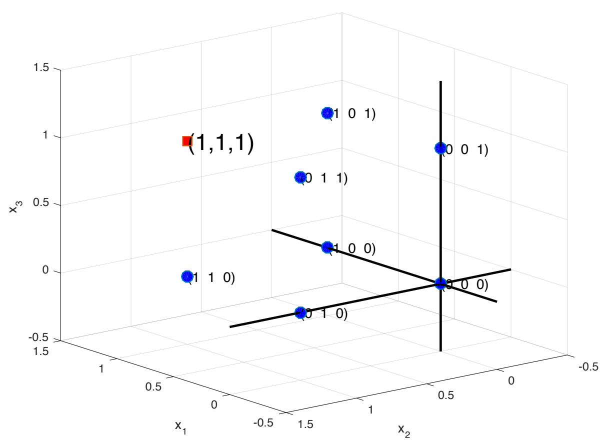
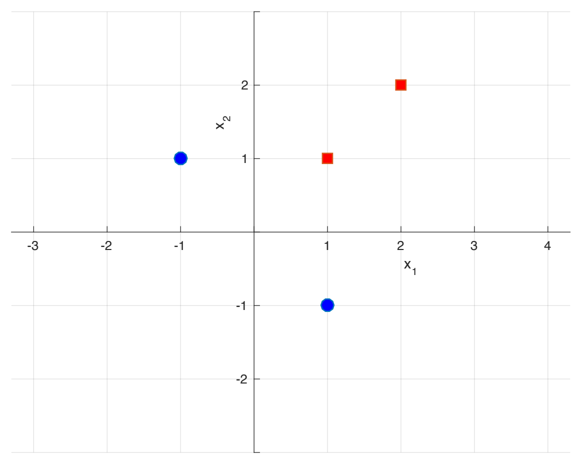
Discussion 1
Probabilistic Machine Learning,Fall 2016

1 Linear Classifiers: Concepts

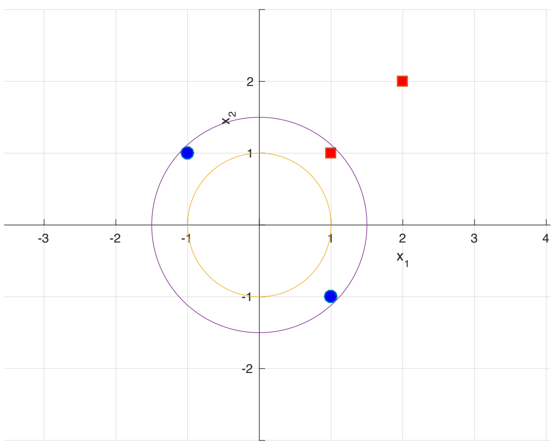


Data for (1)

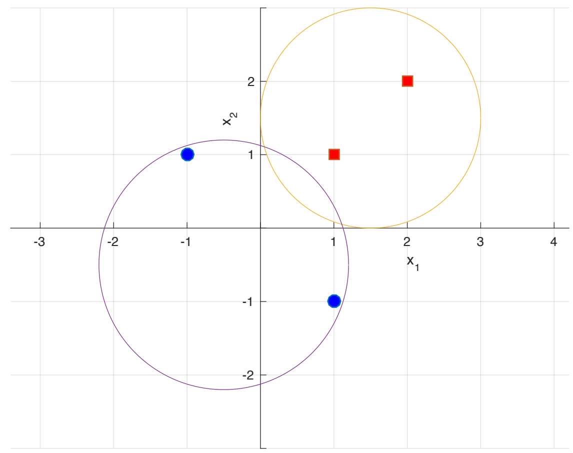


Data for (2)

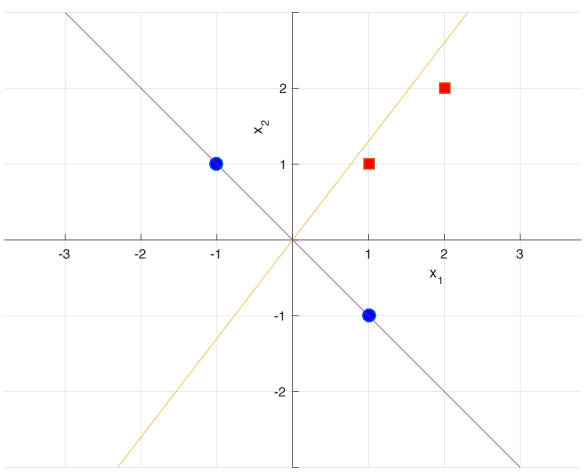
Figure 1: Visualization of Data.



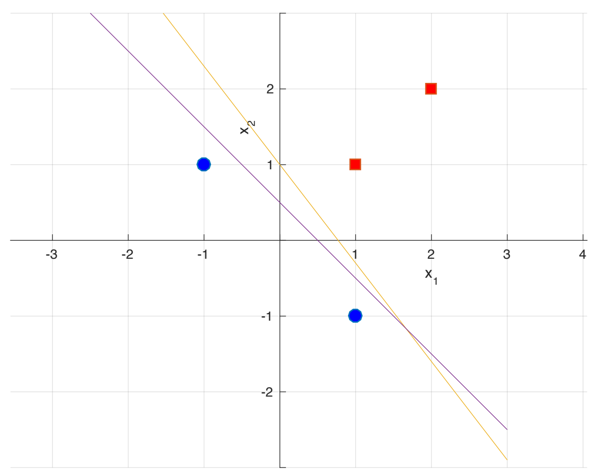
Circle



Circle with offset



Line



Line with offset

Figure 2: Circle and Line Classifiers.

2 Linear Classifiers for Rain Prediction

Suppose we are given a linear classification model that predicts whether or not it is going to rain based upon the temperature (in degrees Celsius) and humidity (expressed as a percentage from 0-100). The model has weights defined such that if the sum of the temperature and the humidity exceeds 110, then it predicts rainfall instead of clear weather.

(a) Assume that an output of +1 corresponds to predicted rainfall. This model has a weight vector θ of length 2 and a nonzero offset θ_0. What are the values of θ and θ_0?

Answer:

$$y = sign(T + H - 110)$$

Thus, $\theta = [1, 1]^T$, $\theta_0 = -110$

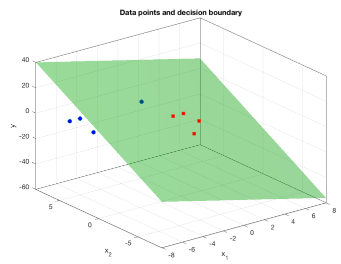
(b) Consider what happens when we feed this model a data point from the planet Mercury (where it never rains) on which the temperature is observed to be 400° C with a humidity of zero. What does this model predict will happen on Mercury? What does this say about the generalization ability of this model?

Answer: When $T = 400, H = 0$ we have

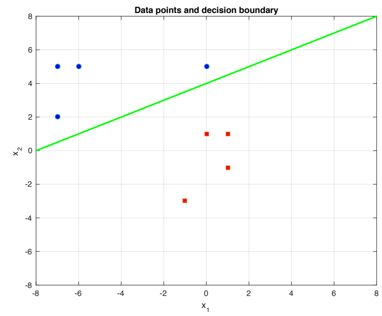
$$y = sign(T + H - 110) > 0.$$

It predicts rainfall, which is false. It means the the clssifier overfit the data on Earth, and can not be generalized to Mercury.

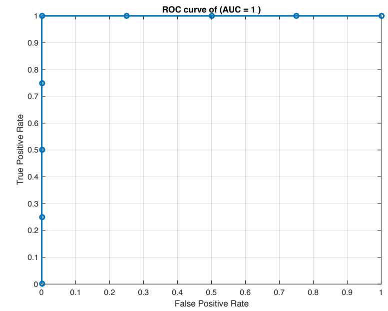
3 ROC and AUC



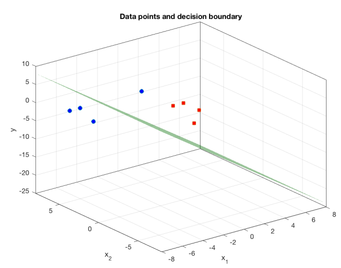
Classifier A: Decision boundary



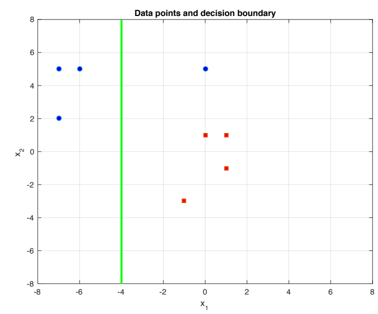
Classifier A: Decision boundary



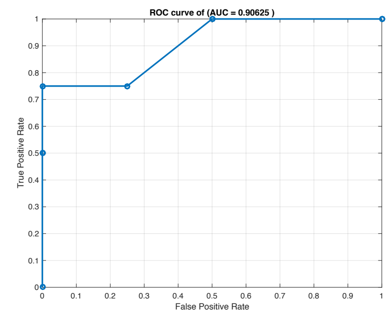
Classifier A: ROC and AUC



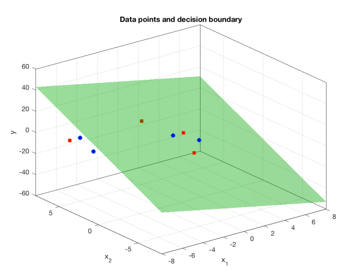
Classifier B: Decision boundary



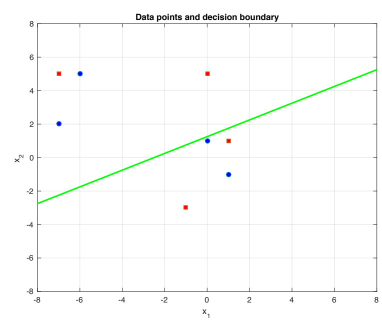
Classifier A: Decision boundary



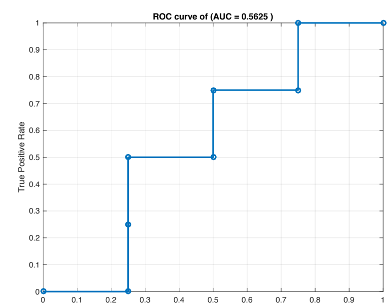
Classifier B: ROC and AUC



Classifier C: Decision boundary



Classifier C: Decision boundary



Classifier C: ROC and AUC

Figure 3: Decision boundary and ROCs.

4 Linear Classifiers for Multple Classes

We want to extend the binary classifier shown above to a multiple classification problem. One potential method for doing this is to divide up the P dimensional input space into K classes. Let $P = 3$ and $K = 3$. This classification problem can be visualized as two or more planes which partition a 3D into several regions. As for the binary case, the region of space which an input point falls into determines its classification.

(a) Assuming our linear classifiers are not collinear, determine the number of regions into which two planes split the 3D input space. Is it equal to K?

Answer:

4

(b) A better way to implement multiple linear classification *this (typo)* is to designate one linear classifier for each class k such that $h_k(x; \theta_k) = \theta_k^T \cdot x$. Then, for any input x, we select the class k such that $h_k(x; \theta_k)$ is maximal. Write an expression for the number of parameters of this model as a function of the number of classes K and the dimensionality of the input P.

Answer:
$(P + 1) \times K$ if bias is considered, otherwise $P \times K$