Machine Learning

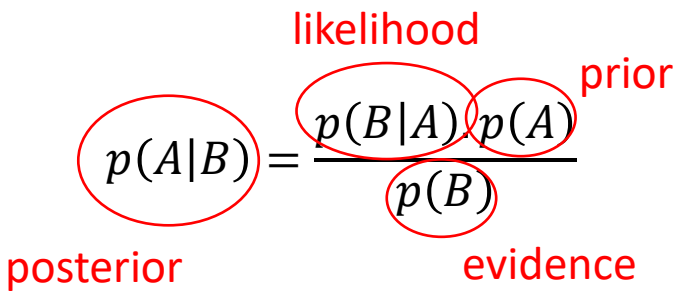
Asieh Daneshi
2024

Naïve Bayes

	Diagnosed with depression	Diagnosed healthy	
Depressed	6	4	=10 depressed
Healthy	2	88	=90 healthy

$$p(depressed|diagnosed\ with\ depression) = \frac{6}{6+2} = 0.75$$

Bayes Rule:



Example

$p(false\ positive) = p(diagnosed\ depressed|healthy) = 0.01$

$p(false\ negative) = p(diagnosed\ healthy|\text{depressed}) = 0.001$

$p(depressed) = 0.1$

$p(healthy) = 1 - p(depressed) = 0.99$

	Diagnosed with depression	Diagnosed healthy
Depressed	?	0.001
Healthy	0.01	?

Sum of each row must be 1

	Diagnosed with depression	Diagnosed healthy
Depressed	1-0.001=0.999	0.001
Healthy	0.01	1-0.01=0.99

If diagnosed with depression, what is the probability of being really depressed?

$$p(depressed|diagnosed\ with\ depression) = \frac{p(diagnosed\ with\ depression|depressed).p(depressed)}{p(diagnosed\ with\ depression)}$$

$$= \frac{0.999 * 0.1}{p(diagnosed\ with\ depression|depressed).p(depressed) + p(diagnosed\ with\ depression|healthy).p(healthy)} =$$

$$= \frac{0.999*0.1}{0.999*0.1+0.01*0.99} = 0.9098$$

Now, let's enjoy some formulas :D

$$p(C_k|x_1, x_2, \dots, x_n) \propto p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

$$p(C_k|x_1, x_2, \dots, x_n) = \frac{p(x_1, x_2, \dots, x_n|C_k).p(C_k)}{p(x_1,x_2,\dots,x_n)}$$

$$p(C_k, x_1, x_2, \dots, x_n) \propto p(x_1, x_2, \dots, x_n|C_k).p(C_k)$$

Assuming that all the likelihoods are independent, we can write:

$$p(x_1, x_2, \dots, x_n|C_k) = p(x_1|C_k).p(x_2|C_k) \dots .p(x_n|C_k) = \prod_{i=1}^{n} p(x_i|C_k)$$

We place this in the previous equation:

$$p(C_k, x_1, x_2, \dots, x_n) \propto p(C_k).\prod_{i=1}^{n} p(x_i|C_k)$$

How are we going to use this for classification?

We had a real output, y, and the output of our model, $\hat{y}$

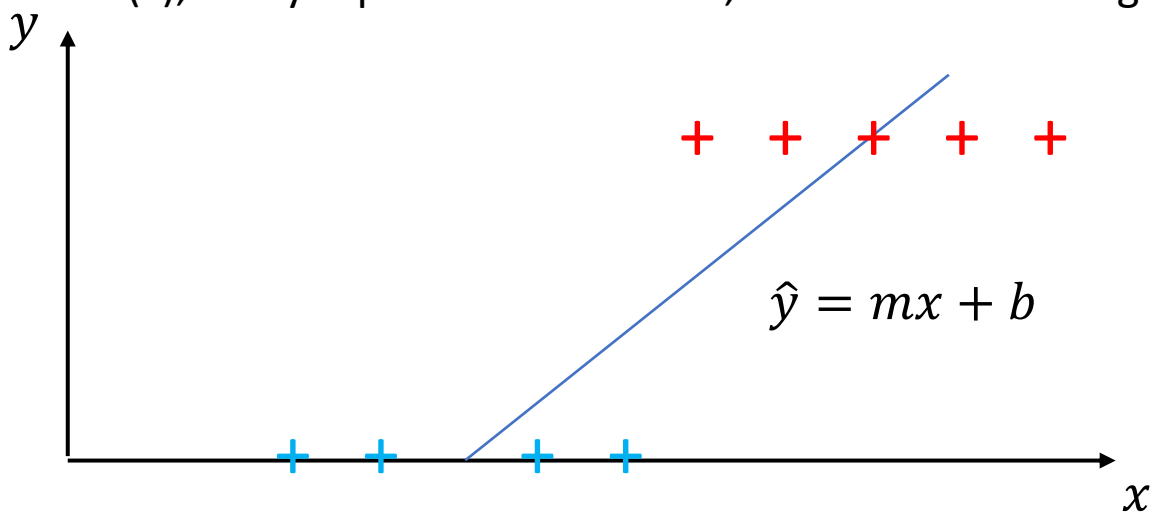
$$\hat{y} = \underset{k \in \{1,k\}}{\operatorname{argmax}}\, p(C_k|x_1, x_2, \dots, x_n) = \underset{k \in \{1,k\}}{\operatorname{argmax}}\, p(C_k). \prod_{i=1}^{n} p(x_i|C_k)$$

What does the above equation mean?

It means that to find $\hat{y}$,we should go through all the categories (1 to k) and compute $p(C_k). \prod_{i=1}^{n} p(x_i|C_k)$ and find the k that makes that maximum. This process is called maximum a posteriori (MAP)

Logistic regression

Imagine that we have one feature (x), and y represents the labels, which are two categories.



We can formulate the line like this:

$$p = mx + b$$

There is a problem in this formula. We know that $mx + b$ can be any value between $-\infty$ and $+\infty$. However, the probability (p) can just take the values between 0 and 1. To solve this, we change the formula to:

$$\frac{p}{1-p} = mx + b$$

This problem is solved, but there is still another problem. $mx + b$ can take negative values too.

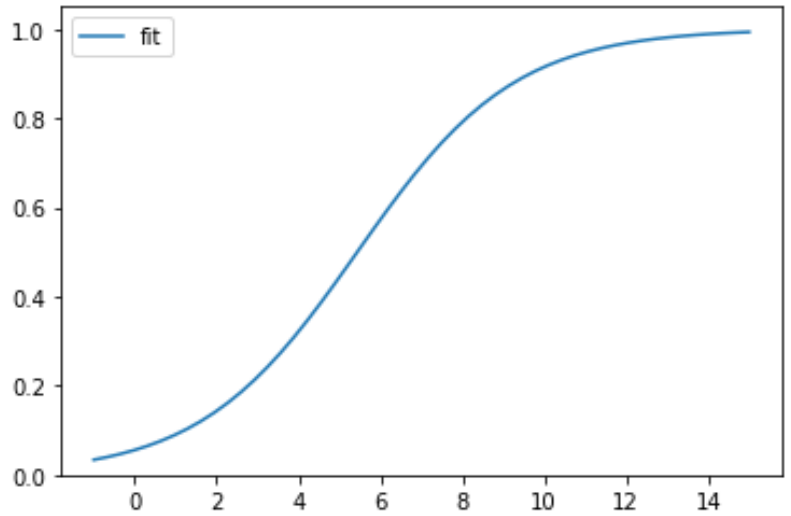
However, the probability can't be negative. We solve this by changing the formula to:

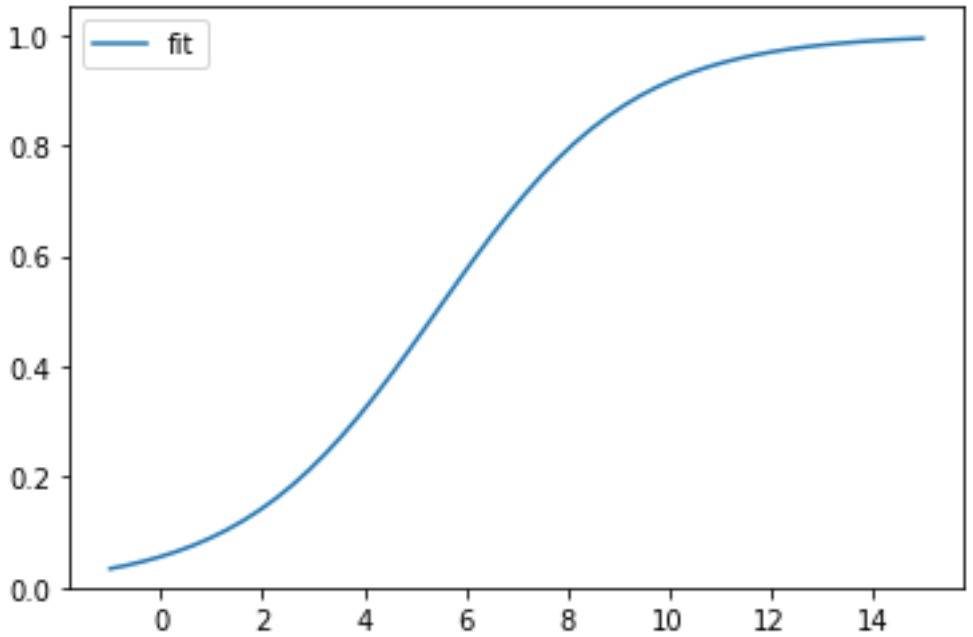
$$\ln\left(\frac{p}{1-p}\right) = mx + b$$

$$\frac{p}{1-p} = \mathrm{e}^{\mathrm{mx+b}}$$

$$p = \frac{e^{mx+b}}{1+e^{mx+b}} = \frac{1}{1+e^{-(mx+b)}}$$

Now, this last formula looks like Sigmoid function: $S(x) = \frac{1}{1+e^{-x}}$

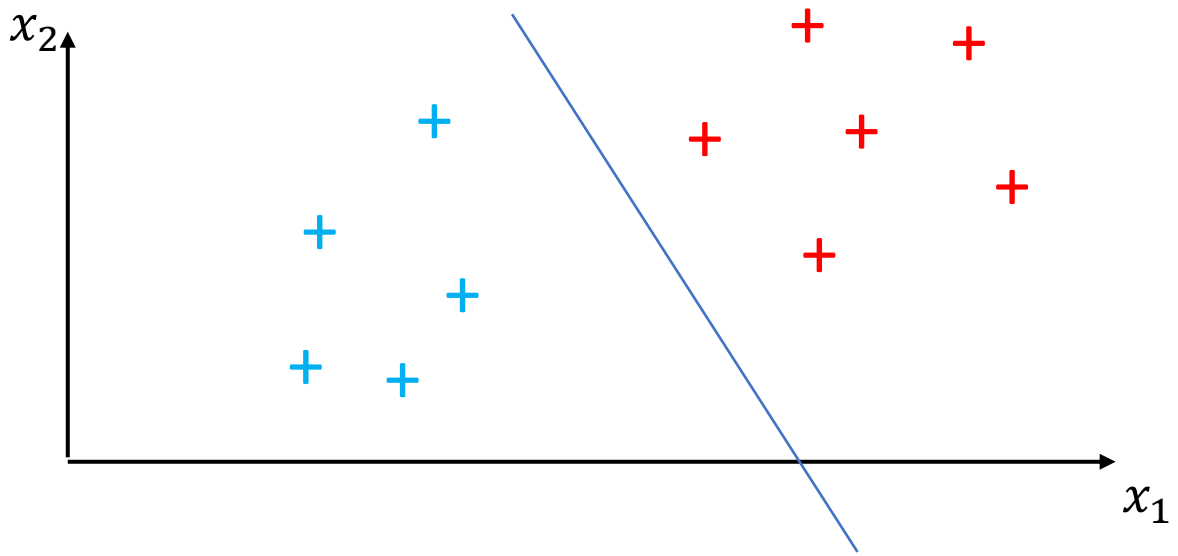




If we only have one feature (x) in our classification, we call it “logistic regression”, otherwise, when we have several features, we call it multiple regression.

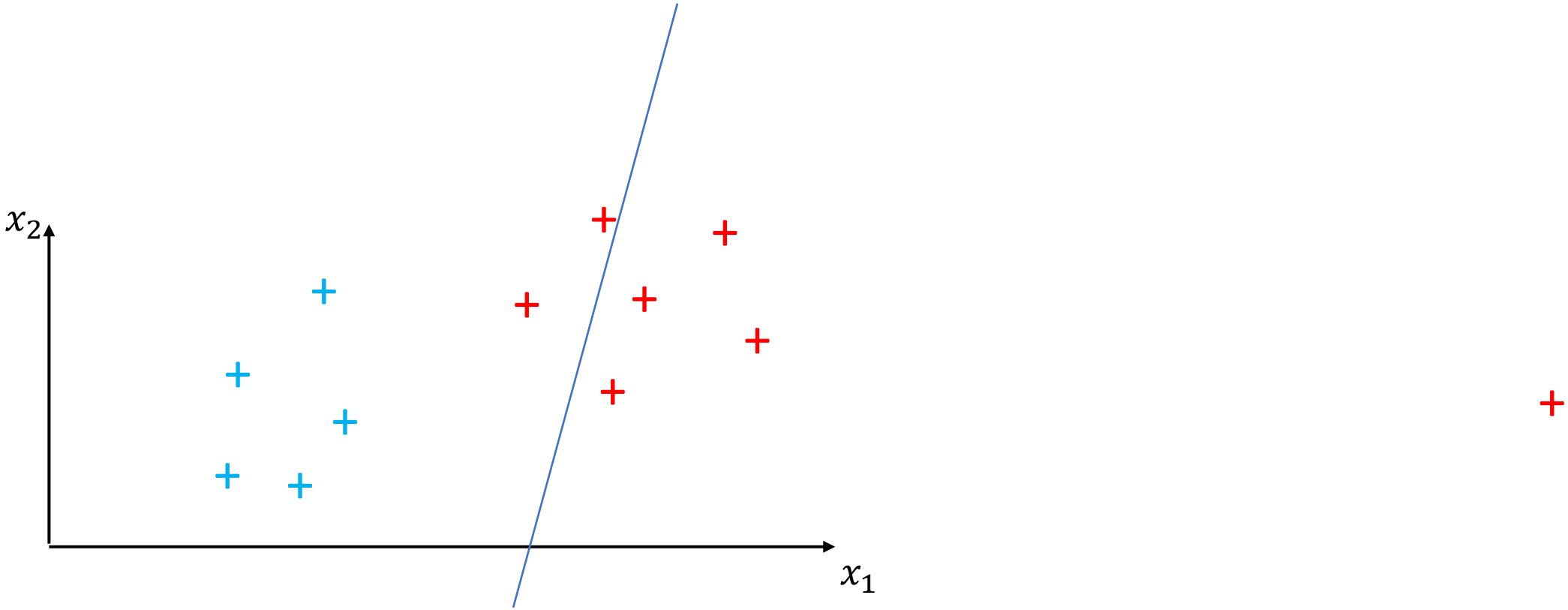
Support Vector Machines (SVM)

Imagine that we have two features (x_0 and x_1 .(The goal of SVM is to find a line that can best divide our data into two classes based on these features.



If we have more than two features, the classifier will be a plane instead of a line, and we call it hyperplane.

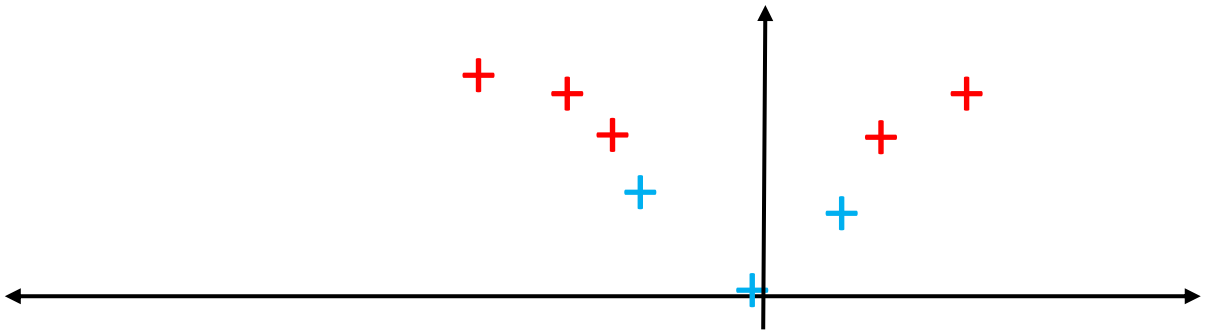
The problem with SVM is that it is not robust against outliers. Even one single outlier can totally change the classifier.


x_2
x_1

There is another problem with SVM, that I am going to explain in the following example. Imagine that we just have one feature (one-dimensional), and our data looks like this:

How are we supposed to separate the two classes using one line?

Well, we use projection. In the following plot, the horizontal axis is our original feature (x_0), and we make up another feature called (x_1) which is x_0^2. It gives us a plot like the following plot:



This process (making up a new feature that helps us to classify the data better) is called “Kerne trick”.