Assignment 1: Regression

Due September 29 at 11:59pm
100 marks total

This assignment is to be done individually.

Important Note: The university policy on academic dishonesty (cheating) will be taken very seriously in this course. You may not provide or use any solution, in whole or in part, to or by another student.

You are encouraged to discuss the concepts involved in the questions with other students. If you are in doubt as to what constitutes acceptable discussion, please ask! Further, please take advantage of office hours offered by the instructor and the TA if you are having difficulties with this assignment.

DO NOT:

- Give/receive code or proofs to/from other students
- Use Google to find solutions for assignment

DO:

- Meet with other students to discuss assignment (it is best not to take any notes during such meetings, and to re-work assignment on your own)
- Use online resources (e.g. Wikipedia) to understand the concepts needed to solve the assignment

---

Question 1 (20 marks)

In lecture we went over an example of modeling coin tossing – estimating a parameter that is the probability the coin comes up heads.

Consider instead the problem of estimating the mean μ of a 1-d Gaussian distribution. Assume the variance σ^2 is known. Suppose you are given data $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$, where each $x_i \in \mathbb{R}$.

1. (8 marks) Derive the maximum likelihood estimate μ_{ML}.
2. (8 marks) Derive the maximum a posteriori estimate μ_{MAP}. For the prior on μ, use a Gaussian distribution with mean a, and the same variance $\sigma_p^2 = \sigma^2$. I.e. $p(\mu) = \mathcal{N}(\mu|a, \sigma_p^2)$.
3. (2 marks) Provide an intuitive explanation for what the prior mean a does in this case. (Write one or two sentences.)
4. (2 marks) What would be the effect of using a variance for the prior σ_p^2 that is much smaller than σ^2? (Write one or two sentences, qualitative answer.)

Question 2 (20 marks)

The sum-of-squares error function for regression (Eqn. 3.12 in PRML) treats every training data point equally. In some instances, we may wish to place different weights on different training data points. This could arise if we have confidence estimates of the accuracy of each training data point.

Consider the weighted sum-of-squares error function:

$$E_{\hat{D}}(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N} \alpha_n \{t_n - \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n)\}^2 \tag{1}$$

with weights $\alpha_n > 0$ on each training data point.

1. (15 marks) Derive the optimal weights $\boldsymbol{w}$ given this weighted sum-of-squares error function.
2. (5 marks) Provide a probabilistic interpretation for the weights α_n. (Consider the relationship between maximum likelihood and the sum-of-squares error function. Provide the precise probabilistic model under which $E_{\hat{D}}$ would arise, and explain what α_n are in that model.)

Question 3 (10 marks)

Show that determinant of a real, symmetric matrix is equal to the multiplication of its eigenvalues. (Read Appendix C of PRML.)

Question 4 (40 marks)

In this question you will implement linear basis function regression with polynomial and Gaussian bases.

Start by downloading the code and dataset from the website. The dataset is the `Housing` dataset from the UCI repository. The task is to predict median house value from features describing a town.

Functions are provided for loading the data[1], and normalizing the features and targets to have 0 mean and unit variance.

```
[t,X] = loadData();
X_n = normalizeData(X);
t = normalizeData(t);
```

For the following, use these normalized features `X_n` and targets.

You may also find the provided function `designMatrix.m` useful.

Polynomial basis functions

Implement linear basis function regression with polynomial basis functions. Use only monomials of a single variable (x_1, x_1^2, x_2^2) and no cross-terms ($x_1 \cdot x_2$).

Perform the following experiments:

1. Create a MATLAB script `polynomial_regression.m` for the following.

   Using the first 100 points as training data, and the remainder as testing data, fit a polynomial basis function regression for degree 1 to degree 7 polynomials. Do not use any regularization. Plot training error and test error (in RMS error) versus polynomial degree. **Put this plot, along with a brief comment on what you see, in your report.**

2. Run your polynomial regression using a degree 1 polynomial. Examine the learned weights. What value is chosen for w_5, the weight on the 5^{th} feature (average number of rooms per dwelling)? What value is chosen for the weight on the 7^{th} feature (weighted distances to five Boston employment centres)? (Don't forget the bias weight.) Do these 2 weights seem reasonable?

   Put the values of all weights and your comments on weights for the 5^{th} and 7^{th} features in your report. You do not need to submit code for this part.

[1] Note that loadData reorders the datapoints using a fixed permutation. Use this fixed permutation for the questions in this assignment. If you are interested in what happens in "reality", try using a random permutation afterwards. Results will not always be as clean as you will get with the fixed permutation provided.

3. Create a MATLAB script `polynomial_regression_1d_vis.m` for the following.

   It is difficult to visualize the results of high-dimensional regression. Instead, only use one of the features (use `X_n(:,2)`) and again perform polynomial regression. Produce plots of the training data points, learned polynomial, and test data points. The code `visualize_1d.m` may be useful. **Put 2 or 3 of these plots, for interesting (low-order, high-order) results, in your report. Include brief comments.**

4. Create a MATLAB script `polynomial_regression_reg.m` for the following.

   Implement L_2-regularized regression. Again, use the first 100 points, and only use the 2nd feature. Fit a degree 8 polynomial using $\lambda = \{0, 0.01, 0.1, 1, 10, 100, 1000\}$. Use 10-fold cross-validation to decide on the best value for λ. Produce a plot of average validation set error versus regularizer value. Use a `semilogx` plot, putting regularizer value on a log scale[2]. **Put this plot in your report, and note which regularizer value you would choose from the cross-validation.**

Gaussian basis functions

Implement linear basis function regression with Gaussian basis functions. You may use the supplied `dist2.m` function. For the centers $\boldsymbol{\mu}_j$ use randomly chosen training data points (use `randperm` in MATLAB). Set $s = 2$. Perform the following experiments:

1. Create a MATLAB script `gaussian_regression.m` for the following.

   Using the first 100 points as training data, and the remainder as testing data, fit a Gaussian basis function regression using 5, 15, 25, ..., 95 basis functions. Do not use any regularization. Plot training error and test error (in RMS error) versus number of basis functions. **Put this plot, along with a brief comment on what you see, in your report.**

2. Create a MATLAB script `gaussian_regression_reg.m` for the following.

   Implement L_2-regularized regression. Again, use the first 100 points (do **not** only use the second feature, use them all). Fit a regression model with 90 basis functions using $\lambda = \{0, 0.01, 0.1, 1, 10, 100, 1000\}$. Use 10-fold cross-validation to decide on the best value for λ. Produce a plot of average validation set error versus regularizer value. Use a `semilogx` plot, putting regularizer value on a log scale (see note previously). **Put this plot in your report, and note which regularizer value you would choose from the cross-validation.**

[2]The unregularized result will not appear on this scale. You can either add it as a separate horizontal line as a baseline, or report this number separately.

Question 5 (10 marks)

We developed the technique of kernel density estimation in lecture. Kernel density estimation can also be used for regression. Consider the kernel density estimate of the joint distribution:

$$p(\boldsymbol{x}, t) = \frac{1}{N} \sum_{n=1}^{N} f(\boldsymbol{x} - \boldsymbol{x}_n, t - t_n)$$

A regression estimate can be created by considering the expectation $y(\boldsymbol{x}) = \mathbb{E}[t|\boldsymbol{x}]$.

This expectation is computed as:

$$y(\boldsymbol{x}) = \frac{\sum_{n=1}^{N} t_n g(\boldsymbol{x} - \boldsymbol{x}_n)}{\sum_{m=1}^{N} g(\boldsymbol{x} - \boldsymbol{x}_m)}$$

where $g(\boldsymbol{x})$ is the marginal distribution of the kernel function $f(\cdot, \cdot)$ used above:

$$g(\boldsymbol{x}) = \int f(\boldsymbol{x}, t) dt$$

This regression estimate is known as the Nadaraya-Watson model, kernel regression, and locally-weighted regression. The derivation is provided in Sec. 6.3.1 of PRML.

1. Create a MATLAB script `nadaraya_watson_regression.m` for the following.

   Implement the Nadaraya-Watson regression model using 1-D triangle kernels:

   $$g(u) = \begin{cases} (1 - |u|/h) & \text{if } |u|/h \leq 1; \\ 0 & \text{otherwise} \end{cases}$$

   Again, use the first 100 points, and only use the second feature. Fit a model using $h = \{0.01, 0.1, 0.25, 1, 2, 3, 4\}$. Use 10-fold cross-validation to decide on the best value for h. Produce a plot of average validation set error versus kernel width. **Put this plot in your report, and note which kernel width you would choose from the cross-validation.**

Submitting Your Assignment

The assignment must be submitted online at `https://courses.cs.sfu.ca`. In order to simplify grading, you must adhere to the following structure.

You must submit two files:

1. You must create an assignment report in **PDF format**, called `report.pdf`. This report must contain the solutions to questions 1-3 as well as the figures / explanations requested for 4-5.

2. You must submit a gzipped tar file of all your code, called `code.tgz`. This must contain a single directory called `code` (no sub-directories, no leading path names), in which all of your files must appear[3]. There must be the 6 scripts with the specific names referred to in Questions 4 and 5, as well as a common codebase you create and name.

   As a check, if one runs

   ```
   tar xzf code.tgz
   cd code
   matlab
   polynomial_regression_1d_vis
   ```

   the script produces the plots in your report from the relevant question.

[3]This includes the data files and others which are provided as part of the assignment.