Homework 3: Independent Component Analysis

Robert Grant
2009-09-20

1 The Assignment

In this homework, we were to implement Independent Component Analysis (ICA) using the algorithm presented in the homework document. Briefly, we were to generate a matrix of received signals X that is the linear combination of several source signals from a matrix U. Using gradient ascent, our algorithm attempts blind source separation by maximizing entropy and generating an estimate of U which we call Y.

2 Implementation

My implementation is contained in three files: `hw3Main.m`, `hw3ICA.m`, and `hw3HasConverged.m`. The core ICA algorithm is implemented in the function file `hw3ICA.m` which calls the function file `hw3HasConverged.m` after every iteration to determine if it has converged or diverged. The script `hw3Main.m` runs experiments by calling the `hw3ICA.m`. Due to space limitations, please reference the extensive comments at the top of these files more more details.

3 Results

First, I reproduce the results for the small example test set in Figure 1. In this graph, as in the graph in the assignment, I ran the ICA algorithm on the test set for 1e6 runs, with $\eta = 0.01$. From the bottom up, the signals shown are plots of the three rows from U, X, and Y, also known as the source signals, the linearly combined signals, and the separated signals. As you can see, the top three signals closely match the bottom three. As expected, due to limitations in the algorithm, they are not in the correct order and one is sign-reversed.

In that test, I simply ran the algorithm for 1e6 runs. To determine convergence (and have a chance of finishing early), I came up with a metric expressing the amount of change in the matrix W as a scalar.

$$c = \sqrt{}(max(max(dW^2))), \tag{1}$$

where c is the absolute value of the element of W that has changed the most between runs. In Figure 2, I show a log-log plot of the value of c over the 1e6 runs of ICA. This plot shows that the matrix W is changing more and more slowly over time, indicating that the algorithm is converging.

In Figures 3 and 4 I show the convergence and results of mixing signals from the provided `sounds.mat` dataset and then trying ICA on them. As can be seen in `hw3main.m`, I combined signals 1, 3, and 4, which are (respectively) Homer Simpson talking, a crazy laugh, a crowd clapping, and a crazy laugh. The graphs (produced after my metric dropped below 1e-10 show that the algorithm was converging, and that it was starting to produce signals that looked very like the originals. This experiment was run with $R_{max} = 1e6$ and $\eta = 0.0001$.

To reproduce my results graphs, simply run `hw3Main` from the MATLAB interactive command prompt in the proper directory. This will not reproduce my convergence graphs, as these rely on plotting from within

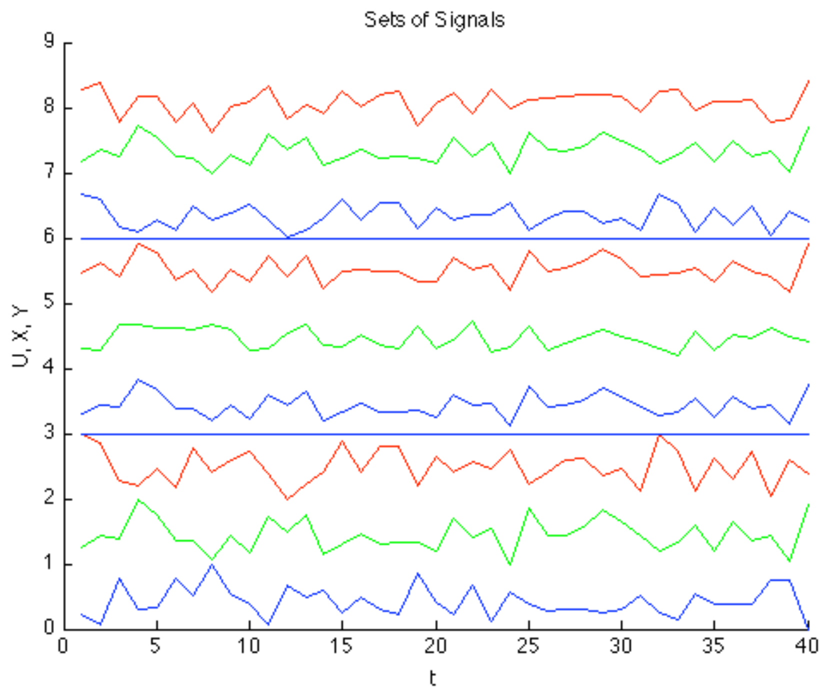

Figure 1: Reproduction of test results.

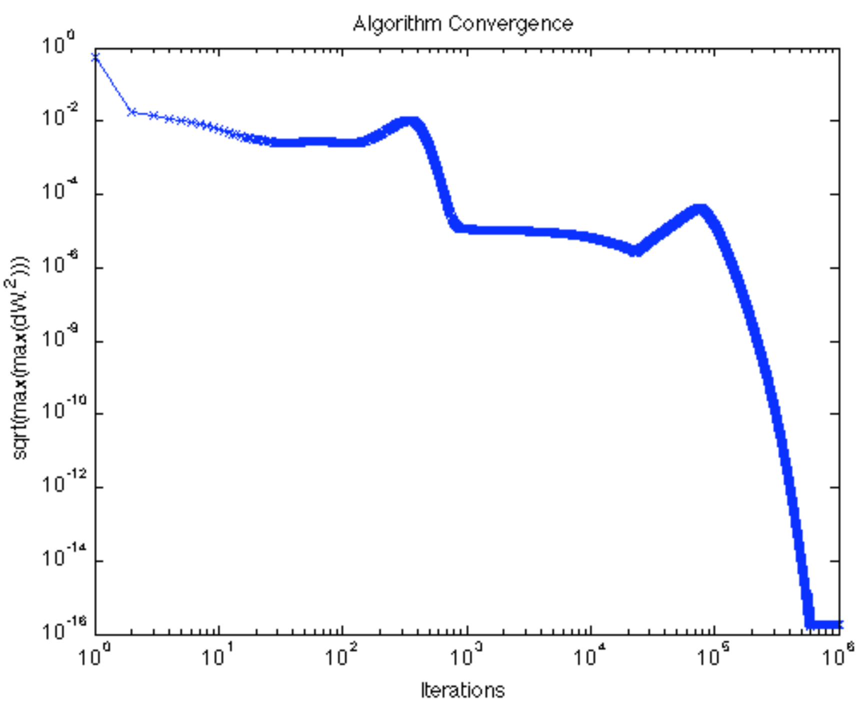

Figure 2: Convergence of example metric on test results.

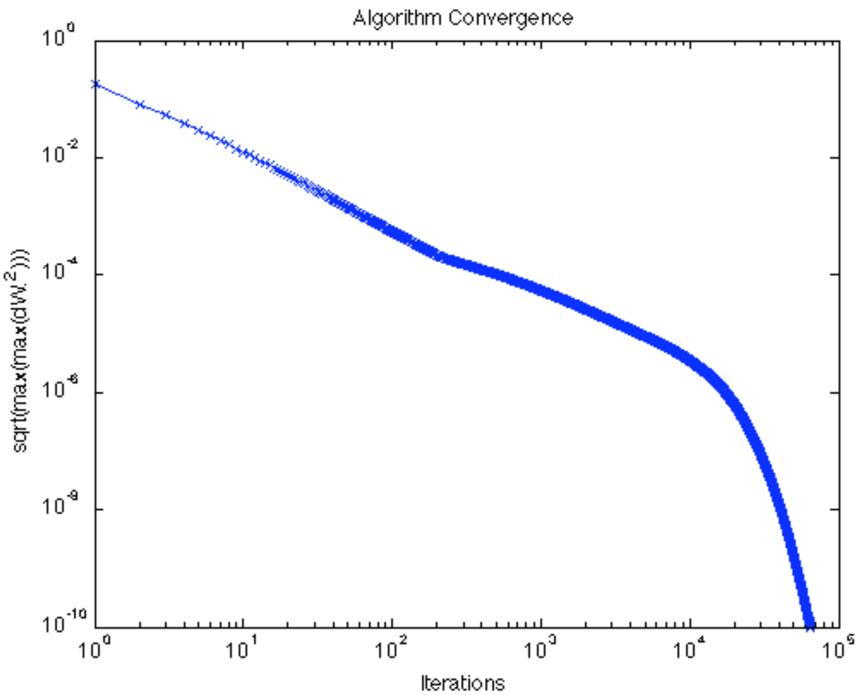

Figure 3: Convergence of metric on real signals.

`hw3ICA.m`, which I have commented out as per instructions for my submission.

4 Other Comments

In my experiments, I noticed that the algorithm seems to be very sensitive to the learning rate η. If η was too large, the algorithm would diverge quickly. If it was instead too small, it would not produce good results in a reasonable amount of iterations. Also, more iterations seem to be better, as far as I went.

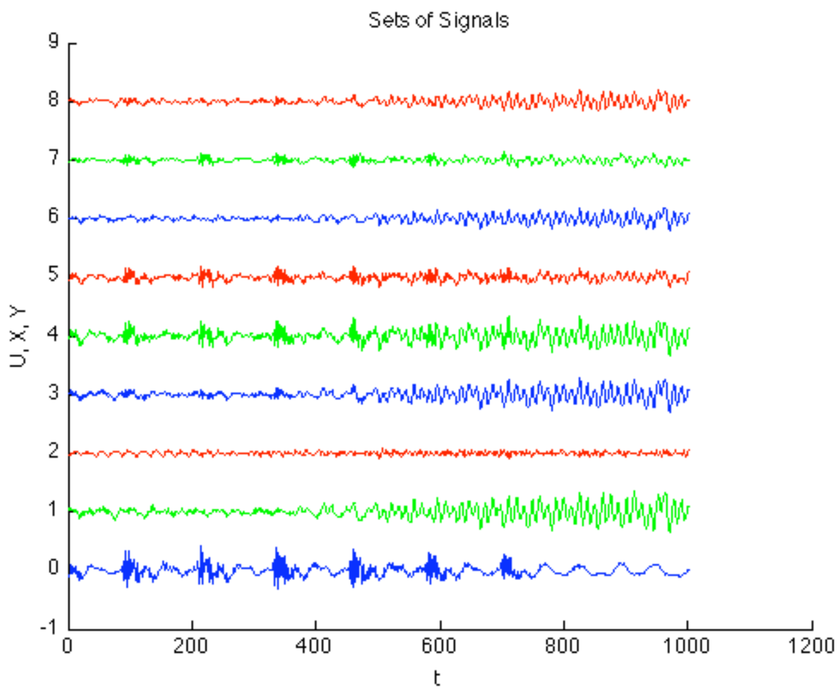

Figure 4: Results from section of real signal separation.