ELL409: Assignment 1

Demo Schedule: 1st week of April 2021

Polynomial Curve Fitting

Polynomial curve fitting is an example of regression. Here you will apply the concepts of linear regression for polynomial curve fitting. In regression, the objective is to learn a function that maps an input variable x to a continuous target variable y.

For this part, you will be provided a personalised input file that contains data of the form (x_i, y_i) for $i = 1, \dots, 100$. The relationship between x and y is of the form:

$$y = w_0 + w_1 x + \cdots + w_M x^M + \epsilon$$

where the noise ϵ is drawn from a Gaussian distribution with zero mean and unknown (but fixed, for a given input file) variance. M is also unknown. You can download your input data file from `https://web.iitd.ac.in/~seshan/a1/<groupno>.txt` (for e.g., `https://web.iitd.ac.in/~seshan/a1/group01.txt`) where `groupno` is your group number listed here: `https://docs.google.com/spreadsheets/d/1vLh7DtW60s6QWMROUY-C-KUstPl0uQ_96ViGVNa-PC8`. The goal is to identify the underlying polynomial (both the degree and the coefficients), as well as obtain an estimate of the noise variance.

Specifically, the following tasks are to be accomplished:

- To begin with, use only the first 20 data points in your file. Solve the polynomial curve fitting regression problem using error function minimisation. Define your own error function other than the sum-of-squares error. Try different error formulations and report the results.
- Use a goodness-of-fit measure for polynomials of different order. Can you distinguish overfitting, underfitting, and the best fit?
- Obtain an estimate for the noise variance.
- Introduce regularisation and observe the changes. For quadratic regularisation, can you obtain an estimate of the optimal value for the regularisation parameter λ? What is your corresponding best guess for the underlying polynomial? And the noise variance?
- Now repeat all of the above using the full data set of 100 data points. How are your results affected by adding more data? Comment on the differences.
- What is your final estimate of the underlying polynomial? Why?

You will be required to give a demonstration of regression, the coefficients you have obtained, and how you have done so. In addition, present visualisations of the data and results in meaningful ways.

Handwritten Digit Recognition

Automated handwritten digit recognition is widely used today, for e.g., recognising postal codes on envelopes for sorting the mail. You will use logistic regression to recognise handwritten digits (0 to 9).

A data set containing 5000 training examples of handwritten digits can be downloaded from here: `https://web.iitd.ac.in/~seshan/a1/handwritten_image_data.rar`. The data has been saved in a tab-delimited text (ASCII) format. Each training example is a 28 pixels by 28 pixels grayscale image of the digit. Each pixel is represented by a floating point number indicating the grayscale intensity at that location. The 28×28 grid of pixels is reshaped into a 784-dimensional vector. Each of these training examples becomes a single row in our data file. Reading the data from this text file into a matrix gives a 5000×784 matrix where every row is a training example for a handwritten digit image. The second part of the training set is a 5000-dimensional vector that contains labels for the training set. The digits "0" to "9" are labelled as "0" to "9" in their natural order.

Using multiple one-vs-all logistic regression models build a multi-class classifier. Since there are 10 classes, you will need to train 10 separate logistic regression classifiers. In (unregularised) logistic regression, the cost function is given by:

$$E(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} [-y_i \log(h(\mathbf{x}_i)) - (1 - y_i) \log(1 - h(\mathbf{x}_i))]$$

where $(\mathbf{x}_i, y_i)$ is the i^{th} training example with $y_i \in \{0, 1\}$, N is the total number of training examples, and $h(\mathbf{x}_i) = g(\mathbf{w}^T\mathbf{x}_i) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}_i}}$.

- Using the above expression you can compute the cost function and its gradient which can be used by an unconstrained optimisation routine to obtain the optimal classifier parameter $\mathbf{w} \in \mathbb{R}^{(d+1)}$ where $(d+1)$ is the dimensionality of the inputs $\mathbf{x}_i$ (including the bias term). After training a separate classifier for each of the K classes, your code should return all the classifier parameters in a matrix $\mathbf{W} \in \mathbb{R}^{K\times(d+1)}$, where each row of $\mathbf{W}$ corresponds to the learned logistic regression parameters for one class.

- The multi-class classifier trained using the steps above can now be used to predict the digit contained in a given image. Using the learned parameter matrix $\mathbf{W}$, make predictions on the training set and determine the training set accuracy.

- Add quadratic regularisation to the cost function so that:

$$E(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}[-y_i\log(h(\mathbf{x}_i)) - (1-y_i)\log(1-h(\mathbf{x}_i))] + \frac{\lambda}{2N}\sum_{j=1}^{d}w_j^2.$$

  Note that the bias term w_0 is not considered for regularisation. Obtain the corresponding expression for the gradient of the cost function and retrain your classifier with this regularised cost function. Comment on the differences.

- During the demonstration you will be asked to show your classifier output and performance on a separate set of test images.

You should prepare a report, compiling all your results and your interpretation of them, along with your overall conclusions. In particular, you should attempt to answer all of the questions posed above. Any graphs or other visualisations should also be included in the report. The report along with your code will have to be submitted via Moodle.