Homework 3

86-595 Neural Data Analysis
Name: Shashank Singh
Email: sss1@andrew.cmu.edu
Due: Tuesday, October 9, 2012

Problem 1

See last few pages for MATLAB code used for each part.

a. See Figure 1.

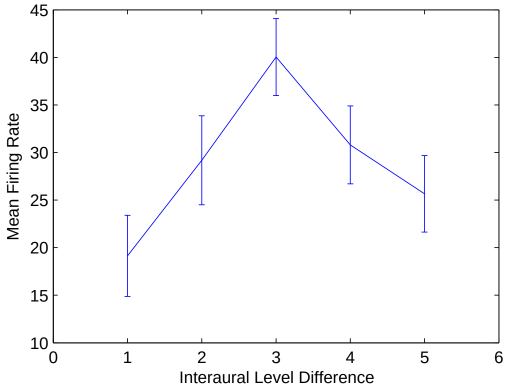

Figure 1: Error bars denote one standard deviation from the mean.

b. See Figure 2.

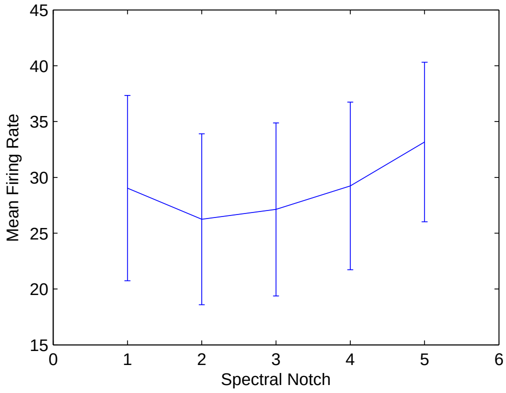

Figure 2: Error bars denote one standard deviation from the mean.

c. See Figure 3.

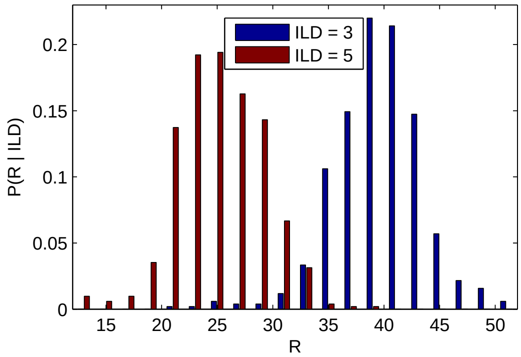

Figure 3: Spike counts were binned in groups of 2.

d. See Figure 4.

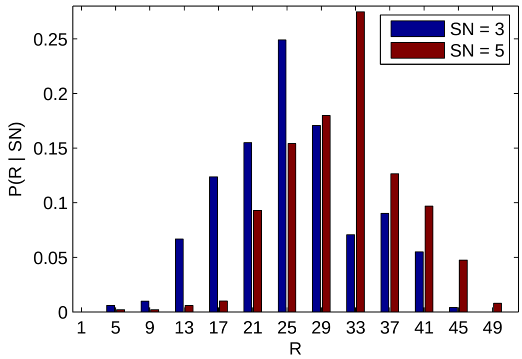

Figure 4: Spike counts were binned in groups of 4.

e. Figures 3 and 4 suggest that spike counts carry more information about ILD; given a spike count is easy to classify the ILD as 3 or 5, except for trials with 32-33 spikes, which occur are less than 5% of trials. On the other hand, there is considerable overlap between the conditional PDF's of $P(R|SN)$, and, for trials with 21-44 spikes (more than 50% of trials), it is difficult to predict whether $SN = 3$ or $SN = 5$ without much error.

f. Since each ILD and SN occurs with uniform probability $\frac{1}{5}$, as shown in Problem 4a. of Homework Set 2,

$$H(ILD) = H(SN) = \log_2(5) \approx \boxed{2.32.}$$

$H(R) \approx \boxed{5.0026}$ (see code). As shown in class,

$$MI(R; ILD), MI(R; SN) \leq \min\{H(R), H(ILD)\} = \min\{H(R), H(SN)\} \approx \boxed{2.32.}$$

g. $H(R|ILD) \approx \boxed{4.0036}$ (see code). $H(R|SN) \approx \boxed{4.8344}$ (see code).

h. As shown in class,

$$MI(R;ILD) = H(R) - H(R|ILD) \approx 5.0026 - 4.0036 = \boxed{0.9990,}$$

$$MI(R;ILD) = H(R) - H(R|SN) \approx 5.0026 - 4.8344 = \boxed{0.1682.}$$

Problem 2

---

a. See Figure 5.

b. See Figure 5.

c. See Figure 5.

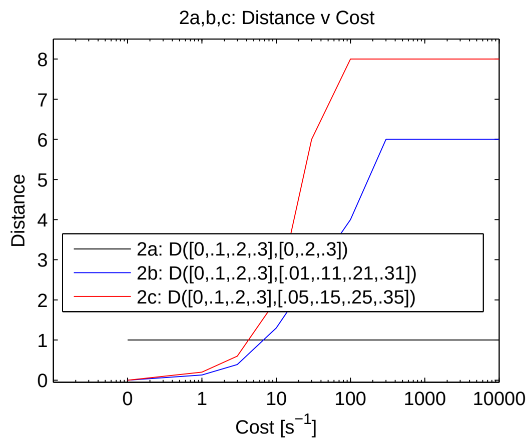

Figure 5: Plot of VP Distance versus cost.

d. The mean distance is $\boxed{31.9167}$ (see code). See figure 6.

e. The mean distance is $\boxed{32.3364}$ (see code). The spike train with $ILD = SN = trial = 1$ is closer to the $SN = 1$ group than the $SN = 5$ group. See figure 7.

f.
```
>> confmat_ild{8}

ans =

    25     0     0     0     0
    22     3     0     0     0
    19     0     0     0     6
    18     0     0     1     6
```

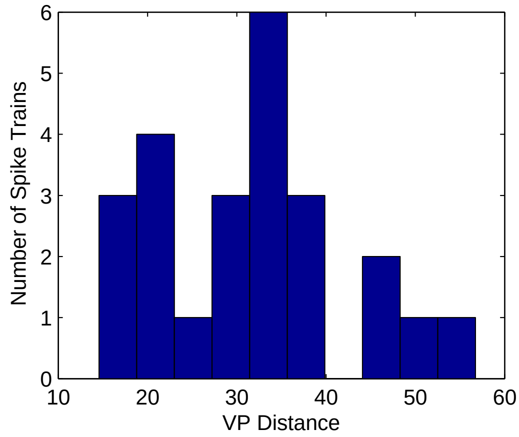

Figure 6: Distribution of VP distances of spikes under identical conditions.

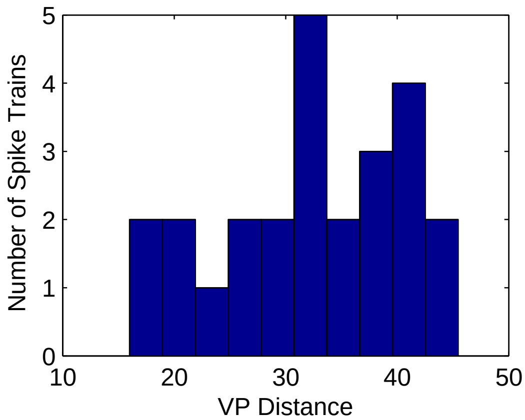

Figure 7: Distribution of VP distances of spikes with $SN = 5$.

```
    15     0     0     0    10
```

Since the entries of `confmat_ild{8}` do not vary much between the rows, there should not be much information about ILD.

g. `>> confmat_sn{8}`

```
ans =

    11    13     1     0     0
     0    23     2     0     0
     0     0    25     0     0
     0     0     7    18     0
     0     0     5     1    19
```

Since most of the spikes were correctly classified, so that the entries of `confmat_sn{8}` lie along the diagonal, there should be a large amount of information about SN.

h. See figures 8 and 9.

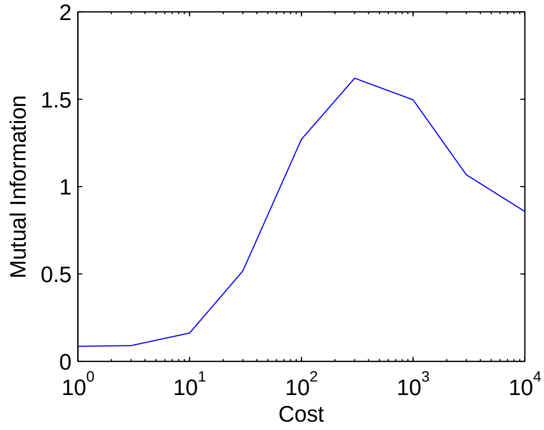

Figure 8: Mutual information between estimated and actual ILD.

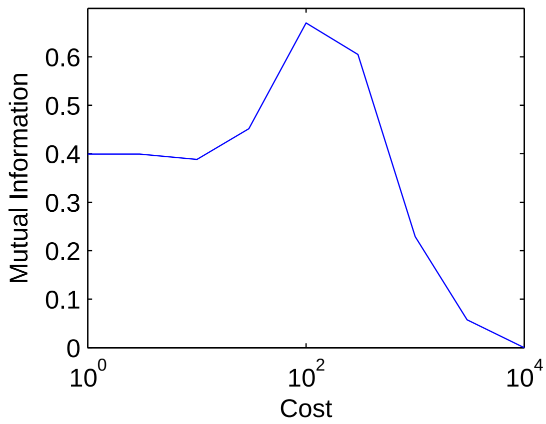

Figure 9: Mutual information between estimated and actual ILD.

i. Problem 1 shows that the neuron's spike count primarily carries ILD information. This is encoded in a tuning curve, and is selective to $ILD = 3$. Thus, the neuron differentiates $ILD = 3$ from ILD values of greater or lesser index, although the tuning curve in Figure 1 suggests the neuron's firing rate fails to distinguish well between $ILD = 2$ and $ILD = 4$ or $ILD = 1$ and $ILD = 5$. Figure 2 shows that the neuron's spike count is essentially independent of SN, and, indeed, Figure 4 showed that SN cannot be accurately determined from spike count. Finally, the mutual information between spike count and ILD was shown to be about 6 times that between spike count and SN.

Problem 2 showed that the neuron's spike timing primarily carries SN information. For large costs, there is little or no correlation between ILD and estimated ILD, because the VP metric places too much value on precise spike timing to accurately measure the distance between train. This is not true, however, for SN, which can be measured reasonably well with a cost of $q = 10^3$, and even somewhat with a cost of $q = 10^4$.

Problem 3

Computing the confusion matrices in part 2h. took quite a while, but probably just because I need to brush up on my matlab. Overall, this homework took a lot longer than the previous two assignments; I think theory tends to be much easier (or at least, much less time consuming) than the programming. However, most of Problem 1 also went pretty quickly.

Problem 2, especially parts g-i, definately taught me the most. It was interesting to use the VP distance to classify spike trains based on their estimated stimuli. It was also helpful, in Problem 1, to apply some information theory to actual data, although 1g was a bit tedious, mostly because the structure of the data was not particularly amenable to the computation.

Parts a-c of Problem 2 probably taught me the least, but they were also very quick and easy.

Code

Problem 1

a.
```
>> for i=1:5
     for j=1:5
       for k=1:102
         d(i,j,k) = length(find(snild_dat(:,1) ==  i ...
                              & snild_dat(:,2) ==  j ...
                              & snild_dat(:,3) ==  k));
       end
     end
   end
>> c = reshape(d,[5 5*102]);
>> errorbar(1:5,mean(c,2),std(c,0,2));
```

b.
```
>> c = reshape(permute(d, [2 1 3]),[5 5*102]);
>> errorbar(1:5,mean(c,2),std(c,0,2));
```

c.
```
>> c = reshape(d,[5 5*102]);
>> x(1,:) = histc(c(3,:),1:2:52); x(2,:) = histc(c(5,:),1:2:52);
>> x(1,:) = x(1,:) ./ sum(x(1,:)); x(2,:) = x(2,:) ./ sum(x(2,:));
>> bar(1:2:52,x'); axis([12 52 0 0.23]);
```

d.
```
>> c = reshape(permute(d, [2 1 3]),[5 5*102]);
>> x(1,:) = histc(c(3,:),1:4:52); x(2,:) = histc(c(5,:),1:4:52);
>> x(1,:) = x(1,:) ./ sum(x(1,:)); x(2,:) = x(2,:) ./ sum(x(2,:));
>> bar(1:4:52,x'); axis([12 52 0 0.28]);
```

e. No code for this part.

f. The following code was used to compute $H(R)$:

```
>> c = histc(d(:),unique(d(:)));
>> c = c ./ sum(c);
>> HR = -sum(c .* log2(c));
```

g. The following code was used to compute $H(R|ILD)$:

```
>> c = reshape(d,[5 5*102]);
>> c = histc(c',unique(d(:)))';
>> for i=1:5, ccond(i,:) = c(i,:)./sum(c(i,:)); end
>> plogcond = ccond .* log2(ccond);
>> plogcond(isnan(plogcond) = 0;
>> HR = mean(-sum(plogcond'));
```

The following code was used to compute $H(R|SN)$:

```
>> c = reshape(permute(d, [2 1 3]),[5 5*102]);
>> c = histc(c',unique(d(:)))';
>> for i=1:5, ccond(i,:) = c(i,:)./sum(c(i,:)); end
>> plogcond = ccond .* log2(ccond);
>> plogcond(isnan(plogcond) = 0;
>> HR = mean(-sum(plogcond'));
```

h. No code for this part.

Problem 2

---

a. The following code was use to calculate `dist_2a`

```
>> costs = [0,1,3,10,30,100,300,1000,3000,10000];
>> spike_train_1 = [0, 0.1, 0.2, 0.3];
>> spike_train_2 = [0, 0.2, 0.3];
>> for i=1:10, dist_2a(i) = HW3_spkd(spike_train_1,spike_train_2,costs(i)); end
```

b.
```
>> spike_train_2 = [0.1, 0.11, 0.21, 0.31];
>> for i=1:10, dist_2b(i) = HW3_spkd(spike_train_1,spike_train_2,costs(i)); end
```

c.
```
>> spike_train_2 = [0.05, 0.15, 0.25, 0.35];
>> for i=1:10, dist_2c(i) = HW3_spkd(spike_train_1,spike_train_2,costs(i)); end
```

d.
```
>> for i=1:5
     for k=1:5
       trains{i,k} = snild_dat2(find(snild_dat2(:,1) ==  i
                                   & snild_dat2(:,2) ==  1
                                   & snild_dat2(:,3) ==  k),4);
     end
   end
>> trains = trains(:);
>> train1 = trains{1};
>> for i = 2:length(trains)
     dist(i-1) = HW3_spkd(train1,train{i},1000);
   end
>> meandist = mean(dist);
```

e.
```
>> for i=1:5
     for k=1:5
       trains{i,k} = snild_dat2(find(snild_dat2(:,1) ==  i
                                  & snild_dat2(:,2) ==  5
                                  & snild_dat2(:,3) ==  k),4);
     end
   end
>> trains = trains(:);
>> for i = 1:length(trains)
     dist(i) = HW3_spkd(train1,train{i},1000);
   end
>> meandist = mean(dist);
```

f. The following code was used to compute each confusion matrix:

```
function cf = confmat(snild_dat,cost)
  cf = zeros(5,5);
  trains = cell(5,5,5);

  for i=1:5
    for j=1:5
      for k=1:5
        trains{i,j,k} = snild_dat(find(snild_dat(:,1) ==  i
                                   & snild_dat(:,2) ==  j
                                   & snild_dat(:,3) ==  k),4);
      end
    end
  end

  dists = zeros(5,5,5);
  for i=1:5 % current ILD
    for j=1:5 % current SN
      for k=1:5 % current trial

        for x=1:5 % comparison ILD
          for y=1:5 % comparison SN
            for z=1:5 %comparison trial
              dists(x,y,z) = HW3_spkd(trains{i,j,k},trains{x,y,z},cost);
            end
          end
        end
        mean_dists = mean(mean(dists,3),2);
        [~, assignment] = min(mean_dists);
        cf(i,assignment) = cf(i,assignment) + 1;
      end
    end
```

```
    end
  end
```

g. The same code was used as in part f., except that the ILD and SN columns of `snild_dat` were swapped.

h. The following code was used to compute each mutual information value for ILD (the code used for SN was essentially identical):

```
>> for i=1:10
     confmat_ild{i} = confmat_ild{i} ./ sum(sum(confmat_ild{i}));
   end
>> for q=1:10
     for i=1:5
       for j=1:5
         mi(i,j) = confmat_ild{q}(i,j) .* log2(confmat_ild{q}(i,j) ...
                                  ./ (0.2 .* sum(confmat_ild{q}(:,j))));
         mi(isnan(mi)) = 0;
       end
     end
     MI(q)=sum(mi(:));
   end
```

i. No code for this part.