AI701: Foundations of Artificial Intelligence
Fall 2022

Assignment-01

September 9, 2022

Instructions:

- Group Assignment. Maximum number of students per group: 3.
- This assignment carries 40 marks. It has three sections with a total of 12 tasks.
- The deadline to submit the assignment is by the end of September 19, 2022 (23:59 UAE time).
- Assignment deliverables: three completed jupyter notebooks and a report. All the required material should be zipped in one folder (per group).

---

1 Linear Regression (10 points)

1.1 Mean Squared Error

Task 1: In the provided notebook, write a function to compute the mean squared error of a given line for the Iris dataset.

1.2 Computing the analytic solution

The best coefficients a and b that minimise the mean squared error (MSE) can be found analytically, using a bit of calculus.
In particular, we set the gradient of the MSE loss function to 0 in order to obtain the least squares estimate for a and b. For MSE denoted by $\mathcal{L}(a, b)$, setting $\frac{\partial \mathcal{L}}{\partial a} := 0$ and $\frac{\partial \mathcal{L}}{\partial b} := 0$, we obtain the least squares estimate

$$a = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$$

$$b = \bar{y} - a\bar{x}$$

where $\bar{x}$ is the mean value of $[x_0, \ldots, x_{N-1}]$ and so on.
Task 2: Use this solution to compute the least squares estimate for the Iris dataset directly.
Task 3: Predict petal widths for new flowers

1.3 Higher dimensional input features

Task 4: Fit a linear regression model to a dataset with higher dimensional input features. We choose to model the housing price as a function of the property's location and its size, using a synthetic dataset. Is the plane a good fit to the data? If not, what is the reason?

2 Logistic Regression (20 points)

2.1 Decision boundary

* The decision boundary is a line which separates out the data points from different clusters.
* If we have a data point $\mathbf{x}$ that we want to classify, where $\mathbf{x} = \{x_1, x_2\}$ – a 2D feature vector. Then the predicted class depends on which side of the line $\mathbf{x}$ lies.
* We represent the decision boundary of the form: $f(\mathbf{x}) = ax_1 + bx_2 + c = 0$.
* This allows us to easily compute which side of the line the point lies from looking at the sign of $f(\mathbf{x}) = ax_1 + bx_2 + c$.
* A positive sign indicates that the data point $\mathbf{x}$ belongs to the blue class, and a negative sign indicates red.

Task 1: Implement the Sigmoid function
Task 2: Implement a function which gives you the predictive probability $p(\mathbf{x})$ of 2D data points $\mathbf{x} = (x_1, x_2)$

2.2 Learning Objective: Binary Cross-Entropy Loss

* We want to find a way to automatically infer the decision boundary.
* We use the Binary Cross Entropy loss to work out the decision boundary.

$$\mathcal{L} = \sum_{i=1}^{N} -y_i \log(p(\mathbf{x}_i)) - (1 - y_i) \log(1 - p(\mathbf{x}_i))$$

Task 3: Implement the Binary cross entropy (BCE) loss.

2.3 Optimization with Gradient Descent (GD)

* Gradient descent is a way to minimize (or maximize) a function, similar to walking down a hill.
* If you want to walk down a hill from a random point, you'd choose a direction which points down, and then take a step.
* That's what we do a lot in machine learning (literally, this is what everyone uses all the time), but rather than take a physical step, we just move the parameters by a certain amount which we call a step.
* For GD to be useful, we need to calculate the gradient of our loss function.
* The gradients for BCE are:

$$\frac{\partial \mathcal{L}}{\partial a} = \sum_{i=1}^{N} (\sigma(ax_1^i + bx_2^i + c) - y^i) x_1^i$$

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_{i=1}^{N} (\sigma(ax_1^i + bx_2^i + c) - y^i) x_2^i$$

$$\frac{\partial \mathcal{L}}{\partial c} = \sum_{i=1}^{N} (\sigma(ax_1^i + bx_2^i + c) - y^i)$$

Note that $(\mathbf{x}^i, y^i)$ refer to the i-th data points in our observations.
Task 4: Compute the gradient.
Task 5: Implement the gradient descent algorithm.
Task 6: Predict output on test set and evaluate.

3 Support Vector Machines (10 points)

The goal is to determine whether a tweet was written by a Democratic or Republican politician, using just the text of the tweet.
Working with text data The features for an SVM can't be words or whole tweets. We need a numerical representation for the words in the texts. One method is to transform the text into TF-IDF vectors.
It will take the tweets, tokenise them into words (using a special tokeniser that knows how best to split up tweets), remove stop words then it will create a sparse matrix representation of all the tweets.
Task 1: Train a linear kernel SVM
Task 2: Use grid search to find the best model