ISTA 421/521 – Homework 3

Due: Tuesday, October 14, 10pm

20 pts total for Undergrads, 25 pts total for Grads

Emanuel Carlos de Alcantara Valente

Undergraduate

Instructions

In this assignment you are required to modify/write 2 scripts in python. Details of what you are to do are specified in problems 2 and 7, below.

Included in the homework 3 release are following sample scripts:

- `approx_expected_value.py` - This script demonstrates how to approximate an expected value through sampling. You will modify this code and submit your solution for problem 2.
- `gauss_surf.py` - This is provided for fun – it is not required for any problem here. It generates a 2d multivariate Gaussian and plots it as both a contour and surface plot.
- `predictive_variance_example.py` - This script demonstrates (a) generating and plotting error bars (predictive variance) and (b) sampling of model parameters from the cov$\{\widehat{\mathbf{w}}\}$ estimated from data. You will run this script in problem 6, and then use it as the basis for a script in problem 7.
- `w_variation_demo.py` - This script is also provided for fun and is not required for the assignment. (It also provides more example python code!) This implements the simulated experiment demonstrating the theoretical and empirical bias in the estimate of variance, $\widehat{\sigma^2}$, of the model variance, σ^2, as a function of the sample size used for estimation.

All problems require that you provide some "written" answer (in some cases also figures), so you will also submit a .pdf of your written answers. (You can use LaTeX or any other system (including handwritten; plots, of course, must be program-generated) as long as the final version is in PDF.)

The final submission will include (minimally) the two scripts and a PDF version of your written part of the assignment. You are required to create either a .zip or tarball (.tar.gz / .tgz) archive of all of the files for your submission and submit your archive to the d2l dropbox by the date/time deadline above.

NOTE: Problems 4 and 8 are required for Graduate students only; Undergraduates may complete them for extra credit equal to the point value.

(FCMA refers to the course text: Rogers and Girolami (2012), *A First Course in Machine Learning*. For general notes on using LaTeX to typeset math, see: `http://en.wikibooks.org/wiki/LaTeX/Mathematics`)

1. [2 points] Adapted from **Exercise 2.3** of FCMA p.90:

   Let Y be a random variable that can take any positive integer value. The likelihood of these outcomes is given by the Poisson pmf (probability mass function):

   $$p(y) = \frac{\lambda^y}{y!} e^{-\lambda} \tag{1}$$

   By using the fact that for a discrete random variable the pmf gives the probabilities of the individual events occurring and the probabilities are additive...

   (a) Compute the probability that $Y \leq 6$ for $\lambda = 8$, i.e., $P(Y \leq 6)$. Write a (very!) short python script to compute this value, and include a listing of the code in your solution.

   (b) Using the result of (a) and the fact that one outcome has to happen, compute the probability that $Y > 6$.

   Solution.

   a)

   Code Listing 1: `poisson.py` script

```
#!/usr/bin/python
import math, sys

def poisson_probability(y, lamb):
        sum = 0.0;
        for i in range(y + 1):
                sum += (math.pow(lamb, i) * math.exp(-lamb)) / math.factorial(i)
        return sum

y = 6
lamb = 8
prob = poisson_probability(y, lamb)
print "The Poisson probability is ", prob
```

```
[emanuel@localhost submit]$ python poisson.py
The Poisson probability is  0.313374277536
```

   As we saw above, $P(Y \leq 6) = 0.3133742$

   b)

   $P(Y > 6) = 1 - P(Y \leq 6) = 1 - 0.3133742 = 0.6866257$

2. [3 points] Adapted from **Exercise 2.4** of FCMA p.90:

   Let X be a random variable with uniform density, $p(x) = \mathcal{U}(a, b)$. Derive $\mathbf{E}_{p(x)}\{1 + 0.1x + 0.5x^2 + 0.05x^3\}$. Work out analytically $\mathbf{E}_{p(x)}\left\{1 + 0.1x + 0.5x^2 + 0.05x^3\right\}$ for $a = -10$, $b = 5$ (show the steps).

   The script `approx_expected_value.py` demonstrates how you use random samples to approximate an expectation, as described in Section 2.5.1 of the book. The script estimates the expectation of the function y^2 when $Y \sim \mathcal{U}(0, 1)$ (that is, y is uniformly distributed between 0 and 1). This script shows a plot of how the estimation improves as larger samples are considered, up to 100 samples.

   Modify the script `approx_expected_value.py` to compute a sample-based approximation to the expectation of the function $1+0.1x+0.5x^2+0.05x^3$ when $X \sim \mathcal{U}(-10, 5)$ and observe how the approximation improves with the number of samples drawn. Include a plot showing the evolution of the approximation, relative to the true value, over 3,000 samples.

Solution.

$\mathbf{E}_{p(x)}\left\{1+0.1x+0.5x^2+0.05x^3\right\} = \int_{-\infty}^{\infty}(1+0.1x+0.5x^2+0.05x^3)(\frac{1}{b-a})dx =$
$(\frac{1}{15})\int_{-10}^{5}(1+0.1x+0.5x^2+0.05x^3)dx =$
$(\frac{1}{15})(x+\frac{0.1x^2}{2}+\frac{0.5x^3}{3}+\frac{0.05x^4}{4})\|_{-10}^{5} = \frac{81.5625}{15} = 5.4375$

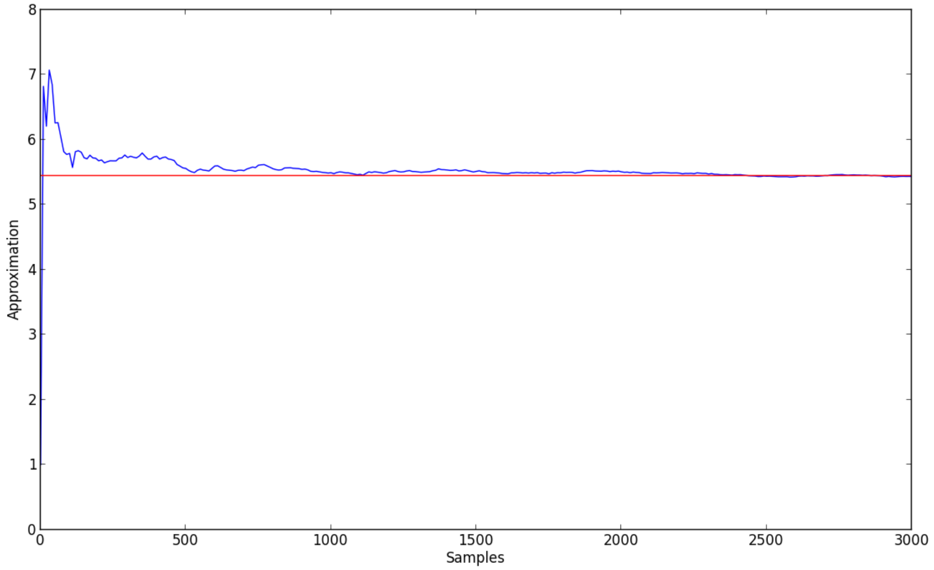


Figure 1: Evolution of the approximation, relative to the true value, over 3,000 samples.

3. [3 points] Adapted from **Exercise 2.5** of FCMA p.91:

Assume that $p(\mathbf{w})$ is the Gaussian pdf for a D-dimensional vector $\mathbf{w}$ given in

$$p(\mathbf{w}) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{w}-\mu)^{\top}\mathbf{\Sigma}^{-1}(\mathbf{w}-\mu)\right\}.$$

By expanding the vector notation and re-arranging, show that using $\mathbf{\Sigma} = \sigma^2\mathbf{I}$ as the covariance matrix assumes independence of the D elements of $\mathbf{w}$. You will need to be aware that the determinant of a matrix that only has entries on the diagonal ($|\sigma^2\mathbf{I}|$) is the product of the diagonal values and that the inverse of the same matrix is constructed by simply inverting each element on the diagonal. (Hint, a product of exponentials can be expressed as an exponential of a sum. Also, just a reminder that $\exp\{x\}$ is e^x.)

Solution.

First, let us define the matrices:

$\mathbf{\Sigma} = \sigma^2\mathbf{I} \implies \Sigma^{-1} = \frac{1}{\sigma^2}\mathbf{I}$

Therefore,
$|\mathbf{\Sigma}|^{1/2} = (\Pi_{d=1}^{D}\sigma^2)^{\frac{1}{2}} = \Pi_{d=1}^{D}(\sigma^2)^{\frac{1}{2}}$

Now, using the properties and hints presented in the beginning of this exercise, we will show how the Gaussian pdf for a D-dimensional vector will be:

$$p(\mathbf{w}) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{w}-\mu)^\top \mathbf{\Sigma}^{-1}(\mathbf{w}-\mu)\right\} =$$

$$= \frac{1}{(2\pi)^{D/2}\Pi_{d=1}^{D}(\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{w}-\mu)^\top \frac{1}{\sigma^2}\mathbf{I}(\mathbf{w}-\mu)\right\} =$$

We can (see the proof in exercise 4) write:

$$(\mathbf{w}-\mu)^\top \mathbf{\Sigma}(\mathbf{w}-\mu) = \prod_{d=1}^{n}(w_d-\mu_d)^2\sigma_d^2$$

So:

$$p(\mathbf{w}) = \frac{1}{\Pi_{d=1}^{D}(2\pi)^{\frac{1}{2}}\Pi_{d=1}^{D}(\sigma^2)^{\frac{1}{2}}}\Pi_{d=1}^{D} \exp\left\{-\frac{1}{2\sigma^2}(w_d-\mu_d)^2\right\}$$

$$p(\mathbf{w}) = \prod_{d=1}^{D}\frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma^2}(w_d-\mu_d)^2\right\}$$

4. [2 points; **Required only for Graduates**] Adapted from **Exercise 2.6** of FCMA p.91:

Using the same setup as in Problem 4, see what happens if we use a diagonal covariance matrix with different elements on the diagonal, i.e.,

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_D^2 \end{bmatrix}$$

Solution.

First, we will show that this expression $\exp\left\{-\frac{1}{2}(\mathbf{w}-\mu)^\top \mathbf{\Sigma}^{-1}(\mathbf{w}-\mu)\right\}$ can written similarly if compared the previous exercise.

If we analize the dimensions of the product, we will know that the result is going to be an scalar. So, let us consider 2 dimensions:

$$(\mathbf{w}_{2\times 2}-\mu_{2\times 2})^\top \mathbf{\Sigma}_{2\times 2} = \begin{bmatrix} w_0-\mu_0 & w_1-\mu_1 \end{bmatrix} \begin{bmatrix} \sigma_0^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix} = \begin{bmatrix} (w_0-\mu_0)\sigma_0^2 & (w_1-\mu_1)\sigma_1^2 \end{bmatrix}$$

Therefore,

$$(\mathbf{w}_{2\times2} - \mu_{2\times2})^\top \mathbf{\Sigma}_{2\times2}(\mathbf{w}_{2\times2} - \mu_{2\times2}) = \begin{bmatrix}(w_0-\mu_0)\sigma_0^2 & (w_1-\mu_1)\sigma_1^2\end{bmatrix}\begin{bmatrix}(w_0-\mu_0)\\(w_1-\mu_1)\end{bmatrix} =$$

$$= \left[(w_0-\mu_0)^2\sigma_0^2 + (w_1-\mu_1)^2\sigma_1^2\right] = \prod_{d=1}^{2}(w_d-\mu_d)^2\sigma_d^2$$

Consequently, we can generalize it for n dimensions:

$$(\mathbf{w}-\mu)^\top \mathbf{\Sigma}(\mathbf{w}-\mu) = \prod_{d=1}^{n}(w_d-\mu_d)^2\sigma_d^2$$

and

$$(\mathbf{w}-\mu)^\top \mathbf{\Sigma}^{-1}(\mathbf{w}-\mu) = \prod_{d=1}^{n}\frac{1}{\sigma_d^2}(w_d-\mu_d)^2$$

So, as the indexes are the same as the previous exercise, we can follow the same steps of it. The expression will be:

$$p(\mathbf{w}) = \prod_{d=1}^{D}\frac{1}{(2\pi\sigma_d^2)^{\frac{1}{2}}}\exp\left\{-\frac{1}{2\sigma_d^2}(w_d-\mu_d)^2\right\}$$

5. [4 points] Adapted from **Exercise 2.9** of FCMA p.91:

Assume that a dataset of N binary values, $x_1, ..., x_n$, was sampled from a Bernoulli distribution, and each sample x_i is independent of any other sample. Explain why this is *not* a Binomial distribution. Derive the maximum likelihood estimate for the Bernoulli parameter.

Solution.

The Binomial distribution defines the probability of observing a certain numbers of trials whilst the Bernoulli distribution gives the probability of just one.

Consider p the success probability, so we will have:

$$\mathbf{L}(p,\mathbf{x}) = \prod_{i=1}^{n}p^{x_i}(1-p)^{1-x_i} = p^{\sum_{i=1}^{n}x_i}(1-p)^{\sum_{i=1}^{n}(1-x_i)}$$

We will apply the natural logarithmic function in our likelihood function:

$$log(\mathbf{L}(p,\mathbf{x})) = \sum_{i=1}^{n}x_i log(p) + \sum_{i=1}^{n}(1-x_i)log(1-p)$$

Taking derivatives:

$$\frac{\partial log(\mathbf{L}(p,\mathbf{x}))}{\partial p} = \frac{(1-p)\sum_{i=1}^{n} x_i - p\sum_{i=1}^{n}(1-x_i)}{p(1-p)}$$

and setting to zero:

$$\frac{(1-p)\sum_{i=1}^{n} x_i - p\sum_{i=1}^{n}(1-x_i)}{p(1-p)} = 0 \iff p = \frac{1}{n}\sum_{i=1}^{n} x_i = \overline{\mathbf{x}}$$

6. [3 points] Adapted from **Exercise 2.12** of FCMA p.91:

   Familiarize yourself with the provided script `predictive_variance_example.py`. When you run it, it will generate a dataset and then remove all values for which $-2 \leq x \leq 2$. Observe the effect this has on the predictive variance in this range. Plot (a) the data, (b) the error bar plots for model orders 1, 3, 5 and 9, and (c) the sampled functions for model orders 1, 3, 5 and 9. You will plot a total of 9 figures. Include a caption for each figure that qualitatively describes what the figure shows. Also, clearly explain what removing the points has done in contrast to when they're left in.

   Solution.

   The figures are Figure 2 and Figure 3.

   They represent those shapes because by removing the points we are increase the uncertainty in that specific region. That is why the errors bars are bigger and the lines from models are a little far from one another in that specific region.

7. [5 points]

   In this exercise, you will create a simple demonstration of how model bias impacts variance, similar to the demonstration in class. Using the same true model in the script `predictive_variance_example.py`, that is $t = 1 + 0.1x + 0.5x^2 + 0.05x^3$, generate 20 data sets, each consisting of 25 samples from the true function (using the same range of $x \in [-12.0, 5.0]$). Then, create a separate plot for each of the model polynomial orders 1, 3, 5 and 9, in which you plot the true function in red and each of the best fit functions of that model order to the 20 data sets. You will therefore produce four plots. The first will be for model order 1 and will include the true model plotted in red and then 20 curves, one each for an order 1 best fit model for each of the 20 data set, for all data sets. The second plot will repeat this for model order 3, and so on. You can use any of the code in the script `predictive_variance_example.py` as a guide. Describe what happens to the variance in the functions as the model order is changed. (tips: plot the true function curve last, so it is plotted on top of the others; also, use `linewidth=3` in the plot fn to increase the line width to make the curve stand out more.)

   Solution.

   As we saw in the class:

   $\mathbf{E}_{(\mathbf{p}(\mathbf{t}|\mathbf{X},\mathbf{w},\sigma^{2}})} = \hat{\sigma^2}\left(1 - \frac{D}{N}\right)$

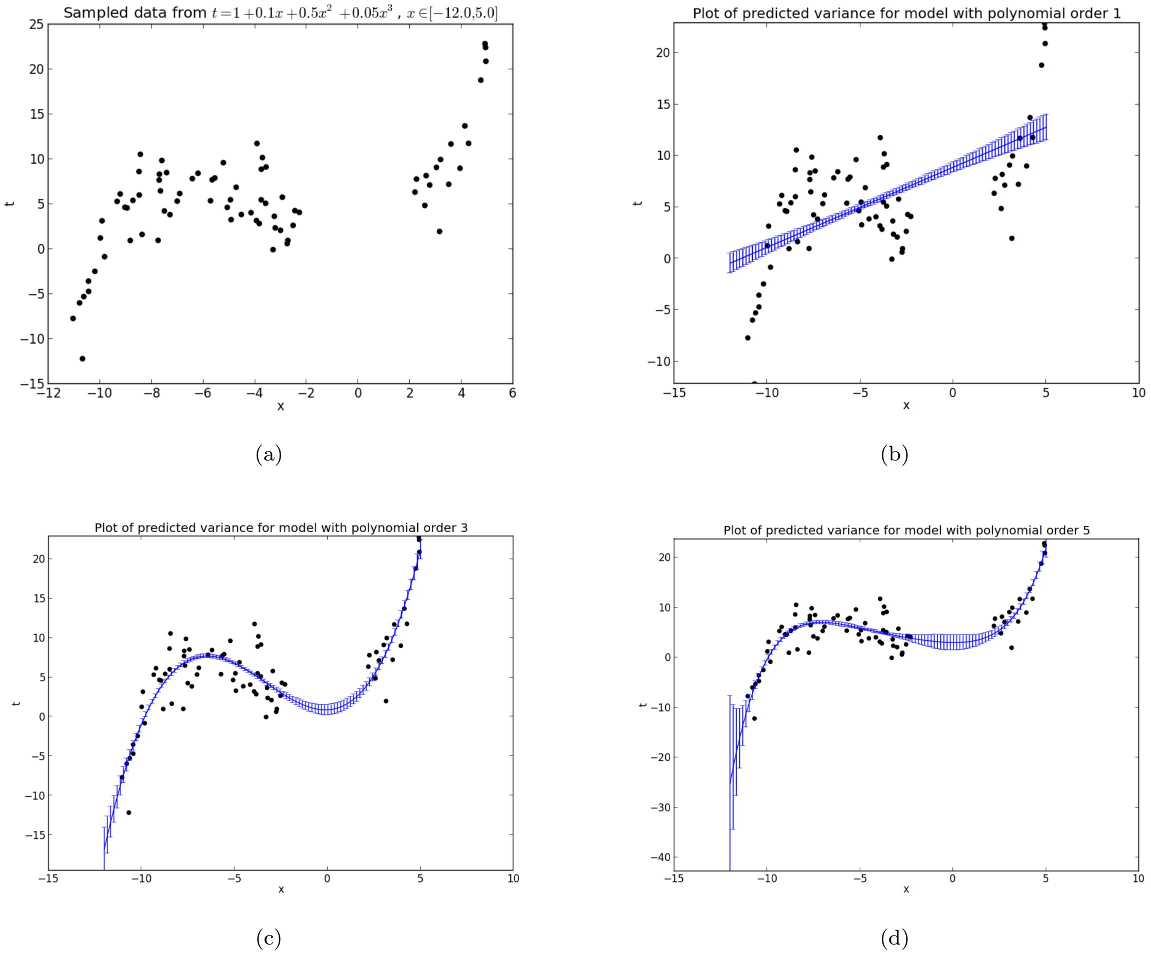
   as more we increase D for (in this case) a fixed elements, the estimated variance will be more biased.

   The plots are in the Figure 4.

8. [3 points; **Required only for Graduates**] Adapted from **Exercise 2.13** of FCMA p.92:

   Compute the Fisher Information Matrix for the parameter of a Bernoulli distribution.

   Solution.



(a)

(b)

(c)

(d)

Figure 2: Exercise 6 - Figures

We have seen that the Fisher information Matrix is computed as the expected value of the matrix of second derivatives of the log likelihood. So we will have:

$$I = \mathbf{E}_{Bern(x)}\left\{-\frac{\partial^2 log(x|p)}{\partial^2 p}\right\}$$

It can be written as:

$$I = -\int\left(\frac{\partial^2 log(x|p)}{\partial^2 p}\right) log(x|p)dx$$

We know that:

$$Bern(x|p) = p^x(1-p)^{1-x}$$

Therefore:

$$log(Bern(x|p)) = xlog(p) + (1-x)log(1-p)$$

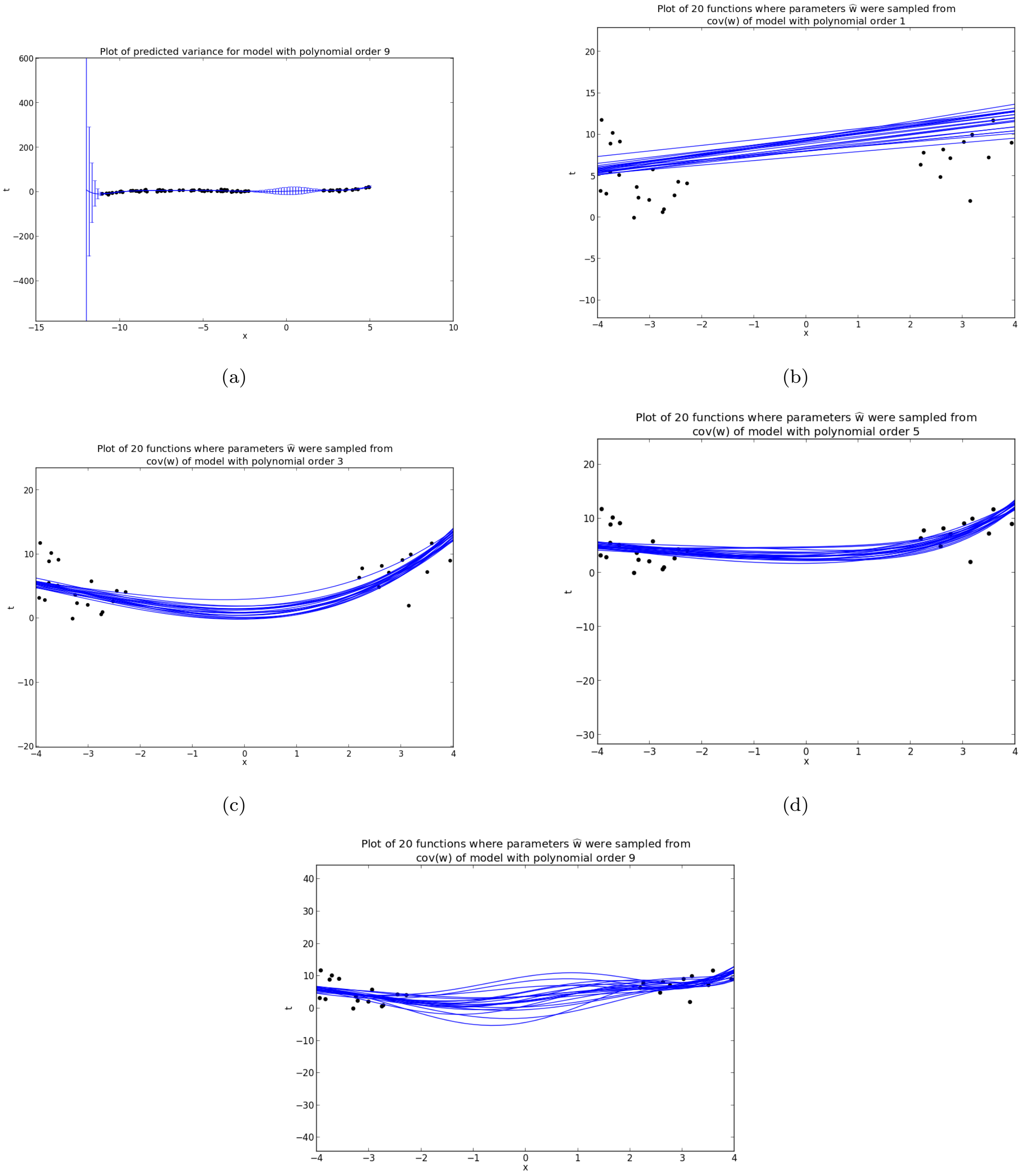
$$\frac{\partial log(Bern(x|p))}{\partial p} = \frac{x}{p} - \frac{(1-x)}{(1-p)}$$

$$\frac{\partial^2 log(Bern(x|p))}{\partial^2 p} = \frac{-x}{p^2} - \frac{(1-x)}{(1-p)^2}$$

The $\mathbf{E}_{Bern(x)} = p$

The Fisher Information Matrix will be:

$$I(x|p) = \frac{p}{p^2} - \frac{1-p}{(1-p)^2} = \frac{1}{p} - \frac{1}{(1-p)} = \frac{1}{p(1-p)}$$



Plot of predicted variance for model with polynomial order 9

(a)

Plot of 20 functions where parameters $\widehat{w}$ were sampled from cov(w) of model with polynomial order 1

(b)

Plot of 20 functions where parameters $\widehat{w}$ were sampled from cov(w) of model with polynomial order 3

(c)

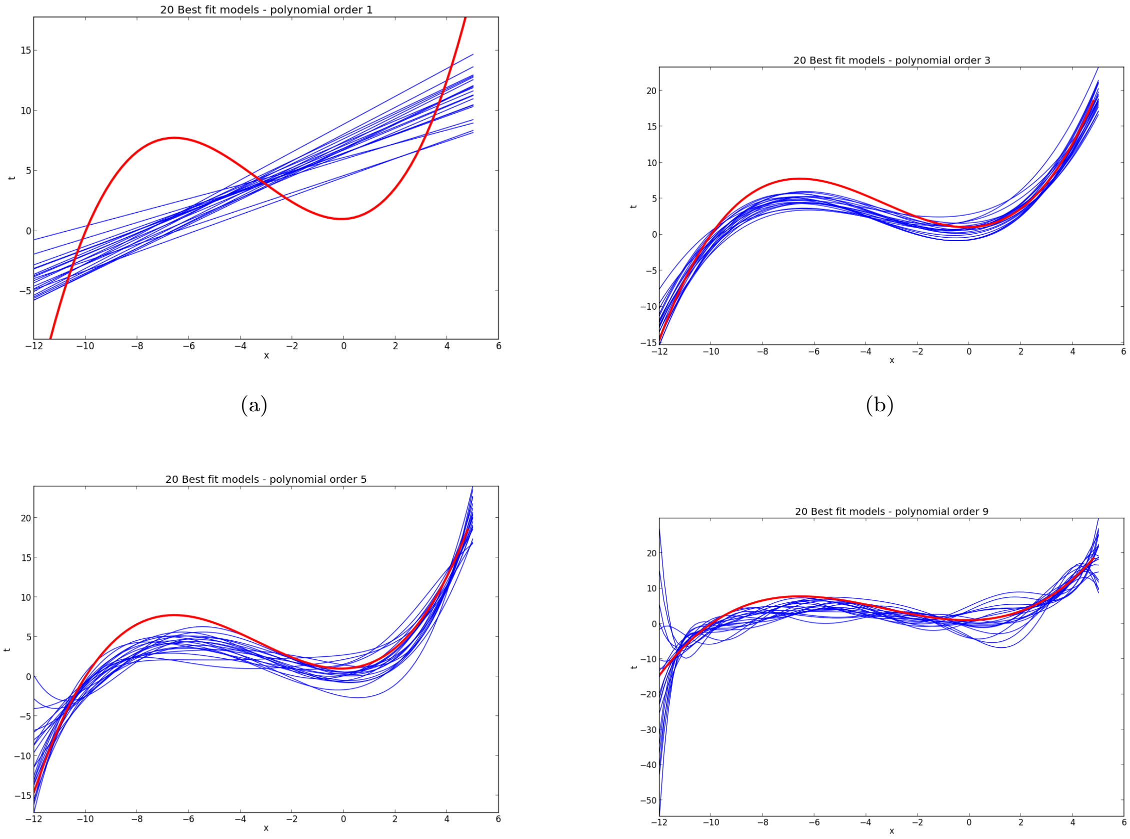
Plot of 20 functions where parameters $\widehat{w}$ were sampled from cov(w) of model with polynomial order 5

(d)

Plot of 20 functions where parameters $\widehat{w}$ were sampled from cov(w) of model with polynomial order 9

(e)

Figure 3: Exercise 6 - Figures (cont)



(a)

(b)

(c)

(d)

Figure 4: Exercise 7 - 20 Best fit models for order 1, 3, 5, and 9