LETTER

JPEG Steganalysis Based on Multi-Projection Ensemble Discriminant Clustering*

Yan SUN[†], Nonmember, Guorui FENG^{†a)}, Member, and Yanli REN[†], Nonmember

SUMMARY In this paper, we propose a novel algorithm called multi-projection ensemble discriminant clustering (MPEDC) for JPEG steganalysis. The scheme makes use of the optimal projection of linear discriminant analysis (LDA) algorithm to get more projection vectors by using the micro-rotation method. These vectors are similar to the optimal vector. MPEDC combines unsupervised K-means algorithm to make a comprehensive decision classification adaptively. The power of the proposed method is demonstrated on three steganographic methods with three feature extraction methods. Experimental results show that the accuracy can be improved using iterative discriminant classification.

key words: JPEG steganalysis, multi-projection, linear discriminant analysis, K-means

1. Introduction

The combat between steganography and steganalysis has become an important subject of information security. The most popular steganalysis mainly includes feature extraction and classifier learning. For JPEG images, early features are directly used in the DCT domain to train the classifier. The CC-JRM (JPEG rich model with Cartesian-calibration) [1] feature uses the idea of feature fusion to fuse the 40 sub-models of inter-block and intra-block statistical properties of DCT model and the subset of 11 sub-models of DCT integral co-occurrence matrices. Later, the researchers proposed discrete cosine transform residual (DCTR) [2] and Gabor filter residual (GFR) [3], which are better feature extraction methods. In this paper, we will use these typical features to train the effective classifier.

As for feature classification, there exists a large variety of various machine learning tools employed in steganalysis. The linear discriminant analysis (LDA) ensemble classifier [4] can maintain a fast running speed under high dimensional condition and a good accuracy. It contains multiple LDA sub-classifiers, and each sub-classifier randomly extracts a part of the feature to construct the feature subspace. In our model, we integrate LDA and K-means clustering [5], [6] to more accurately address the complex issues arising in steganalysis, and utilize ensemble learning to

create the uniform detection model. Self-learning ensemble discriminant clustering [7] successfully utilized the ensemble learning theory to create an ensemble classifier consisting of LDA and k-means classifiers trained on a set of stego and cover images to solve the issues of steganalysis in high-dimensional feature space.

2. Background

Both LDA and K-means are popular feature classification methods in machine learning. The integrated classifier which combines LDA and K-means can solve stego-free image steganalysis problem.

2.1 LDA and K-Means

The linear discriminant analysis (LDA) is one of the most widely used discrimination criterion in the feature classification, which defines a projection vector that makes the within-class scatter S_w smaller and the between-class scatter S_b larger. The LDA method can well reduce the dimensionality of image features, and it has a strong power of discrimination which is widely used to select the feature subspace. K-means algorithm, as a hard clustering algorithm, is a typical representative of the prototype-based objective function clustering method using the iterative adjustment rules.

2.2 Self-Learning Ensemble Discriminant Clustering

Self-learning ensemble discriminant clustering is denoted as SEDC in [7], where the average of each sample point is projected onto the vector obtained by LDA and used as the initial cluster center of the K-means algorithm. The best projection direction $\mathbf{w}$ is given which is defined by $\max J(\mathbf{w})$ as follows:

$$\max_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}}, \quad (1)$$

To obtain $\max J(\mathbf{w})$, we minimize S_w , and maximize S_b . $\mathbf{w}$ can be calculated by

$$\mathbf{w} = \mathbf{S}_w^{-1}(\mathbf{u}_1 - \mathbf{u}_2) \quad (2)$$

where $\mathbf{u}_1$ and $\mathbf{u}_2$ are the means of the cover and stego features.

Manuscript received April 10, 2018.

Manuscript revised August 14, 2018.

Manuscript publicized October 15, 2018.

[†]The authors are with Shanghai Institute for Advanced Communication and Data Science and School of Communication and Information Engineering, Shanghai University, China.

*This work is supported by the National Science Foundation of China under Grant No. (U1536109, 61373151, U1736120, 61572309).

a) E-mail: fgr2082@aliyun.com

DOI: 10.1587/transinf.2018EDL8073

Fig. 1 Block diagram of the proposed MPEDC. The random subspace of each example is constructed by sampling $m_{sub} \ll m$ feature randomly from the entire feature space.

3. Multi-Projection Ensemble Discriminant Clustering

The MPEDC (multi-projection ensemble discriminant clustering) also integrate the LDA and K-means algorithms. For the diversity of discriminant, we try to model the random distribution in the classifier and look for multi-projection direction. The extracted features are used to train a number of sub-classifiers. Figure 1 gives the overall block diagram of the proposed MPEDC.

3.1 Problem Formulation

We note a classification problem with n_{tr} training samples and feature with m -dimension. For $\mathbf{m}_v \subset \{1, \dots, m\}$, $|\mathbf{m}_v| = m_{sub}$ is the number of random subspace. In the v th sampled subset, LDA is trained on the set $\{\mathbf{x}_1^{\mathbf{m}_v}, \dots, \mathbf{x}_i^{\mathbf{m}_v}\}_{i=1}^{n_{tr}}$, and tested on $\{\mathbf{y}_1^{\mathbf{m}_v}, \dots, \mathbf{y}_j^{\mathbf{m}_v}\}_{j=1}^{n_{te}}$, where $\mathbf{x}_i^{\mathbf{m}_v}$ is m_{sub} -dimension feature from the original cover feature $\mathbf{x}_i$ and $\mathbf{y}_j^{\mathbf{m}_v}$ is m_{sub} -dimension feature from the original feature $\mathbf{y}_j$ to be detected, which are batch unlabeled feature data. n_{tr} and n_{te} are the numbers of training and testing samples. We use $\mathbf{u}_1$ and $\bar{\mathbf{u}}$ as means of the cluster in the cover and the unlabeled testing feature, respectively. In the v th subspace, the projection vector $\bar{\mathbf{w}}_v$ is obtained by LDA on the training set.

The centroid of each class $\bar{\mathbf{u}}_j$ means $\bar{\mathbf{u}}_j = \sum_{\mathbf{y}_j^{\mathbf{m}_v} \in \theta_j} \mathbf{y}_j^{\mathbf{m}_v} / n_j$, ($j = 1, 2$), where n_1 and n_2 are the number of cover and stego images to be detected (unknown) [7] and θ_j is the j th cluster. For MPEDC, we assume that the input cover images have labels and the cover images to be detected have the same statistical property, e.g. $\mathbf{u}_1 \approx \bar{\mathbf{u}}_1$. Therefore, $\bar{\mathbf{u}}_2$ can be expressed as

$$\bar{\mathbf{u}}_2 = \bar{\mathbf{u}} - \bar{\mathbf{u}}_1 \approx \bar{\mathbf{u}} - \mathbf{u}_1 \quad (3)$$

Now, in the v th view, total scatter matrix $\bar{\mathbf{S}}_t$ and between-class scatter matrix $\bar{\mathbf{S}}_b$ may therefore be expressed as:

$$\bar{\mathbf{S}}_t = \sum_{i=1}^n (\bar{\mathbf{y}}_i^{\mathbf{m}_v} - \bar{\mathbf{u}})(\bar{\mathbf{y}}_i^{\mathbf{m}_v} - \bar{\mathbf{u}})^T \quad (4)$$

$$\bar{\mathbf{S}}_b = \sum_{j=1}^2 n_j (\bar{\mathbf{u}}_j - \bar{\mathbf{u}})(\bar{\mathbf{u}}_j - \bar{\mathbf{u}})^T \quad (5)$$

where the total scatter matrix $\bar{\mathbf{S}}_t$ means $\bar{\mathbf{S}}_t = \bar{\mathbf{S}}_b + \bar{\mathbf{S}}_w$. In

Fig. 1, we need to get the v th projection vector $\bar{\mathbf{w}}_v$, here

$$\bar{\mathbf{w}}_v = \bar{\mathbf{S}}_w^{-1}(\bar{\mathbf{u}}_1 - \bar{\mathbf{u}}_2) = (\bar{\mathbf{S}}_t - \bar{\mathbf{S}}_b)^{-1}(\bar{\mathbf{u}}_1 - \bar{\mathbf{u}}_2) \quad (6)$$

3.2 Multi-Projection Accessing and K-Means Clustering

MPEDC algorithm micro-rotate the projection vector $\bar{\mathbf{w}}_v$, which are similar to $\bar{\mathbf{w}}_1$, and then project the samples onto multiple vectors approximating ‘the best projection vector’ for integrated classification to get more accurate classification results. $\bar{\mathbf{w}}_v^\psi$ is as follows

$$\bar{\mathbf{w}}_v^\psi = \begin{cases} \bar{\mathbf{w}}_v^\psi + \mathbf{a}^\psi * \bar{\mathbf{w}}_v^\psi, & \psi > 0 \\ \bar{\mathbf{w}}_v, & \psi = 0 \end{cases} \quad (7)$$

where $\bar{\mathbf{w}}_v^\psi$ is the ψ th projection vector obtained randomly from $\bar{\mathbf{w}}_v$. The operation $*$ means the element-by-element multiplication. ψ is a positive integer, which is a parameter related to embedding rate r expressed as

$$\psi = \begin{cases} 5 - \text{round}(10r), & r \leq 0.5 \\ 0, & r > 0.5 \end{cases} \quad (8)$$

where ψ and r are negatively correlated and $10r$ should be an integer. If $10r$ is not an integer, we will round off this value. According to the LDA algorithm, the $\mathbf{a}$ stands for a randomly vector containing either positive or negative elements with values close to zero. Therefore, $\mathbf{a}$ is defined as

$$\mathbf{a} = \frac{2\mathbf{b} - 1}{10^{10r-1}} \quad (9)$$

where $\mathbf{b}$ is used to generate a random vector of m_{sub} dimensions with element values between 0 and 1. When calculating $\mathbf{a}^\psi$, we can get the corresponding $\bar{\mathbf{w}}_v^\psi$. About the choice of parameter ψ , we will explain more specification in the experimental part of the article.

After obtaining multiple projections, MPEDC can project $\bar{\mathbf{u}}_1$ and $\bar{\mathbf{u}}_2$ of each sub-classifier onto the corresponding projection vector respectively as the first cluster center of K-means clustering, i.e., $\bar{\mathbf{u}}_1 \bar{\mathbf{w}}_v^\psi \approx \mathbf{u}_1 \bar{\mathbf{w}}_v^\psi$. Every instance nearest to the clustering centroid will be distributed to the corresponding class.

In each sub-classifier, there will be two categories of cover and stego. MPEDC will re-cluster them with LDA and K-means algorithms, which means these two categories using LDA are projected onto a single vector for the supervised classification. The pseudo code of the iteration process is presented in **Algorithm 1**, where the parameter T is the number of iterations. The above-mentioned algorithm is shown in **Algorithm 2**. The parameter L stands for the number of the sub-classifier. In particular, P_E and τ represents the detection error and the number of experiments, respectively.

4. Experimental Verification

In our experiments, a total of 10,000 JPEG grayscale images

Algorithm 1 Iteration process

```

1: for  $T \leftarrow 1$  to  $t$  do
2:   Get tagged cover and stego images according to the previous clas-
     sification results;
3:   Compute the best projection vector with LDA algorithm by Eq. (2);
4:   Run K-means: obtain the cluster label vector;
5: end for

```

Algorithm 2 The proposed clustering algorithm

Require: $\mathbf{X} = \{\mathbf{x}_1^{m_v}, \dots, \mathbf{x}_i^{m_v}\}_{i=1}^{n_r}$, $\mathbf{Y} = \{\mathbf{y}_1^{m_v}, \dots, \mathbf{y}_j^{m_v}\}_{j=1}^{n_{te}}$, τ , m_{sub} , L , T

Ensure: Cluster label vector and P_E

```

1: for meantime  $\leftarrow 1$  to  $\tau$  do
2:   Form a random subspace
     
$$\mathbf{m}_v \subset \{1, \dots, m\}, |\mathbf{m}_v| = m_{sub} \ll m$$

3:   for  $l \leftarrow 1$  to  $L$  do
4:     Compute  $\mathbf{u}_1, \hat{\mathbf{u}}, \hat{\mathbf{S}}_r, \hat{\mathbf{S}}_b$  by Eqs. (3)–(5)
5:     Compute the projections  $\tilde{\mathbf{w}}_v$  by Eq. (6)
6:     Compute rotated multi-projection  $\tilde{\mathbf{w}}_v^\psi$  by Eqs. (7)–(9)
7:     Compute the projection vector for all samples and their means,
        such as  $\mathbf{u}\tilde{\mathbf{w}}_v^\psi, \mathbf{Y}\tilde{\mathbf{w}}_v^\psi$ 
8:     Run K-means: obtain the cover, stego clusters and their label
        vectors
9:     Run iteration algorithm in Algorithm 1
10:   end for
11: end for

```

from the BOSSBASE 1.01 [8] with the same size 512×512 and quality factors $QF = 75$ and $QF = 95$ are used as the original covers. We performed nsF5 (no-shrinkage F5) [9] and J-UNIWARD [8] steganographic methods on the original images to produce 10,000 stego images using CC-JRM [1], DCTR [2] and GFR [3]. All the results are from the average of $\tau = 10$ times.

4.1 Detection Error Comparisons

In Tables 1–2, we can obviously see the error rates of detecting the features of different steganalysis methods in J-UNIWARD and nsF5 with different embedding rates. For example, the error detection rates of MPEDC for different embedding rates of DCTR features in J-UNIWARD are almost lower than SEDC as $QF = 75$. However, the CC-JRM features with different embedding rates show different characteristics, and the detection rate of SEDC algorithm is lower than that of MPEDC at the embedding rates of 0.2, 0.3 and 0.4, which are respectively 47.9%, 41.4% and 33.7%. With the higher embedding rate, the detection is easier, especially against nsF5. Also, both SEDC and MPEDC methods have the poor performance on J-UNIWARD with lower embedding rates.

From Tables 1–2, we can clearly see that there are a few results that MPEDC is lower than SEDC. When calculating the rotating multi-projection vector, $\mathbf{b}$ is a random vector, so the vector obtained by the rotation has a certain randomness, which may lead to a very small number of cases that have a negative impact on the classification result. Even if our experiment takes the average of 10 experiments ($\tau = 10$), the

Table 1 The detection errors for different steganalysis schemes using SEDC and MPEDC in J-UNIWARD of different payloads with $QF = 75$ and $QF = 95$

QF	Feature	Method	Payload (bpnzac)			
			0.1	0.2	0.3	0.4
75	CC-JRM	SEDC	53.1%	47.9%	41.4%	33.7%
		MPEDC	52.4%	49.4%	42.5%	34.4%
	DCTR	SEDC	52.3%	44.0%	32.1%	24.1%
		MPEDC	51.7%	43.4%	32.0%	21.0%
	GFR	SEDC	46.7%	38.1%	24.4%	18.1%
		MPEDC	49.3%	36.8%	24.2%	14.0%
95	CC-JRM	SEDC	54.1%	54.0%	51.8%	47.2%
		MPEDC	53.0%	52.7%	51.5%	47.2%
	DCTR	SEDC	53.1%	53.0%	48.9%	43.4%
		MPEDC	54.3%	52.4%	48.5%	42.9%
	GFR	SEDC	51.7%	50.7%	45.8%	38.9%
		MPEDC	53.1%	50.4%	43.2%	35.3%

Table 2 The detection errors for different steganalysis schemes using SEDC and MPEDC in nsF5 of different payloads with $QF = 75$ and $QF = 95$

QF	Feature	Method	Payload (bpnzac)			
			0.05	0.1	0.15	0.2
75	CC-JRM	SEDC	45.7%	25.5%	17.6%	10.5%
		MPEDC	43.9%	27.0%	15.5%	7.7%
	DCTR	SEDC	44.0%	31.6%	19.1%	13.1%
		MPEDC	47.7%	32.9%	16.6%	8.0%
	GFR	SEDC	47.4%	36.6%	26.1%	17.7%
		MPEDC	48.1%	35.7%	23.4%	14.3%
95	CC-JRM	SEDC	39.5%	21.3%	12.3%	5.7%
		MPEDC	40.4%	18.7%	6.8%	2.0%
	DCTR	SEDC	45.7%	29.7%	16.1%	7.4%
		MPEDC	48.1%	27.4%	12.0%	3.6%
	GFR	SEDC	49.8%	36.2%	27.4%	18.9%
		MPEDC	47.4%	37.7%	25.1%	15.0%

Table 3 For the different features of the four embedding rates, as shown in the 12 sets of experiments in Tables 1–2, improving AVE of the detection rate of MPEDC compared to SEDC

QF	J-UNIWARD			nsF5		
	CC-JRM	DCTR	GFR	CC-JRM	DCTR	GFR
75	−0.65%	1.10%	0.75%	1.30%	0.65%	1.58%
95	0.68%	0.75%	1.28%	2.73%	1.95%	1.78%

negative effects cannot be completely excluded. Moreover, most of the classifiers do not have a good classification effect on the features of low embedding rate, and the MPEDC algorithm will amplify the negative effects on the features of low embedding rate. As shown in Table 1, when the embedding rate is 0.1 for the GFR ($QF = 75$), the error detection rate of MPEDC is higher than SEDC by 2.6%.

In Table 3, we list the improved average error detection rate (AVE) of MPEDC relative to SEDC under the four embedding rates of the same feature. It can be seen that for the 12 sets of experiments in Tables 1–2, AVE corresponding to the J-UNIWARD QF of 75 is slightly worse, and the other 11 sets of experiments are greatly improved, which also proves the effectiveness of our approach.

4.2 Iterative Weight Definition

In Fig. 2, we can clearly see that when the three of features

Fig. 2 For $QF = 75$, P_E over ten iterations of DCTR against J-UNIWARD (payload = 0.5), GFR against J-UNIWARD (payload = 0.4), and CC-JRM against nsF5 (payload = 0.1) with different dims of sub-classifiers, where iteration L , τ and m_{sub} are 95, 10 and 1100, respectively.

Fig. 3 As the number of ψ increases, the figure shows the value of the error detection rates for GFR, DCTR, CC-JRM ($QF = 95$) against nsF5 with payload = 0.2. The figure contains columns of blue, green, yellow and brown, which respectively represent the error detection rates when ψ is 1, 2, 3 and 4.

DCTR, GFR and CC-JRM are in the first iteration, the detection error rate is reduced by a large margin, while in more than second iterations, although the error rate is reduced, the reduction rate is less. Considering the time complexity and efficiency of our classification, we think that the performance of the classifier is higher when the number of iterations T is 1.

4.3 The Selection of Parameter ψ

The size of ψ has an important relation with the embedding rate as Eq. (8). For example, when the embedding rate is 0.2, the projection rotates slightly three times; when the embedding rate is greater than or equal to 0.5, the projection does not rotate. In Eq. (8), $10r$ should be an integer. When $10r$ is not an integer, we round off the r value. For example, when the embedding ratio is 0.015, the value of 10×0.015

is 0.15, then we take the approximation of r as 0.2 and the number of ψ as 3. In Fig. 3, we can clearly see that yellow bars of three features are smaller than those of the other colors as the number of ψ changes. This shows that when the projection micro-rotates three times, the error rate will be lower.

5. Conclusion

In this paper, we point out the close relationship between LDA and K-means clustering. Then we rotate a projection obtained by the LDA algorithm in a random subspace and yield approximately multiple projections so as to combine LDA and K-means clustering into MPEDC. Experimental results show that the proposed method can effectively detect J-UNIWARD and nsF5 as the state-of-the-art steganographic algorithm. Especially for steganographic features with a high embedded rate, the detection error rate is lower.

References

- [1] J. Kodovský and J. Fridrich, "Steganalysis of JPEG images using rich models," Proceedings of SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics XIV, San Francisco, CA, Jan. 23–25, vol.8303, p.83030A, 2012.
- [2] V. Holub and J. Fridrich, "Low complexity features for JPEG steganalysis using undecimated DCT," IEEE Trans. Inf. Forensics Security, vol.10, no.2, pp.219–228, 2015.
- [3] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang, "Steganalysis of adaptive JPEG steganography using 2D Gabor filters," Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security, pp.15–23, 2015.
- [4] J. Kodovský, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," IEEE Trans. Inf. Forensics Security, vol.7, no.2, pp.432–444, 2012.
- [5] C. Ding and T. Li, "Adaptive dimension reduction using discriminant analysis and K-means clustering," Proceedings of the 24th International Conference on Machine Learning (ICML), pp.521–528, 2007.
- [6] A. Wu, G. Feng, X. Zhang, and Y. Ren, "Unbalanced JPEG image steganalysis via multiview data match," Journal of Visual Communication & Image Representation, vol.34, pp.103–107, 2016.
- [7] B. Cao, G. Feng, Z. Yin, and L. Fan, "Unsupervised image steganalysis method using self-Learning ensemble discriminant clustering," IEICE Trans. Inf. & Syst., vol.E100-D, no.5, pp.1144–1147, 2017.
- [8] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," EURASIP Journal on Information Security, vol.2014, no.1, pp.1–13, 2014.
- [9] J. Fridrich, T. Pevný and J. Kodovský, "Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities," Proceedings of the 9th ACM workshop on Multimedia and security, Dallas, TX, Sept. 20–21, pp.3–14, 2007.