Linear Models of Classification

Andres Ponce

March 18, 2020

The goal with **classification** is "to take an input vector x and to assign it to one of K discrete classes.

If we want to classify a certain input vector, we would have to map it to a certain region of a plane, defined by certain boundaries.[1]

[1] Is this similar to **linear programming**, where we had to optimize a linear combination of the parameters?

If there are D varaiables we are trying to optimize, then our answer will exist in a D dimensional space.

Discriminants

Linear discriminants can be of the form[2]

[2] Here, w is the parameters, and w_0 is the bias.

$$y(x) = w^T x + w_0$$

We also ahve a **decision boundary**, where we know which of the K categories our input belongs to given our discriminant function, and a certain cutoff value C_x.

We can also use a Bayesian approach to determine classification, if we calculate $p(C_k|x)$[3]

[3] The probability that given input x, it belongs in class k.

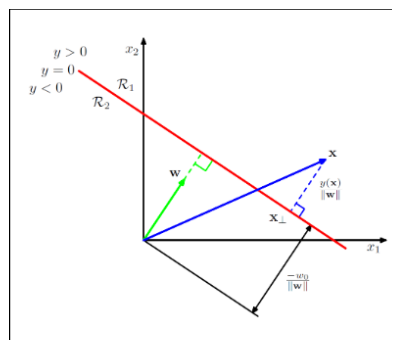

Figure 1: The dotted green and blue lines are the distance between the arbitrary point and the decision boundary.

How do we extend a linear discriminant to other classes? We can have **One-vs-the-rest** or **one-vs-one** qualifiers.

One-vs-the-rest

With this type of classifier functions, we try to distinguish those inputs on a binary case by case basis. That is, we try to one at a time build a classifier that knows points in C_i from those points in other classes. However, when two points "not in their repsective classes" fall in the same area, this area would be defined differently by the areas we're actually testing.

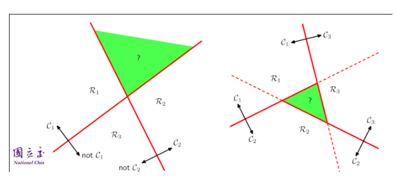

Figure 2: When a point is not in a certain region, it can be ambiguous where it falls, if the same point can be thought of differently by each region.

K-class discriminant

If we use K linear functions of the form

$$y_k(x) = w_k^T(x) + w_{k0}$$