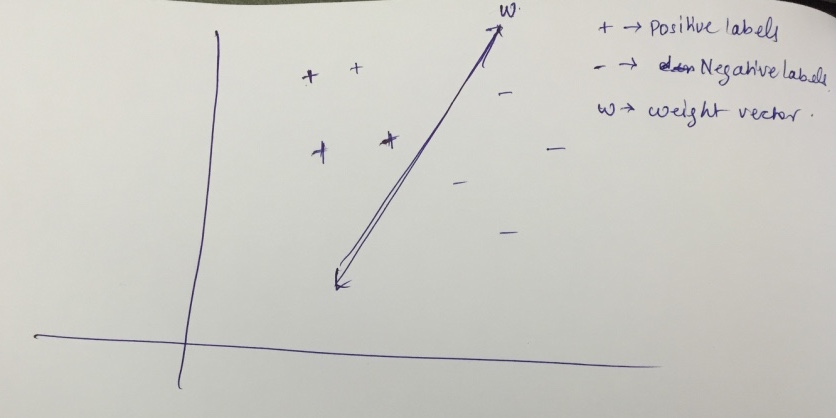
Process Book

Overview and Motivation:

In this project we are attempting to visualize one of the machine learning algorithms called “perceptron”. In machine learning domain this is a fundamental and a very powerful algorithm used to obtain a linear classifier of a data set.

Now taking a glimpse at the below diagram it must be evident that there are set of points labeled +, a set of point labeled – and a line (weight vector) w. The weight vector clearly separates the dataset into two parts. If we denote each example by (x, y , label) and the weight vector by (w1, w1, b), it follows the property that :

label == sign(w1\*x + w2\*y + b)



The algorithm attempts to find the weight vector w, given the data set and its labels.

So what’s the use of the weight vector when we already have the data set and their labels?

If there are any new data points in future whose labels are unknown the weight vector can be used to obtain the label of that new point. There are many practical applications of this algorithm. The positive and negative labels could represent whether a mail is a spam of not, whether it will rain today or not, whether a given flower is a lily or jasmine etc.

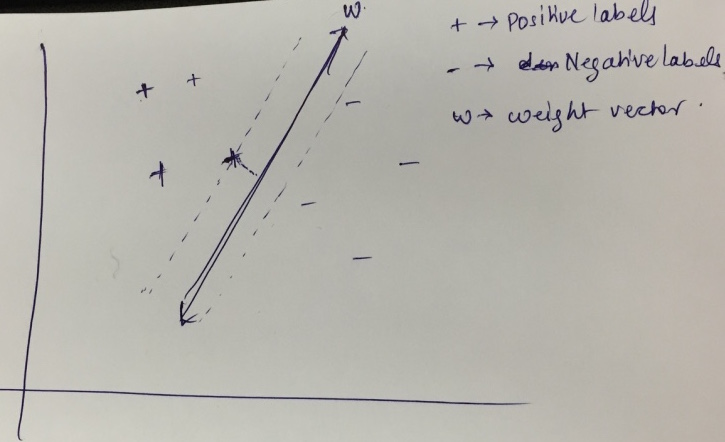
Though it is obvious, the mechanics of the algorithm are effectively understood only with the help of a well-designed visualization. It must be noted that the weight vector changes several times through out the course of the algorithm, but what is important is, how does it change?

Through this project we also address the question of how the perceptron would behave if we took only a subset of the given data.

**Margins:**

A margin is defined as the distance of the closest correctly classified point from a given weight vector.

One important aspect of the perceptron algorithm is the concept of margins. The concept of margins can be better understood by actually seeing the margin of a given weight vector than theorizing it. So, in this project we address the concept of margins and answer some questions like why larger margins are better.



Related Work:

What inspired us most was the fact that when we studied machine -learning course we looked up for interactive resources online, which would enable us to play around with the weight vector and see how it is updated?

In addition the home-works in this course enabled us to learn about d3 and interactive visualization techniques, which drove us towards this project.

Furthermore, we are hoping that this be used later on by students learning machine learning to better understand the perceptron algorithm.

The below website had some cool visualization of machine learning algorithms and it also influenced us in coming up with this project.

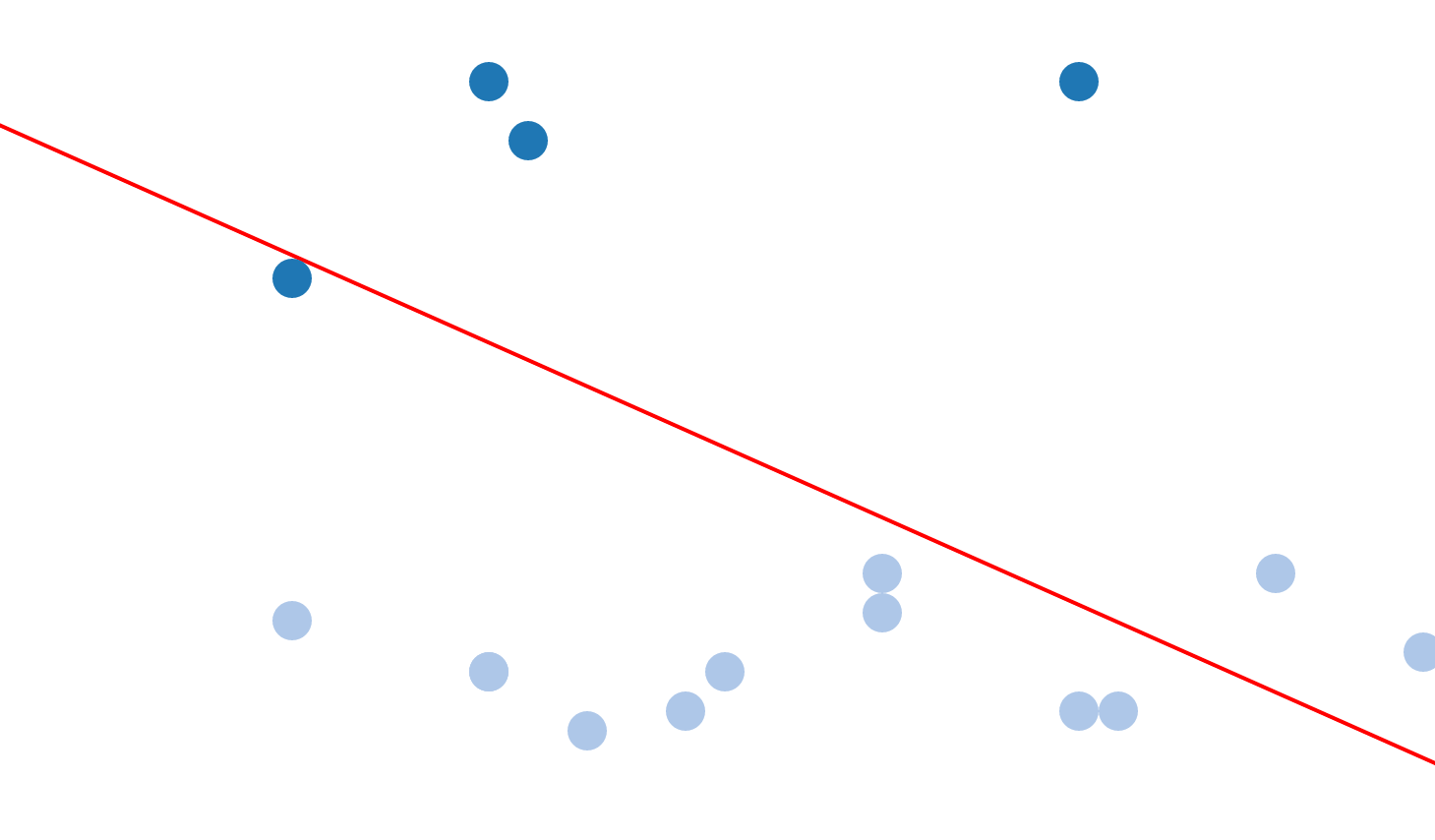
http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

Data: Source, scraping method, cleanup, etc.

One of the most critical aspects of this project is find the right data set that can be used fit all our requirements. Though there are lots of datasets out there it was important to find a dataset that had binary classification, two features (as the visualizations are 2d in nature) and was linearly separable.

After searching and failing to find a good data set we decided to generate synthetic data set that catered to all our needs.

The below snapshot represents a subset of points with binary classification along with the weight vector with perceptron in action.



Later on we decided to be more creative and found a data set named iris data set.

The data set consists of 50 samples from three species of *Iris flowers* ([*Iris setosa*](https://en.wikipedia.org/wiki/Iris_setosa), [*Iris virginica*](https://en.wikipedia.org/wiki/Iris_virginica) and [*Iris versicolor*](https://en.wikipedia.org/wiki/Iris_versicolor)). Four [features](https://en.wikipedia.org/wiki/Features_(pattern_recognition)) were measured from each sample: the length and the width of the [sepals](https://en.wikipedia.org/wiki/Sepal) and [petals](https://en.wikipedia.org/wiki/Petal), in centimeters.

Exploratory Data Analysis:

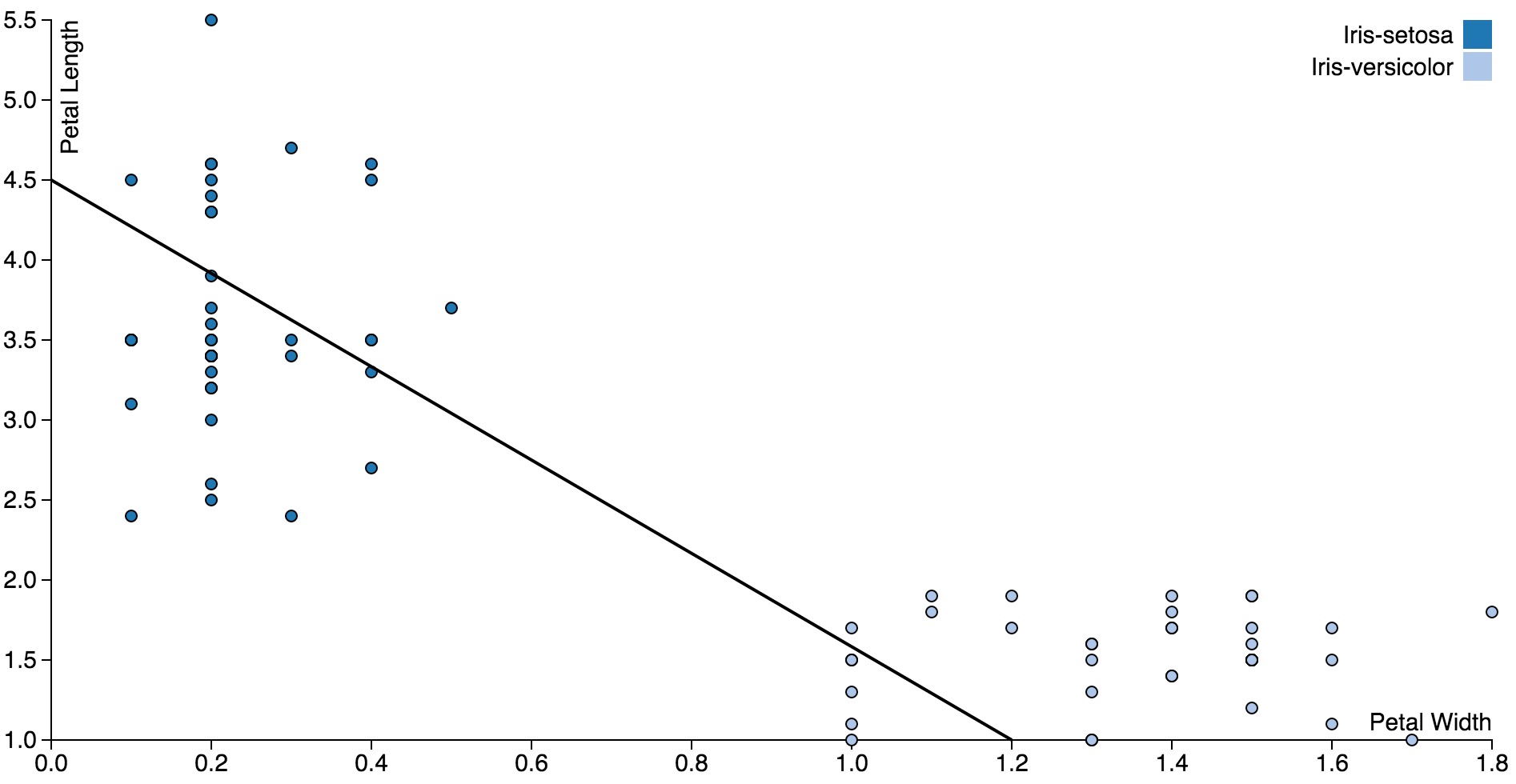
Now, the data set clearly had three classes of flowers Iris-setosa, Iris virginica, Iris versicolor and perceptron works only on binary classified data-set. We initially decided to PCA of this data set and use the top two principal components to run our experiments. But on performing scatter plots for various combinations of the attributes we found that upon selecting two particular attributes named

“ petal length” and “sepal width” the data gives good clusters that could be suitable for perceptron algorithm.

We then went on to pick only two classes named Iris-setosa and Iris versicolor that looked promising to run our experiment.

This gave us an interesting idea of using d3.path to construct a flower and use flowers instead of dots while plotting a data points.

The below image shows the snapshot of the iris data set after cleaning up and processing it to be suitable for out project:



Design Evolution:

We though of 3 different designs for representing perceptron

a: We can draw a scatter plot and use d3 symbols to visualize different classes and use the a line to separate the two regions into different classes. With this approach we can potentially make use of the idea of the brush discussed in the class and define the region of the training data set on which perceptron algorithm must run. Thus, one can observe the accuracy with the variation of the data set.

b: We also though of using a 3d pie graph which is essential has two regions representing two classes and the axis of the pie can be used to represent the weight vector that separates the given data. Though this is one way of representing the perceptron model, it is ineffective in visualizing some key properties like representing the margin.

c: Since we are using a brush to select the region of interest , we thought we could use a bar graph which contains two bars corresponding to the two labels. However for the similar reasons as the pie chart, this is very in effective.

Implementation

As you can see in the images defined above we implemented scatter plot. This representation gives quite explicit understanding of the concept. The interaction elements involve the data points according to which the learning algorithm will update. Right now we are done with implementing the algorithm we are working on adding more design elements.

Evaluation:

We evaluated the flowers data and found out that the data is linearly separable and the sequence of how a classifier can train itself on this data. With the help of visualization technique we were able to figure out the different set of margins possible on this dataset.

We are working on make it more appealing by replacing circles with actual shapes and bringing some more interaction elements in the canvas.