Homework 3: Source Separation

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

1 Task Description

This assignment invites you to gain some hands-on experience in music source separation, using the supervised nonnegative matrix factorization (NMF) approach. The goal is to separate violin and clarinet sounds from their mixtures.

For the training set, you are given clean source signals of single notes of the violin and the clarinet, in the `train` sub-folder of the `audio` folder. There are 45 clips for the violin (from MIDI number 55 to 99; in the `vio` sub-folder), and 40 clips for the clarinet (from MIDI number 50 to 89; in the `cla` sub-folder). See the following link if you need more information about MIDI numbers: `https://newt.phys.unsw.edu.au/jw/notes.html`.

The validation set in the `validation` sub-folder contains the groundtruth source signals of five 5-second violin clips (e.g. '01_vio.wav'), clarinet clips (e.g. '01_cla.wav') and their mixtures (e.g. '01_mix.wav'). The goal of source separation is to recover the source signals of the two instrument from a mixture signal. You can use the validation set to compare the performance of different algorithms.

The test set in the `test` sub-folder contains another five mixtures of violin and clarinet signals. You will have to recover the sources, name the recovered WAV files as for example '06_vio_est.wav' and '06_cla_est.wav', and put them in the `pred` folder, which is empty now but is supposed to have 10 files after you complete the assignment.

In particular, you are asked to address the following questions, which are divided into *basic*, *required*, and *advanced* ones. The number in the braces indicates the weight of that particular question to your overall grade for this homework. There is no bonus task this time, but the maximal grade you can receive is 115, instead of 100.

Again, it's fine to program in either Matlab or Python, and you will have to write a report to describe how you address these questions.

2 Basic Questions

B1: BSS Eval (10%)

This question leads you to get familiar with the BSS Eval toolbox (`http://bass-db.gforge.inria.fr/bss_eval/`) for assessing the performance of source separation. Here, we are going to use three audio files from the `validation` set: '01_vio.wav,' '01_cla.wav,' and '01_mix.wav.' First, use Matlab (or Python) to load the audio files

- a = wavread('01_vio.wav');
- b = wavread('01_cla.wav');
- c = wavread('01_mix.wav');
- n = randn(length(a),1); % normally distributed pseudorandom numbers

Then, compute the source-to-distortion ratio (SDR) for the following inputs and discuss your findings.

- SDR = bss_eval_sources([c'; c']/2,[a'; b']);
- SDR = bss_eval_sources([a'; b'],[a'; b']);
- SDR = bss_eval_sources([b'; a'],[a'; b']);
- SDR = bss_eval_sources([2*a'; 2*b'],[a'; b']);
- SDR = bss_eval_sources((a+0.01*n)',a');
- SDR = bss_eval_sources((a+0.1*n)',a');
- SDR = bss_eval_sources((a+n)',a');
- SDR = bss_eval_sources((a+0.01*b)',a');
- SDR = bss_eval_sources((a+0.1*b)',a');
- SDR = bss_eval_sources((a+b)',a');

B2: Inverse STFT (15%)

This time, we want to compute the STFT of 'vio_64.wav' from the `train` set, manipulate the spectrogram, and then generate the time-domain signal using inverse STFT (ISTFT). Specifically, use whatever window size and hop size you like to compute the STFT of 'vio_64.wav' and then do the following things:

1. set all the frequency components *under* 1,200 Hz to zero in the spectrogram, and then create the time-domain signal by ISTFT (this is equivalent to applying a *high-pass* filter with cutoff frequency 1,200 Hz). Save the file as 'vio_64_hp.wav.'
2. set all the frequency components *over* 1,200 Hz to zero in the spectrogram, and then create the time-domain signal by ISTFT (this is equivalent to applying a *low-pass* filter). Save the file as 'vio_64_lp.wav.'
3. plot the spectrogram of 'vio_64.wav,' 'vio_64_hp.wav,' and 'vio_64_lp.wav' and discuss their differences.
4. listen to the three files and discuss their differences.
5. compute the SDR between 'vio_64.wav' and 'vio_64_lp.wav,' by using the former one as the true source and the latter as the estimated source.

Please note that you do no have to send me your 'vio_64_lp.wav' and 'vio_64_hp.wav.' Just write in the report how you address the question and report your findings.

B3: NMF (15%)

Let's address the following three sub-questions using the Euclidean distance as the cost function for NMF and random initialization for 'W' and 'H.' First, use NMF to learn three templates from the music clip 'vio_64.wav' with whatever window size and hop size you like (e.g. window size 2,048 samples and 50% overlaps). What do the three templates look like in the spectrum? Do they correspond to the pitch of 'vio_64.wav'? Moreover, use ISTFT to get the time-domain signals of the three templates and plot them.

Second, use NMF to learn another three templates from 'vio_88.wav' and compare these templates with the templates learned from 'vio_64.wav.'

Finally, use NMF to learn another three templates from 'cla_64.wav' and compare these templates with the templates learned from 'vio_64.wav.'

3 Required Questions

R1: Source Separation: vio→vio (15%)

We are going to use both the `train` and `validation` sets in this question. Let's focus on only the violin first. You are asked to do the following things:

1. learn three templates for each single note of the violin in the `train` set by NMF, using the Euclidean distance as the cost function. This should give you in total 135 templates. Let's call these 135 templates the *violin dictionary*.
2. plot the violin dictionary and discuss the result (you are supposed to see a figure similar to the one shown in the middle of page 62 of the slides 'lecture09_separation.pdf').
3. use the *violin dictionary* as the pre-trained templates to decompose the file '01_vio.wav' in the `validation` set (i.e. fix the 'W' and compute the 'H').
4. plot the activation matrix 'H' and discuss what you see.
5. create the WAV file from the reconstruction result (i.e. 'W*H') using ISTFT, listen to it, and discuss whether it sounds like the input '01_vio.wav.'

R2: Source Separation: vio→cla (10%)

Let's investigate whether the dictionary for one instrument can be applied to signals of another instrument.

1. use the same *violin dictionary* as the pre-trained templates but this time decompose the file '01_cla.wav' in the `validation` set (i.e. fix the 'W' and compute the 'H').
2. plot the activation matrix 'H' and discuss what you see.
3. create the WAV file from the reconstruction result (i.e. 'W*H') using ISTFT, listen to it, and discuss whether it sounds like the input '01_cla.wav.'

R3: Source Separation: Validation (10%)

After the warm up, we are ready to actually implement a source separation algorithm.

1. learn three templates for each single note of the clarinet in the `train` set using NMF; this should give you in total 120 templates. Let's call these 120 templates the *clarinet dictionary*.
2. use the violin dictionary and the clarinet dictionary together as the pre-trained templates and decompose the file '01_mix.wav' in the `validation` set (i.e. fix the 'W' and compute the 'H').
3. create the WAV file from the reconstruction result using ISTFT, save them as '01_vio_est.wav' and '01_cla_est.wav.'
4. compute the SDRs, SIRs, and SARs by using '01_vio.wav,' '01_cla.wav,' '01_vio_est.wav' and '01_cla_est.wav' as inputs to BSS Eval.
5. repeat the above for all the other four clips ('02,' '03,' '04' and '05') and report the SDRs, SIRs, and SARs for each instrument each clip.

R4: Source Separation: Test (20%)

Use pre-trained violin and clarinet dictionaries to decompose the five clips in the `test` set, and put the estimated sources (e.g. '06_vio_est.wav') in the `pred` folder. Because there are 5 clips, there should be in total 10 files after you complete the assignment.

You may want to listen to the estimated sources yourself to have a sense of how good the separation is. Please feel encouraged to use the validation set to experiment with other settings of NMF, such as the number of templates per note, the cost function (e.g. the KL divergence), initialization method, the widow size and hop size of STFT, and the use of a binary or soft mask for Wiener filtering, etc. Describe your investigations in detail in the report and send us the audio files of the best result you have for the `test` set.

4 Advanced Questions

A1: Score-informed NMF (10%)

You can find something like the musical score of the five mixtures of the `validation` set in the folder `score-info`. Each row corresponds to the information of a note, in the format [onset time, offset time, MIDI number, instrument ID]. The onset and offset times specify the starting and ending points of a note, in million seconds. The instrument ID is either 1 (violin) or 2 (clarinet). For the last few notes the offset time is after 5 seconds, which is beyond the length of the clips you have. You can just set them to 5 seconds.

Use these files to implement a score-informed NMF algorithm by setting constraints on the activation (i.e. by properly initializing 'H') and redo question R3.

A2: Pitch-informed NMF (10%)

You are not given the musical score of the five mixtures in the `test` set, which is a practical situation we encounter in many real-world music signals. However, we can use pitch detection algorithms introduced in the class to estimate the pitch from the audio, and treat the *pitch estimate* as the *pseudo musical score* to inform NMF. A pitch detection algorithm will not let you know the instrument labels of the notes, so you will have to consider both the violin and clarinet templates for all the estimated notes.

Try some pitch detection algorithms and implement such a pitch-informed NMF for the `test` set. Describe your endeavors and findings in the report. If you want, you can submit your estimated sources in a separate folder named as `pred_score`.

5 Submission

The deadline of this homework is May 9, 2016 (Monday) 23:59pm Taiwan time, and there will be a discussion session on May 12. We will also invite some of you to present your solutions in the discussion session. The scoring policy is the same as that in HW1.

Submit your report, along with source codes and estimated sources of the `test` set, to yang@citi.sinica.edu.tw. Please name the estimated sources according to the way specified above (e.g. '06_vio_est.wav') and put them in the `pred` folder. Remember to include a readme to let us know how to run your codes if we want. Compress all the files into a single zip file and name the zip file simply as '[your ID].' Finally, use 'HW3 [your ID]' as the title of the mail, so as to reduce the chance that we miss the mail.