MIT OpenCourseWare
http://ocw.mit.edu

9.35 Sensation And Perception
Spring 2009

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.

Problem Set 1

Solutions guide

Contents

Init

Note that this document was auto-generated using File->(Save and) Publish to HTML. This is a very convenient way to generate a PSET. See the source code for the formatting to use. After generating an HTML file, it was just printed as a PDF.

```
clear all
close all
clc
drawnow
```

1)

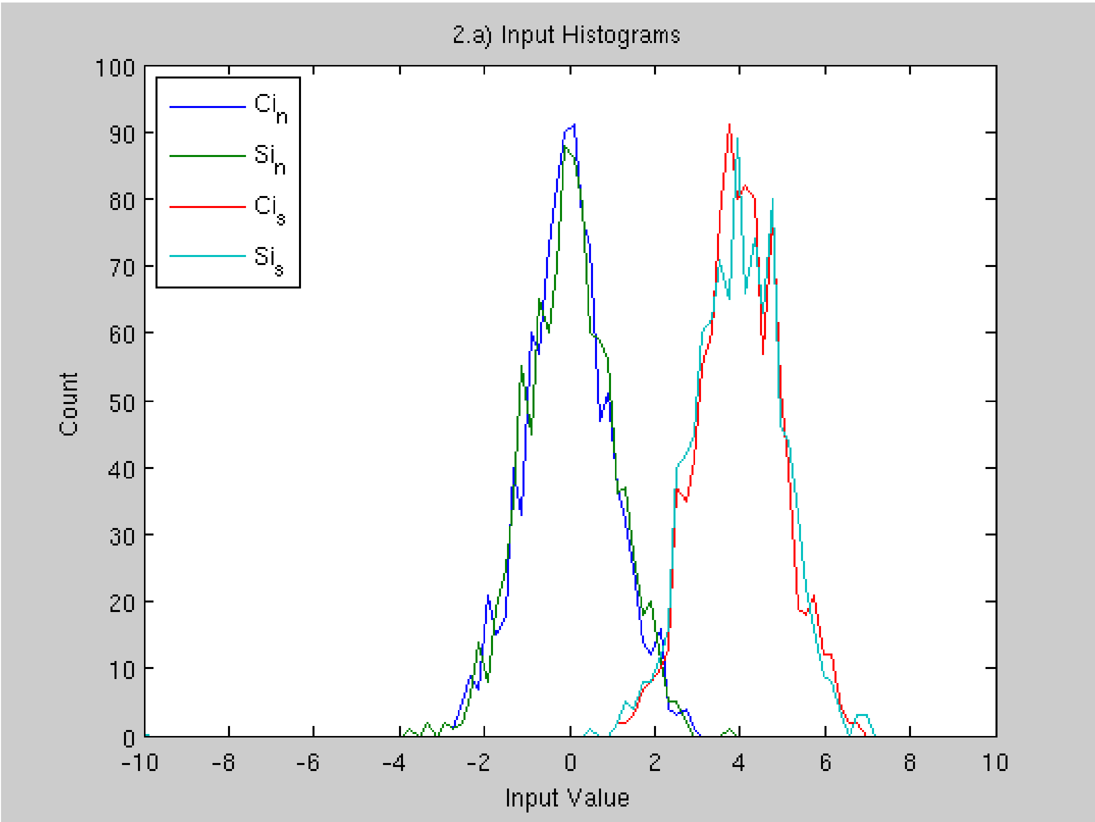
a) After passing through the cornea and lens, photons from the star are focused on the outer segments of photoreceptors, where they are absorbed by rhodopsin pigment molecules. This triggers a chemical cascade which results in the hyperpolarization of the photoreceptor. Downstream, this disinhibits an ON bipolar cell which in turn excites the ON-midget ganglion cell. Additionally, horizontal and amacrine cells integrate information from surrounding photoreceptors and bipolar cells, and help form an antagonistic surround.

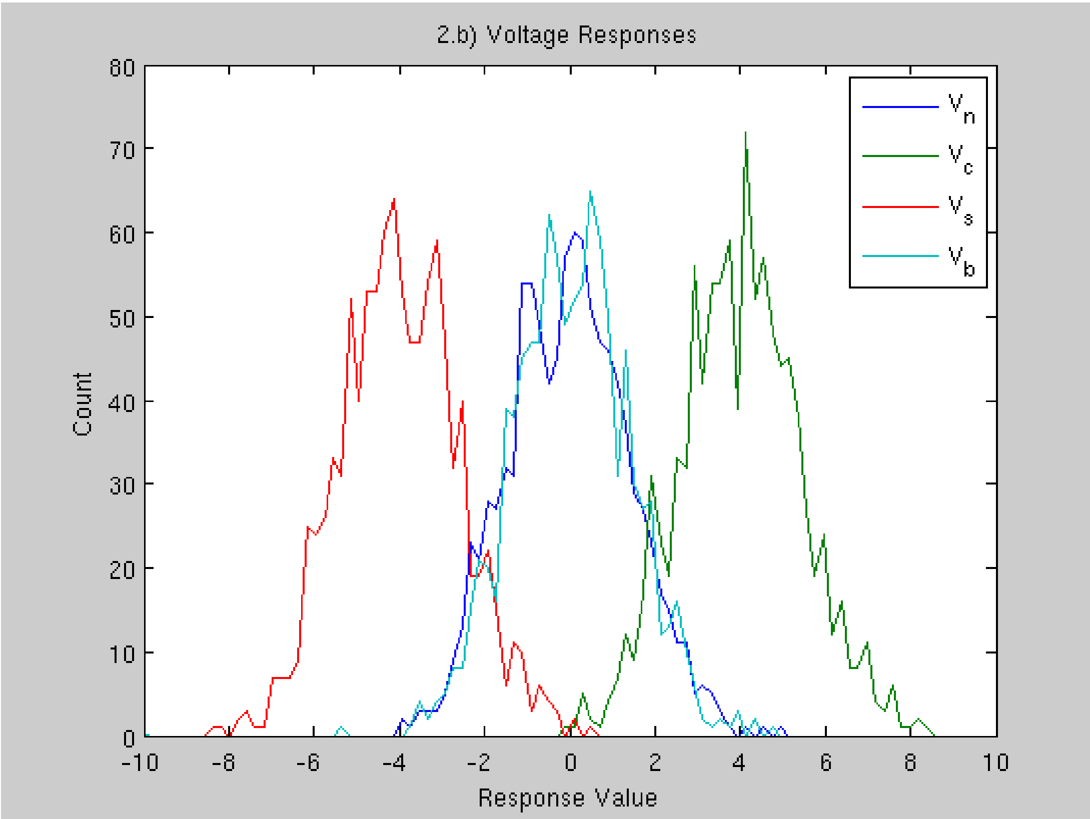
b) ON-bipolar cells in the center of the RF will excite this ganglion cell, while surround activity, through lateral inhibition, will inhibit it. Therefore, light in the center only will increase the cell's firing rate, light in the surround only will decrease it, and light in both will have an antagonistic effect resulting in baseline activity.

2.a,b)

```
% Create our four sample vectors
Ci_n = randn(1000, 1);
```

```
Si_n = randn(1000, 1);
Ci_s = randn(1000, 1) + 4; % Can't just add 4 to Ci_n or Si_n
Si_s = randn(1000, 1) + 4; % since it breaks independence

% Create the four input responses: CENTER - SURROUND
V_n = Ci_n - Si_n;
V_c = Ci_s - Si_n;
V_s = Ci_n - Si_s;
V_b = Ci_s - Si_s;

% Plot histograms
t = linspace(-10, 10);
figure
plot(t, hist(Ci_n, t), t, hist(Si_n, t), t, hist(Ci_s, t), t, hist(Si_s, t));
legend('Ci_n', 'Si_n', 'Ci_s', 'Si_s', 'Location', 'NorthWest');
title('2.a) Input Histograms');
ylabel('Count');
xlabel('Input Value');

input_hist = figure;
plot(t, hist(V_n, t), t, hist(V_c, t), t, hist(V_s, t), t, hist(V_b, t));
legend('V_n', 'V_c', 'V_s', 'V_b');
title('2.b) Voltage Responses');
ylabel('Count');
xlabel('Response Value');
```





Nota bene: to treat every trial as a unique event on which we may perform the statistics of our choice, each trial must be independantly drawn from the same distribution. We can't simply add 4 to simulate a stimulation since now the stimulated trials are dependent on the samples drawn in the non-stimulated condition! For the pure of heart, we must also avoid using the same samples more than once in V_x (ie V_n and and V_c both use Si_n), but at this point, for our purposes, it's not worth creating 4 more independent vectors... kudos if you noticed!

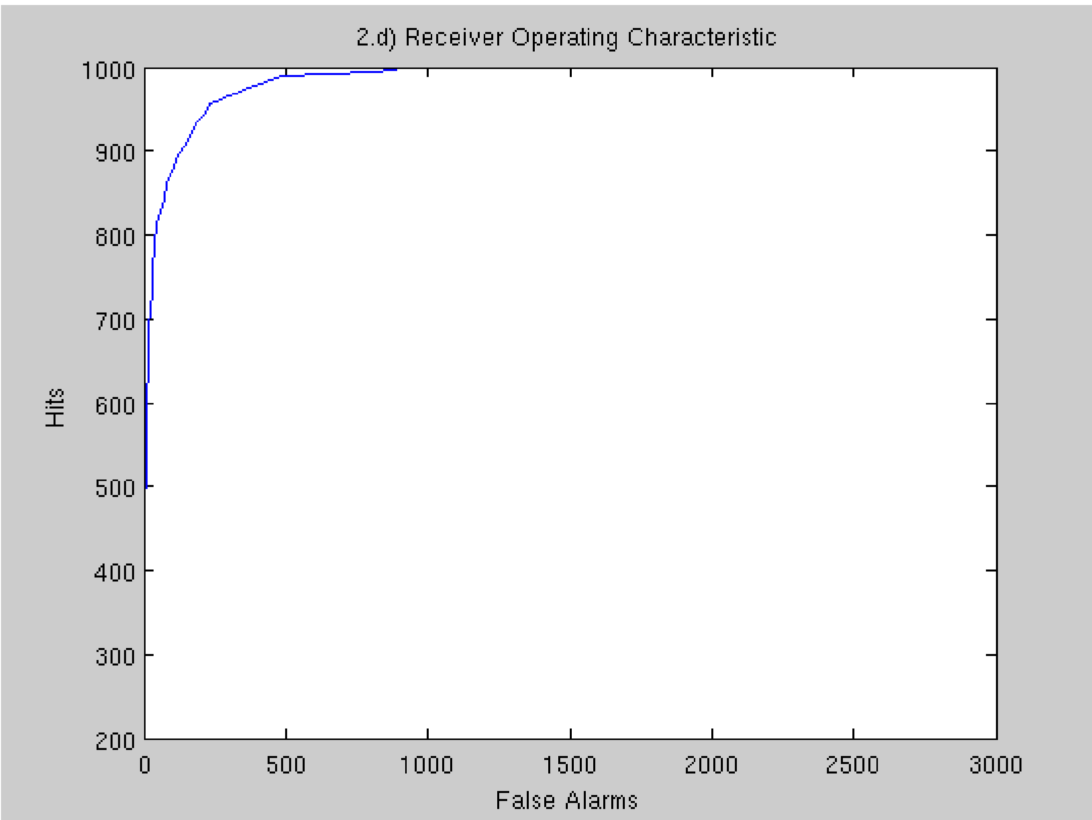
2.c, d)

```
APs = @(vec, thresh) sum(vec>thresh); % Anonymous function for simplicity

thr = 3;
hits = APs(V_c, thr)
FA_Possibles = [V_n; V_s; V_b];
FAs = APs(FA_Possibles, thr)
```

```
hits =

   754


FAs =

    33
```

Anonymous functions are ways of defining a function in-line (ie without writing an m-file). Alternatively, the `for` loop could be written as, `hits(i) = sum(V_c>thr(i));` etc. Anonymous functions are much more convenient when a repeated calculation is needed and you don't want to retype the same thing many times!

```
% ROC data
thr = linspace(-5, 5);

for i = 1:length(thr)
    hits(i) = APs(V_c, thr(i));
    FAs(i) = APs(FA_Possibles, thr(i));
end

roc = figure;
plot(FAs, hits);
xlabel('False Alarms')
ylabel('Hits')
title('2.d) Receiver Operating Characteristic')
```



2.e)

If the input were noisier, the ROC curve would approach a linear function with origin 0 and slope representing the proportion of total possible hits / total possible FA's. Cleaner input would make the ROC curve approach the limit of a step function (all hits with no false alarms). Correspondingly, the sensitivity of a cell is it's ability to separate signal and noise. In other words, the more the ROC curve deviates from a straight line, for a given input, the more sensitive the cell (or process/algorithm etc) is. One way this is typically measured is by the (normalized) area under the ROC curve: highly-sensitive processes will have an area near 1, while noisy processes will have an area close to 1/2.

The sensitivity of a process is its *effective* signal to noise ratio, or, how much noise it adds to the input. Our simulated cells don't add any noise to the input, so the ROC curve simply describes the statistics of the input noise. However, if we had cells of varying sensitivity (meaning they added more or less noise), then we could compare the sensitivities of different cell to the same input via their ROC curves. Specifically, a highly sensitive cell would have a similar ROC curve to a less sensitive cell that had a cleaner signal.

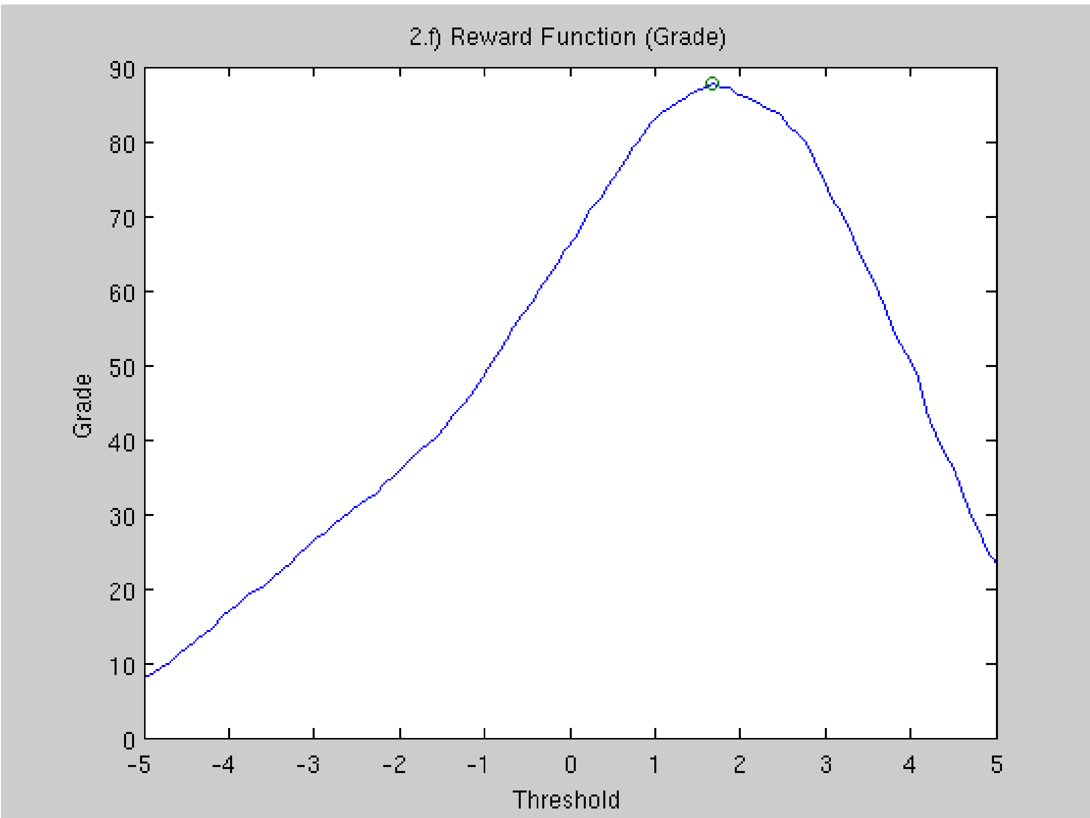
2.f)

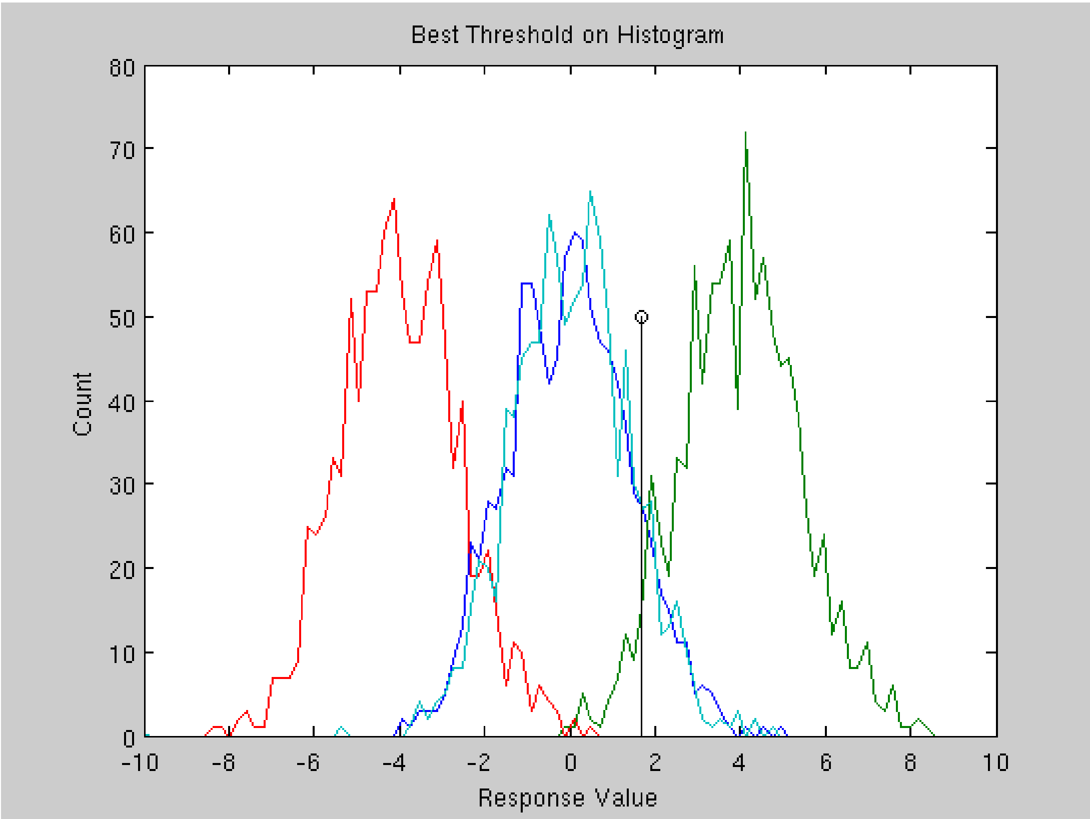
The best threshold should be just under 2. If we only considered V_n and V_c, and our reward function only had multipliers of 1, the expected threshold would be exactly 2 since this lies halfway between the mean of each input response distribution (and they have identical variance). However, we shift the bias by including more FA possibilities in V_s and V_b, and weighing hits and FAs differently. In any case the threshold will vary from run to run, since this is a stochastic process (your grade is a continuosly distributed random variable!)

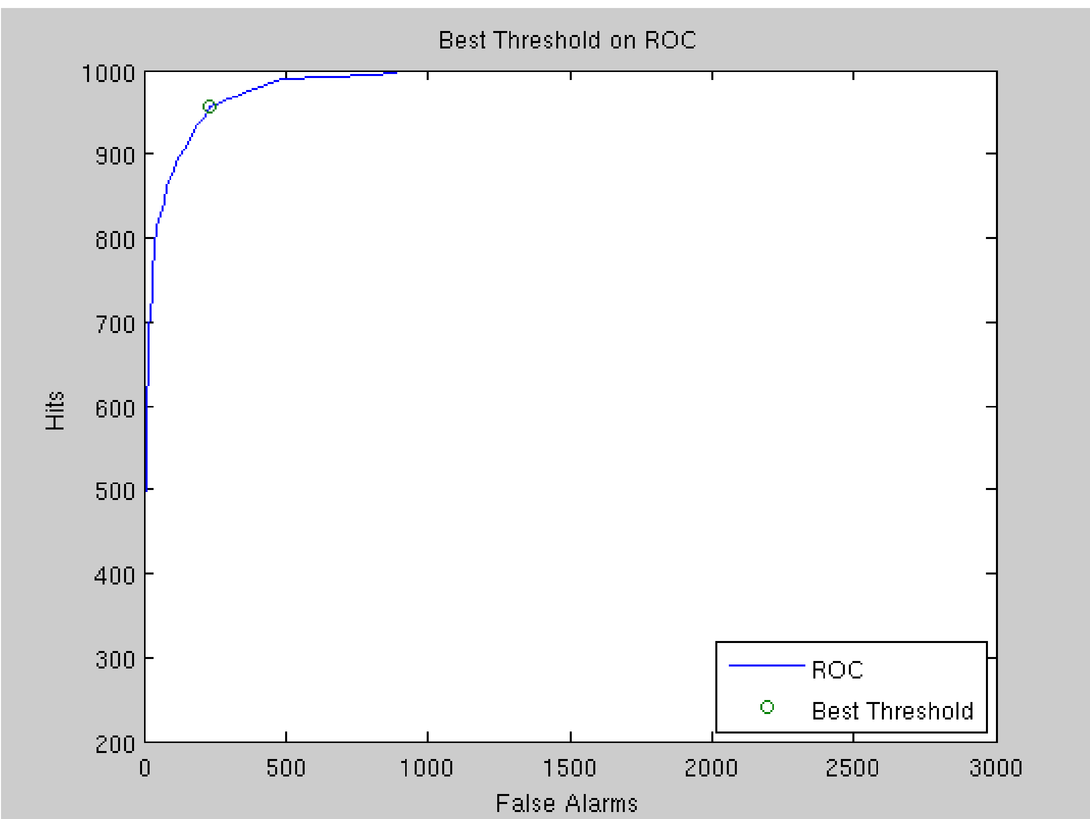
```
% Reward function
grade = hits/10-FAs/30;

[MaxGrade MaxGradeInd] = max(grade);
MaxGrade
MaxGradeThresh = thr(MaxGradeInd)

figure
plot(thr, grade);
xlabel('Threshold')
ylabel('Grade')
title('2.f) Reward Function (Grade)')
hold on
scatter(MaxGradeThresh, MaxGrade');
hold off

figure(input_hist);
hold on
stem(MaxGradeThresh, 50, 'k-');
legend off
hold off
title('Best Threshold on Histogram')
```

```
figure(roc);
hold on
scatter(FAs(MaxGradeInd), hits(MaxGradeInd));
hold off
legend('ROC', 'Best Threshold', 'Location', 'SouthEast');
title('Best Threshold on ROC')
```

```
MaxGrade =

   87.7667


MaxGradeThresh =

    1.6667
```







Deviating from Sensation and Perception into the realm of cognitive psychology, it is also worthwhile to note that, regardless of this cell's performance or their calculation of it, every student believes they deserve an A!

Published with MATLAB® 7.6