The first assignment is meant to exercise your probability and programming abilities. For both parts, you will need to produce a report which you will submit online. In addition, for the programming portion you will need to submit your code in a separate ZIP file. Both parts are due **Friday, Feburary 24 at 11:59pm**. Late submissions will not be accepted. Submissions will be handled through Moodle.

Continuous Probabilistic Models

1. The Naive Bayes assumption for estimation assumes that the dimensions in a given estimation problem are (conditionally) independent. For classification we used this to dramatically reduce the number of parameters that we needed to estimate. The Naive Bayes assumption can also be used for continuous distributions. Assume $x \in \mathbb{R}^D$ is distributed according to a multivariate Gaussian distribution, i.e., $p(x) = \mathcal{N}(x|\mu, \Sigma)$. Given a set of samples $\mathcal{D} = \{x^{(1)}, \ldots, x^{(N)}\}$, derive the Maximum Likelihood estimate of μ and Σ under the Naive Bayes assumption (i.e., that $p(x) = \prod_{i=1}^{D} p(x_i)$). Be sure to show your derivation in detail. What specific structure does Σ have and why?

2. Consider the problem of linear regression where the output is a vector instead of a scalar. That is, let the input be $x \in \mathbb{R}^D$ and the output be $y \in \mathbb{R}^M$. Assume a Gaussian model of the output

$$p(y|x, W) = \mathcal{N}(y|Wx, \Sigma)$$

where $W \in \mathbb{R}^{M \times D}$ is a weight matrix and $\Sigma \in \mathbb{R}^{M \times M}$ is the covariance matrix of the observations.

a) Assume that the covariance matrix has the specific form of a scale identity matrix, that is $\Sigma = \sigma^2 I$ for a scalar variance σ^2. Derive the Maximum Likelihood estimate of W. Prove that the estimation of each row of W depends only on the corresponding dimensions of y. That is, prove that the estimate of row i of W only uses the values of the ith dimension of the output vectors y.

b) Now instead, assume that the covariance matrix is diagonal

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

for a set of scalar variances $\sigma_1^2, \ldots, \sigma_M^2$. Again derive the Maximum Likelihood estimate of W and prove that the estimation of each row of W depends only on the corresponding dimensions of y. That is, prove that the estimate of row i of W only uses the values of the ith dimension of the output vectors y.

c) In the most general case, Σ will have non-zero off diagonal entries. Explain what this implies about the relationship between the output dimensions. Will the estimates of the rows of W remain independent of each other? Why or why not?

3. Computing the mean vector and covariance matrix of a set of samples is a common problem. The standard equations for computing the mean and covariance from N samples is

$$\begin{aligned}\mu_N &= \frac{1}{N}\sum_{i=1}^{N} x_i \\ \Sigma_N &= \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_N)(x_i - \mu_N)^T\end{aligned}$$

The direct implementation of these equations would require two passes through the data, once to compute μ_N and once to compute Σ_N. However, sometimes this isn't possible or convenient. For instance, if N is very large or the data is streaming in and can't be stored. In those cases we would like a recursive form of the equations so that we can compute μ_N and Σ_N from μ_{N-1} and Σ_{N-1}.

a) Prove that the mean vector can be updated as

$$\mu_N = \mu_{N-1} + \frac{1}{N}(x_N - \mu_{N-1})$$

b) Using the result from a), derive a method for recursively computing Σ_N. That is, you should derive a formula or set of formulas which allow one to compute Σ_N from Σ_{N-1} and x_N without needing to use the values of $x_1, \ldots, x_{N-1}$. *Hint:* You may need to compute an auxilliary quantity which you can then combine with μ_N in order to compute Σ_N.

4. Consider the following model

- $x \in \mathbb{R}^N$ has a multivariate Gaussian distribution, i.e., $p(x) = \mathcal{N}(x|\mu_x, \Sigma_x)$, with mean μ_x and variance Σ_x
- $\eta \in \mathbb{R}^M$ has a multivariate Guassian distribution with mean 0 and covariance Σ_η, i.e., $p(\eta) = \mathcal{N}(\eta|0, \Sigma_\eta)$
- $y = Ax + b + \eta$ where $A \in \mathbb{R}^{M\times N}$ and $b \in \mathbb{R}^M$ are constants

a) Assuming that x and η are independent, derive the mean and covariance of y.

b) Assume x and η are dependent but jointly Gaussian such that $z = (x^T, \eta^T)^T$

$$p(z) = \mathcal{N}(z|\mu_z, \Sigma_z)$$

where

$$\mu_z = \begin{bmatrix} \mu_x \\ 0 \end{bmatrix}$$

and

$$\Sigma_z = \begin{bmatrix} \Sigma_x & \Sigma_{x\eta} \\ \Sigma_{x\eta}^T & \Sigma_\eta \end{bmatrix}$$

Derive the mean and covariance of y.

Regression

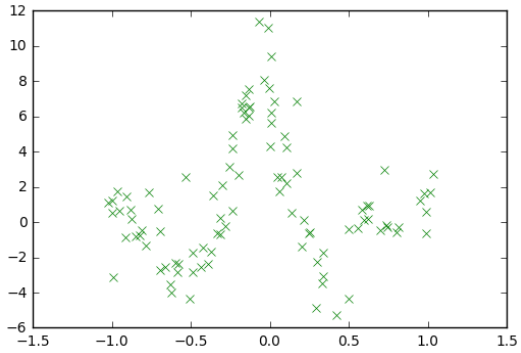
In this assignment we will consider the problem of basis function regression. You'll explore issues of overfitting, generalization and Bayesian prediction. We'll do this by considering a few different datasets and using a set of polynomial basis functions. I.E., $\phi_i(x) = x^{i-1}$ for $i = 1, \ldots, D$ and we will consider different values of D.

What to hand in: As with assignment 1, you will submit a report with your results and analysis. In addition, you should also submit a ZIP file containing all of your source code implementing the methods.

ML and MAP Estimation

Step 1

The data is stored in two files, DATASET1_INPUTS.TXT and DATASET1_OUTPUTS.TXT which contain the input values (i.e., values of x) and the output values (i.e., values of y) respectively. These files are simple text files which can be loaded with the LOADTXT function in NumPy and the LOAD function in Matlab/Octave. Once you've loaded the data, plot it to produce a plot that looks like



What to include in your report: Include your plot of the data.

Step 2

Using all the available data, fit polynomial functions for values of $D = 1, \ldots, 20$ using the Maximum Likelihood estimate of w. For each value of D compute the *mean squared error* on the training data $s_D^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - f_D(x_i))^2$ where $f_D(x) = \sum_{i=1}^{D} w_i \phi_i(x)$ is the estimated function. Plot the mean squared error as a function of D.

What to include in your report: Include the plot you generated. Based on this plot, what value of D do you think is best and why?

Step 3

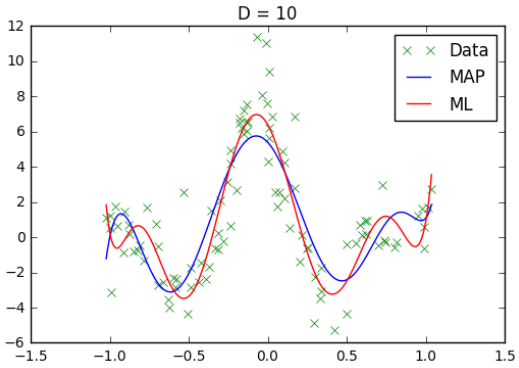
Repeat step 2, but this time use the MAP estimate of w with $\lambda = 0.001$. Plot the mean squared error as a function of D for both MAP and ML estimates.

What to include in your report: Include the plot with the MSE for both MAP and ML estimation.

After a point the MSE for the MAP case stops decreasing even though the ML case continues to decrease. Explain why this happens.

Step 4

For each value of $D = 1, \ldots, 20$ plot the estimate function for both MAP and ML estimates along with the data. For instance, for $D = 10$ you should produce a plot which looks like this.



What to include in your report: Looking at the MAP function in these plots subjectively, what value of D do you think is best and why? Include a plot for that value of D.

In Step 2 you identified a "best" value of D according to the MSE on the training data with the ML estimate. Include the plot you created for this value of D. Looking at the plot, does the ML estimated function look like a good fit for this data? What about the MAP estimated functions?

Describe the differences between the MAP and ML estimated functions as D increases. Use these differences to argue for the use of MAP estimation in general.

Finally, along with the plots above, include plots for $D \in \{5, 10, 15, 20\}$.

Step 5

Implement 10-fold cross validation. That is, randomly divide the data into ten pieces. Then take the data in each piece in turn to be the validation set while the data in the remaining 9 pieces are used as the training data. Perform the MAP estimation on each set of training data and compute the MSE on the held-out validation set. This will give you 10 MSE values: average these to get a single final MSE value for each value of D. Do this for $D = 1, \ldots, 20$ and plot the 10-fold cross-validate mean squared error as a function of D.

What to include in your report: Include the plot you generated.

Based on this curve, if you had to pick a single value of D, what would it be? Is this choice an obvious one? That is, based on the curve is it clearly obvious that a single value is the best?

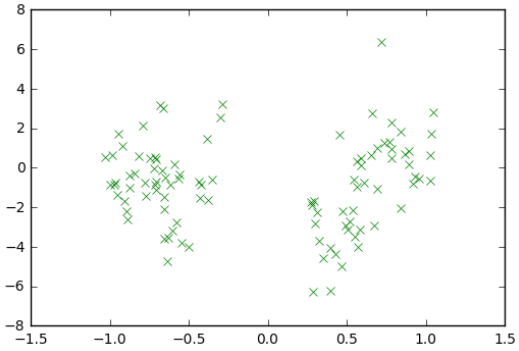
Include the plot you generated for this value of D in Step 4 if it's not already included in the report.

Bayesian Regression

MAP and ML estimation are good for providing point estimates. That is, given a value of x they will provide a prediction for the value of y. However, they aren't able to provide an estimate of uncertainty with that prediction. Bayesian regression makes this possible by retaining the uncertainty inherent in the estimation of the parameters w and passing it on during prediction time.

Step 6

The data is stored in two files, DATASET2_INPUTS.TXT and DATASET2_OUTPUTS.TXT which contain the input values (i.e., values of x) and the output values (i.e., values of y) respectively. They can be loaded in the same way as before. Again, the first step is to plot the data.



What to include in your report: Include your plot of the data.

Step 7

Implement Bayesian linear regression using the same polynomial basis functions as described above. Specifically, the model looks like

$$\begin{aligned} p(y|x,w) &= \mathcal{N}(y|w^T\phi(x),\sigma^2) \\ p(w) &= \mathcal{N}(w|0,\tau^2 I) \end{aligned}$$

and your goal is to compute

$$\begin{aligned} p(y|x,\mathcal{D}) &= \int p(y|x,w)p(w|\mathcal{D})dw \\ &= \int \mathcal{N}(y|w^T\phi(x),\sigma^2)\mathcal{N}(w|\mu_w,\Sigma_w)dw \\ &= \mathcal{N}(y|\mu_{\mathcal{D}}(x),\sigma^2_{\mathcal{D}}(x)) \end{aligned}$$

where $\mathcal{D}$ is the training data,

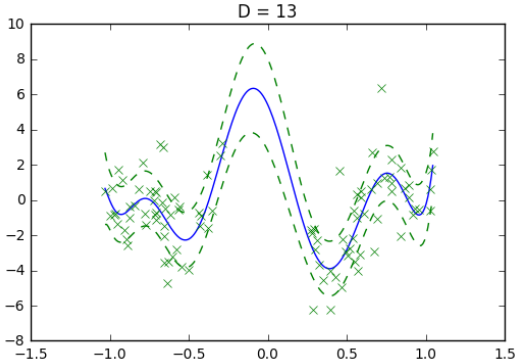
$$\mu_w = \sigma^{-2}\Sigma_w\Phi^T Y$$
$$\Sigma_w = (\tau^{-2}I + \sigma^{-2}\Phi^T\Phi)^{-1}$$

are the mean and variance of the posterior of the weights, Y is a column vector of the training data outputs, Φ is a matrix of the basis functions evaluated on the training inputs and

$$\mu_{\mathcal{D}}(x) = \mu_w^T\phi(x)$$
$$\sigma^2_{\mathcal{D}}(x) = \sigma^2 + x^T\Sigma_w x$$

are the predicted mean and variance at an input point x. More details on this model were presented in lecture and can be found in Section 7.6.2 of the textbook.

Using $D = 13$ with observation standard deviation $\sigma = 1.5$ and prior standard deviation $\tau = 1000$ compute the predicted mean $\mu_{\mathcal{D}}(x)$ and standard deviation $\sigma^2_{\mathcal{D}}(x)$ at a range of input values. Use these values to plot the mean function as well as curves which indicate plus or minus one standard deviation of the prediction at that value of x. That is, you should plot one curve which is $\mu_{\mathcal{D}}(x)$ and two other curves which are $\mu_{\mathcal{D}} \pm \sigma_{\mathcal{D}}(x)$. Your plot should look something like this



In a separate plot, plot the predicted standard deviation as a function of x for $x \in [-1.1, 1.1]$.

What to include in your report: Include both of the plots generated for this part.

When using Bayesian regression with this model, the standard deviation fluxuates as a function of x. Explain this fluctuation in terms of the data. In particular, draw connections between points where the standard deviation is particularly high and characteristics of the data.

The standard deviation also as a minimum value. Explain what this value is and why.

Imagine the values of y that were used in training were badly corrupted but the values of x were unchanged. How would this impact the predicted standard deviation shown? Does this seem reasonable? Explain which modelling assumption is responsible for this behaviour.