Dual perceptron algorithm

Why it works?

Perceptron Learning Algorithm(PLA) is used in the classification of linearly separable data. In online version of PLA, if the class label is wrong, the weight vector is pushed towards or away from training sample x_i depending on whether label is positive or negative. The intuition is that, if it is misclassified the sample lies on the wrong side of the line/plane(classifier). So by subtracting or adding x_i to the weight vector, the line is moved towards the sample until it is correctly classified.

In PLA, the weight vector is a linear combination of the training samples[1]. While in PLA, the weight vector is estimated as follows for N training samples.

$$w = \sum_{i=1}^{N} \alpha_i * x_i * y_i \text{ - (1)}$$

In Dual PLA, the weight vector given by (1), where α counts the number of times a training instance is misclassified. While in online PLA, we added or substracted x, n times where n is the number of misclassifications, in Dual PLA we are keeping track α which is equal to n and therefore the summation over the entire training set i.e (1) gives us the weight vector to be considered.

Now in online PLA, if $w * x_i * y_i \geq 0$ that means the instance is correctly classified and is on the right side of plane. Otherwise, i.e $w * x_i * y_i < 0$ it means instance is misclassified and on the wrong side of the plane. It follows from the hypothesis function $h(x) = sign(w * x)$ which predicts class label as 1 if $h(x)$ is positive and 0 if negative. Therefore, if prediction and true class label are same(+ * + or - * -) the product $w * x * y$ would be greater than or equal to 0 and otherwise lesser than 0.

In the dual version, the hypothesis function is computed in a similar way with weight vector given by (1) and by the dot product between weight vector and x(where x is a training sample with x_i, y_i). This is multiplied with the true class label and if the result is negative, the corresponding α_i is incremented as the hypothesis function got it wrong. Incrementing α will add/substract(depending upon y_i to the weight vector which is same as in online PLA i.e moving the separator plane towards or away from the misclassified sample.

$$h(x) * y = sign(w * x) = sign(\sum_{t=1}^{N} \alpha_i * x_i * y_i * x_t * y_t)$$

Is it better?

Here α_i denotes the number of times a training sample x_i has been misclassified. Therefore for data which is easier to classify will have low α and which are difficult to be classified will have high α as it will be misclassified more number of times. So from the weight vector being used in Dual PLA it shows that training samples which are classified more times are given higher priority over the ones with lower α as the former has more information about the distribution.

Compared to the earlier version of PLA, this one finds a linear seperator among the points as well information about the content of training samples. Therefore it is much more

expressive but the computation of the vector can be expensive with large size of training set.

References

[1] *Kernel methods and factorization for image and video analysis*. Ranjeeth Dasineni