CS 280
Programming Assignment 4
Cocktail Party Problem

Instructor: Pros Naval

Individual Submissions
Due: December 7, 2018

1. Introduction

Independent Component Analysis (ICA) is a statistical signal processing method for finding independent signal sources given only measurements of observed data that are mixtures of the unknown sources without any prior knowledge of the mixing mechanism.

The cocktail party problem is a classic example for demonstrating usefulness of ICA. In this programming assignment, you are to write code in Python or Matlab/Octave that will accept as inputs several sound sources that contain a mixture of speech and music and unmix them into their independent components.

2. Independent Component Analysis

Let $x_1(t), x_2(t), ..., x_m(t)$ denote the m sensor (microphone) measurements of source signals $s_1(t), s_2(t), ..., s_n(t)$ where $m \leq n$. Construct matrices $X = [x_1, x_2, ..., x_m]^T$ and $S = [s_1, s_2, ..., s_n]^T$ whose rows represent $x_1(t), x_2(t), ..., x_m(t)$ and $s_1(t), s_2(t), ..., s_n(t)$ respectively. The ICA model assumes that the observed signals X are linear transformations of the latent source signals S, i.e.,

$$x_i = \sum_{j=1}^{n} a_{ij} s_j \qquad i = 1, ..., m$$

or more compactly

$$X = AS$$

where the mixing matrix A is an $m \times n$ matrix with unknown coefficients a_{ij}. Furthermore, the source signals are assumed to be mutually statistically independent and non-gaussian.

The ICA solution is obtained in an unsupervised learning process that computes the de-mixing matrix $W = A^{-1}$ which is used to unravel the independent component signals Y from the mixture signals X:

$$Y = WX$$

Independent Component Analysis is formulated as an optimization problem by setting up some measure of statistical independence of the non-gaussian independent components for its objective function with the aim of solving for the de-mixing matrix W. Non-gaussianity is quantified by the negentropy

$$J(y) = H(y_{gaussian}) - H(y)$$

where $y_{gaussian}$ is a Gaussian random vector with the same covariance matrix as y and H is the entropy of the random vector y with density $p(y)$ defined as $H(y) = -\int p(y) \log p(y) dy$. Due to the difficulty in computing for the negentropy, the following approximation is often used:

$$J(y) \approx [E\{G(y)\} - E\{G(v)\}]^2$$

where v is the zero mean and unit variance standard Gaussian variable and y is a random variable with zero mean and unit variance. Suitable choices for the contrastive $G(y)$ include

$$\begin{aligned} G_1(y) &= \frac{1}{a_1} \log\cosh(a_1 y) \\ G_2(y) &= \frac{1}{a_2} \exp(-a_2 y^2/2) \\ G_3(y) &= \frac{1}{4} y^4 \end{aligned}$$

where $1 \leq a_1 \leq 2, a_2 \approx 1$ are constants.

The choice of $G(y)$ is dictated by the following:

- G_1 is a general-purpose contrastive function
- G_2 is suitable when the independent components are super-Gaussian or when robustness is very important
- G_3 is useful only for estimating sub-Gaussian independent components when there are no outliers

Piecewise linear approximations for G_1 and G_2 may be used in cases where computational overhead must be reduced. Super-(sub-) Gaussian densities are densities with positive (negative) kurtosis. The kurtosis of a random variable y is defined as $kurt(y) = E\{y^4\} - 3(E\{y^2\})^2$ which simplifies to $E\{y^4\} - 3$ if y has unit variance.

Preprocessing steps for ICA models involves the following

1. Centering: the input x_i is centered by subtracting from it the mean:

   $$x_i \leftarrow x_i - E(x_i)$$

2. Whitening: the matrix with zero mean X is passed through the whitening matrix V to remove the second order statistic of the input matrix:

   $$Z = VX$$

   where the whitening matrix is the inverse square root of the covariance matrix of the X, i.e.,

   $$V = 2(C_X)^{-1/2}$$

   where $C_X = E[XX^T]$ is the covariance matrix of X. The rows of the whitened matrix Z denoted by z are uncorrelated and have unit variance, i.e., $E[zz^T] = I$

3. Unmixing Speech and Music

In this Programming Assignment, you are to write Matlab/Octave or Python code that unravels the independent components from five audio sources. FastICA codes are available at http://research.ics.aalto.fi/ica/fastica/

The script should perform the following:

1. Load the audio files `mic1.wav` to `mic5.wav` found in the folder `Audio_Data`. These files are synchronized audio recordings captured by five microphones positioned at five different locations.

2. Form the mixture matrix X from the input files.

3. Invoke the appropriate ICA command that ”unmixes” the five independent components from the mixture of audio signals. Experiment on the following:

   - appropriate input sampling rate
   - whether centering is required
   - whether whitening is necessary
   - appropriate contrastive function $G(y)$

4. Save the independent components $\hat{s}_i(t), i = 1, ..., 5$ as audio files in wav format. Label them as `shat[1-5].wav`.

5. Reconstruct the mixture signals $\hat{x}_i(t),\ i = 1, ..., 5$ and measure the residuals for each one. Print out the residual values.

6. Save the reconstructed mixture signals as audio files in wav format. Label them as `recon[1-5].wav`.

4. Deliverables

Deliverables for this Programming Assignment:

1. Octave/Matlab or Python script

2. Unmixed audio files

3. Reconstructed audio files

4. Describe your efforts in the determination of the following:

   (a) What is the sampling rate for the input files ?

   (b) Is centering necessary ?

   (c) Is whitening required ?

   (d) What is the appropriate contrastive function $G(y)$?

   (e) What are the residuals of the reconstructed mixture signals ?

The deadline for submission is **December 7, 2018**. Email Octave/Matlab or Python script and audio files to `submit2pcnaval@gmail.com` with ”[CS280: PA4 Submission] Your Name ” on the subject line. **Do not email DropBox/GoogleDrive/etc. links**.