1. Soft-Margin SVM:

We first construct the Lagrangian formula for this optimization problem:

Then, we can find and accordingly:

We can finally substitute back and into the Lagrangian formula and eliminate the w and together to get the final version of the dual form of the soft-margin SVM:

2. SVM, RBF Kernel and Nearest Neighbor:

1. As 0, will go to zero for all , since and thus the exponent goes to negative infinity and goes to zero. For , . Thus, , which remains as a significant term. Next, since b is a constant and it doesn’t depend on as 0, it can be ignored accordingly. Therefore, if dividing and as 0, all terms belong to either will be cancelled out and thus .

3. Decision Tree and Adaboost:

1. We first determine the proportion of each type of points: and . Then, we can use the formula:

.

The sample entropy is 1 for this dataset.

2. The maximum information gain for the split is and . Thus, has points (1,2), (2,1), (3,4), and (4,6) and contains points (5,3), and (6,5). Thus, the maximum information gain is calculated as:

Thus, the maximum information gain is 0.459 and the splitting rule is and .

3. Since is already pure, which means it contains only blue points and thus we don’t need to split that anymore.

For , we can apply the split at , which means the split rule becomes and for . Thus, has (2,1) and has (1,2), (3,4), (4,6)

Thus, the maximum information gain is calculated as:

Thus, the maximum information gain is 0.811 and the splitting rule is and .

There’s no further splitting since both and contains only one type of point.

4.

:

for all samples,

,

,

.

:

[0.1, 0.5, 0.1, 0.1, 0.1, 0.1],

, where , otherwise 0,

,

.

5.

For adaboost, the final ensemble classifier after T rounds is given by the weighted sum of the weak classifiers :

Where is the prediction of the t-th decision stump and is weight of the t-th decision stump.

Thus, using the above calculation to substitute and :

4. Learning Theory:

1. Since and confidence level , we can apply the Hoeffding’s inequality and solve for the minimum number for achieving the accuracy:

.

2.

a. .

Proof:

Consider two points x1 and x2, with x1<x2. We can show that for any labeling of these two points, we can find parameters w and to correctly classify them:

First, if both points are labeled 1, we can choose w and such that and .

Second, if both points are labeled 0, we can choose w and such that and .

Third, if is labeled 0 and is labeled 1, we can choose w and such that and .

However, if we have 3 points, there can indeed be configurations (especially considering the strict ordering in 1D, suppose x1<x2<x3) where the middle point’s label differs from the outer points, making it impossible for a single linear threshold to correctly classify all possible label configurations. Thus, it cannot be shattered by this classifier properly when we have 3 points. Thus, .

b. .

Proof:

In k dimensions, we can place k+1 points in such a way that they form a triangle in 2D or a tetrahedron in 3D to k dimensions. These k+1 points can be shattered because, in the augmented k+1 dimensional space, they are in general position and can be separated in all possible ways by a hyperplane. However, for any set of k+2 points in , there does not exist a set of weights and a bias that can classify all possible labels of these points correctly. This is because, in the augmented space, k+2 points would require a hyperplane in at least k+2 dimensions to separate all possible labels, which exceeds the dimensionality of the space created by our affine transformation. Hence, the VC dimension of the set of affine classifiers in k dimensions, , is k+1.

c. .

Proof:

For any two points, if we want them to have different labels, we can adjust c such that one of the points falls in a region where and the other falls in a region where . If we want them to have the same label, we can position both either in the positive or negative part of the cosine wave. Thus, 2 points can be shattered. However, if there are three points to be shattered, this cosine function’s ability will be constrained. Consider three points x1<x2<x3 in X=R. Due to the periodicity of the cosine function, it's hard to find a single c that can satisfy an arbitrary labeling of these three points because the sign change imposed by the cosine function's periodicity cannot match arbitrary label assignments across three points. For example, if the labels are assigned in a way that requires two adjacent points to have the same label (either 0 or 1) and the third to have a different label, but in a non-consecutive ordering that doesn't align with the cosine's periods (e.g., 101 or 010), it might not be possible to align the cosine function's sign changes with these labels due to its fixed periodic behavior. Thus, .

5. 3.

A graph of a number

Description automatically generated with medium confidence