Proposal for CTPS570 Project

Amirkhosro Vosughi (11463709)

(Feature's coordinate rotation in Decision Tree)

I decide to expand the ID3 algorithm for finding decision tree in order to be able to decrease the depth of table.

Problem & Motivation:

The ID3 algorithm that was introduced in the class, find the minimum entropy based on decision boundary that contains one feature. As a results, when the actual decision boundary between different classes it not aligned with the axis of features, it lead to many leaf in the decision tree. For example see the figure 1 in the below. So we need to somehow overcome this problem

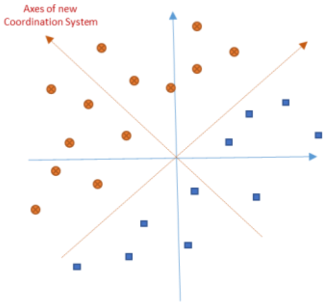

Fig 1. Data set with boundary (Classes' label determined by circle and square)

Methodology:

My idea to overcome the above problem is to consider the decision boundary as linear function of features (such as $x_1\alpha_1 + x_2\alpha_2 + \cdots + \alpha_n\alpha_n > \beta$). So the problem is that how we can find appropriate linear function. Let explain the idea 2 features set ($n = 2$). To find α_i's we can first calculate entropy in the boundaries as ID3 suggest. Next we can compare the minimum entropies of the horizontal boundary (E_h) and the vertical boundary (E_v). Consider the case that those minimum entropies are too close to each other, that means it is more likely that the best boundary with minimum entropy is the one with 45 degree with horizontal axes that is shown in figure 1. So if we rotate the coordinate of the features around the origin by 45 degree and repeat the ID3 for new set of features, we supposedly end up with smaller entropy. After that finding β is not too hard using ID3 in new coordinate. For other values of E_h and E_v, we can find best amount of rotation angle ($\theta = \tan^{-1} E_v/E_h$ or some similar function) to find the minimum entropy. This procedure can be continued until we reach to the leaf.

Final product:

To check the above procedure, I need to write an ID3 algorithm and then modify it to expant coordinate transformation. So we can compare the results for two cases and see how it works. I am really optimistic that it will give good results at least for simple examples. I will try to write the code in Matlab and test in on some simple training data that I will create randomly. I think the bottleneck of this work would be finding rotation angle θ in each step.

I know this proposal might be a little ambitious for course project. But I think it might give us good results. So please let me know about your comments about this proposal.