

1a) The decision boundary can be nonlinear. Observe that the VP keeps track of a series of weight vectors and has a flag counter that determines how long the vector was left untouched (not updated on weights). If there are multiple weight vectors that are left isolated on a certain quadrant, drawing a decision boundary would cause a separation from the rest of the quadrants thus the decision boundary would be nonlinear.

1b) The decision boundary is linear. The AVGP uses a simple algorithm to average the weights of weight vector. (Averages of all weights into a single vector instead of a multitude of matrices). The singular vector makes it clear that it is linear.

2) Given that the  $n$  training examples have an importance weight then it simply becomes a matter of transforming the perceptron algorithm to be like that of the voted perceptron. Notice that the voted perceptron's algorithm goes ahead and keeps track of the vote [which is the weight of the surviving vector]. The concept is simply rearranged with the training data. Notice that the misclassification cost would simply be weighted on priority or importance. The learning rate can also be lowered to keep the outputs neutrally scaled (possible normalization attempt?). The learning rate can be parallelized with in terms of increasing dependently on a more important example. The training portion of the perceptron then relies on the importance or priority of examples when it comes to weight vectors.

3) The importance of an issue with perceptron going about learning an imbalanced data set points that one majority class might be favored over the other. Notice that this example states 90% of the examples are negative, this is likely to have the perceptron learn that most data point will be negative. You can mitigate this by balancing the data set with oversampling regarding the positive side or under sampling the negative side. Regarding the perceptron algorithm you can have weights for classes. This acts almost as a reinforcement style of learning such as encouraging the perceptron not to misclassify upon classifying.

4) Observe that the  $\text{Score}(i) = w * f(x_i)$  and  $\text{score}(j) = w * f(x_j)$  if candidate  $i$  is better than  $j$  then clearly the score follows suite. Thus  $\text{score}(i) > \text{score}(j)$  would imply  $w * F(x_i) > w * F(x_j) \rightarrow w(F(x_i) - F(x_j)) > 0$ . If mistake is presented you could update the weights in the following manner :  $W_{\text{new}} = w_{\text{old}} + (LR)(f(x_i) - f(x_j)) \quad y = +1$ .

## 5) View the picture for complete proof

Problem 5Proof

$$\begin{aligned}
 & \textcircled{a} \quad \|x - c_+\|^2 \leq \|x - c_-\|^2 \quad \textcircled{1} \\
 & \|x - c_+\|^2 \leq \|x - c_-\|^2 \quad \textcircled{2} \quad // \text{Square both sides} \\
 & \|x\|^2 - 2x \cdot c_+ + \|c_+\|^2 \leq \|x\|^2 - 2x \cdot c_- + \|c_-\|^2 \\
 & -2x \cdot c_+ + \|c_+\|^2 \leq -2x \cdot c_- + \|c_-\|^2 \\
 & 2x \cdot c_+ - 2x \cdot c_- + \|c_+\|^2 - \|c_-\|^2 \leq 0 \quad \textcircled{3} \quad // \text{Isolate terms} \\
 & w = [2(c_+ - c_-) \cdot x] \quad \text{---} \times \text{---} \\
 & b = [\|c_+\|^2 - \|c_-\|^2] \quad \text{---} \times \text{---}
 \end{aligned}$$

$$\textcircled{b} \quad w = 2 \underbrace{\sum_{i=1}^n x_i}_{n+} - 2 \underbrace{\sum_{i=1}^n x_i}_{n-}$$

$$\alpha = \frac{2}{n+} \quad \text{for positive Ex.} \quad \alpha = \frac{2}{n-} \quad \text{for negative Ex.}$$

Support vectors



6) Those who are familiar with machine learning will know that learning is the collaboration of representation, evaluation and optimization. When it comes to machines though the greatest assistance given to users comes in the form of classification. Being able to get an answer given data that might have some pattern or relatability is extremely useful. This answer or result comes in the form of a label. Which are the models predicted output given a data set usually in the form of a vector. Not all models are the same and can be judged by optimization potential and performance measures.

Most ML newbies make the mistake of not discerning the training data and testing data. Using both makes the model inconclusive since that invalidates the process of being able to give an output that isn't just memorized.

Generalization comes in handy when contributing to a dataset and model. The application should technically be refined constantly in order to attack a certain situation or dilemma. The free lunch

Zeid Al-Ameedi

04/04/2020

CPTS 315 HW3

principle states this perfectly in that no ML model individually can be random guessing on every possible application. Don't forget to refine and retune constantly!

The frustration that comes with machine learning is simply stated as not being a one shot process of building a dataset and running a learner, but rather an iterative process of running the learner, analyzing the results, modifying the data and/or the learner, and repeating.

The most important takeaway from this article is that Machine learning is usually applied to observational data, where the predictive variables are not under the control of the learner, as opposed to experimental data, where they are.