Homework 2 Solutions
Kernel SVM and Perceptron

CMU 10-701: Machine Learning (Fall 2014)
https://piazza.com/cmu/fall2014/1070115781/home
OUT: Sept 25, 2014
DUE: Oct 8, 11:59 PM

Problem 1: SVM decision boundaries [Zichao - 30pts]

Support Vector Machines are powerful tools for classifications. SVM can perform non-linear classification using the kernel trick. In this question, you are to examine the decision boundaries of SVM with different kernels.

1. Recall that the soft-margin primal SVM problem is

$$\min \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum_{i=1}^{n}\xi_i \tag{1}$$

$$\text{s.t.} \forall i = 1, \cdots, n: \tag{2}$$

$$\xi_i \geq 0 \tag{3}$$

$$(\mathbf{w}\cdot\mathbf{x_i} + b)y_i - (1-\xi_i) \geq 0. \tag{4}$$

For hard-margin primal SVM, $\xi_i = 0, \forall i$. We can get the kernel SVM by taking the dual of the primal problem and then replace the product of $\mathbf{x_i}\cdot\mathbf{x_j}$ by $k(\mathbf{x_i}, \mathbf{x_j})$, where $k(.,.)$ can be any kernel function.

Figure 1 plots SVM decision boundaries resulting from using different kernels and/or different slack penalties. In Figure 1, there are two classes of training data, with labels $y_i \in \{-1, 1\}$, represented by circles and squares respectively. The SOLID circles and squares represent the support vectors. Label each plot in Figure 1 with the letter of the optimization problem below and explain WHY you pick the figure for a given kernel. (Note that there are 6 plots, but only 5 problems, so one plot does not match any of the problems.)

2 pts for choice and 2pts for reason

(a) A soft-margin linear SVM with $C = 0.1$. **[4 pts]**
Solution: Corresponds to Fig. 1.4. The decision boundary of linear SVM is linear. In comparison with Fig. 1.3(problem b), the line does not separate the two classes strictly, which corresponds to the case C is small and more errors are allowed.

(b) A soft-margin linear SVM with $C = 10$. **[4 pts]**
Solution: Corresponds to Fig. 1.3. The decision boundary of linear SVM is linear. In comparison with Fig. 1.4(problem a), the line separates two classes strictly, which corresponds to the case C is big.

(c) A hard-margin kernel SVM with $\mathbf{K}(\mathbf{u}, \mathbf{v}) = \mathbf{u}\cdot\mathbf{v} + (\mathbf{u}\cdot\mathbf{v})^2$. **[4 pts]**
Solution: Corresponds to Fig. 1.5. The decision function of quadratic kernel is given by $f(\mathbf{x}) = \sum_i \alpha_i(\mathbf{x_i}\cdot\mathbf{x} + (\mathbf{x_i}\cdot\mathbf{x})^2) + b$. Hence the decision boundary is $f(\mathbf{x}) = 0$. Since $f(\mathbf{x})$ is second order function of $\mathbf{x}$, the curve can be ellipse or hyperbolic curve. Fig. 5 is hyperbolic curve.

(d) A hard-margin kernel SVM with $\mathbf{K}(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{1}{4}\|\mathbf{u}-\mathbf{v}\|^2\right)$. **[4 pts]**
Solution: Corresponds to Fig. 1.1. We can write out the decision function as $f(\mathbf{x}) = \sum_i \alpha_i \exp\left(-\gamma\|\mathbf{x_i}-\mathbf{x}\|^2\right) + b$. If γ is large, then the kernel value is quite small even if the distance

between the $\mathbf{x}$ and $\mathbf{x_i}$ is small. This makes the classification hard with few supporting vectors. If Fig. 1.1 corresponds to the case γ is large (=4), then it is hard to classify many circle point in the middle in Fig. 1.1. Hence, Fig. 1.1 corresponds to $\gamma = \frac{1}{4}$.

(e) A hard-margin kernel SVM with $\mathbf{K}(\mathbf{u}, \mathbf{v}) = \exp\left(-4\|\mathbf{u} - \mathbf{v}\|^2\right)$. **[4 pts]**

Solution: Corresponds to Fig. 1.6. Using similar argument, we can conclude that if γ is large, there are more support vectors.

2. You are given a training dataset, as shown in Fig 2. Note that the training data comes from sensors which can be error-prone, so you should avoid trusting any specific point too much. For this problem, assume that we are training an SVM with a quadratic kernel.

(a) Where would the decision boundary be for very large values of C (i.e., $C \to \infty$)? Draw on figure and justify your answer. **[3 pts]**

Solution: The decision boundary is the curve (a) in Fig. 2 When $C \to \infty$, it becomes hard margin SVM, hence the decision boundary must separate the two class. Given it's a quadratic kernel (with the analysis from 1.1.c), we can conclude the decision boundary is like curve (a) in Fig. 2.

(b) For $C \approx 0$, indicate in the figure where you would expect the decision boundary to be? Justify your answer. **[3 pts]**

Solution: The decision boundary is the curve (b) in Fig. 2 When $C \to 0$, the decision boundary tends to find the max margin between the two classes regardless of several noises. Any linear or second order curve that appears in the middle is okay for this problem.

(c) Which of the two cases above would you expect to work better in the classification task? Why? **[4 pts]**

Solution: The second decision boundary tends to work better. We can see that the two circle points near the square points are noises. Put $C \to \infty$ will make the classification decided by the noises, which doesn't represent the real decision boundary.

Problem 2: Understanding the Likelihood Function, Bit by Bit (Ben, 30 points)

You are receiving a random stream of 0 or 1 bits, and you would like to know the probability that the next received bit is 1. This can be thought of as flipping a (possibly unfair) coin and finding the probablity of the coin being heads.
Let

$$H_i = \begin{cases} 1 & \text{if the } i\text{th bit received is 1} \\ 0 & \text{if the } i\text{th bit received is 0} \end{cases}$$

Let $P(H_i = 1) = p_H$. Then, $P(H_i = 0) = 1 - p_H$. Let $N_H = \sum_{i=1}^{N_{\text{bits}}} H_i$ be the number of received 1s and N_{bits} be the total number of bits received. We observe $H_1, \ldots, H_{N_{\text{bits}}}$.

1. (a) (1 point) Give the name of the distribution and list its parameters for the random variable H_i (i.e., $H_i \sim$ Distribution(parameters)). Give the name of the distribution and list its parameters for the random variable N_H.

$H_i \sim$ Bernoulli(p_H). (+0.5)
$N_H \sim$ Binomial(N_{bits}, p_H). (+0.5)

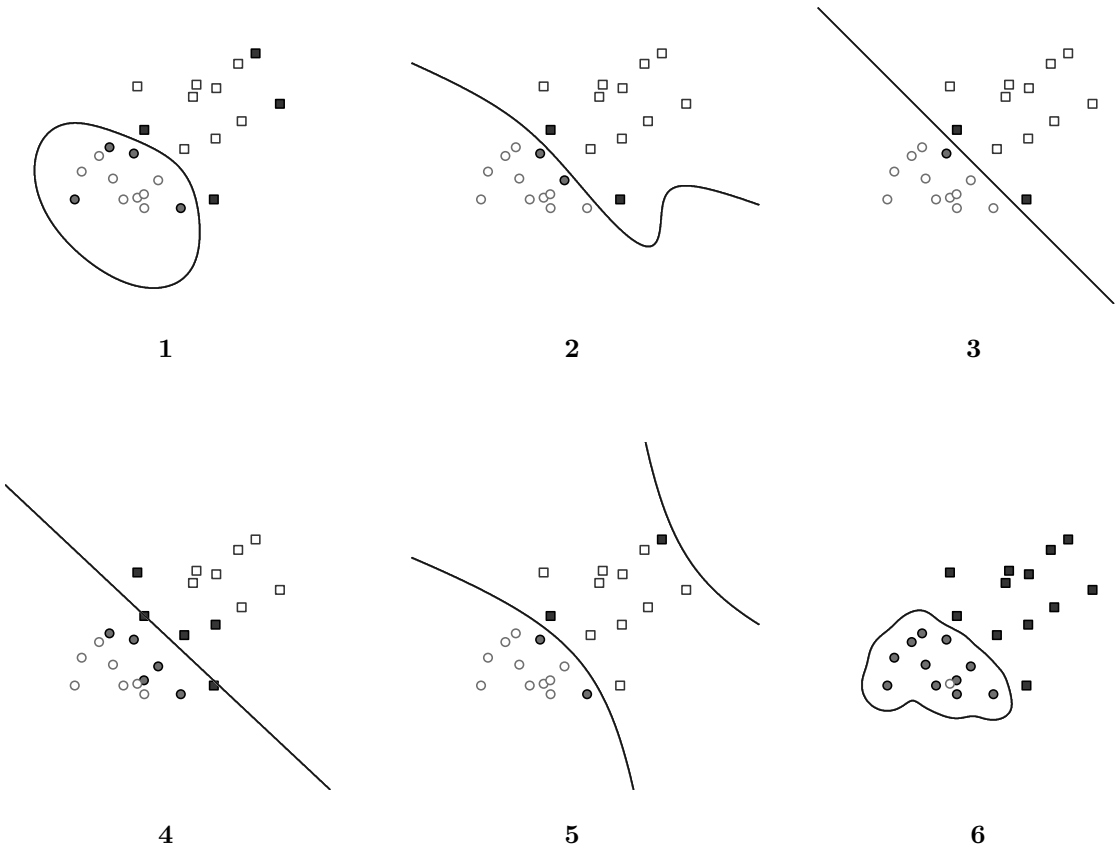

Figure 1: Induced Decision Boundaries

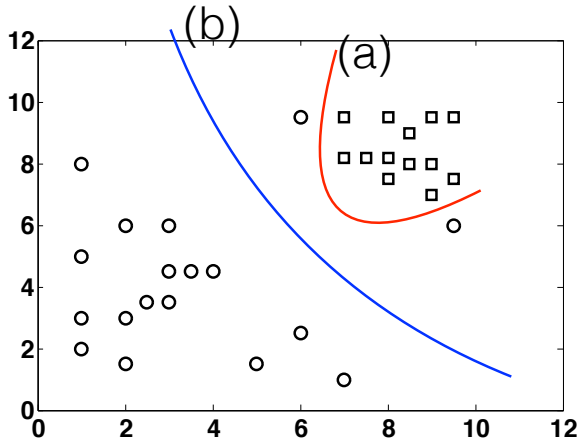

Figure 2: Training dataset

(b) Since we do not know the true parameter p_H^*, we would like to estimate it from observed data. One estimate is the maximum likelihood estimate (MLE). This maximizes the likelihood function of p_H given the data.

i. (1 point) Give the form of the likelihood function, $\mathcal{L}(H_1, \ldots, H_{N_{\text{bits}}}; p_H)$. Use the symbols N_H and N_{bits} where appropriate.

$\mathcal{L}(H_1, \ldots, H_{N_{\text{bits}}}; p_H) = P(H_1, \ldots, H_{N_{\text{bits}}}; p_H)$
$= \prod_{i=1}^{N_{\text{bits}}} P(H_i; p_H)$; since the bits are independent
$= \prod_{i=1}^{N_{\text{bits}}} {p_H}^{H_i}(1-p_H)^{1-H_i}$; if $H_i = 1$, $P(H_i; p_H) = p_H$, if $H_i = 0$, $P(H_i; p_H) = 1 - p_H$
$= {p_H}^{\sum_{i=1}^{N_{\text{bits}}} H_i}(1-p_H)^{\sum_{i=1}^{N_{\text{bits}}}(1-H_i)} = {p_H}^{N_H}(1-p_H)^{N_{\text{bits}}-N_H}$

$\mathcal{L}(H_1, \ldots, H_{N_{\text{bits}}}; p_H) = {p_H}^{N_H}(1-p_H)^{N_{\text{bits}}-N_H}$ (+1)

ii. (2 points) Let $N_{\text{bits}} = 3$, and say we observed four different sequences of bits: 111, 110, 001, 000. Plot $\mathcal{L}(H_1, \ldots, H_{N_{bits}}; p_H)$ (y-axis) vs. p_H (x-axis) for each sequence on the same axes. (You will have four curves on the same plot. Sample p_H from 0 to 1 in 0.01 increments.) On each curve, put a star at the p_H that has the highest likelihood. These are the maximum likelihood estimates. Do they make intuitive sense given the data? Why?
[For your convenience, you can use the given hw2_ques2_1_b_ii.m as a starting script to make the plot. Simply fill in the missing code where it instructs you. Include a figure of the plot in your write-up. Do not include extra .m files in your homework submission.]

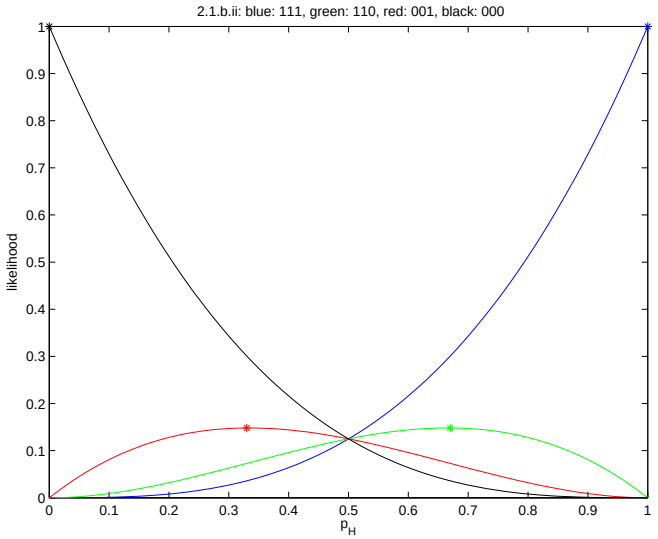

Figure 3: Question 2.1.b.ii

(+1.5 for Figure 3)
For 111, $\hat{p}_H = 1$, because we've only seen 1s. $\hat{p}_H$ is the MLE. For 000, $\hat{p}_H = 0$, because we've only seen 0s. For 110, $\hat{p}_H = \frac{2}{3}$, since we've seen two 1s and 0; for 001, we have $\hat{p}_H = \frac{1}{3}$. These makes sense because the best information we have is the number of 1s seen out of N_{bits}. (+0.5)

iii. (2 points) Using calculus, calculate the area under the curve for each sequence. You can check the approximate correctness of these answers by summing up the curves in 2.1.b.ii. Is the likelihood function a valid probability distribution over p_H? Explain.

For 111:
$\int_0^1 {p_H}^3 dp_H = \frac{{p_H}^4}{4}|_0^1 = \frac{1}{4}$

For 110:
$\int_0^1 {p_H}^2(1-p_H)dp_H = \int_0^1 {p_H}^2 - {p_H}^3 dp_H = \frac{{p_H}^3}{3} - \frac{{p_H}^4}{4}|_0^1 = \frac{1}{3} - \frac{1}{4} = \frac{1}{12}$

For 001:
$\int_0^1 (1-p_H)^2 p_H dp_H = \int_0^1 (1-2p_H+{p_H}^2)p_H dp_H = \int_0^1 p_H - 2{p_H}^2 + {p_H}^3 dp_H = \frac{{p_H}^2}{2} - \frac{2}{3}{p_H}^3 + \frac{1}{4}{p_H}^4|_0^1$
$= \frac{1}{2} - \frac{2}{3} + \frac{1}{4} - 0 = \frac{3}{4} - \frac{2}{3} = \frac{1}{12}$

For 000:
$\int_0^1 (1-p_H)^3 dp_H = \frac{(1-p_H)^4}{4}(-1)|_0^1 = -\frac{(1-1)^4}{4} - \frac{-(1-0)^4}{4} = 0 + \frac{1}{4} = \frac{1}{4}$ (+1)

No, the likelihood function is *not* a valid probability distribution (otherwise, all these areas should sum to 1). This is because we fix the data $(H_1, \ldots, H_{N_{\text{bits}}})$ and vary the parameter p_H. Note that it is called the likelihood *function* and not *distribution*. If p_H was fixed and we instead integrated over the random variables representing the data, the likelihood function would sum to 1 (i.e., this is the actual probability distribution of the data). (+1)

(c) (2 points) Now compute $\hat{p}_H$, the MLE, for any N_{bits}. Write out and label each step clearly, providing movitation for the steps (e.g., why are you taking the derivative?). Just writing down the MLE form will not be given full credit. You should only need to use N_{bits}, $H_1, \ldots, H_{N_{\text{bits}}}$, N_H, p_H, and $\hat{p}_H$ as symbols.

$\hat{p}_H = \arg\max_{p_H} \mathcal{L}(H_1, \ldots, H_{N_{\text{bits}}}; p_H)$
$= \arg\max_{p_H} {p_H}^{N_H}(1-p_H)^{N_{\text{bits}}-N_H}$ (+0.5)
To find the maximum, find where the function $\mathcal{L}$ has 0 slope (we know $\mathcal{L}$ is concave). This is equivalent to taking the derivative and setting it to 0. (+0.5)

$\frac{d\mathcal{L}}{dp_H} = \frac{d}{dp_H}[{p_H}^{N_H}(1-p_H)^{N_{\text{bits}}-N_H}]$ — Use the chain rule.
$= {p_H}^{N_H}(N_{\text{bits}}-N_H)(1-p_H)^{N_{\text{bits}}-N_H-1}(-1) + N_H {p_H}^{N_H-1}(1-p_H)^{N_{\text{bits}}-N_H} = 0$ — setting the derivative to 0.
$\Rightarrow (N_{\text{bits}}-N_H){p_H}^{N_H}(1-p_H)^{N_{\text{bits}}-N_H}(1-p_H)^{-1} = N_H {p_H}^{N_H}(1-p_H)^{N_{\text{bits}}-N_H} p_H^{-1}$
$\Rightarrow p_H(N_{\text{bits}}-N_H) = (1-p_H)N_H$
$\Rightarrow p_H N_{\text{bits}} = N_H \Rightarrow p_H = \frac{N_H}{N_{\text{bits}}}$
The maximum of $\mathcal{L}$ is when $p_H = \frac{N_H}{N_{\text{bits}}}$, so
$\hat{p}_H = \frac{N_H}{N_{\text{bits}}}$ (+0.5 for work, +0.5 for answer)

2. We now consider the case where we receive the stream of bits with additive noise (i.e., channel corruption). The noise model can be written:

$$O_i = H_i + \epsilon_i$$

where $i = 1, \ldots, N_{\text{bits}}$, O_i is the observed bit with noise, H_i is defined in the same manner as in 2.1, and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, where σ^2 is given.

(a) (3 points) Give the form of the likelihood function, $\mathcal{L}(O_1, \ldots, O_{N_{\text{bits}}}; p_H)$. Determine if the log of

this function is concave. Why is concavity important? (One way of checking for concavity is to see if the second derivative is negative for all values of p_H.)

$\mathcal{L}(O_1, \ldots, O_{N_{\text{bits}}}; p_H) = P(O_1, \ldots, O_{N_{\text{bits}}}; p_H)$
$= \prod_{i=1}^{N_{\text{bits}}} P(O_i; p_H) = \prod_{i=1}^{N_{\text{bits}}} \sum_{t=0}^{1} P(O_i, H_i = t; p_H)$ — We are expanding to include H_i's with marginalization. Thus, we'll get to treat the H_i's as "known" to capture the ϵ_i's.
$= \prod_{i=1}^{N_{\text{bits}}} \sum_{t=0}^{1} P(O_i \mid H_i = t; p_H) P(H_i = t; p_H)$ — Now making use of conditional probability.
Note when $H_i = 0$, then $O_i \sim \mathcal{N}(0, \sigma^2)$ and
when $H_i = 1$, then $O_i - 1 \sim \mathcal{N}(0, \sigma^2)$.
$= \prod_{i=1}^{N_{\text{bits}}} [\mathcal{N}(O_i; 0, \sigma^2)(1 - p_H) + \mathcal{N}(O_i - 1; O, \sigma^2) p_H]$
$= \prod_{i=1}^{N_{\text{bits}}} \left[\mathcal{N}(O_i; 0, \sigma^2) + \big(\mathcal{N}(O_i - 1; 0, \sigma^2) - \mathcal{N}(O_i; 0, \sigma^2)\big) p_H\right]$

So $\mathcal{L}(O_1, \ldots, O_{N_{\text{bits}}}; p_H) = \prod_{i=1}^{N_{\text{bits}}} \left[\mathcal{N}(O_i; 0, \sigma^2) + \big(\mathcal{N}(O_i - 1; 0, \sigma^2) - \mathcal{N}(O_i; 0, \sigma^2)\big) p_H\right]$
(+0.5 for the work, +0.5 for the answer)

Let $\alpha_i = \mathcal{N}(O_i; 0, \sigma^2)$ and $\beta_i = \mathcal{N}(O_i - 1; 0, \sigma^2)$, since these are just constants, given from the data.
$\log \mathcal{L}(O_1, \ldots, O_{N_{\text{bits}}}; p_H) = \sum_{i=1}^{N_{\text{bits}}} \log(\alpha_i + (\beta_i - \alpha_i) p_H)$
$\frac{d}{dp_H} \log \mathcal{L} = \sum_{i=1}^{N_{\text{bits}}} \frac{\beta_i - \alpha_i}{\alpha_i + (\beta_i - \alpha_i) p_H}$
$\frac{d^2}{dp_H{}^2} \log \mathcal{L} = \sum_{i=1}^{N_{\text{bits}}} \frac{(\alpha_i + (\beta_i - \alpha_i) p_H) 0 - (\beta_i - \alpha_i)(\beta_i - \alpha_i)}{(\alpha_i + (\beta_i - \alpha_i) p_H)^2} = \sum_{i=1}^{N_{\text{bits}}} \frac{-(\beta_i - \alpha_i)^2}{(\alpha_i + (\beta_i - \alpha_i) p_H)^2}$
(+1 for the work, +0.5 for the answer)

The second derivative of $\log \mathcal{L}$ with respect to p_H is always negative. Thus, $\log \mathcal{L}$ is a concave function. Concavity is important because if we cannot find a closed form solution for the MLE, we can rely on fast optimization methods that are guaranteed to find the maximum $\log \mathcal{L}$ at a global optimum (and not get stuck in local minima).
(+0.5)

(b) (3 points) As in 2.1.c, compute the MLE of p_H, $\hat{p}_H$, for the likelihood function, $\mathcal{L}(O_1, \ldots, O_{N_{\text{bits}}}; p_H)$. Write out the analytical form, if possible. Otherwise, describe how to obtain the MLE without a closed form.

Setting the derivative ($\frac{d}{dp_H} \mathcal{L}$), calculated in 2.2.a, to zero cannot be solved analytically (p_H is entangled on the dependence of each O_i). However, in 2.2.a we found that the log-likelihood function is concave. Thus, we can use an optimaization method, like gradient descent, to solve for the p_H that produces the maximum log-likelihood.
(+1 for why we can't solve analytically, +1 for concavity, +1 for optimization method)

(c) (4 points) Let the true parameter $p_H^* = 0.6$, $\sigma^2 = 0.1$. Generate 5 datasets from the model with different trial sizes (i.e., vary N_{bits}): $\{100, 500, 1000, 2500\}$. Plot $\log \mathcal{L}(O_1, \ldots, O_{N_{\text{bits}}}; p_H)$ vs. p_H for each dataset on the same axes. For each curve, place a star at the p_H with the maximum log-likelihood. (There will be 4 curves on the same plot. Again, sample p_H from 0 to 1 with 0.01 increments.) What happens when you increase σ^2? [You can use the given hw2_ques2_2_c.m as a

starting script. Include the plot as a figure in your write-up.]

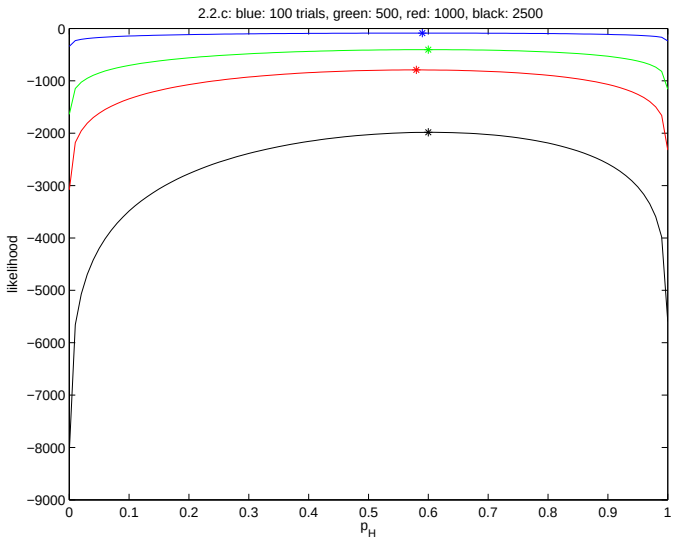

Figure 4: Question 2.c

As you increase σ^2, the MLE's performance would become worse overall. Also, an MLE based on smaller trial sizes would perform worse than an MLE based on large trial sizes (the more data, the better the estimator). In the limit as σ^2 becomes unimaginably large, the data gives no information about the underlying p_H, and the MLE (or any estimator) will perform poorly.
(+2 for the plot, +2 for the explanation)

(d) (3 points) Let us define another estimator, $\bar{p}_H$, as the number of O_i greater than 0.5 divided by N_{bits}. This estimator takes a threshold of the observed data, and is not the MLE. To compare $\hat{p}_H$ with $\bar{p}_H$, plot $\hat{p}_H$ vs. trial size (the same trial sizes as in 2.2.c) and $\bar{p}_H$ vs. trial size on the same axes. (There will be two curves on the same plot.) Comment on which estimator does a better job. What happens when you increase σ^2? [You can use the given hw2_ques2_2_d.m as a starting script. Include the plot as a figure in your write-up.]

Both estimators do a fairly good job, especially at high trial sizes (2,500 trials). When σ^2 increases modestly ($\sim$1), both estimators perform poorly for small trial sizes. However, the MLE will perform much better than $\bar{p}_H$ for large trial sizes. This is because the MLE is much better at accounting for outliers (i.e., O_i with a large noise contribution) than $\bar{p}_H$.
(+2 for the plot, +1 for the explanation)

3. Assume the same model in 2.2, $O_i = H_i + \epsilon_i$, $i = 1, \ldots, N_{\text{bits}}$, but now $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$, where the variance grows for each consecutively received bit.

(a) (3 points) Write out the form of the likelihood function $\mathcal{L}(O_1, \ldots, O_{N_{\text{bits}}}; p_H)$. Is the log-likelihood function concave?
The likelihood function takes the exact same form as 2.2.a, except to change σ^2 to σ_i^2.

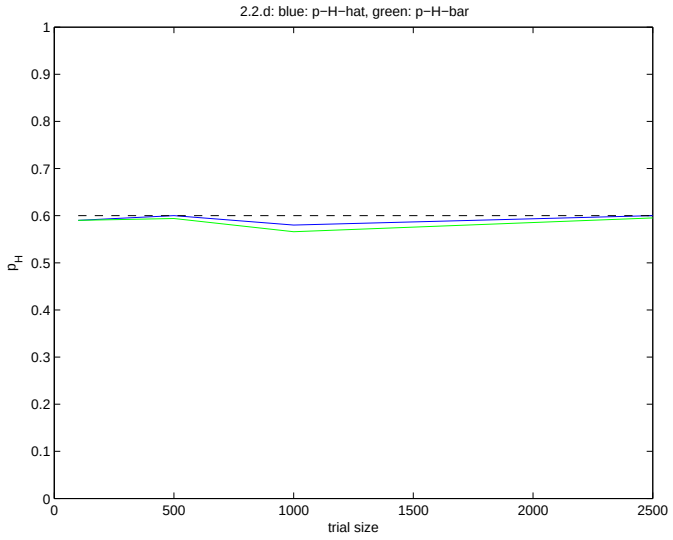

Figure 5: Question 2.2.d

So $\mathcal{L}(O_1, \ldots, O_{N_{\text{bits}}}; p_H) = \prod_{i=1}^{N_{\text{bits}}} \left[\mathcal{N}(O_i; 0, \sigma_i^2) + \left(\mathcal{N}(O_i - 1; 0, \sigma_i^2) - \mathcal{N}(O_i; 0, \sigma_i^2)\right) p_H\right]$
Since the double derivative also remains the same (only the αs and the βs would change), this function is still concave.
(+2 for the likelihood, +1 for concavity)

(b) (3 points) Repeat 2.2.c with the new model, where $p_H^* = 0.6$ and $\sigma_i^2 = \frac{i}{N_{\text{bits}}}$. Plot the $\log \mathcal{L}(O_1, \ldots, O_{N_{\text{bits}}}; p_H)$ vs. p_H for each trial size. (There will be 4 curves on the same plot.) [You can use the given hw2_ques2_3_b.m as a starting script. Include the plot as a figure in your write-up.]

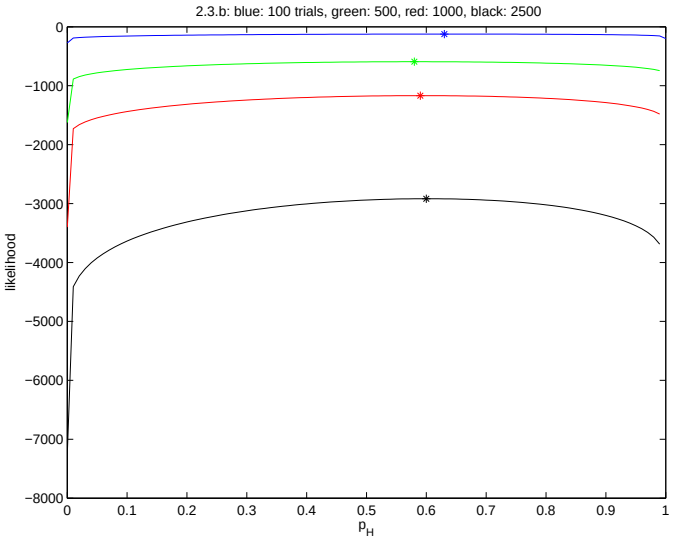

Figure 6: Question 3.b

(+3 for the plot)

(c) (3 points) Repeat 2.2.d with the new model. Plot p_H vs. trial size for both estimators. (There will be two curves on the same plot.) Comment on the differences between $\hat{p}_H$ and $\bar{p}_H$. [You can use the given hw2_ques2_3_c.m as a starting script. Include the plot as a figure in your write-up.]

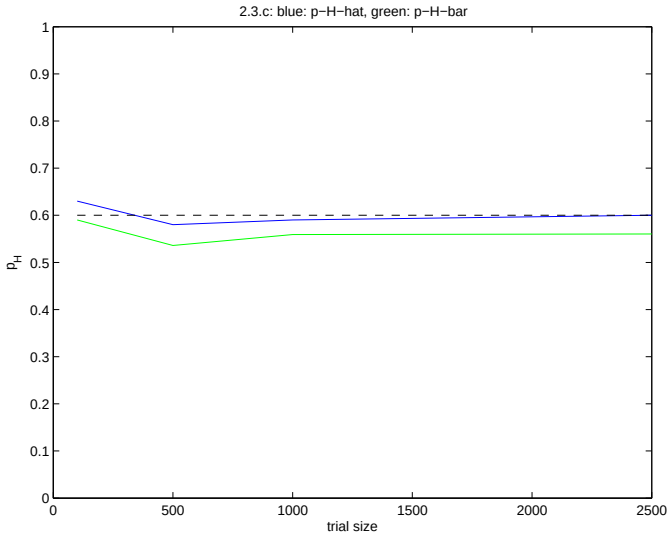

Figure 7: Question 3.c

$\hat{p}_H$ is less biased than $\bar{p}_H$ for the incrementally-increasing variance. This is because $\bar{p}_H$ equally weights every observation, while $\hat{p}_H$ is flexible enough to rely heavily on the first trials and all but ignore the last trials (which have high variance).
(+2 for the plot, +1 for the explanation)