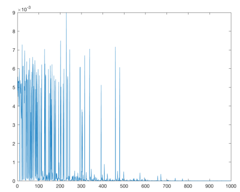
Homework 3; Psych 186B, Winter 2018

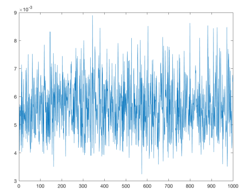
Marshall Briggs

6 Februrary 2017

For the following parts A-D, I exacted the following procedure to obtain my results:

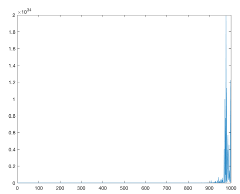
1. I generated two sets of n = 1, 20, 40, 60, 80, 100, ... normalized, length=1 vector pairs, f and g, each f_i-g_i representing a single vector pair.
2. I used those vectors to build the linear associator A.
3. I then tested the distribution of g', to make sure I constructed my vectors and associator correctly.
4. Next, I performed n * 10 learning trials with procedurally chosen f vectors (randomly in parts A-C, consecutively in part D), and applied the Widrow-Hoff algorithm during each trial. That is, I computed the square of the error between g' and g, and applied a gradient descent technique with that error result to the linear associator.
5. I then plotted the number of each trial with the length of the squared error for that trial, to quantify how successfully the associator was able to learn over time.

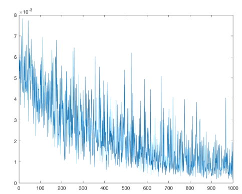




(a) (b)

$$k_i = \frac{1}{(f(:, vi)' * f(:, vi))} - \epsilon \qquad k_i = \frac{2}{(f(:, vi)' * f(:, vi))} - \epsilon$$





(c) (d)

$$k_i = \frac{5}{(f(:, vi)' * f(:, vi))} - \epsilon \qquad k_i = \frac{\frac{1}{9}}{(f(:, vi)' * f(:, vi))} - \epsilon$$

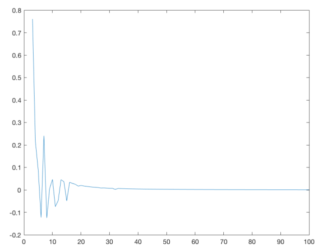
Figure 1: Oscillation of supervised learning algorithm, at different values of k_i

A. Oscillations

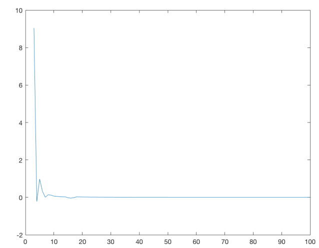
Oscillations in the linear associator A's predictive ability can be measured with the length of the squared error vectors. Given our knowledge of the Widrow-Hoff learning algorithm, we would expect that our linear associator would become more accurate as the number of learning trials increase.

After examination of the results, we discover that this measurement of accuracy is *highly* dependent on the value we choose for k. Thus, we see that (in figure (a), (b), and (c)), for the numerators of k_i= 1, 2, 5 respectively, as k increases, the oscillation of sq error length is seen to increase. In figure (a), $\overline{e^2}$ is seen to converge on $\overline{e^2} = 0$. In figure (b), $\overline{e^2}$ does not seem to converge at all, and rather just oscillates up and down. In figure (c), the oscillations of $\overline{e^2}$ seem to increase as the trial number increases, to infinity.

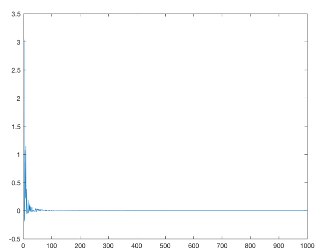
In contrast, in figure (d) the numerator of k_i is decreased. It seems possible that this $\overline{e^2}$ may eventually converge to $\overline{e^2} = 0$, but I included this figure to further illustrate that decision of k_i has significance on both whether $\overline{e^2}$ converges at all, and if so, on how quickly this happens..



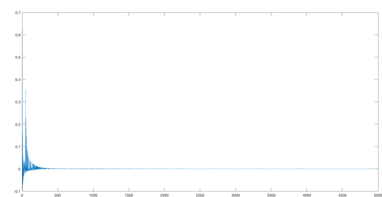
(a) number of pairs = 10, converge point t ≈ 26



(b) numpairs = 50, convergence point t 65



(c) numpairs = 100, convergence point t ≈ 90



(d) numpairs = 500, convergence point t ≈ 500

Figure 2: Convergence of supervised learning algorithm, at different numbers of learned pairs

B. Convergence

To measure convergence, I used a method similar to log-likelihood convergence, but less complicated. Before each learning trial, I measured the mean of the percentage differences in $\overline{e^2}$ so far. Then, I computed the percentage difference from the mean $\overline{e^2}$: $abs|\dfrac{(\overline{e^2}[i] - mean(\overline{e^2}))}{mean(\overline{e^2}))}|$, and measured the new mean of the percentage differences seen so far (including the current learning trial). Then, I computed the percentage difference between the two percentage mean differences, to quantify the impact of the current trial on the overall mean percentage differences between each trial and the mean of the trials seen so far.

In other words, as the number of trials increases, if $\overline{e^2}$ converges as expected, the %Δ between the post-learning mean difference and the pre-learning mean difference should converge to 0, i.e., *each consecutive trial should have less and less of an impact on the overall mean % difference between trials.* At this point, $\overline{e^2}$ can be said to have converged. For my system, the convergence point appears to be around t ≈ numpairs.

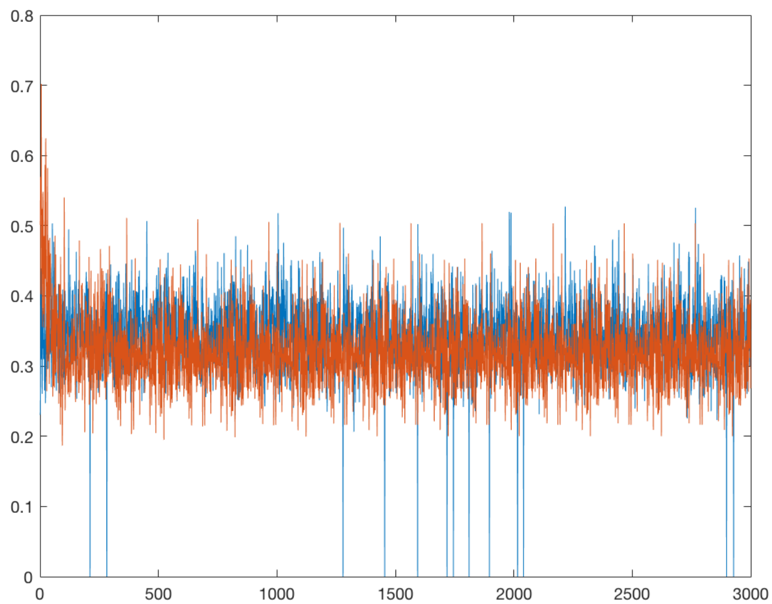
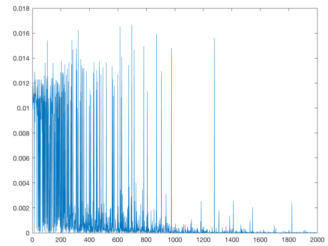


Figure 3: Performance of Widrow-Hoff algorithm (red line) vs. performance of an unsupervised linear associator pairing randomly (blue line). numpairs = 300, dimensions = 100

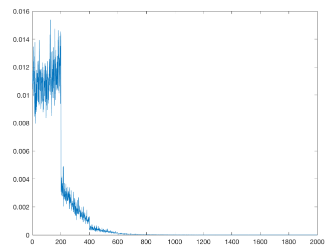
C. Breakdown

I tested many different dimensions and across many values for the number of vector pairs learned by the linear associator and I had difficulty seeing the Widrow-Hoff algorithm ever perform worse than random chance. Perhaps I wasn't using the correct method to simulate randomness. My method was to pair f_i with a random g_j and measure the length of the sq error.

In figure 3, I've shown that, under certain circumstances numpairs = 300 and dimensions = 100, supervised learning may perform just as poorly as random chance. My other experiments seem to show that, as dimensionality of f and g increase, the supervised linear associator becomes more accurate.



(a) Number of learning trials vs. performance of Widrow-Hoff, with random trial selection



(b) Number of learning trials vs. performance of Widrow-Hoff, with consecutive trial selection

Figure 4: Supervised learning with random input selection vs. with consecutive input selection

D. Consecutive values

As shown in the above figures, if we present the trials to be learned by the linear associator consecutively, it will learn an order-based bias such that it is far better at predicting the f-g relationships than if the learning trials were presented randomly. This learning accelerates rapidly at around t = numpairs.