Pattern Recognition Techniques

Project Assignment 2009/2010

SAFE-Audio: Stego Analysis For Evaluation of Audio

by **João Carlos Ferreira Gonçalves**

jcgonc@student.dei.uc.pt

Abstract. In this project several kinds of discriminating classifiers are used to help in the topic of steganography. This topic comprises techniques used to hide confidential information behind innocent data. The original (unmodified) source is called cover and the data with hidden information is labeled stego. As in the project's proposal paper, the task comprises of the development and study of different classifiers and feature selection to help discriminating between stego and cover data on audio files, with the help of MatLab software. Considered feature reduction techniques are the Kruskal Wallis Test, the Principal Component Analysis, the Linear Discriminant Analysis and the Generalized Discriminant Analysis, including other studied tools. Used classifying algorithms are Bayesian Inference, Radial Basis Functions, Feed forward Neural Networks and Support Vector Machines.

Keywords. Steganography, recognition, discriminative classifiers, audio

1. Introduction

In this project, the task of classification is done using two classes, being the one labeled (1) the cover and the one labeled (2) stego. This kind of classification is also known as binary classification. The stego observations correspond to the positives, while the remaining, cover, are the negatives.

Two file formats are analyzed in search of stego information, **MP3** and **WAV**. These formats, which are audio files, are used to carry one or two kinds of information, only cover or cover with hidden data, stego. For both of them, several algorithms can be used to retrieve features required for the development of classifiers, to help identifying these two kinds of files. Next, the two most used audio formats are discussed briefly:

- **WAV** (short for wave) files usually hold **uncompressed PCM** (**pulse code modulation**, i.e., voltage levels sampled at regular intervals), with one or more channels per sample, various sample rates, bit depth and possible different encodings (compressed). The uncompressed variant is usually the biggest in terms of memory storage, but with the highest fidelity (also called **lossless encoding**, because the file preserves original source information);
- **MP3** (short for **MPEG Audio Layer 3**) files are **lossy** files (loose detail in reference to the original source). These files use a **modified form of the Discrete Cosine Transform** to change the original information (amplitude per sample, mono channel case) to a new space of frequencies per window. From this new space, the MP3 codec applies more techniques (**psychoacoustics**, **masking effects**) to attenuate (or eliminate) unperceivable frequency to the human ear. Finally, it applies a quantizer to further reduce the amount of bits required for coding.

On both the file formats, confidential information can be hidden. Depending on the format, this information can be hidden using different techniques, for example:

- On the WAV using the least significant bit of the sample values (more complex ways can be devised).
- On the MP3 file using modulation on some unperceivable frequency by the human ear.

2. Features

Using external methods (not studied in this project), several kinds of features (attributes) are extracted from the audio files. In the case of the MP3 files, there are four kinds of attributes, grouped by extraction method, and these are referred as "feature groups" on this document. The attributes are summarized in table 1.

Number of features	Description	Group ID
81	Inter-frame Markov features	IM
161	Inter-frame joint density features	IJ
576	Derivative spectrum analysis feature for 576 sub-bands	2DS
4	Moment statistical features of shape parameter (beta)	Bm

Table 1. Features and number from the MP3 file.

The above corresponds to a total of 742 features, stored in the file "*data\mp3_features.csv*". This file stores 1994 samples (6 were removed from the original file, as they were broken), where 997 are of each class.

In the case of the WAV, 58 features were extracted using the second derivative based mel-cepstrum (D-MD). The features are stored in 5 files, one for the cover class and the remaining 4 for stego. Each stego file was hidden data using the algorithms/tools Hide4PGP V4.0, Invisible Secrets, LSB matching and Steghide. The files are also referred as "feature groups" and are resumed in table 2.

Filename	ID	Samples	Tool used	Class
cover6000mono_orig.txt		4390		1
hide4pgp25mono_orig.txt	1	6000	Hide4PGP V4.0	2
invislbe50stero_orig.txt	2	4886	Invisible Secrets	2
lsbmatching50_orig.txt	3	6000	LSB matching	2
steghide1005_orig.txt	4	1003	Steghide	2
steghide993_orig.txt	5	993	Steghide	2

Table 2. Features and number from the MP3 file.

The features referenced before are used to build the classifiers according to the target class (stego or cover) for both the MP3 and WAV files.

With the purpose of not prioritizing one class over the other, the same amount of samples are randomly (from an uniform distribution) chosen from each class before the feature reduction phase. This is done with the MatLab function "**selectxfromn.m**".

Finally, the label (1) corresponds to the class "cover" (negatives) and the label (2) to the class "stego" (the positives).

3. Feature reduction and selection

As one can tell, in the case of the MP3, it's not practical to design a complex (like a SVM) classifier using the entire set of features (742). The learning/training time could last a significant amount of time, as well as the testing stage (while this should be shorter). Also, problems may arise because some features may confuse the classifier, so these must be eliminated. In the case of the WAV data set, this one has a much smaller amount of attributes (58), so the task of selecting/reducing this amount is simpler (but still recommended).

A number of algorithms exist to reduce and select the amount of required but still significant attributes. It's obvious that this step can't be too naïve, as it can seriously affect the discrimination done by the next stage, the classifiers. This is because it is possible that some information in the removed features, if present, would be helpful to the performance of the classifiers.

First of all, there is a difference between feature reduction and feature selection. The first, **reduction**, comprises of some algorithm which takes all of the initial attributes (feature space) and for instance, using a linear combination of those, transforms (or maps) that space in another projected space with different features (which quite possibly have no real world meaning). Examples of tools of this kind are the PCA, LDA and GDA (more do exist). The second, **selection**, as the name says, corresponds to some manual or automatic selection of input features. In this case, the chosen features exist in the initial feature set and are exactly the same. Examples of algorithms are the Kruskal-Wallis test and various feature selection search algorithms.

Some of the available tools used for this purpose are summarized below. All of them were studied in the course of development of this project, but only some were chosen to be used in the final version. The algorithms are the following:

- The **PCA** (**Principal Component Analysis**) finds the **orthogonal linear transformation** of input features which, when projected, transforms the input space to a new coordinate system. This transformed coordinate system has the **maximum variance of the input data in it's the first coordinate**, the second greatest variance in the second coordinate and so on. It's a non supervised algorithm, so it doesn't have in account the classes' separability.
- The **Kruskal-Wallis test** (KW-t) is used for **ranking medians** among groups of a population. Basically, in the case of feature elimination, it serves as a tool to check if the features are from a similar distribution (i.e. have same median and consequently, in this test they have the same rank). Using this algorithm, the features with the similar rank values are very likely from a similar distribution and usually removed (selected) from the initial feature set. Like the PCA, this tool is also unsupervised, consequently it doesn't give any information regarding the most discriminating features.

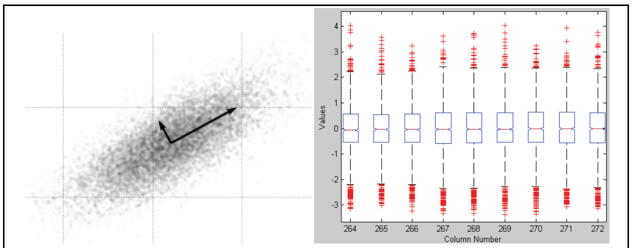

Figure 1. Left is an example of PCA with two features and resulting projection (major and second major variance axes) . The right is KW's test, with the middle frequencies of 2DS from MP3, all very similar between them, as easily seen.

- The **LDA** (**Linear Discriminant Analysis**) which finds the linear combination of input features which best divides two or more classes. Unlike PCA, this method is supervisionized, consequently it uses additional information (the sample's classes, labels) to train the linear transform which maximizes the ratio between "class separation" and "within-class scatter" on the projected data. This tool is also known as "**Fisher's Linear Discriminant**" and it also includes a linear classifier, except that in this project only the feature reduction stage is used.
- The **GDA** (**Generalized Discriminant Analysis**). The GDA is "kernelized" version of the Linear Discriminant Analysis (LDA). It produces the kernel data projection which maximizes the class separability of the input feature space. Basically it has the same goal like the LDA, but using the kernel trick to map the input space to a greater dimension.

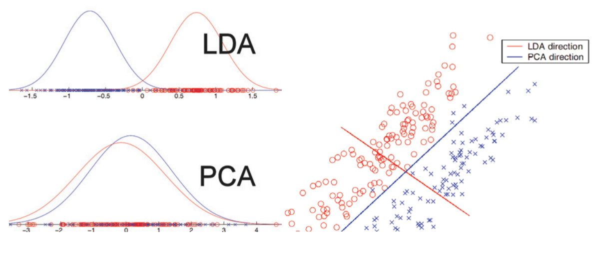

Figure 2. Comparison between both PCA's and LDA's first projected axis (output space). The PCA's direction matches with the feature's maximum variance direction. LDA's direction is almost perpendicular, but fits with the direction which best separates data.

While the first two algorithms (KW-t and PCA) were tested during this project, they proved insufficient in one primary aspect: eliminating features not helpful in discriminating between the two classes, stego and cover.

For instance, it is possible that one of the last coordinates of the PCA projected space is good separating classes. If these coordinates are removed, the classifier stage may have problems while discriminating data. Additionally, it may happen that both the linear combination of features produced by the PCA and the problem referred above is worst than not applying PCA at all (for instance by using only the first output space axis (blue) in figure 2).

In the case of the KW test, as it tests for similarity of features (independently of their classes), their ranks are not in any way signal of discriminating performance. For an example, in figure 1, the shown feature's statistical distributions are very similar. However, one or more of them may not be used in the classifier (eliminated using KW-t) and be in fact an excellent discriminating feature. This happens because KW-t (as well as the PCA) are not supervised tools.

After analyzing both the PCA and KW test and studying the feature elimination problem, one simple tool was devised for helping choose the best features in discriminating the two classes. Basically, the algorithm compares the two histograms (per class) of each feature and ranks the features according to the highest histogram difference.

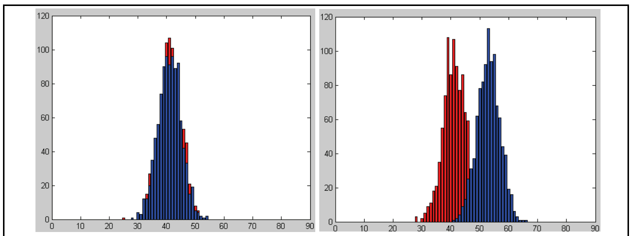

Figure 3. Two histograms from one feature, each color corresponds to one class. Left figure corresponds to example "a", right to example "b".

For example, in the figure 3 there are two features, “a,” and “b”, both following a normal distribution, where each feature’s class histogram (using 81 bins) is drawn on a bar plot. The two features correspond to a population size of 2000 samples, 1000 per class.

For the figure “a”, the feature follows the same normal distribution for both classes. The area of the difference between both histograms is 146 (using 81 bins). For the figure “b”, the area of the difference between both histograms is 1746 (using 81 bins). Consequently, for a population of 2000 samples, the feature “b” (with 1746 identified samples in all population) is a much better discriminating feature than “a” (with 146 identified samples). A better way to compare features is to divide the number of unique elements (difference between classes’ histograms) by the total amount of samples (for example, in the “b” feature, score is 1746/2000 which is 0,873.

In the limit, the amount of bins may be infinite (and the histograms would approach density functions). In this case, any possible differences between the features distribution (per class) may be detected by the tool. Logically, this is the best any classifier can do, finding each one of these unique (to each class) samples (and using only one feature). However, a classifier which finds the separating hyperplane (or surface) in this case would be extremely specific, something which is not desired.

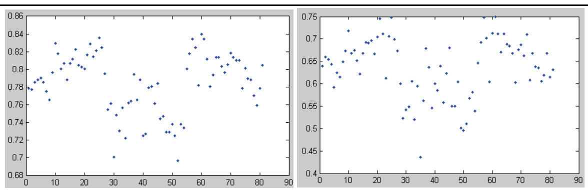

Figure 4. Comparison between the score of described histogram tool using 32000 bins (left) and the F-Score achieved by a Feed forward Neural Network with two layers, 10 neurons each (right), using the 81 IM features from the MP3 dataset . Both figures are correlated. Naturally, because the classifier is somewhat generic, it’s scoring is lower than the histogram tool.

Basically, this algorithm is equivalent to run a classifier in each feature, choosing the ones which give best results. However, it is much faster. It would be interesting to further develop this tool with the help of Kruskal-Wallis or ANOVA tests, to find correlated features and eliminate those, selecting only the best discriminating and uncorrelated features.

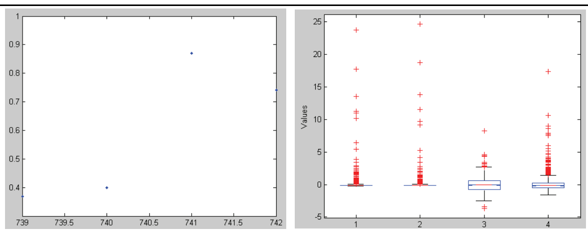

Figure 5. Histogram tool score (left) and the KW-t (right) for the MP3’s Bm feature group. In this case the attributes 741 and 742 were selected.

This algorithm was run for all features in the MP3 dataset and used to choose the best discriminating features. On average (depending on the feature group), all of those with discriminating ratio below 50% were eliminated (2DS and Bm feature groups). On better feature groups (like the IM and IJ) the selection threshold was higher, selecting the feature with score above 75% or more (IM group for instance, figure 4).

The tool was also used to study the best features in the WAV dataset, but in this case, all of them were judged to be important (none was considered as a confusing one, however some maybe).

For both MP3 and WAV datasets, following the feature selection all features are further processed in a feature reduction tool (first stage) with the algorithms PCA, LDA and GDA. In this stage, several conclusions were taken regarding the considered reduction tool. These are commented on the section “Results”. On the next stage, the projected features are used in the classifiers. The classifiers were also tested with only the selected features (no feature reduction) to study for possible problems in projected features.

4. Classification

Four kinds of classifiers are used in the project:

- Bayes / Bayesian classifier, using the Gaussian Mixture model (mlcgmm) for the classes’ conditional distributions. However the Expectation-Maximization Algorithm for Gaussian mixture model (emgmm) was also considered, but it’s not present in the results;
- Radial Basis Networks, with user configurable maximum amount of neurons and spread. Because the output layers of these networks, using MatLab’s functions follow a linear function, the outputs below 1.5 are set to (1) and the outputs above 1.5 are set to (2). This way the network’s predicted values only have two possible values (binary classification);
- Feed Forward Neural Networks, using one or two layers with configurable amount of neurons and with the Hyperbolic tangent sigmoid transfer function (tansig) on all layers. This way only the values 1 or 2, the labels for each class (cover/stego), appear at the classifier’s outputs;
- Support Vector Machines, using the Sequential Minimal Optimizer (SMO) to train the SVM classifier with L1-soft margin. The used kernel is the RBF Gaussian with user configurable spread. The DLL / Mex32 used was a recompilation of the STPRTool’s source using Intel’s C++ Compiler 10.1, maximum floating unit precision with SSE3 vectorization (for the fastest execution).

These classifiers are discussed on the “Results” section.

5. Classifier validation and evaluation

The classifiers are trained using the TRAIN set and evaluated using the TEST set. None of the samples present in the TRAIN dataset are present in the TEST set. These sets are built using K-Fold cross validation (MatLab supplies this function), where the number of folds is user configurable.

The metrics used to evaluate the classifiers are the **F-Score** (also known as F1-score and F-measure) and **AUC** (Area Under roc Curve). The F-Score is a geometric mean of the classifier precision and recall:

$$F_{score} = \frac{2 \cdot (precision \cdot recall)}{precision + recall}$$

Where

$$Precision = \frac{tp}{tp + fp}$$

And

$$Recall = \frac{tp}{tp + fn}$$

The *tp* variable corresponds to the amount of true positives (correct positives predicted by the classifier), the *fn* to the number of incorrectly predicted negatives (by the classifier) and the *fp* to the wrong (incorrect) quantity of predicted positives by the classifier. There is also the variable *tn* which stands for the amount of correctly predicted negatives.

The used AUC metric is a trapezoidal integration of the function TPR (true positive ratio) vs. TNR (false positive ratio). Both are correlated (usually have similar values) and high scores are a signal of a good classifier.

These metrics are computed for each feature set in the end of the K-Fold validation, by merging all the generated folds TEST sample's targets and comparing with the classifier prediction "ypred".

6. The MatLab code and Graphical User Interface

For an easy and intuitive project analysis, a GUI was built for each dataset. Consequently, two GUIs exist, one for the MP3 case and another for the WAV. Each of them has the required options for configuring and launching the classification of stego and cover data on the corresponding dataset. There is an option for using the custom selected features (using the histogram tool) which when disabled, uses all the group's features. The GUIs are executed from the files "**mp3configfig.m**" and "**wavconfigfig.m**".

Useful are also the "batch" versions of the code. These were made for long time execution and collection of results, for instance to study the four types of classifiers in all MP3 features group's combinations. The configuration and execution are done in the files "**mp3_project_launcher.m**" and "**wav_project_launcher.m**". In the end of the execution, both used metrics (F-Score and AUC) are displayed in a matrix, where rows correspond to feature groups (MP3) or file's ID (WAV) and columns to the classifiers (using the classifier order, variable "classifiers").

7. Results

Using the "batch" versions of the code, all four types of classifiers were run with both all and none feature reduction techniques referred before. The used features were, for the MP3, all possible combinations of the feature groups (using "a priori" custom feature selection with the histogram tool), for example: IM, IJ, 2DS, Bm, IM+IJ, IM+2DS, … , IM+IJ+2DS+Bm. For the WAV, only five feature groups were tested, one for each stego hiding technique (file). As said before, no feature selection was applied on the WAV files' features, only feature reduction.

All 1994 samples of the MP3 data set were used (997 per class). For the WAV dataset, a maximum of 2000 samples were used per class (depending on the amount of available samples per file). As said before, on the WAV study the training and testing samples were uniformly chosen from the initial population.

The parameters used for the classifiers were the following: for the SVM, RBF kernel with spread of 4 and regularization constant of 10; for the RBF networks, spread of 128 and maximum amount of neurons of 200; the FFNN had two layers, 20 neurons each.

The feature reduction algorithms also have personalization options.

The PCA was run on the "automatic" mode. In this, the number of projected coordinates is computed using the PCA's eigenvalues derivative. When the eigenvalues derivative reaches a specified value (0.02 was the used value), the eigenvalue's coordinate number corresponds to the amount of used axis. For a better example see figure 6. Higher threshold values produced worse results (as less features were generated from the PCA) and lower threshold values did not improve much the classification.

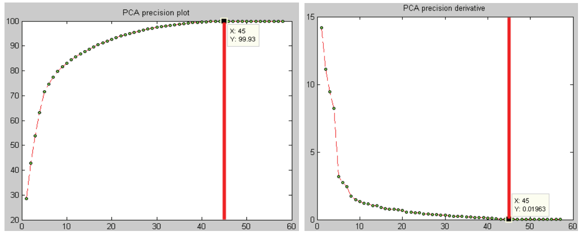

Figure 6. PCA's amount of variance per number of output axis (left) and PCA's eigenvalue derivative per output coordinate (right). If the eigenvalue derivative threshold is set to 0.02, the code selects the first 45 projected features from the PCA's output space. The shown features are the ones from the WAV file "hide4pgp25mono_orig".

The LDA algorithm was run with 2 projected linear combinations (coordinates) of features on both the WAV and MP3 data sets.

The GDA always produced 1 coordinate (one kernel projection of all the input features). However it was tested with 2 RBF spread (s) values: 1 and 4.

The results, for the WAV dataset F-Score and AUC, can be seen on the tables 3 and 4 (annex). The AUC scores for the WAV dataset are extremely similar to the F-Scores (1% difference). The table 3 results (F-Score) are summarized on the next graphics.

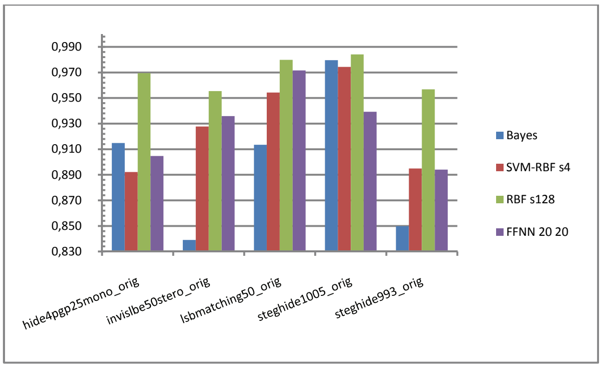

Figure 7. F-Scores for the WAV files, using no feature reduction.

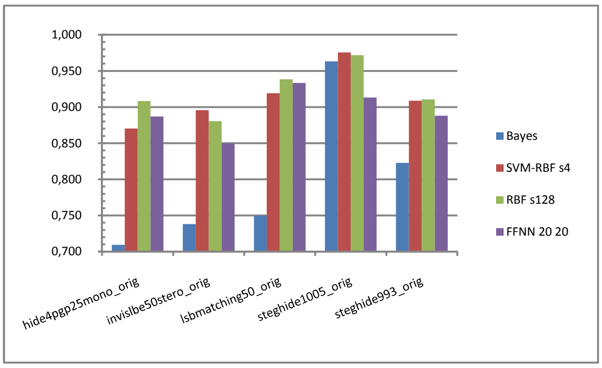

Figure 8. F-Scores for the WAV files, using PCA with eigenvalue derivative threshold set to 0,02.

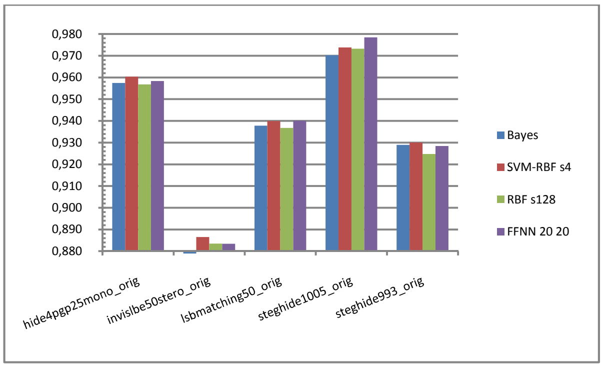

Figure 9. F-Scores for the WAV files, using LDA with a projected space of 2 dimensions.

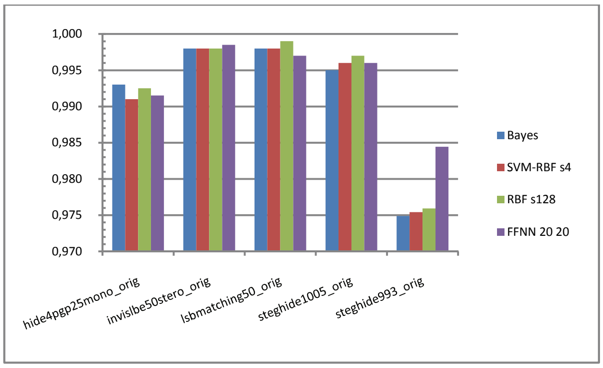

Figure 10. F-Scores for the WAV files, using GDA with a RBF kernel and spread = 1.

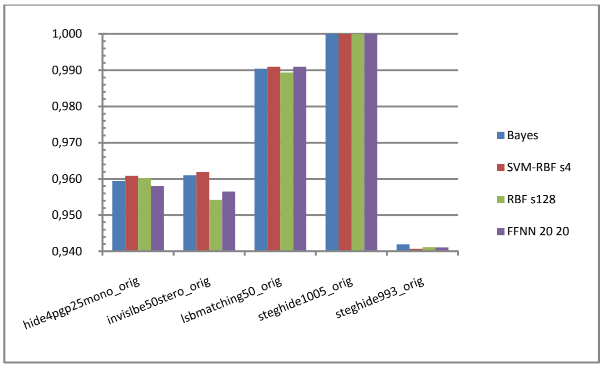

Figure 11. F-Scores for the WAV files, using GDA with a RBF kernel and spread = 4.

The graphical results, for the MP3 dataset F-Score, are shown in the next figures.

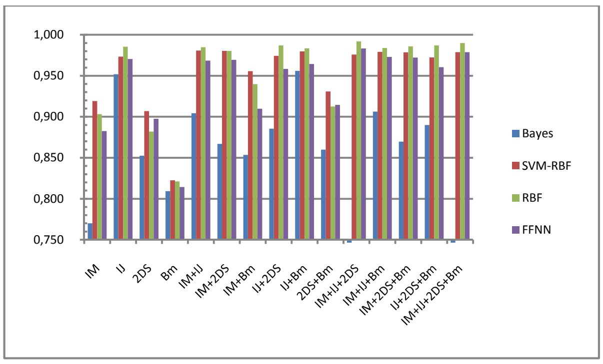

Figure 12. F-Scores for the MP3 feature groups, using no feature reduction.

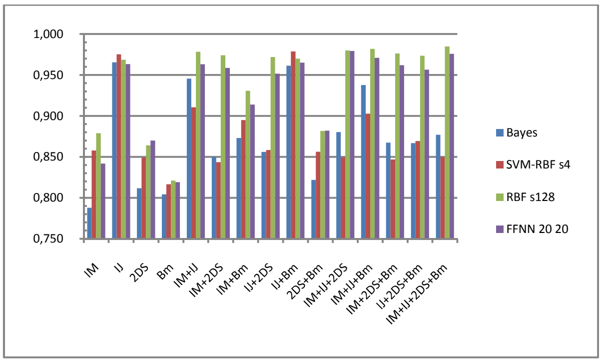

Figure 13. F-Scores for the MP3 feature groups, using PCA with eigenvalue derivative threshold set to 0,02.

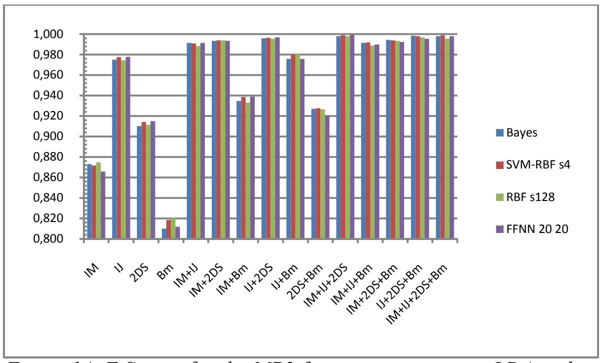

Figure 14. F-Scores for the MP3 feature groups, using LDA with a projected space of 2 dimensions.

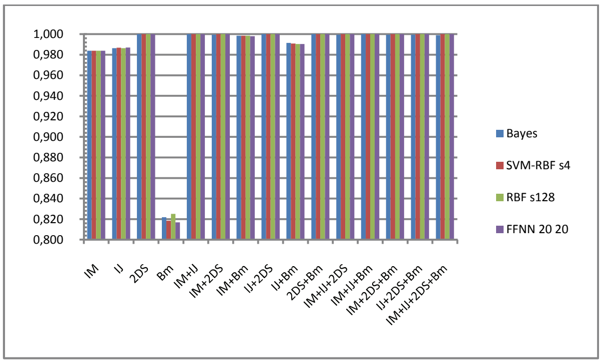

Figure 15. F-Scores for the MP3 feature groups, using GDA with a RBF kernel and spread = 4.

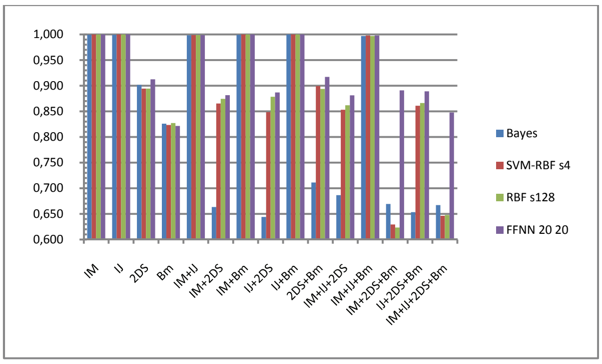

Figure 16. F-Scores for the MP3 feature groups, using GDA with a RBF kernel and spread = 1.

Like the WAV results, the AUC and F-Scores scores are extremely alike (on average, within less than 1% difference).

8. Conclusions on the results

First of all, it’s natural to comment on the feature reduction. If this stage is imperfect, it’s very hard a classifier to have good results.

On both the WAV and MP3 datasets, the worst results were using the PCA for feature reduction. The main reason, as said before, is that the PCA is not a supervised tool (section 3). It doesn’t guarantee at all that the first coordinates of its transformed space are the most discriminating ones. It’s a good method of feature reduction but only when the initial features are good, from which “steghide993_orig” is a good example of.

The LDA gave very good results, in spite of being still a feature reduction where its projected space is a linear combination of the input features. On the file “invislbe50stero_orig” it had some problems, because the initial feature space has the two classes somewhat overlapped. On the remaining WAV files it had a good performance in all classifiers, all above an F-Score of 93%, and for the MP3, on more advanced feature combinations, scores above 95%. In this last dataset, using the combined features of IM+IJ+2DS all classifiers had scores above 99.8%. It’s easy to conclude that this simple feature reduction tool is highly recommended.

The GDA was the only feature reduction tool to both give F-Scores and AUCs of 100%. While it takes much more time than the LDA to compute a kernel projection of the input feature space, the results are quite impressive. It only has the problem that it requires (like the classifiers) some fine tuning, because of its RBF Gaussian spread value. Several spread options were tested, from 0.5 to 16 (in the results only two are present, 1 and 4), each one with its advantages and disadvantages. Basically, the RBF spread (the same applies to the classifiers SVM and RBF) controls how specific or generic is the kernel projection. Smaller spread values made the GDA very specific and it generally had problems on both the WAV and MP3 datasets (with a spread of 0.5 it had worst results than the LDA). On the other hand, a higher RBF spread forced the GDA to be more generic and to have a smoother feature projection space. Using the spread of 1 unit gave in average the best results in all datasets.

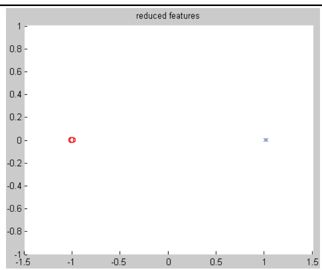

Figure 17. GDA 1D feature projection for the MP3’s custom IM features, using a RBF spread of 1 unit. Naturally, all classifiers have an F-Score of 100% with this single projected feature.

Before going into the classifiers, it remains to refer the feature selection stage, which applies only to the MP3 dataset. Not present in the results, it’s logical to assume that when some features are manually removed from the feature space, some problems may arise during the both the feature reduction and classification stages. This happens because, like said in the section 3, some discriminating features which could be useful are not present to maximize the classifier’s performance. In the case of the PCA, removing the “confusing” (bad discriminating) features helped the classifiers, but on the other hand, if they were present in the LDA and GDA, these could give even better results. This happens because these two tools find the best linear / kernel projection which maximizes class separation and minimizes class scattering (the optimization goal of these algorithms). This can be checked by running the GUI for MP3, enabling / disabling the custom features option, changing the feature reduction tools and analyzing the results.

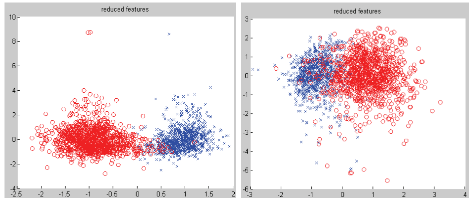

Figure 18. LDA 2D feature projection for all the MP3’s 2DS features (left) and only the 2DS custom features, manually selected using the histogram tool (right).

Following the feature reduction stage comes the classifiers. On most results, the Bayesian classifier had the worst results. It’s natural, because it’s quite simple it’s inference. Using the Gaussian Mixture Model this classifier has a very generic class discrimination. The figure 19 is an example of this. Consequently, it can’t be very specific and adapt to the more complex feature spaces. However, in better feature reduction methods and with good features it had excellent results, for instance MP3’s LDA projection of IM+IJ+2DS where it had a score of 99.4%.

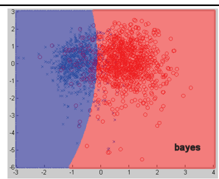

Figure 19. Bayes classifier classification domain and the population samples, for the LDA projection of MP3’s 2DS features.

The Support Vector Machine has the problem of, like the GDA, requiring fine tuning of the RBF Gaussian spread. It’s results on general were good, but depending on the dataset and feature space it’s results changed. The same conclusion taken on the GDA applies to the SVM: smaller spread values make the SVM more specific and bigger spreads make it more generic (figure 20). On average, with simple reduction methods like PCA and LDA the SVM had one of the best results.

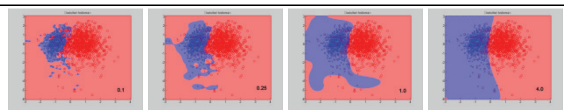

Figure 20. From left to right, SVM-RBF with spread values of 0.1, 0.25, 1.0 and 4.0, respectively.

The same conclusion as with the SVM applies to the RBF networks. Higher spread values makes them more generic, while smaller spreads makes them more specific. Also, when the spread is small, these networks required more neurons to successfully learn the best feature’s discriminating hyperplane/surface, where the network kind of “memorizes” the training set. On the other hand, with higher spreads, these networks need very few neurons, for instance, with spreads of 16 units, less than 10 neurons are required for a good performance. Figure 21 shows the effect of the RBF Gaussian spread on the separating surface of the feature space. This network also had excellent F-Scores on most dataset’s feature groups, but like the SVM, its performance was internally linked with the classes’ feature space dispersion. On some feature groups it was the best, on others not. On general, was taken the conclusion that high Gaussian spread values give the best results (and consequently, very generic networks).

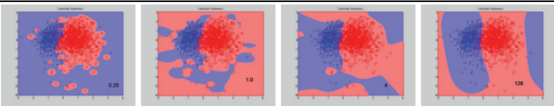

Figure 21. From left to right, RBF network with spread values of 0.25, 1.0, 4.0 and 128, respectively.

Last, remains the Feed Forward Neural Network. This kind of networks also has parameters like the spread of the classifiers referred before. These parameters also require fine tuning, depending on the feature set and are the amount of hidden layers and neurons on each one of these layers. Consequently, more layers and more neurons make the network more specific, as it has more maneuvering space for following the feature space distribution (see figure 22). Like the SVM and RBF networks, best results were got with less layers and neurons, but depending on the feature space, the FFNN had or not the best performance from all the classifiers. Additionally, these networks have the problem of hitting a minimal local center of their combined neural functions, which sometimes makes these networks stall with bad classification results.

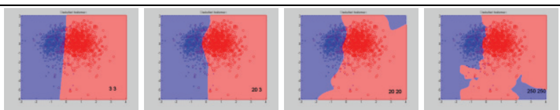

Figure 22. From left to right, FFNN with layer configuration of [3 3], [20 3], [20 20] and [250 250] neurons, respectively.

From the WAV datasets, on general the file with the best scores was the "steghide993_orig". With the GDA feature reduction algorithm, all the WAV datasets had excellent results.

On the MP3 dataset, all possible combinations of the feature groups were checked. As already said before, the best F-Score and AUC came from the combination of the IM+IJ+2DS feature groups. The adding of the Bm features usually (without the use of the LDA and GDA feature reduction) lowered the scores, because they are what is called "confusing" features. Without any feature reduction or with LDA / PCA the feature IJ alone could give F-Scores above 95% on all classifiers.

9. References

- Bishop, M. Cristopher : Pattern Recognition and Machine Learning
- Theodoridis, Sergios; Koutroumbas, Konstantinos : Pattern Recognition
- Sá, J. P. Marques de : Pattern Recognition - Concepts Methods and Applications
- Ribeiro, B. : TRP Lecture Notes
- Franc, V and Hlávač, V: Statistical Pattern Recognition Toolbox for Matlab - User's guide
- Feature Mining and Intelligent Computing for MP3 Steganalysis
- Rita Ferreira, Bernardete Ribeiro, Catarina Silva, Qingzhong Liu and Andrew H Sung, Building Resilient Classifiers for LSB Matching Steganography
- Krenn, Robert: Steganography and Steganalysis
- http://en.wikipedia.org/wiki/Linear_discriminant_analysis
- http://en.wikipedia.org/wiki/Principal_component_analysis
- http://en.wikipedia.org/wiki/Kruskal–Wallis_one-way_analysis_of_variance
- http://en.wikipedia.org/wiki/MP3
- http://en.wikipedia.org/wiki/WAV

10. Annex

The following table stores the F-Score results from the WAV dataset.

Reduction	File	Bayes	SVM	RBF	FFNN
none	hide4pgp25mono_orig	0,915	0,892	0,969	0,905
none	invislbe50stero_orig	0,839	0,928	0,955	0,936
none	lsbmatching50_orig	0,913	0,954	0,980	0,972
none	steghide1005_orig	0,980	0,974	0,984	0,939
none	steghide993_orig	0,850	0,895	0,957	0,894
PCA d0,02	hide4pgp25mono_orig	0,709	0,870	0,908	0,887
PCA d0,02	invislbe50stero_orig	0,738	0,896	0,880	0,850
PCA d0,02	lsbmatching50_orig	0,750	0,919	0,938	0,933
PCA d0,02	steghide1005_orig	0,963	0,976	0,972	0,913
PCA d0,02	steghide993_orig	0,823	0,909	0,911	0,888
LDA 2DD	hide4pgp25mono_orig	0,957	0,960	0,957	0,958
LDA 2DD	invislbe50stero_orig	0,879	0,887	0,884	0,883
LDA 2DD	lsbmatching50_orig	0,938	0,940	0,937	0,940
LDA 2DD	steghide1005_orig	0,970	0,974	0,973	0,978
LDA 2DD	steghide993_orig	0,929	0,930	0,925	0,928
GDA 1D s1	hide4pgp25mono_orig	0,993	0,991	0,993	0,992
GDA 1D s1	invislbe50stero_orig	0,998	0,998	0,998	0,999
GDA 1D s1	lsbmatching50_orig	0,998	0,998	0,999	0,997
GDA 1D s1	steghide1005_orig	0,995	0,996	0,997	0,996
GDA 1D s1	steghide993_orig	0,975	0,975	0,976	0,984
GDA 1D s4	hide4pgp25mono_orig	0,959	0,961	0,960	0,958
GDA 1D s4	invislbe50stero_orig	0,961	0,962	0,954	0,957
GDA 1D s4	lsbmatching50_orig	0,990	0,991	0,989	0,991
GDA 1D s4	steghide1005_orig	1,000	1,000	1,000	1,000
GDA 1D s4	steghide993_orig	0,942	0,941	0,941	0,941

Table 3. F-Scores for all the WAV files, using none or PCA, LDA and GDA feature reduction and the 4 types of considered classifiers.

Reduction	File	Bayes	SVM	RBF	FFNN
none	hide4pgp25mono_orig	0,920	0,894	0,970	0,908
none	invislbe50stero_orig	0,829	0,928	0,955	0,936
none	lsbmatching50_orig	0,915	0,955	0,980	0,972
none	steghide1005_orig	0,980	0,975	0,984	0,940
none	steghide993_orig	0,844	0,893	0,956	0,889
PCA 0,02	hide4pgp25mono_orig	0,709	0,870	0,908	0,887
PCA 0,02	invislbe50stero_orig	0,738	0,896	0,880	0,850
PCA 0,02	lsbmatching50_orig	0,750	0,919	0,938	0,933
PCA 0,02	steghide1005_orig	0,963	0,976	0,972	0,913
PCA 0,02	steghide993_orig	0,823	0,909	0,911	0,888
GDA 1D s1	hide4pgp25mono_orig	0,993	0,991	0,992	0,991
GDA 1D s1	invislbe50stero_orig	0,998	0,998	0,998	0,998
GDA 1D s1	lsbmatching50_orig	0,998	0,998	0,999	0,997
GDA 1D s1	steghide1005_orig	0,995	0,996	0,997	0,996
GDA 1D s1	steghide993_orig	0,975	0,976	0,976	0,984
LDA 2D	hide4pgp25mono_orig	0,959	0,961	0,960	0,959
LDA 2D	invislbe50stero_orig	0,897	0,902	0,900	0,905
LDA 2D	lsbmatching50_orig	0,939	0,948	0,948	0,945
LDA 2D	steghide1005_orig	0,971	0,975	0,975	0,978
LDA 2D	steghide993_orig	0,930	0,933	0,930	0,933
GDA 1D s4	hide4pgp25mono_orig	0,960	0,961	0,961	0,959
GDA 1D s4	invislbe50stero_orig	0,961	0,962	0,954	0,957
GDA 1D s4	lsbmatching50_orig	0,991	0,991	0,990	0,991
GDA 1D s4	steghide1005_orig	1,000	1,000	1,000	1,000
GDA 1D s4	steghide993_orig	0,940	0,938	0,939	0,939

Table 4. AUC scores for all the WAV files, using none or PCA, LDA and GDA feature reduction and the 4 types of considered classifiers.

The next table has the F-Score results from the MP3 dataset.

Reduction	File	Bayes	SVM	RBF	FFNN
none	IM	0,770	0,919	0,903	0,883
none	IJ	0,952	0,973	0,985	0,970
none	2DS	0,853	0,907	0,882	0,898
none	Bm	0,809	0,822	0,821	0,814
none	IM+IJ	0,904	0,981	0,985	0,968
none	IM+2DS	0,867	0,980	0,980	0,969
none	IM+Bm	0,853	0,956	0,940	0,910
none	IJ+2DS	0,886	0,974	0,987	0,958
none	IJ+Bm	0,956	0,980	0,983	0,964
none	2DS+Bm	0,860	0,931	0,913	0,914
none	IM+IJ+2DS	0,050	0,976	0,992	0,983
none	IM+IJ+Bm	0,906	0,979	0,984	0,973
none	IM+2DS+Bm	0,870	0,979	0,986	0,972
none	IJ+2DS+Bm	0,890	0,972	0,987	0,960
none	IM+IJ+2DS+Bm	0,043	0,979	0,990	0,979
PCA 0,02	IM	0,788	0,858	0,879	0,842
PCA 0,02	IJ	0,966	0,975	0,969	0,963
PCA 0,02	2DS	0,812	0,849	0,864	0,870
PCA 0,02	Bm	0,804	0,816	0,821	0,819
PCA 0,02	IM+IJ	0,945	0,911	0,978	0,963
PCA 0,02	IM+2DS	0,850	0,844	0,974	0,959
PCA 0,02	IM+Bm	0,873	0,895	0,931	0,914
PCA 0,02	IJ+2DS	0,856	0,858	0,972	0,952

PCA 0,02	IJ+Bm	0,961	0,979	0,970	0,965
PCA 0,02	2DS+Bm	0,822	0,856	0,882	0,882
PCA 0,02	IM+IJ+2DS	0,880	0,849	0,980	0,979
PCA 0,02	IM+IJ+Bm	0,938	0,903	0,982	0,971
PCA 0,02	IM+2DS+Bm	0,867	0,847	0,976	0,962
PCA 0,02	IJ+2DS+Bm	0,867	0,869	0,974	0,957
PCA 0,02	IM+IJ+2DS+Bm	0,877	0,850	0,985	0,976
LDA 2D	IM	0,873	0,872	0,875	0,866
LDA 2D	IJ	0,975	0,977	0,974	0,978
LDA 2D	2DS	0,910	0,914	0,911	0,915
LDA 2D	Bm	0,810	0,818	0,821	0,812
LDA 2D	IM+IJ	0,991	0,991	0,988	0,991
LDA 2D	IM+2DS	0,993	0,994	0,994	0,993
LDA 2D	IM+Bm	0,935	0,939	0,933	0,939
LDA 2D	IJ+2DS	0,996	0,996	0,995	0,997
LDA 2D	IJ+Bm	0,976	0,980	0,981	0,976
LDA 2D	2DS+Bm	0,927	0,928	0,927	0,920
LDA 2D	IM+IJ+2DS	0,998	0,999	0,998	0,999
LDA 2D	IM+IJ+Bm	0,991	0,992	0,989	0,990
LDA 2D	IM+2DS+Bm	0,994	0,994	0,993	0,992
LDA 2D	IJ+2DS+Bm	0,998	0,998	0,997	0,995
LDA 2D	IM+IJ+2DS+Bm	0,998	0,999	0,995	0,998
GDA 1D s4	IM	0,984	0,984	0,984	0,984
GDA 1D s4	IJ	0,986	0,987	0,986	0,987
GDA 1D s4	2DS	1,000	1,000	1,000	1,000
GDA 1D s4	Bm	0,822	0,818	0,825	0,817
GDA 1D s4	IM+IJ	1,000	1,000	1,000	1,000
GDA 1D s4	IM+2DS	0,999	1,000	1,000	1,000
GDA 1D s4	IM+Bm	0,998	0,998	0,998	0,998
GDA 1D s4	IJ+2DS	1,000	1,000	1,000	1,000
GDA 1D s4	IJ+Bm	0,991	0,991	0,990	0,990
GDA 1D s4	2DS+Bm	1,000	1,000	1,000	1,000
GDA 1D s4	IM+IJ+2DS	0,999	1,000	1,000	1,000
GDA 1D s4	IM+IJ+Bm	1,000	1,000	1,000	1,000
GDA 1D s4	IM+2DS+Bm	0,999	1,000	1,000	1,000
GDA 1D s4	IJ+2DS+Bm	0,999	1,000	1,000	1,000
GDA 1D s4	IM+IJ+2DS+Bm	0,999	1,000	1,000	1,000
GDA 1D s1	IM	0,999	1,000	1,000	1,000
GDA 1D s1	IJ	0,999	1,000	1,000	1,000
GDA 1D s1	2DS	0,902	0,895	0,894	0,913
GDA 1D s1	Bm	0,826	0,823	0,827	0,822
GDA 1D s1	IM+IJ	0,998	0,999	0,999	0,999
GDA 1D s1	IM+2DS	0,664	0,865	0,874	0,882
GDA 1D s1	IM+Bm	0,999	1,000	1,000	1,000
GDA 1D s1	IJ+2DS	0,644	0,849	0,879	0,887
GDA 1D s1	IJ+Bm	1,000	1,000	1,000	1,000
GDA 1D s1	2DS+Bm	0,711	0,899	0,894	0,917
GDA 1D s1	IM+IJ+2DS	0,687	0,853	0,862	0,881
GDA 1D s1	IM+IJ+Bm	0,997	0,998	0,997	0,998
GDA 1D s1	IM+2DS+Bm	0,669	0,629	0,623	0,891
GDA 1D s1	IJ+2DS+Bm	0,653	0,861	0,866	0,889
GDA 1D s1	IM+IJ+2DS+Bm	0,667	0,646	0,648	0,848

Table 5. F-Scores for all the MP3 file, using none or PCA, LDA and GDA feature reduction and the 4 types of considered classifiers.

Reduction	File	Bayes	SVM	RBF	FFNN
none	IM	0,775	0,919	0,903	0,882
none	IJ	0,952	0,973	0,985	0,970
none	2DS	0,852	0,908	0,887	0,899
none	Bm	0,808	0,829	0,830	0,821
none	IM+IJ	0,906	0,981	0,985	0,968
none	IM+2DS	0,862	0,980	0,980	0,969
none	IM+Bm	0,858	0,956	0,940	0,911
none	IJ+2DS	0,882	0,974	0,987	0,959
none	IJ+Bm	0,956	0,980	0,983	0,964
none	2DS+Bm	0,860	0,931	0,914	0,915
none	IM+IJ+2DS	0,508	0,976	0,992	0,983
none	IM+IJ+Bm	0,909	0,979	0,984	0,973
none	IM+2DS+Bm	0,865	0,979	0,986	0,972
none	IJ+2DS+Bm	0,887	0,972	0,987	0,961
none	IM+IJ+2DS+Bm	0,507	0,979	0,990	0,979
PCA 0,02	IM	0,788	0,866	0,877	0,842
PCA 0,02	IJ	0,965	0,975	0,968	0,963
PCA 0,02	2DS	0,808	0,836	0,872	0,874
PCA 0,02	Bm	0,802	0,826	0,830	0,827
PCA 0,02	IM+IJ	0,947	0,917	0,978	0,963
PCA 0,02	IM+2DS	0,845	0,827	0,974	0,958
PCA 0,02	IM+Bm	0,876	0,903	0,931	0,915
PCA 0,02	IJ+2DS	0,851	0,847	0,972	0,952
PCA 0,02	IJ+Bm	0,961	0,979	0,970	0,965
PCA 0,02	2DS+Bm	0,817	0,844	0,886	0,886
PCA 0,02	IM+IJ+2DS	0,876	0,834	0,980	0,979
PCA 0,02	IM+IJ+Bm	0,939	0,910	0,982	0,971
PCA 0,02	IM+2DS+Bm	0,862	0,830	0,976	0,962
PCA 0,02	IJ+2DS+Bm	0,862	0,858	0,974	0,956
PCA 0,02	IM+IJ+2DS+Bm	0,872	0,834	0,985	0,976
LDA 2	IM	0,875	0,873	0,875	0,867
LDA 2	IJ	0,975	0,977	0,974	0,978
LDA 2	2DS	0,912	0,917	0,914	0,917
LDA 2	Bm	0,809	0,828	0,830	0,818
LDA 2	IM+IJ	0,991	0,991	0,988	0,991
LDA 2	IM+2DS	0,994	0,994	0,994	0,993
LDA 2	IM+Bm	0,935	0,938	0,934	0,939

LDA 2	IJ+2DS	0,996	0,996	0,995	0,997
LDA 2	IJ+Bm	0,976	0,980	0,981	0,976
LDA 2	2DS+Bm	0,928	0,928	0,928	0,920
LDA 2	IM+IJ+2DS	0,998	0,999	0,998	1,000
LDA 2	IM+IJ+Bm	0,991	0,992	0,989	0,990
LDA 2	IM+2DS+Bm	0,994	0,994	0,993	0,992
LDA 2	IJ+2DS+Bm	0,998	0,998	0,997	0,996
LDA 2	IM+IJ+2DS+Bm	0,998	0,999	0,995	0,998
GDA 1 s1	IM	0,999	1,000	1,000	0,999
GDA 1 s1	IJ	0,999	1,000	1,000	1,000
GDA 1 s1	2DS	0,905	0,900	0,899	0,912
GDA 1 s1	Bm	0,832	0,832	0,832	0,827
GDA 1 s1	IM+IJ	0,998	0,999	0,999	0,999
GDA 1 s1	IM+2DS	0,539	0,871	0,876	0,887
GDA 1 s1	IM+Bm	0,999	1,000	1,000	1,000
GDA 1 s1	IJ+2DS	0,510	0,861	0,883	0,891
GDA 1 s1	IJ+Bm	1,000	1,000	1,000	1,000
GDA 1 s1	2DS+Bm	0,647	0,904	0,899	0,917
GDA 1 s1	IM+IJ+2DS	0,577	0,861	0,867	0,887
GDA 1 s1	IM+IJ+Bm	0,997	0,998	0,997	0,998
GDA 1 s1	IM+2DS+Bm	0,547	0,504	0,496	0,893
GDA 1 s1	IJ+2DS+Bm	0,523	0,870	0,872	0,892
GDA 1 s1	IM+IJ+2DS+Bm	0,549	0,523	0,522	0,857
GDA 1 s4	IM	0,984	0,984	0,984	0,984
GDA 1 s4	IJ	0,986	0,987	0,986	0,987
GDA 1 s4	2DS	1,000	1,000	1,000	1,000
GDA 1 s4	Bm	0,829	0,830	0,831	0,827
GDA 1 s4	IM+IJ	1,000	1,000	1,000	1,000
GDA 1 s4	IM+2DS	0,999	1,000	1,000	1,000
GDA 1 s4	IM+Bm	0,998	0,998	0,998	0,998
GDA 1 s4	IJ+2DS	1,000	1,000	1,000	1,000
GDA 1 s4	IJ+Bm	0,991	0,991	0,990	0,990
GDA 1 s4	2DS+Bm	1,000	1,000	1,000	1,000
GDA 1 s4	IM+IJ+2DS	0,999	1,000	1,000	1,000
GDA 1 s4	IM+IJ+Bm	1,000	1,000	1,000	1,000
GDA 1 s4	IM+2DS+Bm	0,999	1,000	1,000	1,000
GDA 1 s4	IJ+2DS+Bm	0,999	1,000	1,000	1,000
			1,000	1,000	1,000

Table 6. AUC scores for all the MP3 file, using none or PCA, LDA and GDA feature reduction and the 4 types of considered classifiers.