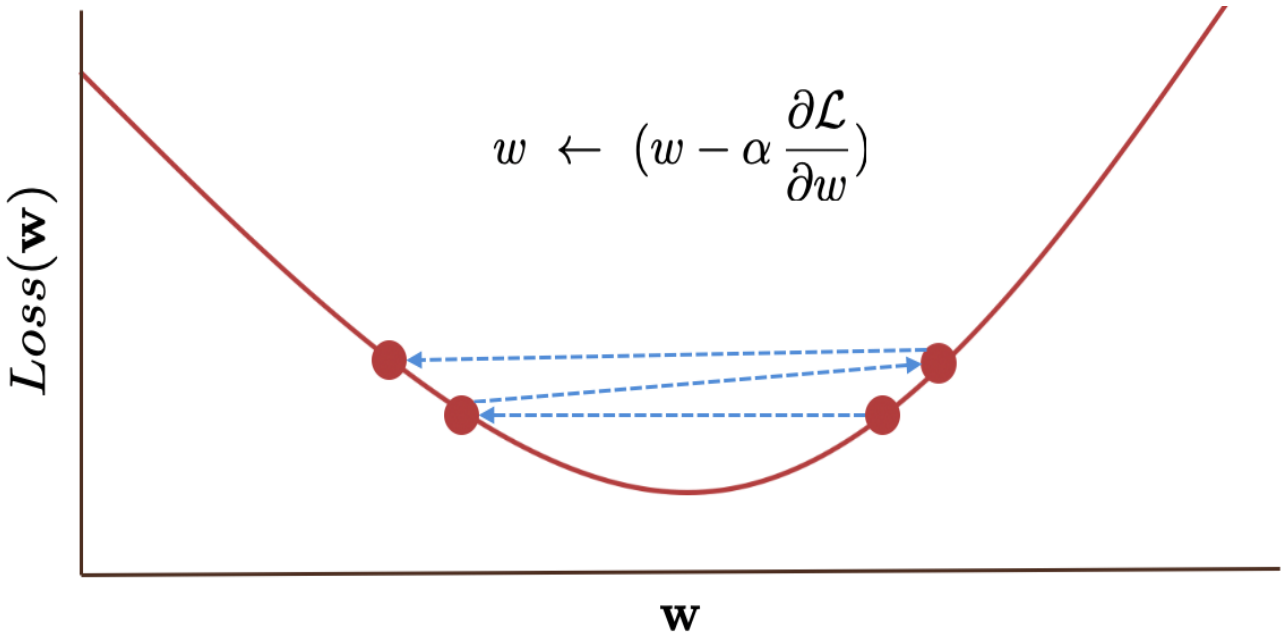
To-Do Date: May 25 at 11:59pm

Topic A: Basic Linear Supervised Models Overview



We start our exploration of machine learning (ML) with a set of linear, supervised models. In supervised learning, we start with training data consisting of known (input, output) pairs, where the inputs are vectors of numerical feature-values, and the type of model depends upon the type of output we have:

- If the output is a single real-numbered value, like the life expectancy of a patient or the monetary value of some commodity, then we have a regression problem.

- If the output is a discrete categorical value, like whether a patient will become ill from some disease or not, of whether a commodity falls into one of three basic price ranges, then we have a classification problem.

A model is linear if fits a linear function to the input data. In general such models derive the linear function in a way that seeks to minimize some established error measure. Once we have fit such a linear function, it can be used to predict output values for novel inputs of the same type as the original data. If the original data-set is representative of the new data, the predictions of the linear model will then be useful guidance as we seek to predict things about how that new data will behave, or seek to understand the categories in which it will fall.

We will start by examining linear regression models, where the predicted outcome is real-valued, and is represented by a basic linear function on the inputs (a line, plane, or higher-dimensional linear object). We will also look at extensions to the model, where the function that we generate is no longer merely linear, but is represented by some higher-degree polynomial. We will consider how such models can be built, along with elementary methods, based upon gradient descent, in which error is measured and the function is adjusted in ways that are mathematically designed to reduce that error. We will look at consider a common issue that can arise in all ML models, namely over-fitting, a phenomenon that happens when we reduce error on our original training data but actually suffer significantly worse performance on new data. We will also examine practical techniques for dealing with over-fitting, including cross-validation and regularization.

Finally, we will examine our first classification model, the perceptron, which also fits a linear function to the data, but uses that not as a predictor of future values, but as a separator, dividing one class of data from another.

Learning Outcomes

Students will be able to:

1. Identify the components of a supervised learning method, and prepare data to be used by such a method.
2. Identify the components of linear and polynomial models, and apply them to data.
3. Describe the difference between regression and classification methods, and the purposes of each.
4. Explain the mathematical underpinnings and practical use of gradient descent methods for regression and classification.
5. Use cross-validation and other techniques to identify when ML models may be over-fitting, and correct some of these issues by using regularization methods.
6. Step through and implement basic algorithms for fitting linear regression models and for doing perceptron-based classification.
7. Install and use Python libraries, writing basic Python code to use those libraries.
8. Use existing Python libraries, in combination with their own code, to perform regression analysis on data, perform cross-validation testing, and apply regularization to existing models.