TDT4173: Machine Learning and Case-Based Reasoning

Assignment 1

January 21, 2016

- **Delivery deadline: February 04, 2016** by 22:00.
- Solutions must be submitted individually.
- Deliver your solution on *itslearning* before the deadline.
- Please upload your report as a PDF file, and package your code into an archive (e.g. zip, rar, tar).
- The programming tasks may be completed in the programming language of your choice.
- Your code is part of your delivery, so please make sure that your code is well-documented and as readable as possible.

Purpose: Gain insight into (a) basic machine learning concepts, and (b) how a simple regression problem can be solved using linear regression.

1 Theory

Basic Concepts

1. Give two examples of relevant machine learning problems and describe them as "well-posed learning problems".
2. What is inductive bias (for a learning method)? Why is it so important in machine learning? The candidate elimination algorithm for learning in version spaces and learning of decision trees with ID3 are two different learning methods. What can you say about the inductive bias for each of them?
3. What is overfitting? Explain the relationship between a learner's bias and how susceptible it is to overfitting.

Concept Learning

Candidate elimination (CE) is a learning method that tries to generate a description which is consistent with all positive — and no negative — examples in the training set. In this task, we will use the attributes and values in Figure 1.

Attributes	Values
Body	Hair, Scales, Feathers
Birth	Live, Egg
Eats-Meat	True, False
Flies	True, False
Teeth	Pointed, Flat, None

Figure 1: Attributes and possible values used to describe animals.

The task is to learn the class *Mammal* from the training examples in Figure 2.

Example	Body	Birth	Eats-Meat	Can-Fly	Teeth	Mammal
1	Hair	Live	False	False	Flat	True
2	Feathers	Egg	True	True	None	False
3	Hair	Live	False	False	Pointed	True

Figure 2: Training examples.

1. Use the CE algorithm for version space learning and show S and G after each new training example. After learning on the training examples, use the system to classify the test examples in Figure 3.

Example	Body	Birth	Eats-Meat	Can-Fly	Teeth	Mammal
1	Hair	Live	False	False	None	?
2	Feathers	Egg	False	True	Pointed	?
3	Scales	Egg	True	False	Flat	?

Figure 3: Test examples.

2. What can the system say about the classification of the three new examples? Explain why.

3. Assume that the system can ask for another training example. Which criteria should the system use to choose the training example? Give an example combination of attribute values that satisfies these criteria.

2 Programming

Linear Regression

In this programming assignment you are required to implement a linear regression model in order to fit a plane using gradient descent. The dataset is stored in a comma-separated values (CSV) file and includes N examples (x_i, y_i). The data has already been split up into a training and test set. Each input vector $x_i = (x_{i1}, x_{i2}, \dots x_{ip})^T$ is defined in terms of two features ($p = 2$) and will be used to predict the associated real-valued output y_i. Assuming the relationship between the input vector and output variable is linear, we can find a (hyper)plane that fits the data reasonably well using a linear regression model. This model, or hypothesis space, can be seen in Equation 1, where W is a weight vector and b is the bias term[1].

$$f(x_i) = f_{W,b}(x_i) = b + \sum_{j=1}^{p} W_j x_{ij} \tag{1}$$

Now that we have a model we need something that allows us to evaluate the quality of the current hypothesis. The mean squared error (MSE) function is one such *loss*, or error, function that measures the average squared differences between model predictions $f(x_i)$ and target values y_i over all observations. The loss function can be seen in Equation 2, where n is the number of examples included in the computation.

$$L(W, b) = \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 \tag{2}$$

Ideally, we want to pick the hypothesis that yield the lowest loss over the examples. The most popular way of estimating this is called *least squares*. In this assignment you will perform least squares using an iterative optimisation algorithm called gradient descent. In order to use gradient descent we need to find out how the loss function changes when W or b are allowed to vary. The results of these partial derivatives can be seen in Equation 3.

$$\frac{\partial L}{\partial W} = \frac{2}{n} \sum_{i=1}^{n} (f(x_i) - y_i) x_i \qquad \frac{\partial L}{\partial b} = \frac{2}{n} \sum_{i=1}^{n} (f(x_i) - y_i) \tag{3}$$

With these partial derivatives we can specify the update rules used to alter the trainable parameters. This can be seen in Equation 4, where α is the learning rate (decides how much the parameters should change). *The rules are repeated until convergence or until the maximum number of iterations has been reached.*

$$\begin{aligned} W &\leftarrow W - \alpha \frac{\partial L}{\partial W} \\ b &\leftarrow b - \alpha \frac{\partial L}{\partial b} \end{aligned} \tag{4}$$

1. Implement linear regression and procedures for optimising it using gradient descent.

2. Train the model on the training dataset

[1] There are several alternative ways to express the linear regression model. Another popular approach employed by Andrew Ng in CS 229 is to incorporate the bias, or intercept term, into the weight vector W, where this new vector is called θ. In other words, $\theta = (b, w_1, w_2, \dots w_p)^T$. Consequently, an extra dimension of all ones must be prepended to x_i.

(a) Show initial parameters (W, b, and α), comment on your choice.

(b) Show intermediate results after the 5th and 10th iterations.

(c) Show the final results and plot how the loss function changed over the course of optimisation. Discuss your result.