

### Independent components analysis

While studying Independent Component Analysis (ICA) in class, we made an informal argument about why Gaussian distributed sources will not work. We also mentioned that any other distribution (except Gaussian) for the sources will work for ICA, and hence used the logistic distribution instead. In this problem, we will go deeper into understanding why Gaussian distributed sources are a problem. We will also derive ICA with the Laplace distribution, and apply it to the cocktail party problem.

Reintroducing notation, let  $s \in \mathbb{R}^n$  be source data that is generated from  $n$  independent sources. Let  $x \in \mathbb{R}^n$  be observed data such that  $x = As$ , where  $A \in \mathbb{R}^n$  is called the *mixing matrix*. We assume  $A$  is invertible, and  $W = A^{-1}$  is called the *unmixing matrix*. So,  $s = Wx$ . The goal of ICA is to estimate  $W$ . Similar to the notes, we denote  $w_j^T$  to be the  $j^{\text{th}}$  row of  $W$ . Note that this implies that the  $j^{\text{th}}$  source can be reconstructed with  $w_j$  and  $x$ , since  $s_j = w_j^T x$ . We are given a training set  $\{x^{(1)}, \dots, x^{(m)}\}$  for the following sub-questions. Let us denote the entire training set by the design matrix  $X \in \mathbb{R}^{m \times n}$  where each example corresponds to a row in the matrix.

(a) **Gaussian source**

For this sub-question, we assume sources are distributed according to a standard normal distribution, i.e  $s_j \sim \mathcal{N}(0, 1), j = \{1, \dots, n\}$ . The likelihood of our unmixing matrix, as described in the notes, is

$$\ell(W) = \sum_{i=1}^m \left( \log |W| + \sum_{j=1}^n g'(w_j^T x^{(i)}) \right),$$

where  $g$  is the cumulative distribution function, and  $g'$  is the probability density function of the source distribution (in this sub-question it is a standard normal distribution). Whereas in the notes we derive an update rule to train  $W$  iteratively, for the cause of Gaussian distributed sources, we can analytically reason about the resulting  $W$ .

Try to derive a closed form expression for  $W$  in terms of  $X$  when  $g$  is the standard normal CDF. Deduce the relation between  $W$  and  $X$  in the simplest terms, and highlight the ambiguity (in terms of rotational invariance) in computing  $W$ .

**Answer:**

(b) **Laplace source.**

For this sub-question, we assume sources are distributed according to a standard Laplace distribution, i.e  $s_i \sim \mathcal{L}(0, 1)$ . The Laplace distribution  $\mathcal{L}(0, 1)$  has PDF  $f_{\mathcal{L}}(s) = \frac{1}{2} \exp(-|s|)$ . With this assumption, derive the update rule for a single example in the form

$$W := W + \alpha(\dots).$$

**Answer:**

(c) **Cocktail Party Problem**

For this question you will implement the Bell and Sejnowski ICA algorithm, but assuming a Laplace source (as derived in part-b), instead of the Logistic distribution covered in class. The file `mix.dat` contains the input data which consists of a matrix with 5 columns, with each column corresponding to one of the mixed signals  $x_i$ . The code for this question can be found in `p4_ica.py`.

Implement the `update_W` and `unmix` functions in `p4_ica.py`.

You can then run `p4_ica.py` in order to split the mixed audio into its components. The mixed audio tracks are written to `midex.i.wav` in the output folder. The split audio tracks are written to `split.i.wav` in the output folder.

To make sure your code is correct, you should listen to the resulting unmixed sources. (Some overlap or noise in the sources may be present, but the different sources should be pretty clearly separated.)

Note: In our implementation, we **anneal** the learning rate  $\alpha$  (slowly decreased it over time) to speed up learning. In addition to using the variable learning rate to speed up convergence, one thing that we also do is choose a random permutation of the training data, and running stochastic gradient ascent visiting the training data in that order (each of the specified learning rates was then used for one full pass through the data).