ISTA 421/521 – Homework 3

Due: Friday, October 5, 5pm

20 pts total for Undergrads, 25 pts total for Grads

Ken Youens-Clark

Graduate

Instructions

In this assignment you are required to modify/write 3 scripts in python. Details of what this will involve are specified in problems 2, 5 and 6, 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.
- `predictive_variance.py` - This script is partially implemented; you will need to fill in two functions (as described in problem 5). When you run it, it demonstrates (a) generating and plotting error bars (predictive variance) and (b) sampling of model parameters from the $\texttt{cov}\{\widehat{\mathbf{w}}\}$ estimated from data. You will also use this script as the basis for a script you will write for problem 6.
- `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.
- `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, $\widehat{\sigma^2}$, 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 3 and 7 are required for Graduate students only; Undergraduates may complete them for extra credit equal to the point value.

(FCML refers to the course text: Rogers and Girolami (2016), *A First Course in Machine Learning, Second Edition*. 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 FCML:

   Let Y be a random variable that can take any non-negative 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 $Y \geq 5$ and $Y \leq 10$ for $\lambda = 7$, i.e., $P(5 \leq Y \leq 10)$. 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 $Y < 5$ or $Y > 10$.

   Solution.

2. [3 points] Adapted from **Exercise 2.4** of FCML:

   Let X be a random variable with uniform density, $p(x) = \mathcal{U}(a, b)$.
   Work out analytically $\mathbf{E}_{p(x)} \left\{35 + 3x - 0.5x^3 + 0.05x^4\right\}$ for $a = -4$, $b = 10$ (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 FCML. 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 1000 samples.

   Modify the script `approx_expected_value.py` to compute a sample-based approximation to the expectation of the function $35+3x-0.5x^3+0.05x^4$ when $X \sim \mathcal{U}(-4, 10)$ 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 5,000 samples.

   Solution.

$$\begin{aligned}
p(x) = \mathcal{U}(a, b) &= \frac{1}{b-a} = \frac{1}{10-(-4)} = \frac{1}{14} \\
\mathbf{E}_{p(x)} \left\{35 + 3x - 0.5x^3 + 0.05x^4\right\} &= \int_{x=-4}^{10} 35 + 3x - 0.5x^3 + 0.05x^4 p(x) dx \\
&= \int_{x=-4}^{10} \frac{35 + 3x - 0.5x^3 + 0.05x^4}{14} dx \\
&= \left[\frac{35x + \frac{3x^2}{2} - \frac{x^4}{8} + \frac{x^5}{25}}{14} \right]_{-4}^{10} \\
&= \frac{65}{14} - \frac{67.8}{14} \\
&= \frac{-2.8}{14} \\
&= -0.2
\end{aligned} \tag{2}$$

3. [2 points; **Required only for Graduates**] Adapted from **Exercise 2.6** of FCMA:

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

$$p(\mathbf{w}) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{w}-\boldsymbol{\mu})^\top \mathbf{\Sigma}^{-1}(\mathbf{w}-\boldsymbol{\mu})\right\}. \tag{3}$$

Suppose we use a diagonal covariance matrix with different elements on the diagonal, i.e.,

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

Does this assume independence of the D elements of $\mathbf{w}$? If so, show how by expanding the vector notation of Eqn. 4 and re-arranging. You will need to be aware that the determinant of a matrix that only has entries on the diagonal 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.

Yes, the elements of $\mathbf{w}$ are independent.

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

$$\mathbf{\Sigma}^{-1} = \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma_2^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sigma_D^2} \end{bmatrix}$$

$$\begin{aligned} p(\mathbf{w}) &= \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{w}-\boldsymbol{\mu})^\top \mathbf{\Sigma}^{-1}(\mathbf{w}-\boldsymbol{\mu})\right\}. \\ &= \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}\sum_{d=1}^{D}\frac{1}{\sigma_d^2}(w_d-\mu_d)^2\right\} \\ &= \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2\sigma_d^2}\sum_{d=1}^{D}(w_d-\mu_d)^2\right\} \\ &= \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \prod_{d=1}^{D}\exp\left\{-\frac{1}{2\sigma_d^2}(w_d-\mu_d)^2\right\} \\ p(\mathbf{w}) &= \prod_{d=1}^{D}\frac{1}{(2\pi\sigma_d^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma_d^2}(w_d-\mu_d)^2\right\} \end{aligned}$$

4. [4 points] Adapted from **Exercise 2.9** of FCML:

   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.

5. [6 points] Adapted from **Exercise 2.12** of FCML:

   Familiarize yourself with the provided script `predictive_variance.py`. It is mostly implemented, but you will have to fill in the details for two functions:

   - `calculate_prediction_variance`, which calculates the *variance* for a prediction at x_{new} given the design matrix, $\mathbf{X}$, the estimated parameters, $\mathbf{w}$, and target responses, $\mathbf{t}$.
   - `calculate_cov_w`, which calculates the estimated covariance of $\mathbf{w}$ given the design matrix, $\mathbf{X}$, the estimated parameters, $\mathbf{w}$, and target responses, $\mathbf{t}$.

   Once implemented, then you can run the script.

   When you run the script, it will generate a dataset based on a function (implemented in `true_function`) and then remove all values for which $-2 \leq x \leq 2$. Three groups of plots will be generated:

   (a) First is a plot of the data (this will be generated by Part 5a of the script, starting on line 70).

   (b) Next, the script will plot the error bar plots for predictions of values for model orders 1, 3, 5 and 9 (this will be generated by Part 5b of the script, starting on line 145).

   (c) Finally, in Part 5c (starting line 197), the script samples model parameters $\mathbf{w}$ from the covariance $\text{cov}(w)$ and plots the resulting functions (again, for model orders 1, 3, 5 and 9).

   In total, you will plot 9 figures. You must include the plots in your submission and do the following: Include a caption for each figure that qualitatively describes what the figure shows; contrast the figures within group (b) with each other; do the same for group (c). Also, clearly explain what removing the points has done in contrast to when they're left in.

   Solution.

6. [5 points]

   In this exercise, you will create a new script that demonstrates how model bias impacts variance, similar to the demonstration in Lecture 9 (starting slide 28). In your submission, you will name the file for your script `model_bias_variance.py`. You will copy the functions `true_function` and `sample_from_function` from `predictive_variance.py`. Using `true_function` (which computes $t = 5x + x^2 - 0.5x^3$), generate 20 data sets, each consisting of 25 samples from the true function (using the same range of $x \in [-4.0, 5.0]$ and `noise_var` $= 6$), using the function `sample_from_function`. 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 (in blue) of that model order to each of 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 blue 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.py` as a guide. In your written answer, describe what happens to the variance in the functions in the plots as the model order is changed. (tips: plot the red 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 red true model curve stand out more.)

   Solution.

7. [3 points; **Required only for Graduates**] Adapted from **Exercise 2.13** of FCML:

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

   Solution.