Manish Tripathi <mtripathi@pivotal.io>

Thursday Thunder: Week 8, Learning 8: Support Vector Machines - Robust to Outliers

Manish Tripathi <mtripathi@pivotal.io> Thu, Mar 10, 2016 at 1:38 PM
To: PDS-ALL <PDS-ALL@pivotal.io>

'August' Data Scientists (We have one April DS too- PJ),

So back to haunt you this week. Actually I realized that few weeks back a couple of team members talked about how they understand Random Forests, Boosting , SVM from the implementation perspective and from an overall framework, but not very sure of the exact theory that undergoes of them all.

So I thought this week I can write about SVM in brief. Mainly why they work well and why SVM's are robust to outliers etc. Consider this as a primer on how to explain SVM's to customer if asked. :-) .

So here we go-

Ok Dude. So what are SVM's:

Well SVM as most of us know is a machine learning algorithm used for Supervised Classification. For Regression purpose, we have something called as Support Vector Regressor*.*

Historically SVM's are a modification on the first AI algorithm called as Perceptron

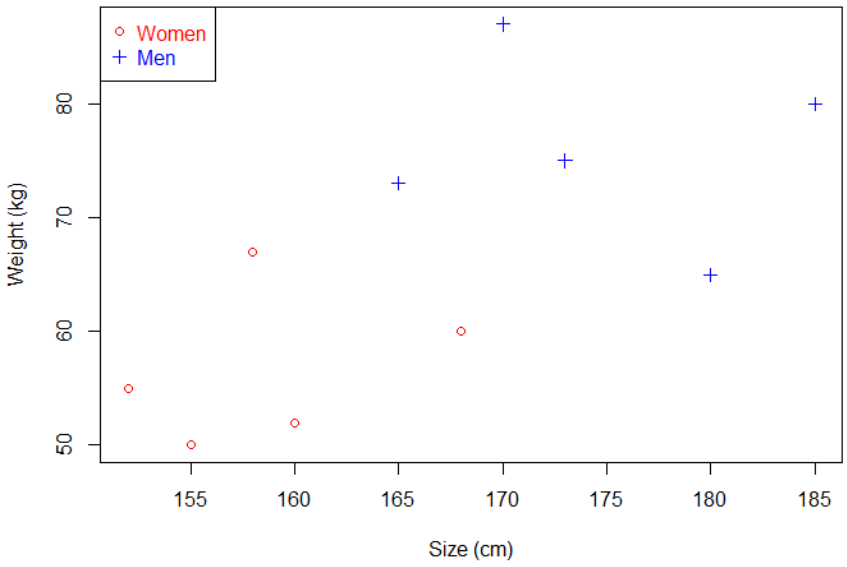
Hmm. So why are the called Support Vector Machines

Well actually they are not always supposed to be called Support Vector Machines. The 'Machines' term is to be used for algorithm which uses a non linear kernel. For all other purpose it should be called as Support Vector Classifier (that's why you would see sklearn uses SVC as one of the module name).

If you put non linear kernel (basically getting a non linear decision boundary) then it should be called Support Vector Machines.

Ok. Got it. But why Support Vectors?

Well before that lets see how Support Vector Classifier works. Given a below data points belonging to two different categories (men vs women) , we want to get a decision boundary which separates them easily based on two features.



Since you see the points can be easily separated by a straight line, (hyperplane) you don't need a non linear decision boundary here. But what can be the possible hyperplanes here?

Aaaargh.. What is meant by Hyperplane? These jargons are confusing...

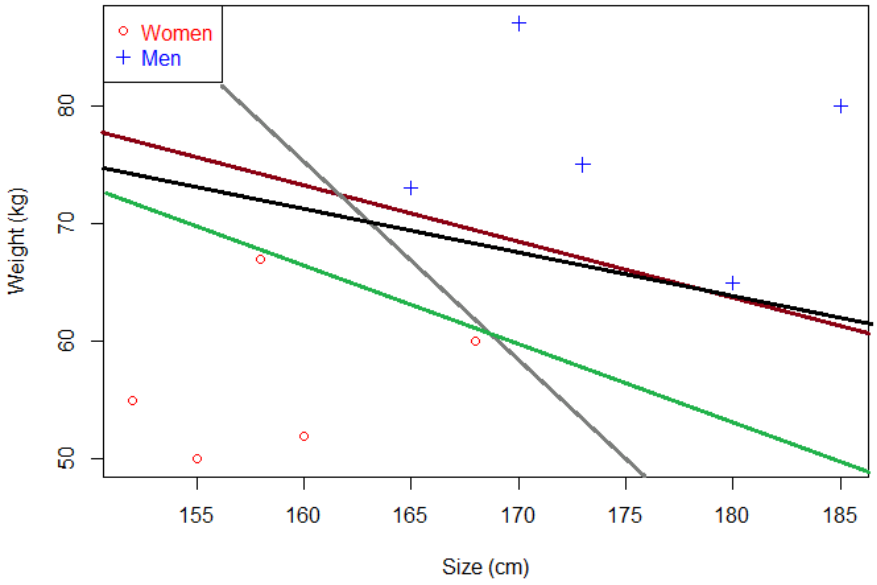
Ha ha. Yeah. They are. So here is a simple way to understand that:

An hyperplane is a generalization of a plane.

- in one dimension, an hyperplane is called a point
- in two dimensions, it is a line
- in three dimensions, it is a plane
- in more dimensions you can call it an hyperplane.

Ok. So then we would have just one hyperplane separating these lines?

No. There are infinite such planes. See below.

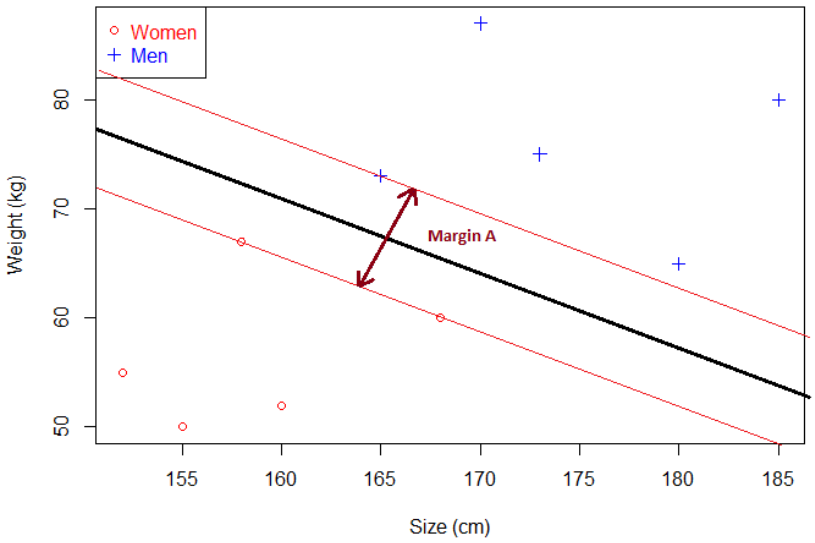


Whoa, Then which one should we choose?

That's where SVM or SVC here helps. The objective of Support Vector is to find that hyperplane which maximizes the 'margin' of the hyperplane with respect to the 'Support Vectors'.

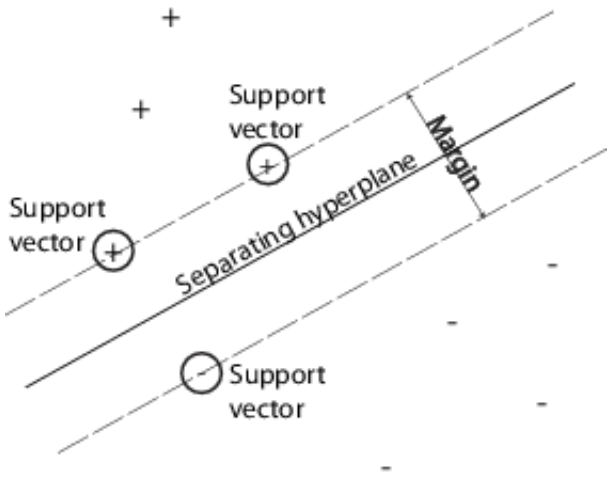
Aah.. What is margin, what is Support Vector?

Good Question. So see the figure below. If you choose the black line as your hyperplane, then Margin is the distance of the closest point to the hyperplane on both sides. *Basically the margin is a no man's land. There will never be any data point inside the margin. But in real life it's not possible , hence we have a soft margin classifier which is what is called as Support Vector Classifier.*



Ok. Then why it is called Support Vector?

Before that you need to understand what is meant by 'Support Vector' in Support Vector Machines. Look at the figure above. The points which are on the red line (at the margin boundary) are called as Support Vectors. Basically Support Vectors are those data points which are at or within the margin boundary of a hyperplane. See below.



Ok. So I got it what does Support Vector mean. But why is this algorithm called as Support Vector Machine/Classifier. There are other data points too outside the margin.

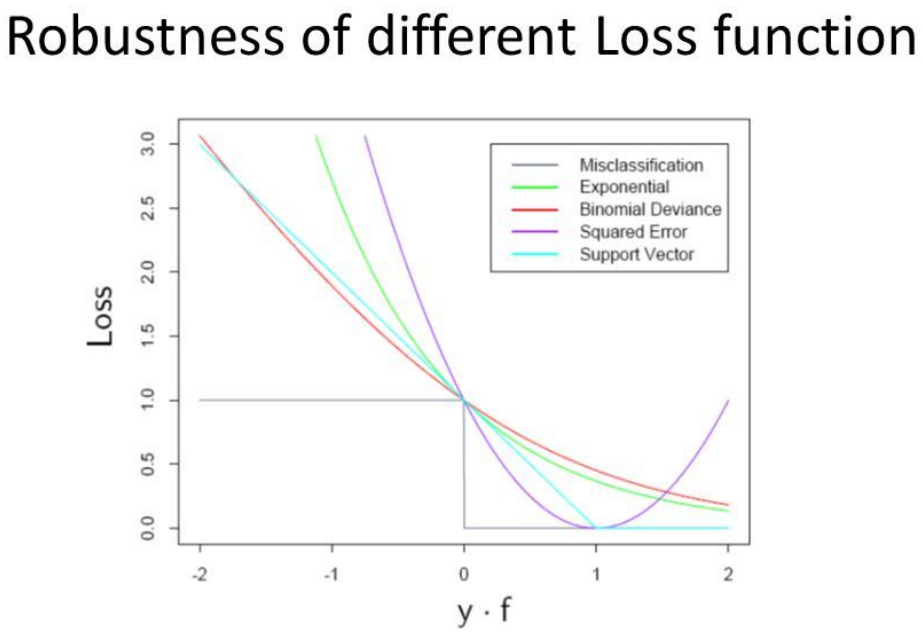
Great question. So the reason is because this algorithm finds a hyperplane or decision boundary by ***ONLY*** focusing on Support Vectors. Or in other words, even if you have million data points, your hyperplane is only defined by the Support Vector points.

What?. What does that supposed to mean?

It means that the loss function in SVM, which is also called as Hinge loss, is not affected by any point outside the margin. The hyperplane is only defined by the Support Vectors.

Oooooh.. So is that why SVM's are robust to outliers?.

Bingo. Now you are thinking. See below.



The y-axis is the loss of the corresponding loss function. X-axis (y.f) is the linear combination of the features(your model equation). If you see , SVM loss (hinge loss, since it is shaped like a hinge), goes to zero when y.f is more than equal to 1.

This is what is the margin defined for SVM. What it means is any point which is not a Support Vector i.e outside the margin, doesn't contribute anything to SVM hyperplane loss function (loss is zero).

Hence any outlier or far away point will not have any effect on SVM decision boundary. That is why they are pretty robust to outliers. If you see Binomial Deviance loss function above (that is logistic regression loss function)***,*** it is very similar to SVM loss. Hence Logistic regression is very similar to SVM***,*** just that it puts a sigmoid transformation to get probabilities instead of 0,1 output as SVM.

Hmm.. SO that means we cant get probabilities from SVM.

You can. Just modify the output like Logistic regression. This transformation used in SVM is called as Platt Scaling.

Cool. Enough for today. Stop your rant!