Homework 1: Instrument Classification

Yi-Hsuan Yang
Research Center for IT Innovation, Academia Sinica, Taiwan
yang@citi.sinica.edu.tw

1 Task Description

Instrument recognition/classification is a fundamental task in music information retrieval (MIR). In this assignment, we would like to implement a classifier for that. While there are many instruments in the world, we consider the following four in this homework: acoustic guitar, violin, piano, and human singing voice.

For the training set, you are given 200 audio examples for each instrument, totalling 800 clips. These files are put under different subfolders in the `audio` folder, so you know their groundtruth labels. You are also given 200 non-labeled clips for testing. They are also in the `audio` folder. All of these 1,000 clips are sampled at 16kHz. Each clip is associated with only one instrument.

The task is to build a multi-class classifier from the training set to discriminate the four instruments, and then to apply the classifier to the test set.

In particular, you are asked to address the following questions. The number in the braces indicates the weight of that particular question to your overall grade for this homework. We provide you some example Matlab codes for you to start the programming. It's fine if you want to program in Python. You will have to write a report to describe how you address these questions.

- **Q1** (10%): Randomly pick one audio clip and compute the spectrogram with the following two settings: 1) window size = 1,024 samples, hop size = 512 samples, 2) window size = 2,048 samples, hop size = 1,024 samples. Plot the resulting two spectrograms and discuss their differences.
- **Q2** (10%): Randomly pick two audio clips of different instruments and compute their spectrograms using window size = 1,024 samples, hop size = 512 samples. Plot the two spectrograms and discuss their differences. What are the possible features we can use to discriminate the two instruments?
- **Q3** (20%): Follow the example Matlab code and train an SVM classifier based on the MFCC features. Report the accuracy obtained for the validation set and show the 4×4 confusion table. What do you see from the confusion table? (Hint: You can use 20% of the training set for validation and the remainders for training, as shown in the example Matlab code. Or, you can also use cross validation here and report the confusion table summing from the different folds, if you like).
- **Q4** (10%): Test the effect of feature normalization. What's the classification accu-

racy you get if you do not perform feature normalization?

- **Q5** (10%): Test the effect of parameter tuning. For RBF-kernel SVM, we need to tune the values of the parameters C and γ using the validation set. Try different values of C and γ when training your MFCC-based SVM classifier. What's the best and worst accuracy you get? (Hint: We only want to let you know that parameter tuning is important. It's good enough to use a small search range for the values for C and γ, like those shown in the example Matlab code.)

- **Q6** (40%): Use your classifier to predict the class of the audio clips in the test set, and then store the prediction result in csv format. Use your creativity and anything you learned from the course to make the classifier better. For example, you can try to use features other than MFCC, and other classification algorithms. (Even if you do not try any other features or classification algorithms, you will still need to submit the prediction result anyway.) Describe what you have attempted to improve the classifier in the report and the things or insights you have learned in these experiments. We will let you know the classification accuracy you get for the test set. It's good if you can obtain good accuracy, but classification accuracy is not everything — we won't judge your efforts only based on the classification accuracy. We are interested in what you have tried and have learned from these attempts.

2 Submission

The deadline of this homework is March 28, 2016 (Monday), 23:59pm Taiwan time. Your score will be deducted by 20 (e.g. 75→55) for being one day late, deducted by 40 for being two days late, and by 60 after that. Plagiarism is surely not allowed at all.

We will spend some time discussing this homework on March 31, during the last hour of the course. We will invite 2 or 3 of you to give a 5-min presentation of what you did. The presentation can be based on the report you write so no need to prepare slides.

Submit your report, along with source codes and prediction result of the test set, to my email address (yang@citi.sinica.edu.tw). Please zip all these files into a single zip file (hopefully a small file), and please use "HW1 [your ID]" as the title of the mail, so as to reduce the chance that I miss the mail. No need to send us the features you computed.

The report can be written in either English or Chinese, though the former is preferred. Please care about readability of your report and make it easy for us to evaluate what you have done to address each of these questions. There is no need to copy and paste your codes to the report, unless there is really something special in the codes that you want to share with us. The writing should be clear and concise, and the results should be presented in a systematic way.

The report will be evaluated in terms of technical strength (e.g. your familiarity with those we taught in the class), novelty (e.g. interesting ideas or findings), and in part readability (e.g. how you document the results).

We won't necessarily look into your source codes, unless there is a concern of plagiarism. But, please also include a simple readme to let us know how to run your codes if we want.

3 Bonus Task: Instrument Tracking

For those of you who are interested, we prepare a bonus task for you. It's a bonus so it's up to you whether you want to do it. As an incentive the bonus task may add up to 20 to your score of this homework.

The bonus task is about instrument tracking, which is arguably more important in real-life applications than instrument classification. For instrument tracking, we want to predict frame-by-frame the occurrence of an instrument. Moreover, there can be multiple instruments at the same time (e.g. guitar, bass, drums, and vocal), making it a multi-label instead of a multi-class classification problem. There will also be *unseen* instruments that are not in your training set. It's more challenging but also more interesting.

Given that the high variability of human voice, we consider only the following three instruments in this task: acoustic guitar, violin, piano. We prepare a validation set and a test set for this task in the `#bonus` folder.

There are 16 audio clips for validation and 12 clips for testing. The sampling rate of these files is also 16kHz. The groundtruth frame-level class labels of the validation set are available to you. You will have to submit your prediction of the test set to us.

The groundtruth labels are frame-level labels for a hop size of 512 samples. Therefore, you also need to use 512 samples as your hop size when computing the features. The groundtruth labels for a clip is a T by 3 matrix (stored in csv format), where the vertical axis corresponds to different frames, and the horizontal axis corresponds to acoustic guitar, violin, piano, respectively. The values are either 0 or 1. For example, if a row reads [0, 1, 1], it means we can hear violin and piano for that frame.

The tracking task is a multi-label classification task. Moreover, the prediction is to be made per frame, not per song, so pooling is not needed. Some more hints:

- Use the 800 audio clips you have from the instrument classification task to train three binary classifiers, one for each target instrument (i.e. not including voice).
- For a given target instrument, the classifier is trained by using the audio samples of the instrument as the positive examples, and the audio samples from the other instruments as the negative examples. You can include the clips of singing voice as the negative examples for all the three binary classifiers.
- To avoid class imbalance, you might want to sample the negative examples such that the numbers of positive and negative examples are the same.
- The classifiers are to be trained on the frame-level. Therefore, we need to use frame-level features now, instead of pooling them.
- There can be too many frame-level feature vectors. For efficiency, it may be good enough to randomly sample a few frame-level feature vectors in training.
- While the performance metric considered in the instrument classification task is classification accuracy, the performance metric considered in the instrument tracking task is precision, recall, and f-score for each of the three target instruments.
- Actually listen to the audio clips may help you gain some ideas.

Please describe your approach(es) and findings in the report.