Classification using Supervised Learning Algorithms
Peter LeBrun
Feb 2 2015

Classification Problems

For this paper, I chose two sets of data from the UCI Machine Learning repository. Set 1 is Haberman's Survival Dataset – a dataset used to predict whether someone with breast cancer survived at least 5 years after their initial diagnosis. Descriptions of the data are below (# Positive Nodes means the number of cancerous lymph nodes found near the breast tissue). As you can see from the table, the major difference between the classifications is the average number of positive nodes for that particular class.

Set 2 is data being used to determine whether a bill is real or fake. The attributes are the (Variance/Skewness/Curtosis/Entropy) of a Wavelet Transformed Image of the bill. There are several major differences between the classes here. For instance, the median variance for fake bills is 2.55, whereas the median variance for real bills is -1.81. We see a similar pattern for skewness, and a small difference in curtosis. Entropy Is basically the same for both groups. We will take this into account as we turn our continuous features into discrete features for decision trees.

Breast Cancer Survival (1 = Survived >= 5 Years)

Class		Age	Year (+ 1900)	# Positive Nodes
0 (n = 81)	Min	34.00	58.00	0.00
	Max	83.00	69.00	52.00
	Median	53.00	63.00	4.00
	Mean	53.68	62.83	7.46
	StDev	10.17	3.34	9.19

Class		Age	Year (+ 1900)	# Positive Nodes
1 (n = 224)	Min	30.00	58.00	0.00
	Max	77.00	69.00	46.00
	Median	52.00	63.00	0.00
	Mean	52.02	62.86	2.79
	StDev	11.01	3.22	5.87

Bank Note Counterfeits (0 = Counterfeit)

Class		Variance	Skewness	Curtosis	Entropy
0 (n = 763)	Min	-4.29	-6.93	-4.94	-8.55
	Max	6.82	12.95	8.83	2.45
	Median	2.55	5.68	0.70	-0.55
	Mean	2.28	4.26	0.80	-1.15
	StDev	2.02	5.14	3.24	2.13

Class		Variance	Skewness	Curtosis	Entropy
1 (n = 609)	Min	-7.04	-13.77	-5.29	-7.59
	Max	2.39	9.60	17.93	2.14
	Median	-1.81	0.17	0.37	-0.66
	Mean	-1.87	-0.99	2.15	-1.25
	StDev	1.88	5.40	5.26	2.07

Analysis

For this project, I fit five models to each of the classification problems in an effort to find the best performing model. I am ashamed to admit that I was somewhat limited by the package I was using, and by my own computer.

SVMs (See Appendix: Images, SVM)

Support Vector Machines performed very well for the bank note data set, obtaining accuracy near 0.99. This was regardless of the regularization parameter, suggesting that we have a pretty wide margin between the real and fake notes' wavelet transform images. I would imagine there exists some dimensionality in which these classes are linearly separable.

The breast cancer survivors did not perform as well. The best performance we got was about 0.72 (this was with a linear kernal and a small regularization parameter around 0.01). This suggests that the data are not easily classifiable. Given that the only parameter that likely has predictive value is the number of cancerous nodes, we don't have much to go on here. I was unable to run the Polynomial kernel, as my computer kept timing out when I went to run it.

Decision Tree/Boosting (See Appendix: Images, Trees/Boost)

This is where I started running into issues. I spent all my time learning sklearn and came to find that its Decision Tree Classifiers don't allow for pruning (I accept that you likely have to dock me points for this :/). I turned the continuous variables into booleans by splitting on the mean of each value (see the tables above).

For the bank notes, decision trees gave us about 0.83 accuracy, and the boost gave us a slight increase over that – about 0.84. I'm sure that with pruning we could have increased this for the initial decision trees. I couldn't get sklearn to print out the trees, but I would bet that for bank notes, the only nodes making a difference would be the variance and the skewness.

For breast cancer, I am guessing that the root node was the number of positive cancerous lymph nodes. We got accuracy around 0.62 using decision trees, and about the same using Boosting. However, as the learning rate grew, the performance of AdaBoost decreased significantly.

K-Nearest Neighbor (See Appendix: Images, KNN)

The bank notes did very well using KNN – with accuracy around 100%. As suggested in the section on SVM, I am guessing that the data is linearly separable in some dimension. Interesting to note that as K grew, performance decreased. This makes sense – if the data is linearly separable, then we really only need two clusters given that we are only training a binary classifier here.

The Breast Cancer Survivors improved in performance as we increased K. This suggests that what we've got here is not two distinct “classes” of data, but rather a more complex and nuanced set of instances identifying breast cancer survivors. It really begs the question: what if there were more features?

Neural Networks (See Appendix: Images, NN)
This was probably the final major roadblock I encountered: sklearn's neural networks only do *unsupervised* learning. So, I worked with PyBrain. Unfortunately, the validation metrics I collected were not calculated the same as for the other four models. I was also limited by the performance of my computer, and so I had to use somewhat simpler models in the interest of time.

The breast cancer survivor group had a Mean Squared Error of about 0.2 regardless of the number of hidden nodes we used, but the bank notes started at about 0.22 (two hidden nodes) and decreased to about 0.18 (eight hidden nodes).

This suggests a number of improvements that we could have made, namely more complex neural networks (additional hidden layers, even more nodes in our hidden layers), using different activation functions (we used a basic sigmoid function here), or even the inclusion of bias.

Final Thoughts

I had a large number of takeaways from this project. First of all, these models are far more complicated to actually work with and understand than the books or the videos suggest. I think this is a function of experience with the models (and the packages) - you could say there is a “learning curve”.

Secondly, I chose data sets that appeared simpler to work with. This was primarily due to my own lack of experience with Machine Learning – I wanted data sets that I didn't find intimidated and that wouldn't require a lot of pre-processing, so that I could spend more time with sklearn and pybrain. While this greatly reduced the time required for me to understand the data (and I did need the time to better learn the packages) , I think it did a disservice to the models we have available. Most of the analysis we needed could have been done in excel for these data sets. Running a neural network on these data sets was about the equivalent of using a snow plow on a small walkway.

Additionally, there was a very significant time investment just to learn the packages – between the inner workings of sklearn, to the very cryptic documentation (and paltry stackoverflow questions!) of pybrain, and then the nuances of numpy, scipy, and matplotlib. These are things I underestimated.

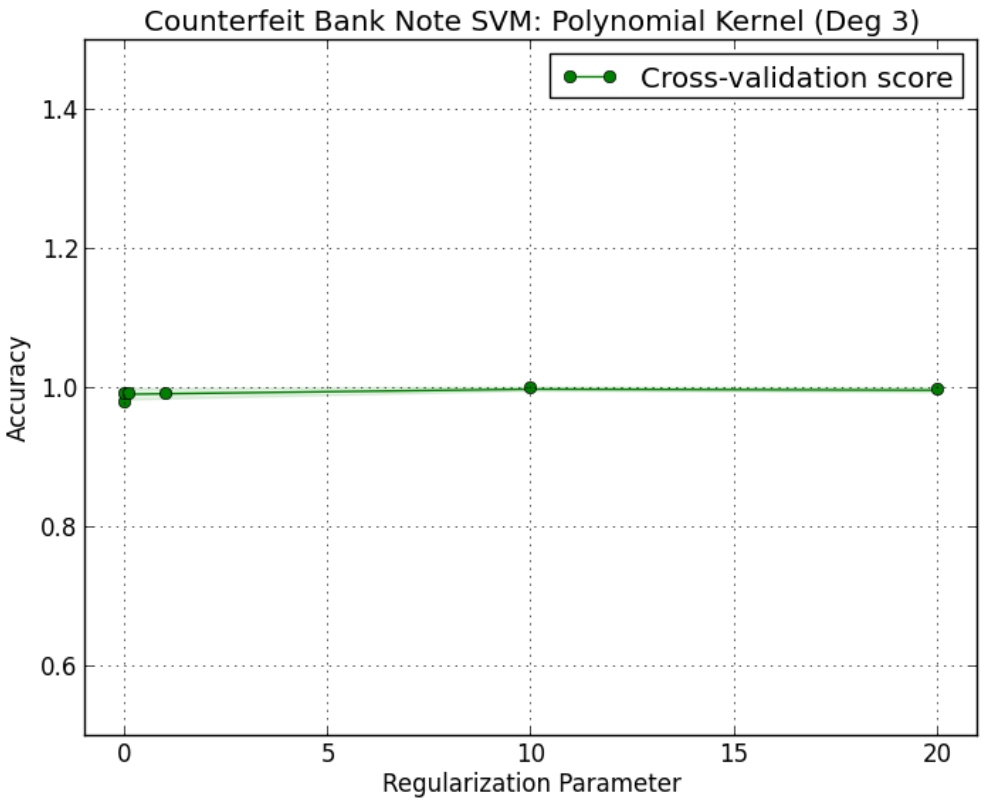
In terms of the models themselves, it seems there are a number of questions I could have asked myself beforehand that would have changed my approach to the problem. For instance – checking how easily the data could be linearly separated. That alone tells us a lot about which models are going to work best – the counterfeit bank notes responded very well to the SVM and KNN models, whereas the breast cancer survivors did not respond particularly well to any of our models. Though – we did get about 80% classification accuracy using KNN, in terms of predicting cancer survival, 80% is probably pretty good.

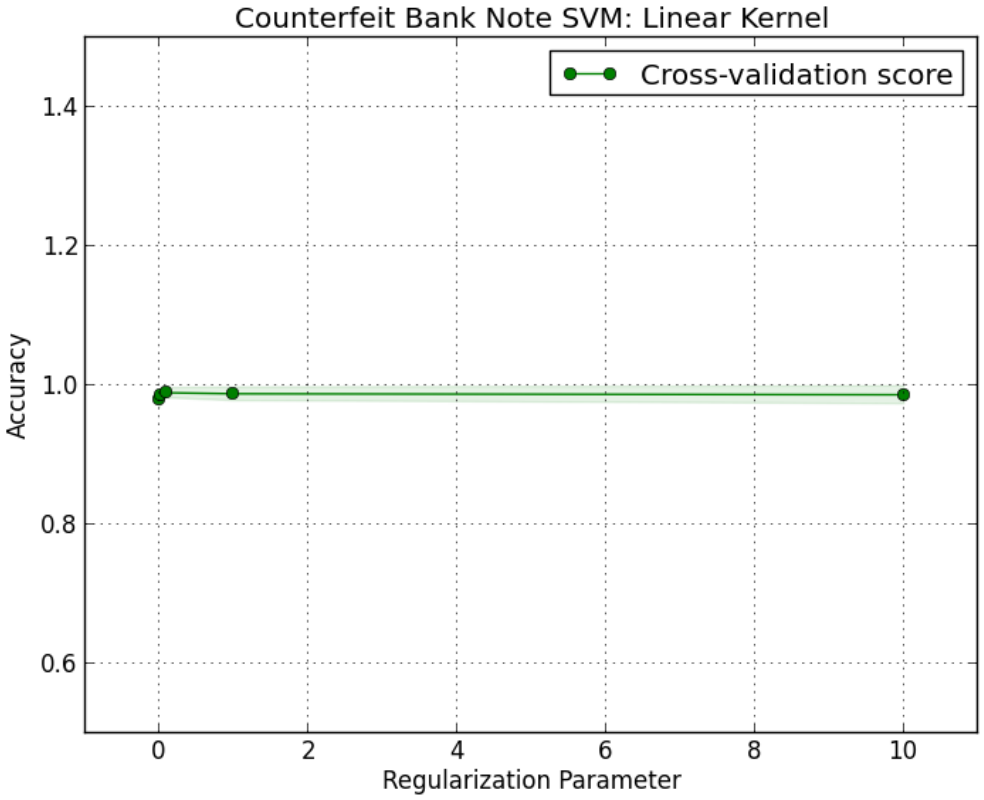
How could we have improved these? My first thoughts are pretty obvious: more data, and more features. Specifically, I could have combined features on the breast cancer survivors (i.e. age > 50 and positive nodes > 20, etc). Or, if I were performing the experiment in the lab, I could have collected more data there. For instance – maybe “diet high in fiber” or “exercises > 60 min/week” - I am sure that features such as these would have proved very valuable for our needs.

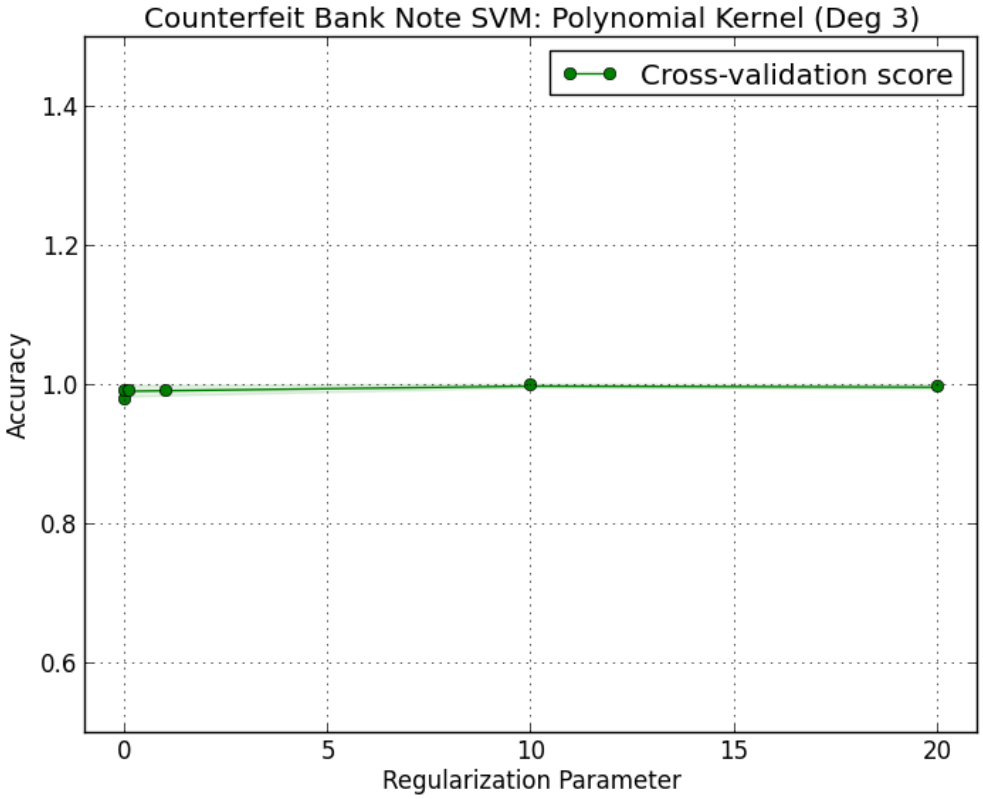
Appendix: Citations

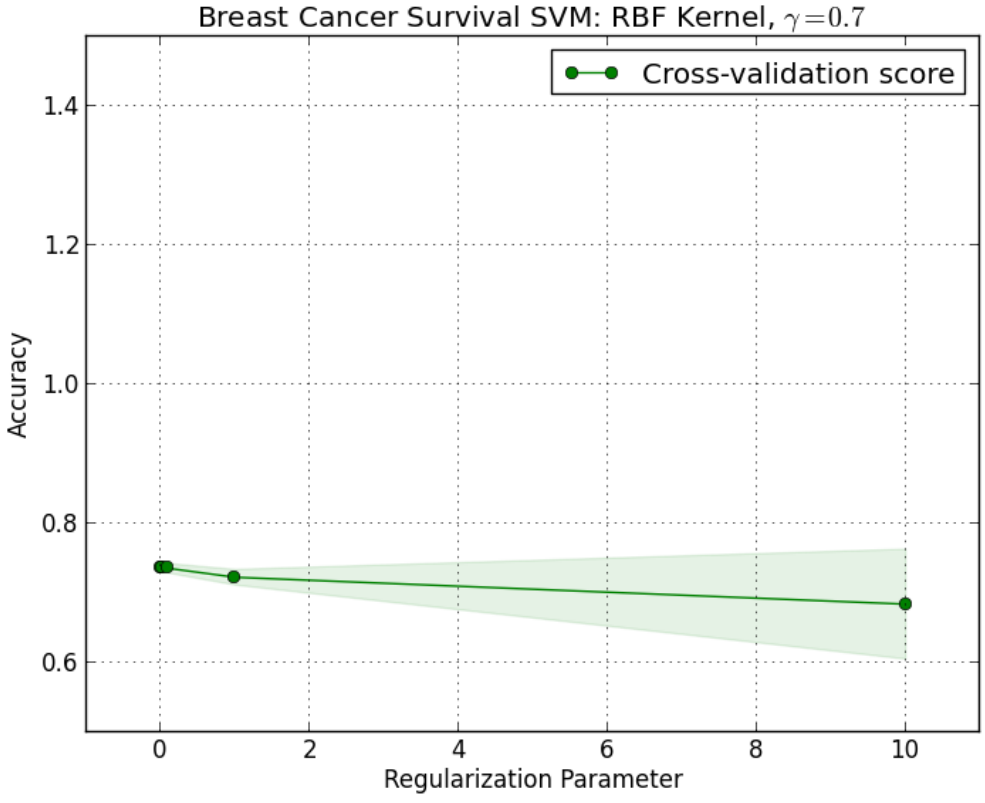
All data sets from Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

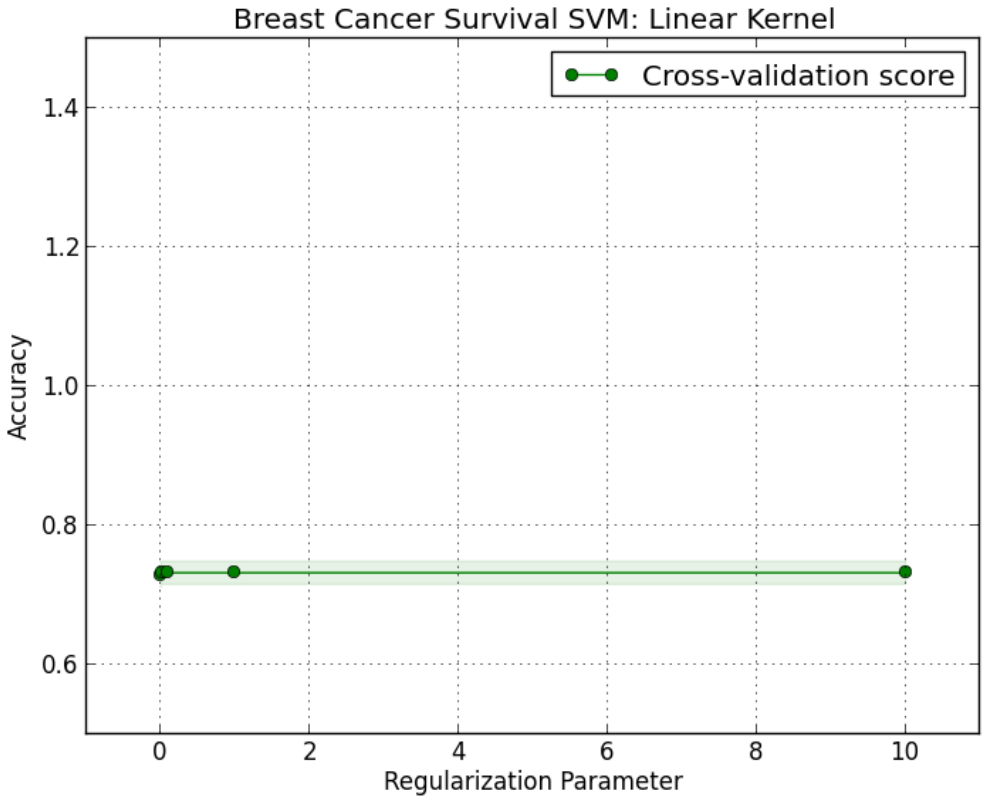
Appendix: Images, SVM



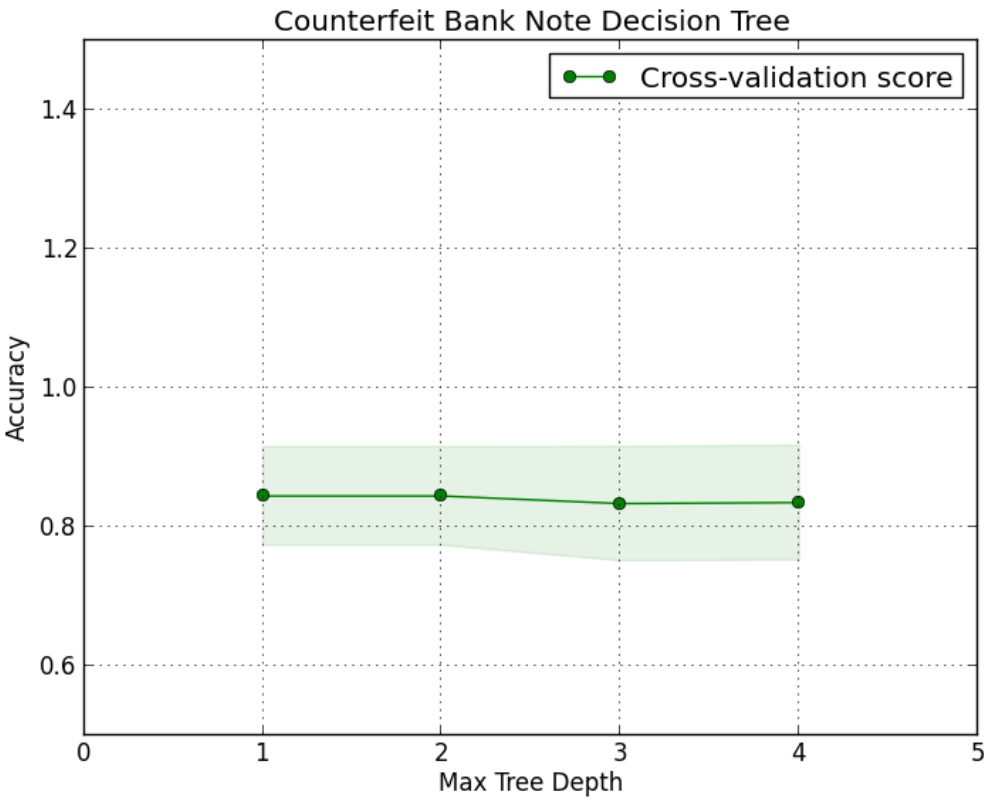


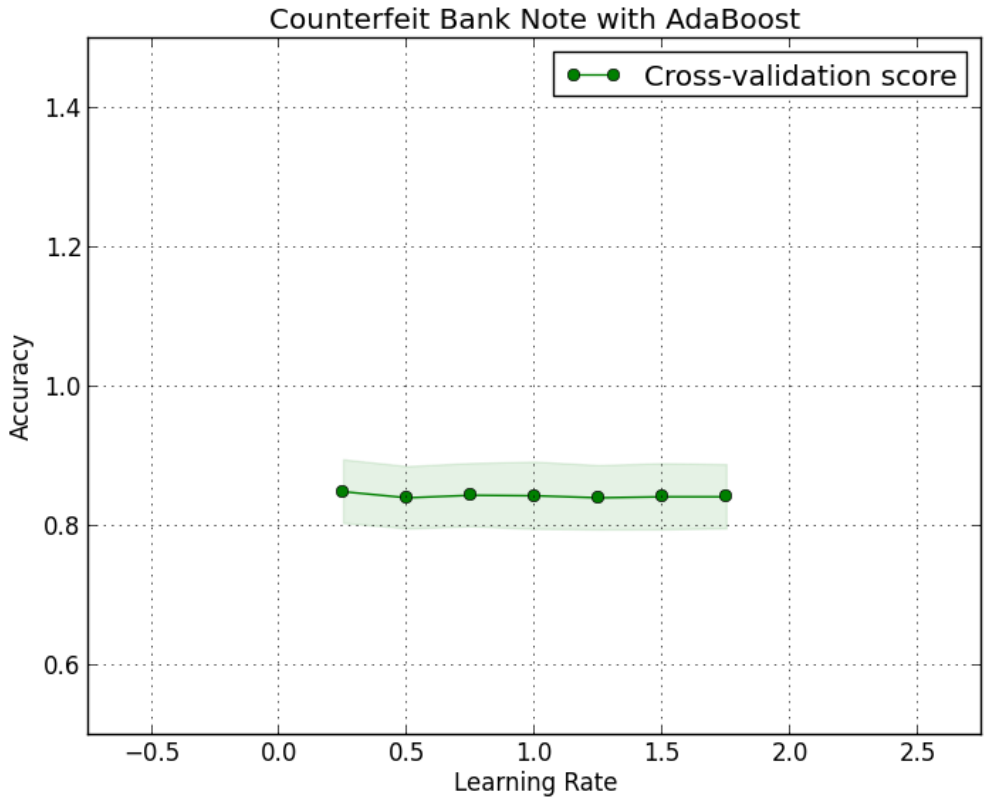

Counterfeit Bank Note SVM: Polynomial Kernel (Deg 3)
Cross-validation score
Accuracy
Regularization Parameter
1.4
1.2
1.0
0.8
0.6
0
5
10
15
20

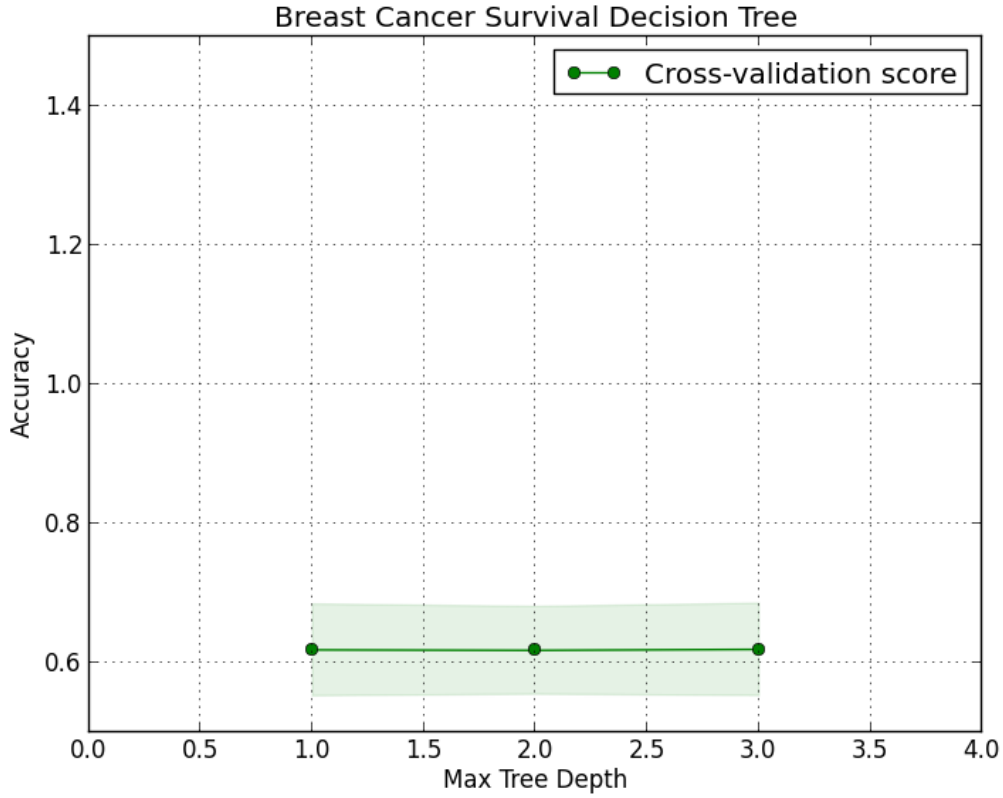

Breast Cancer Survival SVM: RBF Kernel, $\gamma=0.7$
Cross-validation score
Accuracy
Regularization Parameter
1.4
1.2
1.0
0.8
0.6
0
2
4
6
8
10

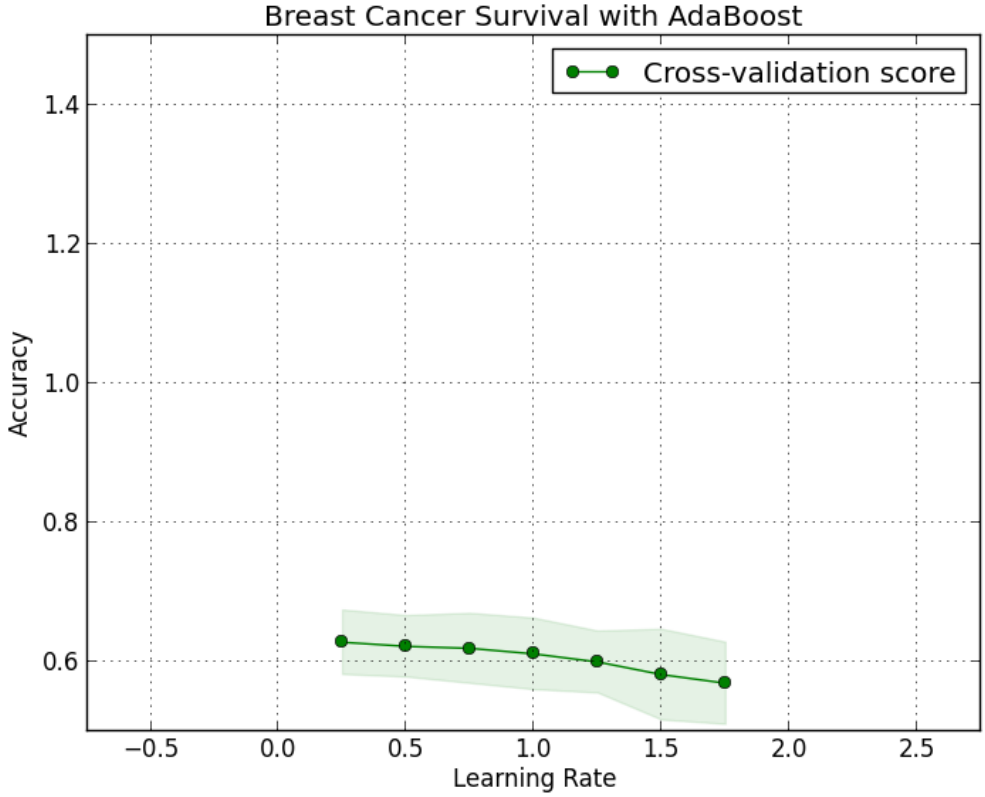

Breast Cancer Survival SVM: Linear Kernel
Cross-validation score
1.4
1.2
1.0
0.8
0.6
Accuracy
0
2
4
6
8
10
Regularization Parameter

Appendix: Images, Decision Tree/Boost

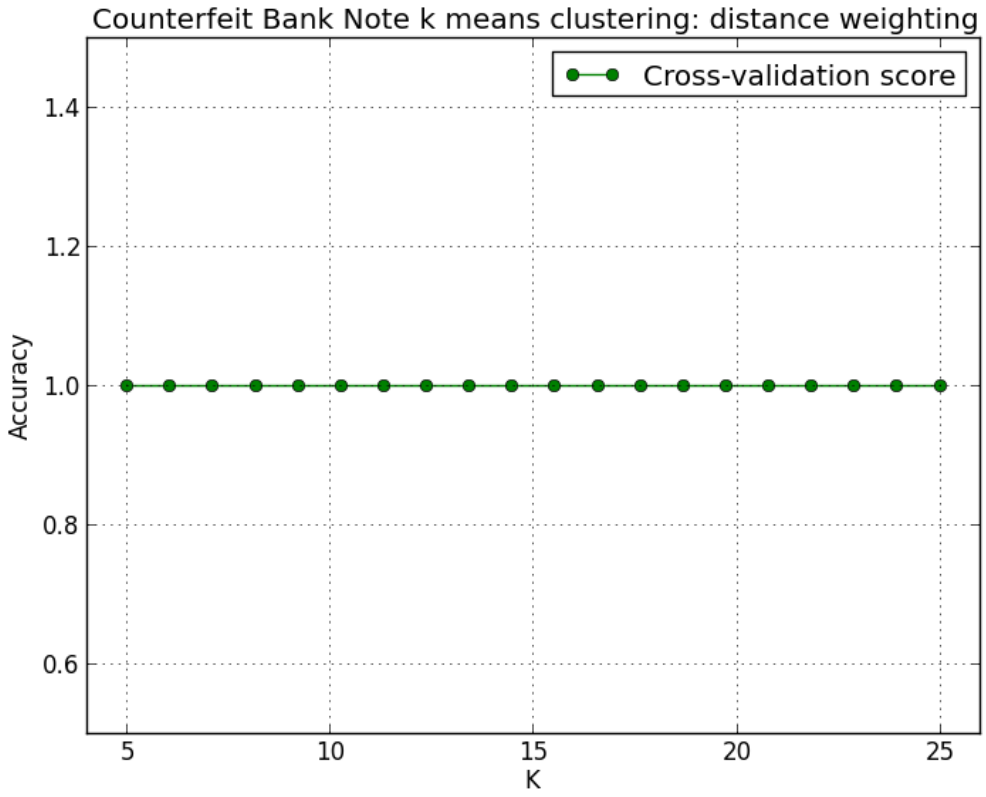


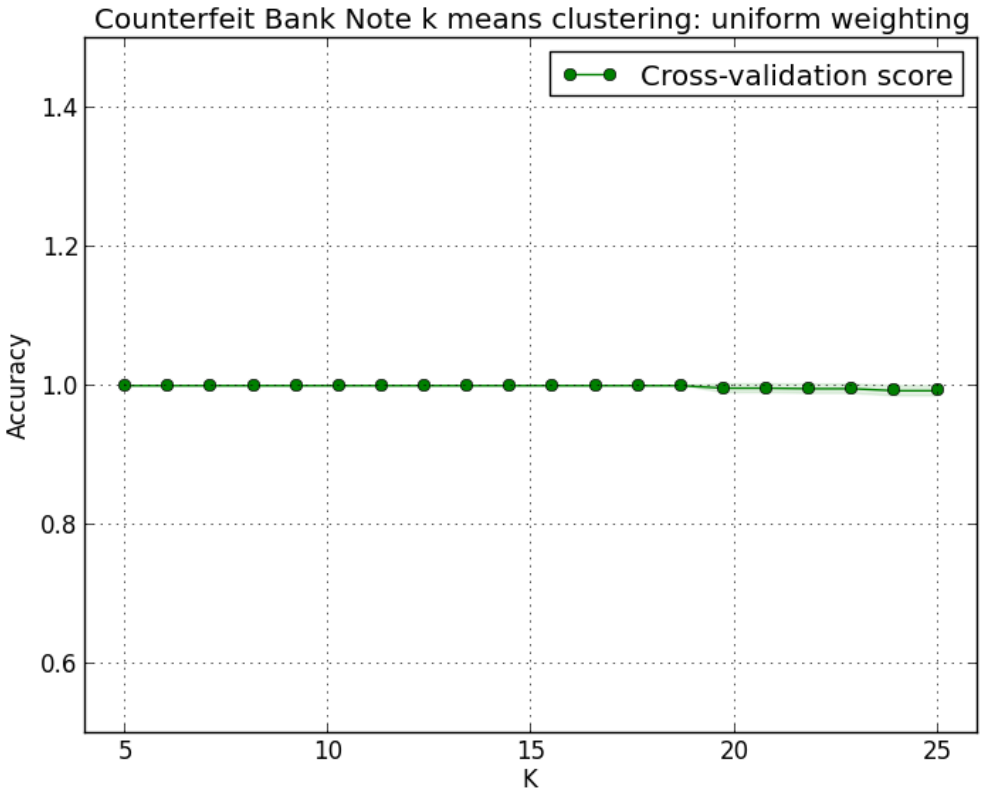


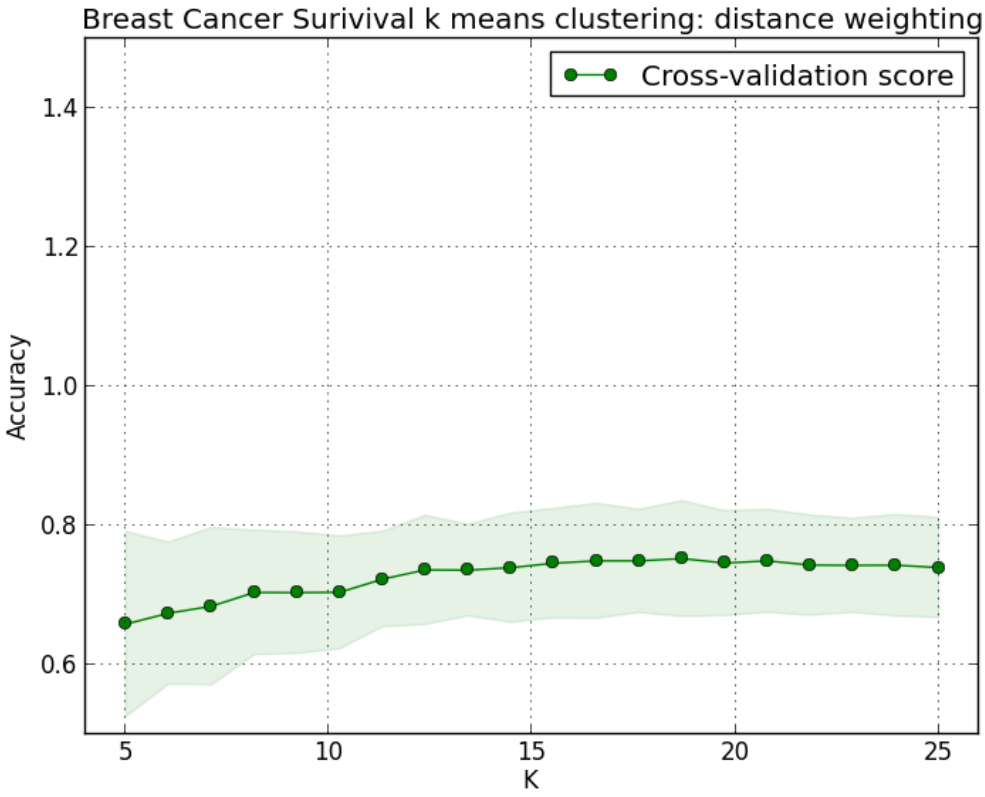

Breast Cancer Survival Decision Tree
Cross-validation score
Accuracy
Max Tree Depth
1.4
1.2
1.0
0.8
0.6
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

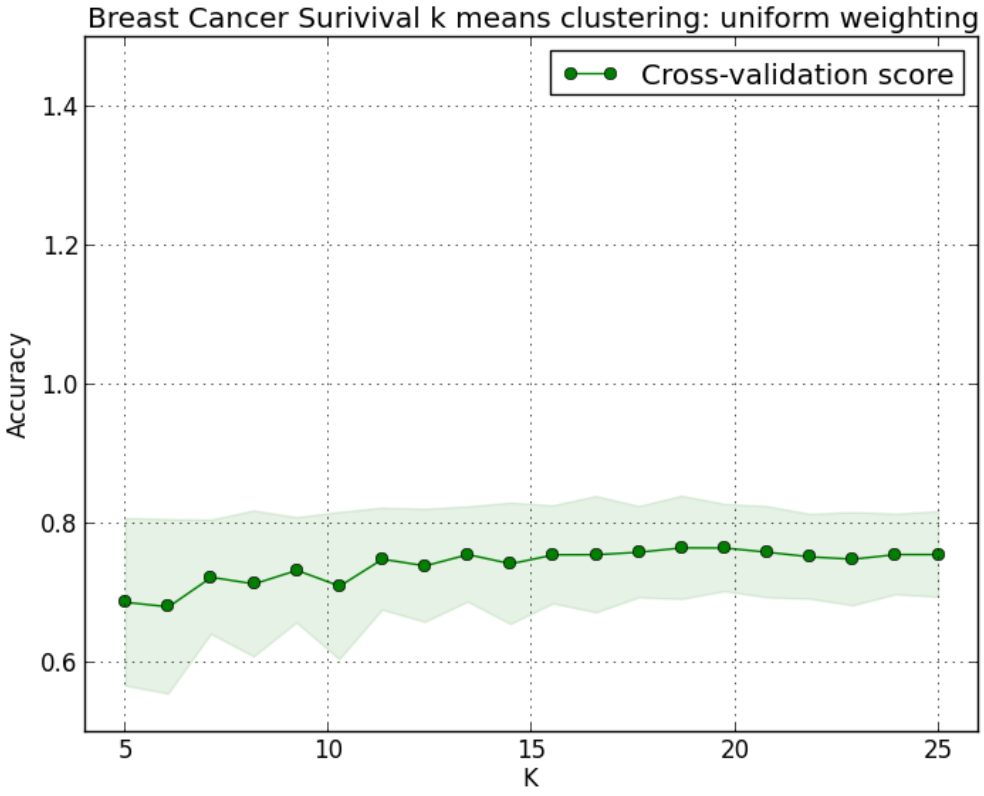

Breast Cancer Survival with AdaBoost
Cross-validation score
Accuracy
Learning Rate
1.4
1.2
1.0
0.8
0.6
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

Appendix: Images, KNN






Breast Cancer Surivival k means clustering: distance weighting
Cross-validation score
Accuracy
K
1.4
1.2
1.0
0.8
0.6
5
10
15
20
25


Breast Cancer Surivival k means clustering: uniform weighting
Cross-validation score
Accuracy
K
1.4
1.2
1.0
0.8
0.6
5
10
15
20
25

Appendix: Images, NN