camera JPEGs). However, our clustering algorithm *cannot* be compared directly with anything (that we know of) in the literature, because it attacks such a different problem. For one thing, the detectors of Ref. 4 – and practically all other published steganalysis – work on a single image at a time and do not take into account larger numbers of images. To our knowledge, there is no successful work attacking this *pooled steganalysis* problem, as described in Ref. 1.

On the other hand, most steganalyzers require some form of training on both cover and stego images so that the stego algorithm (or a short list of potential algorithms) should be known, and often their performance is only good if the training set is from a nearly identical source to the images under analysis. In contrast, the clustering method is completely untrained and completely ignorant of the embedding algorithm. We cannot even make a fair comparison between our proposed method and the universal steganalyzer described in Ref. 19, because that too requires training on cover images. Notwithstanding the foregoing, the results of Tab. 2 indicate that the rough sizes of payloads detectable by our clustering method are on a par with those detectable by conventional means, at least for the set of seven cameras we tested.

5.2 Experiments with Seven Actors, Zero or One Guilty

We now make the detection problem harder, by removing the assumption that exactly one actor is performing steganography. Instead, we suppose that either all actors are innocent, or exactly one is guilty (we leave the case of multiple felons to future work). In that case, it is not adequate to accuse the "most outlying" actor, as in the previous section. Instead, we must make some measurement of how outlying they are, and on that basis decide whether or not to accuse.

At this stage, we pursue only a simple idea. Consider the "height" of each merger in the agglomerative clustering algorithm, which is the distance between the two most-recently merged clusters. Let us make an accu-

Table 3. Confusion matrics for the identification of up to one guilty actor, with each experiment repeated 100 times per guilty actor. Every actor transmits 50 images: on the left, the guilty actor embeds 0.3 bpnc in 30% of their images (overall accuracy of identification 25.3%); on the right, 0.4 bpnc in 40% (overall accuracy 90.4%). Clustering using centroid linkage and linear MMD.

Guilty	Identified actor							
actor	**none**	**A**	**B**	**C**	**D**	**E**	**F**	**G**
none	100	0	0	0	0	0	0	0
A	96	4	0	0	0	0	0	0
B	51	0	49	0	0	0	0	0
C	68	0	0	32	0	0	0	0
D	99	0	0	0	1	0	0	0
E	93	0	0	0	0	7	0	0
F	91	0	0	0	0	0	9	0
G	100	0	0	0	0	0	0	0

Guilty	Identified actor							
actor	**none**	**A**	**B**	**C**	**D**	**E**	**F**	**G**
none	100	0	0	0	0	0	0	0
A	11	89	0	0	0	0	0	0
B	2	0	98	0	0	0	0	0
C	0	0	0	100	0	0	0	0
D	22	0	0	0	78	0	0	0
E	11	0	0	0	0	89	0	0
F	7	0	0	0	0	0	93	0
G	24	0	0	0	0	0	0	76

Table 4. False positive and negative, incorrect identification rate, and overall accuracy, when identifying zero or one guilty actors. No accusation is made if the final agglomeration height is not twice the previous. Clustering using centroid linkage and linear MMD.

N		Guilty actor's proportion of stego images @ payload per image							
		10% @ 0.1bpnc	20% @ 0.2bpnc	25% @ 0.25bpnc	30% @ 0.3bpnc	40% @ 0.4bpnc	50% @ 0.5bpnc	60% @ 0.6bpnc	70% @ 0.7bpnc
100	False positive	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	False negative	100.0%	99.9%	98.7%	80.0%	0.7%	0.0%	0.0%	0.0%
	Incorrect positive	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Overall accuracy	12.5%	12.6%	13.6%	30.0%	99.4%	100.0%	100.0%	100.0%
50	False positive	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	False negative	100.0%	99.9%	98.1%	85.4%	11.0%	0.0%	0.0%	0.0%
	Incorrect positive	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Overall accuracy	12.5%	12.6%	14.1%	25.2%	90.4%	100.0%	100.0%	100.0%
20	False positive	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	False negative	99.3%	99.7%	97.1%	91.1%	36.4%	4.0%	0.4%	0.0%
	Incorrect positive	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Overall accuracy	12.9%	12.8%	15.0%	20.2%	68.1%	96.4%	99.2%	100.0%

sation only if the height of the final agglomeration is at least twice the height of the penultimate agglomeration. Depending on the linkage used, this may be similar to asking whether the distance of the majority to the final actor's features is at least as much as the distance between any two other actors' features (though it is usually a stricter condition than that). If the agglomeration has such a property, we accuse the final actor in the same way as in Subsect. 5.1, otherwise we state that no actor is guilty. Having identified linear MMD and centroid linkage as the best choice, we will report results only for this combination.

Two of the resulting confusion matrices, using the same image sources as in the previous experiments, $N = 50$ images per actor, and centroid clustering with linear MMD, are displayed in Tab. 3. We see that there are no false accusations but many false negatives when the payload is 0.3 bpnc in 30% of images, and still a few false negatives when the payload is 0.4 bpnc in 40% of images. We repeated the experiments for $N = 100, 50, 20$. We measured false positives and negatives separately from incorrect accusations (when one user is guilty, but the wrong one accused), and overall accuracy, and these results are in Tab. 4. It is clear that this decision procedure is far too conservative, never making a false accusation but often failing to accuse a guilty actor. Nonetheless, high accuracy is achieved for overall relative payloads of 0.16. In future work, we will investigate more finely-tuned procedures for deciding whether to accuse an actor.

5.3 Experiments with Thirteen Actors, One Guilty

The accuracy of the clustering paradigm will depend on the homogeneity of the actors' sources. In our final experiments, we broadened the sources to include six more digital cameras:

- **Actor H** uses a Canon EOS 400D, which has native resolution 3906 × 2602;
- **Actor I** uses a Pentax K20D, which has native resolution 4688 × 3124;
- **Actor J** uses a Canon EOS 7D, which has native resolution 5202 × 3465;
- **Actor K** uses a Canon Digital Rebel XSi, which has native resolution 4290 × 2856;
- **Actor L** uses a Leica M9, which has native resolution 5216 × 3472;
- **Actor M** uses a Nikon D70, which has native resolution 3039 × 2014.

Table 5. Confusion matrics for the identification of one guilty actor out of 13. Every actor transmits 50 images. Left, images sources remain different sizes, 0.7 bpnc embedded in 70% of guilty actor's images (overall accuracy 43%). Right, image sources have been cropped to the same size, 0.3 bpnc in 30% of guilty actor's images (overall accuracy 85.1%). Clustering using centroid linkage and linear MMD.

Guilty	Identified actor												
actor	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
A	64	0	0	0	0	0	0	0	0	0	36	0	0
B	0	43	0	0	0	0	0	0	0	0	57	0	0
C	0	0	64	0	0	0	0	0	0	0	36	0	0
D	0	0	0	71	0	0	0	0	0	0	29	0	0
E	0	0	0	0	18	0	0	0	0	0	82	0	0
F	0	0	0	0	0	18	0	0	0	0	82	0	0
G	0	0	0	0	0	0	21	0	0	0	79	0	0
H	0	0	0	0	0	0	0	55	0	0	45	0	0
I	0	0	0	0	0	0	0	0	2	0	98	0	0
J	0	0	0	0	0	0	0	0	0	100	0	0	0
K	0	0	0	0	0	0	0	0	0	0	100	0	0
L	0	0	0	0	0	0	0	0	0	0	100	0	0
M	0	0	0	0	0	0	0	0	0	0	97	0	3

Guilty	Identified actor												
actor	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
A	97	0	0	0	0	0	0	0	0	0	0	0	3
B	0	97	0	0	0	0	0	0	0	0	0	0	3
C	0	0	78	0	0	0	0	0	0	0	0	0	22
D	0	0	0	93	0	0	0	0	0	0	0	0	7
E	0	0	0	0	94	0	0	0	0	0	0	0	6
F	0	0	0	0	0	90	0	0	0	0	0	0	10
G	0	0	0	0	0	0	79	0	0	0	0	0	21
H	0	0	0	9	0	0	0	51	1	0	0	0	39
I	0	0	0	0	0	0	0	0	93	0	0	0	7
J	0	0	0	0	0	0	0	0	0	92	0	0	8
K	0	0	0	1	0	0	0	0	0	0	62	0	37
L	0	0	0	0	0	0	0	0	0	0	0	80	20
M	0	0	0	0	0	0	0	0	0	0	0	0	100

These cameras are generally newer, and take photos in higher resolution, than the first seven. Experiments were conducted, similarly to those in Subsect. 5.1, measuring accuracy of identification when exactly one actor out of thirteen is guilty. Initial results were very disappointing: actor K, or sometimes actor M, were consistently falsely accused unless the embedded payload was extremely high. See for example the confusion matrix in Tab. 5, left, which shows an overall accuracy of only 43% despite a payload of 0.7 bpnc in 70% of the guilty actor's images. This occurs because the steganalysis features from those two cameras are outliers, compared with the others, to an extent that exceeds the effects of steganographic embedding.

Greater sensitivity to cover source, rather than payload, is a defect of the PF-274 features. However, we were able to reduce this substantially by ensuring that all images were the same size. We repeated the experiments after cropping all images to a central region 1792 × 1200 (the size of the smallest). For simplicity of implementation the cropping was done *before* embedding, but since the payload is proportional to the number of nonzero DCT coefficients, which is turn is usually proportional to the image size, the effect would have been the same were the cropping done by a detector receiving full-size images. The same experiments were repeated with much better results: not as good as those in Subsect. 5.1, but this is to be expected since identification of one guilty actor out of thirteen is a harder problem (has more potential errors) than identification of one guilty actor out of seven. Tab. 5, right, shows the confusion matrix for an overall relative payload of 0.09 with equal-sized images, and at higher rates the identification is near-perfect.

We compared cropped and uncropped images, using centroid clustering and linear MMD, for $N = 100$, 50, 20, and the results are displayed in Tab. 6. The difference is striking. This demonstrates that the PF-274 features are improperly scaled, which is not so surprising considering that their design did not take image size into account. It suggests further research for improving the feature set, which is independent of further research on clustering with larger numbers of sources.

Curiously, when we cropped the images and repeated the experiments from Subsects. 5.1 and 5.2 we did not see much difference in the results. Perhaps this is because the seven cameras' resolutions do not, apart from actor B, vary much.

6. FURTHER WORK

Steganalysis has heretofore been rooted in the paradigm of supervised learning – classification – which always requires training data in the form of large and representative sets of cover and stego images; even the universal anomaly detector of Ref. 19 requires a representative training set of cover images. The proposed technique, which we believe to be entirely different from previous approaches, is from unsupervised learning – clustering –

Table 6. Overall accuracy, after 100 repetitions with each of 13 guilty actors, of the proposed method. Original images, and images cropped to identical sizes, are compared.

MMD kernel	Cluster linkage	N	Guilty actor's proportion of stego images @ payload per image							
			10% @ 0.1bpnc	20% @ 0.2bpnc	25% @ 0.25bpnc	30% @ 0.3bpnc	40% @ 0.4bpnc	50% @ 0.5bpnc	60% @ 0.6bpnc	70% @ 0.7bpnc
			uncropped images							
Linear	Centroid	100	7.7%	7.7%	7.7%	7.7%	7.7%	7.7%	12.8%	45.0%
Linear	Centroid	50	7.7%	7.7%	7.8%	7.7%	7.8%	7.8%	12.5%	43.0%
Linear	Centroid	20	7.7%	7.8%	8.0%	7.8%	8.0%	9.4%	15.1%	42.8%
			cropped images							
Linear	Centroid	100	8.5%	26.9%	66.7%	90.8%	100.0%	100.0%	100.0%	100.0%
Linear	Centroid	50	8.5%	25.9%	54.8%	85.1%	99.9%	100.0%	100.0%	100.0%
Linear	Centroid	20	8.5%	22.5%	46.9%	74.2%	95.1%	99.8%	100.0%	100.0%

and so by nature it requires no training. As a corollary, it is universal in the sense that it may detect new or unknown embedding algorithms.

We certainly could not expect an untrained detector, on inhomogeneous data, to be as sensitive as a detector specifically trained and tested on a single cover source. Nonetheless, we seem to be able to detect relative payloads in the region of 4–16% with very high accuracy. Everything hinges, of course, on the nature of the actor's sources and the chosen feature set: it is necessary that difference in source causes less variation than embedding payload. Furthermore, the universality of the detector also depends on the sensitivity of the features to new or unknown embedding algorithms.

Consequently, there are two ways to carry this work forward. First, we intend to investigate the state-of-art in clustering techniques: there are alternatives to agglomerative clustering including hierarchical divisive clustering (an inverse of the agglomerative idea, iteratively breaking the set of actors into smaller clusters) and flat clustering such as k-means. A key aim is to improve the admittedly crude decision procedure from Subsect. 5.2, when it is not known whether any guilty actor exists. The problem domain can be enlarged to the detection of multiple guilty actors, and it would be beneficial to run large-scale experiments. We must also examine the distance metric: it is curious that linear MMD, which essentially amounts to distance between feature set centroids, performed more effectively than Gaussian MMD, yet the reverse is usually true in binary classification.[20] Perhaps alternative kernels, or alternative normalization, would improve the Gaussian MMD accuracy.

Second, we should consider the features themselves. It is already demonstrated[4] that the PF-274 feature set is highly sensitive to certain attributes of the cover source, particularly JPEG quantization levels. We deliberately standardized the quality factor of all the images we tested, in the certain knowledge that the method will fail if quality factor varies much because the features will be more affected by difference in source than by embedding. We also noted, in Subsect. 5.3, that the features appear sensitive to large variation in image size, though this can be mitigated by cropping. In order to use steganalysis features in the domain of multiple sources, it becomes vital to reduce these variations. Creating feature sets with a more uniform response over different cover sources has not been important in the past, because steganalysis was typically benchmarked on single cover sets; clustering can perhaps encourage and inform the design of new feature sets with more uniformity.

Indeed most steganalysis literature has, usually implicitly, considered the problem of a single actor sending a single object. We cannot imagine that a steganalyst could be presented with such a simple problem in practice, and an advantage of this approach is that it generalizes the steganalysis problem to a more realistic scenario. The problem of multiple objects was presented in Ref. 1 but has barely been attacked, and the multiple sources problem has barely been investigated by some publications testing a small number of different image sets individually or perhaps cross-training. Clustering provides a way forwards. Although multiple signals present many opportunities for error (accusing an innocent party, being confused by innocent objects mixed with stego objects) we have turned them to our advantage by using the innocent majority to calibrate our expectations, and thus identify the guilty minority.

In future work we may be able to quantify the variation in a set of sources, and quantify the embedding signal in terms of payload, and thus begin to consider what steganographic capacity could mean in this context. Note that we do *not* expect the Square Root Law of Capacity[15,21] necessarily to apply in this situation: the detector does not have information about the actors' cover sources, so it may be the *rate* of steganographic embedding which affects how the guilty actors' feature sets move apart from the innocent actors', not the *root rate*.

Finally, we could switch roles and consider the embedder's embedding strategy, a problem from batch steganography:[1] is it safer for a guilty actor to embed large payloads in a few images, or small payloads in many images? Such a question can easily be tested against a clustering detector.

ACKNOWLEDGMENTS

The first author is a Royal Society University Research Fellow.

This material is based upon work supported by the ITC-A of the US Army under Contract No. W911NF-10-1-0070. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the ITC-A of the US Army. The effort is also sponsored by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-10-1-3016. The U.S Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright notation thereon. The work of the second author is also supported by Czech Ministry of Education grants 6840770038 and ME10051: AMVIS-AnomalyNET.

REFERENCES

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

[2] Farid, H. and Lyu, S., "Detecting hidden messages using higher-order statistics and support vector machines," in [*Proc. 5th Information Hiding Workshop*], *Springer LNCS* **2578**, 340–354 (2002).

[3] Harmsen, J. and Pearlman, W., "Higher-order statistical steganalysis of palette images," in [*Security and Watermarking of Multimedia Contents V*], *Proc. SPIE* **5020**, 131–142 (2003).

[4] Pevný, T. and Fridrich, J., "Multiclass detector of current steganographic methods for JPEG format," *IEEE Transactions on Information Forensics and Security* **3**(4), 635–650 (2008).

[5] Fridrich, J., Goljan, M., and Du, R., "Reliable detection of LSB steganography in color and grayscale images," in [*Proc. 3rd ACM Workshop on Multimedia and Security*], 27–30 (2001).

[6] Ker, A., "A fusion of maximum likelihood and structural steganalysis," in [*Proc. 9th Information Hiding Workshop*], *Springer LNCS* **4567**, 204–219 (2007).

[7] Lee, K., Westfeld, A., and Lee, S., "Category attack for LSB steganalysis of JPEG images," in [*Proc. 5th International Workshop on Digital Watermarking*], *Springer LNCS* **4238**, 35–48 (2006).

[8] Kodovský, J. and Fridrich, J., "Quantitative steganalysis of LSB embedding in JPEG domain," in [*Proc. 12th ACM Workshop on Multimedia and Security*], 187–198 (2010).

[9] Gu, G., Perdisci, R., Zhang, J., and Lee, W., "BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in [*Proc. 17th USENIX Security Symposium*], (2008).

[10] Bas, P. and Doërr, G., "Evaluation of an optimal watermark tampering attack against dirty paper trellis schemes," in [*Proc. 10th ACM workshop on Multimedia and Security*], *MM&Sec '08*, 227–232, ACM (2008).

[11] McQuitty, L., "Similarity analysis by reciprocal pairs for discrete and continuous data," *Educational and Psychological Measurement* **26**, 825–831 (1966).

[12] Ward, J., "Hierarchical grouping to optimize an objective function," *Journal of American Statistical Association* **58**(301), 236–244 (1963).

[13] Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A., "A kernel method for the two-sample-problem," in [*Advances in Neural Information Processing Systems 19*], Schölkopf, B., Platt, J., and Hoffman, T., eds., 513–520, MIT Press (2007).

[14] Pevný, T. and Fridrich, J., "Benchmarking for steganography," in [*Proc. 10th Information Hiding Workshop*], *Springer LNCS* **4437**, 251–267 (2008).

[15] Ker, A., Pevný, T., Kodovský, J., and Fridrich, J., "The square root law of steganographic capacity," in [*Proc. 10th ACM Workshop on Multimedia and Security*], 107–116 (2008).

[16] Steinwart, I., "On the influence of the kernel on the consistency of support vector machines," *Journal of Machine Learning Research* **2**, 67–93 (2001).

[17] Fridrich, J., Pevný, T., and Kodovský, J., "Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities," in [*Proc. 9th ACM Workshop on Multimedia and Security*], 3–14 (2007).

[18] Cox, T. and Cox, M., [*Multidimensional Scaling*], Chapman and Hall, 2nd ed. (2001).

[19] Pevný, T. and Fridrich, J., "Novelty detection in blind steganalysis," in [*Proc. 10th ACM workshop on Multimedia and Security*], *MM&Sec '08*, 167–176, ACM (2008).

[20] 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).

[21] Ker, A., "A capacity result for batch steganography," *IEEE Signal Processing Letters* **14**(8), 525–528 (2007).