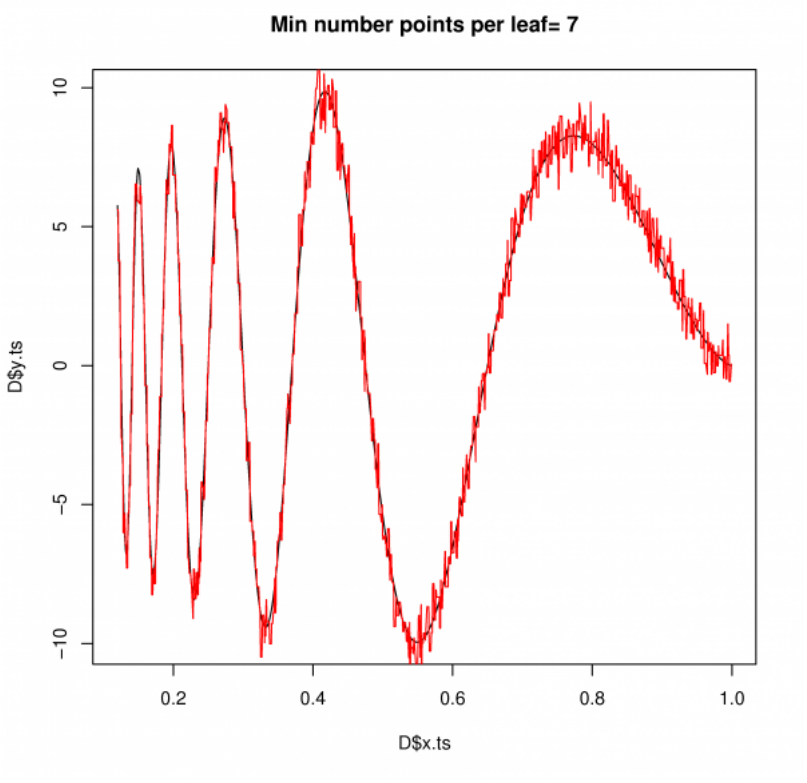
When to use regression trees/forests?

As I was looking for a fine regression algorithm for my problem. I found out one can do that with simple decision trees as well, which is usually used for classification. The output would be something like:



The red *noise* would be the prediction states of such a tree or forest.

Now my question is, why at all to use this method, when there are alternatives, that really try to figure out the underlying equation (such as the famous *support vector machines* SVM). Are there any positive / unique aspects, or was a regression tree more a nice-to-have-algorithm?

regression machine-learning svm random-forest

asked Jan 23 at 11:46

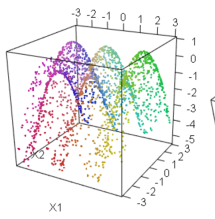
user3085931
108 4

1 Answer

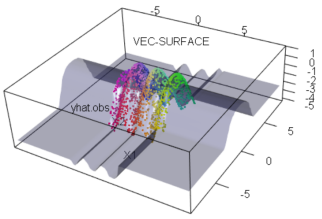
As your figure exemplifies, single decision trees would under perform SVM in most problems. But an ensemble of trees as random forest, is certainly a useful tool to have. Gradient boosting is another great tree derived tool.

SVM and random forest(RF) algorithms are not alike, of course. But both are useful for the same kind of problems and provide similar model structures. Of course the explicitly stated model structures of forest/trees as hierarchical ordered binary step-functions are quite different from SVM regression in a Hilbert space. But if focusing on the actual learned structure of the mapping connecting a feature space with a prediction space the two algorithms produce models with similar predictions. But, when extrapolating outside the feature space region represented with training examples, the *"personality"* of the model takes over and SVM and RF predictions would strongly disagree. See the example below. That's because both SVM and RF are predictive models, your can see that both SVM and RF did a terrible job extrapolating.

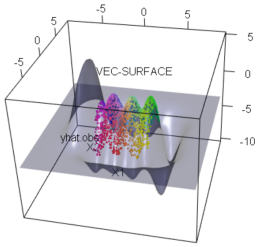
$y = sin(x_1\pi) - .5{x_2}^2$



training samples



random forest model(grey surface)



SVM model (grey surface)

So No SVM is not trying to figure out *the underlying true equation* and certainly not anymore than RF. I disagree with your platonist view-point, expecting real life problems to be governed by some algebra/calculus math, that we humans coincidentally happen to teach each other in high school/uni. Yes in some cases, such simple stated equations are fair approximations of the underlying system. We see that in classic physics and accounting... But that does not mean the equation is the true hidden reality. The *"all models are wrong, but some are useful"* would be one statement in a conversation going further from here...

I does not matter if you use SVM, RF or any other appropriate estimator. You can always inspect the model structures and perhaps realize the problem can be described by some simple equation or even develop some theory explaining the observations. It becomes a little tricky in high dimensional spaces, but it is possible.

In general rather consider RF over SVM, when:

- You have more than 1 million samples
- Your features are categorical with many levels(not more than 10 though)
- You would like to distribute the training on several computers
- Simply when a cross-validated test suggests RF works better then SVM for a given problem.

.

```
library(randomForest) #randomForest
library(e1071) #svm
library(forestFloor) #vec.plot and fcol
library(rgl) #plot3d

#generate some data
X = data.frame(replicate(2,(runif(2000)-.5)*6))
y = sin(X[,1]*pi)-.5*X[,2]^2
plot3d(data.frame(X,y),col=fcol(X))

#train a RF model (default params is nearly always, quite OK)
rf=randomForest(X,y)
vec.plot(rf,X,1:2,zoom=3,col=fcol(X))

#train a SVM model (with some resonable params)
sv = svm(X, y,gamma = 1, cost = 50)
vec.plot(sv,X,1:2,zoom=3,col=fcol(X))
```

edited Jan 25 at 0:26

answered Jan 24 at 23:59

Soren Havelund Welling
2,871 5 17

thanks for your precise answer. Concluding (and making it very short) you want to tell, there is no rule for let's say *use this category of algorithms, for these kind of problems etc.*, since their drawbacks are not predictable for every case. Means whatever I'm trying, I have to evalute in the end and if possible, compare with every available algorithm evaluation there is. – user3085931 Jan 26 at 14:22

However I don't get the meaning from one of your points: *You would like to distribute the training on several computers*. In the end I want to have a very accurate prediction. If I can distribute my training, but the result is distinctively worse than my alternatives, why should I prefer then ? – user3085931 Jan 26 at 15:12

1 To answer the first comment, you typically start with simple ML models (linear regression / logistic regression) and see if it works good enough. If not you, try to realize whats missing and try a more complex models or some data fix. With experience you may realize typical pro's and con's pick a good model first time and/or you just learn to brute force running thousands of model candidates including grid search and nested cross-validation to assist model selecting. – Soren Havelund Welling Jan 26 at 15:52

1 The biggest data set so far I used RF on was 16GB. 5000 rows some 500.000 columns. To compute 500 trees on one PC took a day. To compute 3 trees on each of 80 nodes in a cluster took minutes. Each tree can be combined afterwards. You cannot distribute SVM as easily. Lot of times big is not needed, to do good enough. Just dump 90% of samples or features. *"but the result is distinctively worse than my alternatives, why should I prefer then ?"* -If you are a **server nerd**, you go with bigger for the fun :) , otherwise you go with what get you the best results of course. – Soren Havelund Welling Jan 26 at 16:02

1 If you expect your data structure to be some kind of smooth yet complex polynomial surface, and you have limited data points to train on. SVM would probably be quite superior. – Soren Havelund Welling Jan 26 at 16:10

One last question: if my function to learn is e.g. the velocity of an object to track (in image processing), then pretty sure I'm clearly getting a polynominal function - which is more or less what I am looking for in this special case. However, doesn't everything you want to figure out by regression underlie a polynominal structur ? For a better understanding, might you could give a simple example where this wouldn't be the case? – user3085931 Jan 27 at 7:28

Add Another Answer