Andrey Kurenkov
Project #1
CS 4641

Supervised Learning Report

Datasets

Australian Sign Language Signs: This is a set of numeric data collected from different people performing a total of 95 different signs several times. The classification problem is to be able to classify the sign being performed by a person given measurements of the state of their hands. The data was recorded using a Fifth Dimension Technologies gloves on both hands as well as magnetic position trackers attached to each hand. Thus, each instance in the dataset has the classification of what sign it is as well as a set of frames from the recording containing the 3D location and 3D rotation of both hands as well as a 1D measurement of the degree to which each finger is bent on each hand. There are 27 instances for each sign, with 3 instances gathered from 9 different people, for a total of 2565 instances.

Despite the clear structure of the data, the data are provided only as a sequence of frames for each sign recording and not a consistent set of features. To formulate the dataset as a classification problem, each measured value from each numbered frame was treated as a feature. Additionally, the number of frames is not the same for all the instances in the dataset, with the average number of frames being 57. To have a consistent number of features, all instance were converted to have 80 frames by ignoring any extra frames for instances with more than 80 frames or by adding additional features all having a value of -1 for instance with fewer than 80 frames. Since there are 22 different values in each frame and 80 frames for each instance, there are 760 feature attributes in the classification problem that are all numeric.

Email Folders: This dataset was constructed for this project using data exported from my own Gmail account. The exported data was in the mbox format and contained all information about the emails received and sent by me in Gmail. The classification problem created using this data was the task of classifying which Gmail folder an email should be placed in given features about the email extracted from its mbox information. There are 11 folders with varying numbers of emails in my Gmail account, so given the the mbox data there could only be 11 classes but many possible sets of features.

Folder	**Number of Emails**
Academic	76
Personal-Programming	120
Professional-TA	727
Professional-Research	354
Professional-RISS	177
Trash	539
Professional	135
Group work-SolarJackets	748
Group work-IEEE	44
Financial	63
Personal	243

Table 1. Number of emails in each folder of my Gmail dataset

One set of features was settled on early and the experiments were primarily ran with them. These features include the sender of the email, the domain from which the email was sent, as well as 550 binary features of the form "Does this email contain the word X." The 550 words were selected to be the 50 most frequently occurring words in each folder that also occur at least 25% of times in emails within that folder. Some experimentation with these numbers was performed, so the number of words and percent threshold were chosen to create the best classification rates possible.

Why are these interesting datasets?

The datasets are interesting both because of potential applications of well performing classifiers for their data as well as the nature of the data and the challenges they pose to machine learning. The signs dataset allows for training a classifier of sign language gestures, which has an obvious use: translating speech in sign language to a verbal language or written speech. Though classification based on video may be more applicable in more situations, showing that a classifier can work well with these sensors proves that the problem is fit for machine learning.

As with the emails dataset, the number of features greatly exceeds the number of classes which may result in overfitting due to it being easier to get a more specific hypothesis. It is also notable for representing data that has a temporal aspect, or alternatively a specific sequence to the data, which is more often seen with Hidden Markov Models than supervised learning. Lastly, it is interesting to see that adding default meaningless values to instances where those values don't have any actual recorded value did not hinder learning successful classifiers. My intuition is that these extra feature values could still be valuable for learning by representing more information about the length of a given gesture.

The emails dataset was created with the specific intention of determining whether writing an application to automatically suggest a folder to put an email into is a viable idea. Using my own emails in this project creates a realistic dataset for such an application and allows me to test these algorithms on data not specifically gathered for the purpose of machine learning. It is also interesting for the similarity of this task to the notable machine learning application of spam filtering, although the multiple classes of email and lack of obvious spam-like features make this task significantly more challenging.

There are also several elements that make this dataset interesting from a machine learning perspective. The question of what features should be included and overall design of the classification problem are interesting problems in their own right. Deciding on initial features through intuition and performing quick experiments to refine how many and which features are included gave me some experience with how a large collection of data can be converted into a set of instances for machine learning. A notable result was that including features that intuitively seem like they should be strong classifiers, such as words that occur almost exclusively in one folder, may result in worse overall learning since the data is not as representative of the dataset as a whole. Beyond the classification problem itself, this dataset is interesting because it represents learning about text data unlike the signs dataset, it has classes with widely varying numbers of instances unlike my second dataset, and it has a very large number of feature relative to the classes. These interesting aspects suggested that it should reveal some traits of supervised learning algorithms other datasets would not, and in particular that my second dataset wont. A notable aspect of this dataset is also that all the algorithms perform at least somewhat well and none completely succeed, which means the dataset has features that are fit for all the algorithms but are not sufficient for fully accurate classification.

Learning Curves

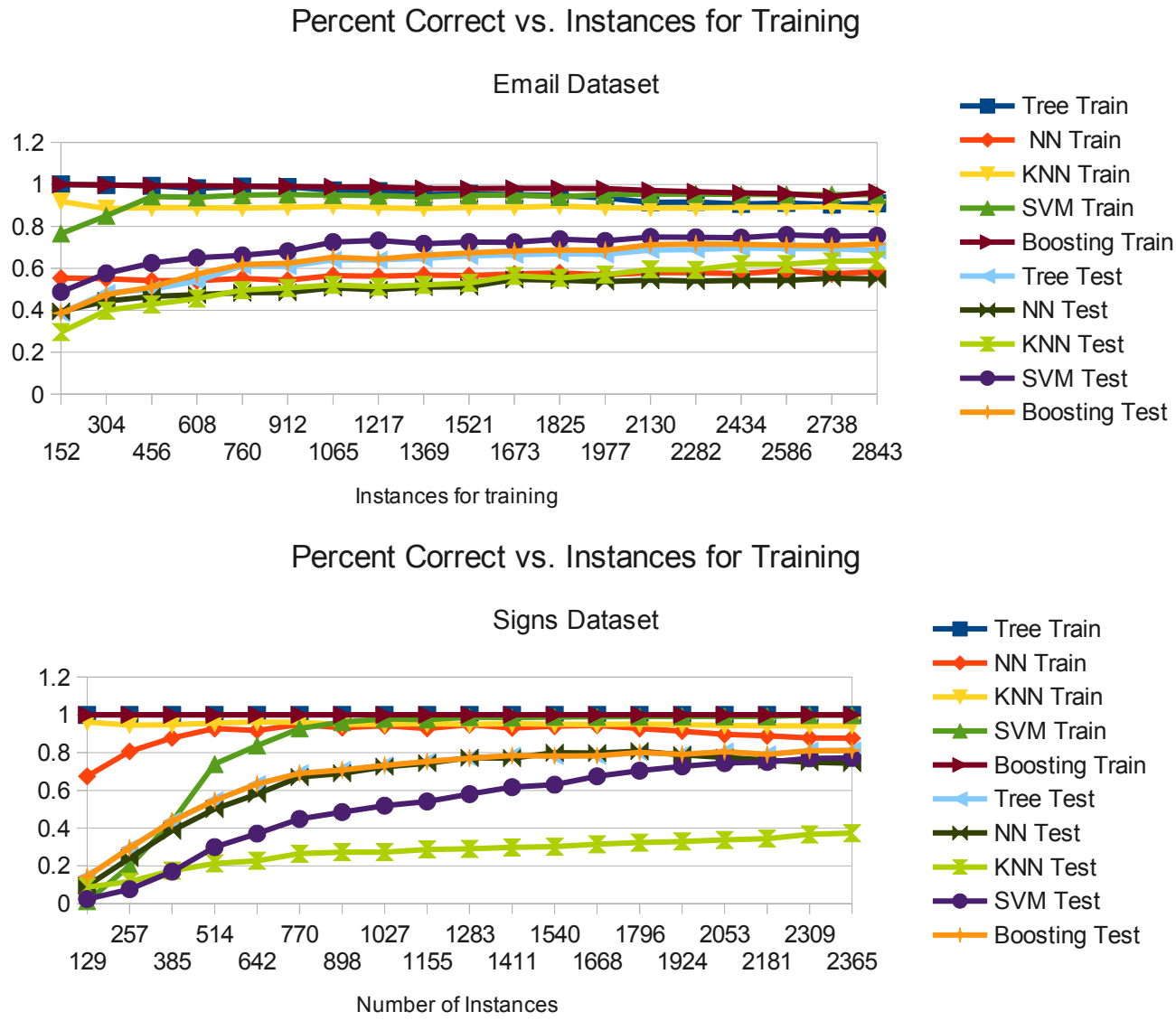


Figure 1. Learning curves for both datasets

Before addressing the behavior of each algorithm separately, it is useful to examine the performance of all the algorithms given varying numbers of instances for training and constant test sets. All algorithms were ran with the best parameters found in other experiments, though unlike the other experiments they were tested with a constant test set rather than cross validation and so performed worse overall.

As seen above, the signs dataset has a better maximum success rate but has a larger variance of performance, with kNN performing significantly worse than the other algorithms. That kNN performs so poorly indicates that it is not enough to merely find examples with similar feature values for the signs dataset, but that some more detailed measure of how much the motions are similar is required. Conversely, for the emails dataset this is not a problem and just finding examples with similar feature values performs relatively well. This makes some intuitive sense, since to classify the continuous motion data of the signs dataset the classifier needs to consider the change between frames but

disregard any offsets for the whole motion, unlike the emails dataset where there is no meaningful relationship to be extracted between features.

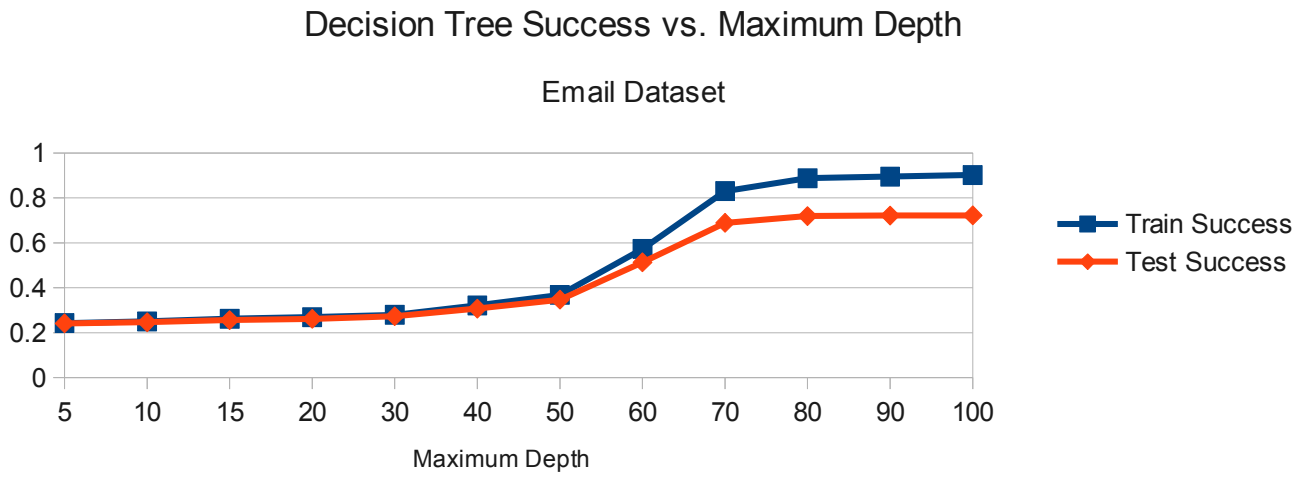
Another interesting thing to note is that both datasets rarely have a problem with overfitting, and as will be shown later this is not due to pruning and remains the same for independent variables different form number of instances. This makes sense for both datasets, but for different reasons. For the emails dataset, it is clear that it actually underfits the data in most cases as it usually does not achieve flawless classification for the training data. Therefore, even given unlimited learning it does not overfit but rather only gets better. A different formulation of the classification problem that avoids underfitting is necessary to get better performance and possibly have overfitting be a problem. The signs dataset does not at all underfit except with Neural Nets, but since the signs have mostly constant motions that classification problem is really to account for noise or minor offsets in those motions. Given 27 instances of the motion it is difficult to "learn the error" instead of the core motion, and so overfitting would require an extreme amount of learning that was only attempted with Neural Nets.

Another aspect that can be seen in the general performance of the algorithms is the difference between which algorithms perform best and worst on them, with SVM performing significantly better than other algorithms for the email dataset but being among the lowest performing on the signs dataset. SVM was run with the Radial Basis Function kernel, which is an infinitely dimensional kernel that learns to classify using the gaussian distance metric between feature values in instances. My intuition is that this is a more robust way of doing close to what kNN, and so once again it does not have impressive performance on the signs dataset but works better for the email dataset.

Decision Trees

Decisions trees was one of the best performing algorithms for the email dataset, and one of the worst performing for the sign dataset. It is important to note that all data except the learning curves was obtained with cross validation due to the lack of a provided test set, so the results are slightly different. One interesting results is that despite two forms of pruning being tested for both datasets, neither form of pruning improved performance whatsoever. The obvious reason for this is that neither dataset suffers from overfitting as discussed before, and so pruning was of no benefit.

The trees still undefit for the emails dataset, but they perform quite well among all the algorithms. This



makes sense, since given the features I created for the dataset the main way to classify an email is to determine who among many possibilities was the sender, from which domain it was sent, and which words it contains. This is entirely possible with decision trees, and furthermore few other methods seem possible given these features so other algorithms should perform no better.

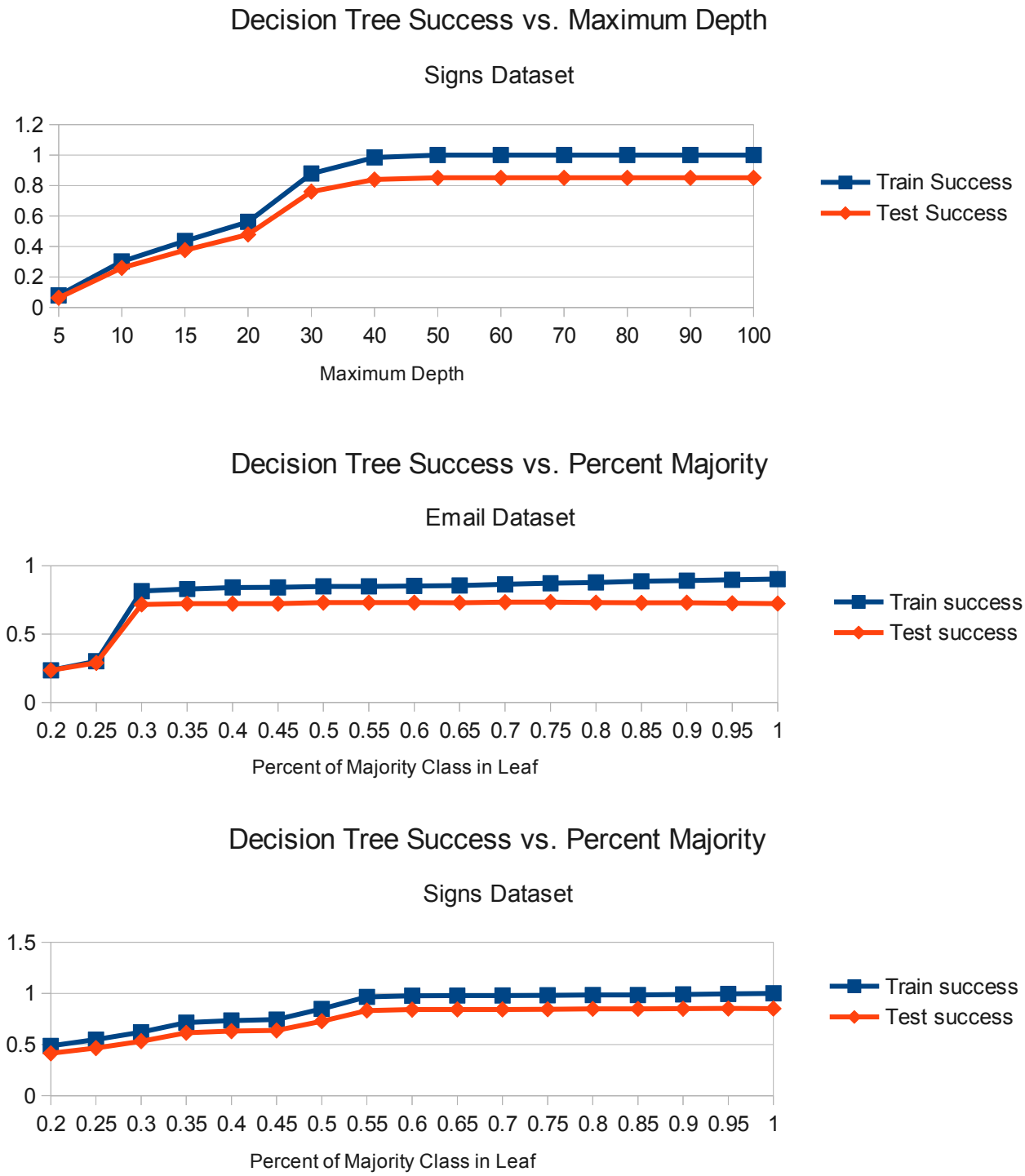


Figure 2. The results of two different pruning methods tested with both datasets

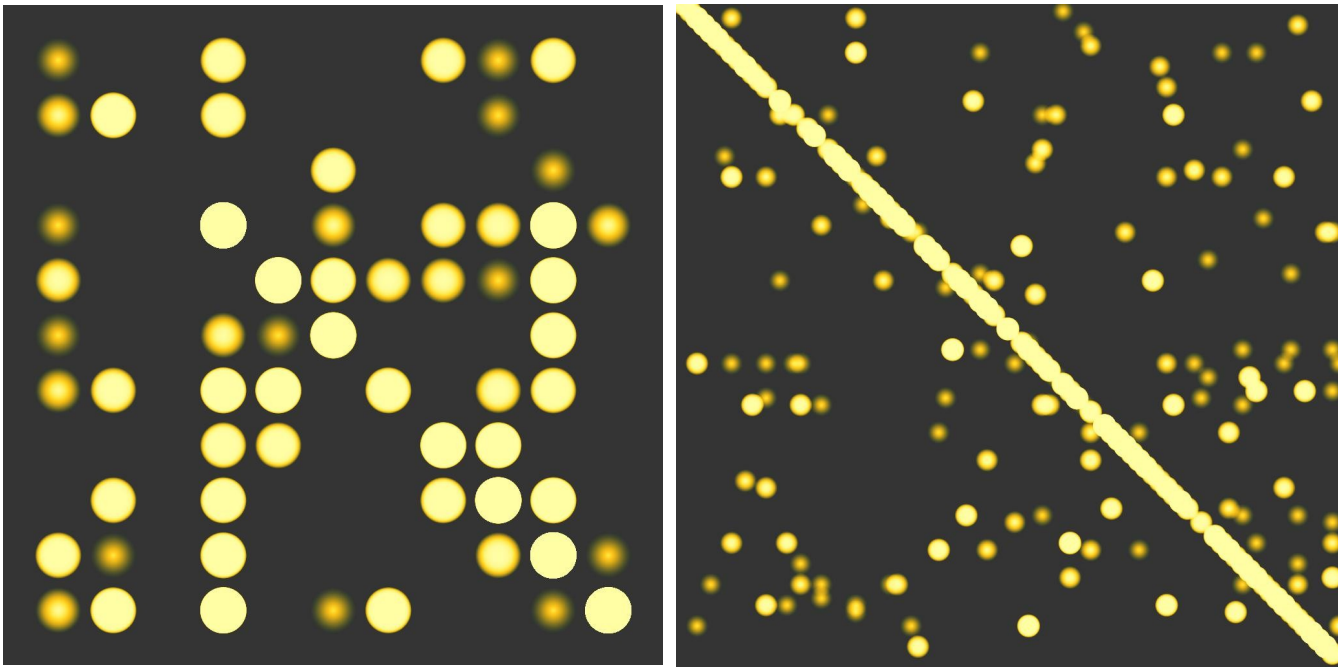


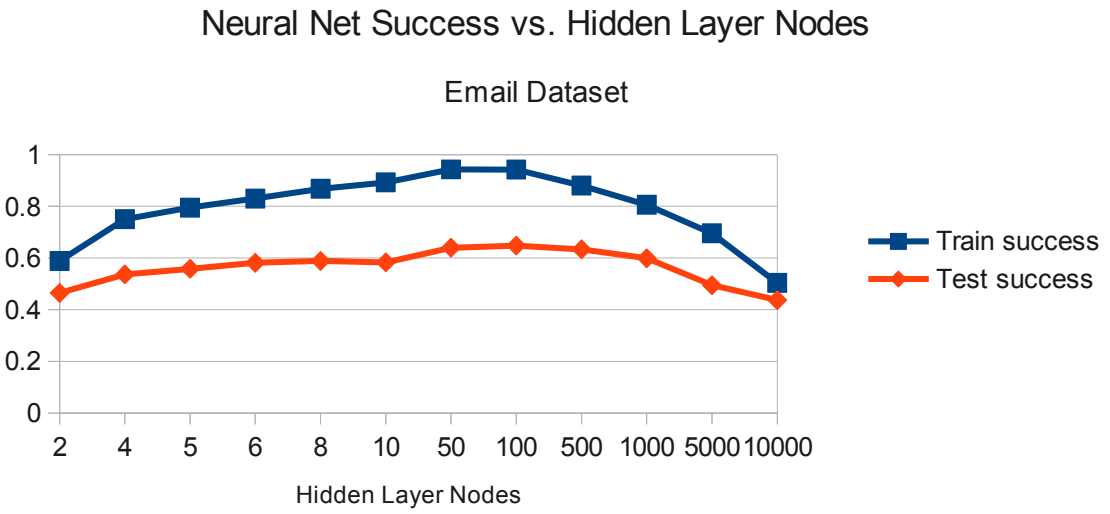
Figure 3. Heat maps representing the confusion matrices of the email data(left) and signs data(right). Higher opacity circles represent higher values, and correct classifications are on the diagonal.

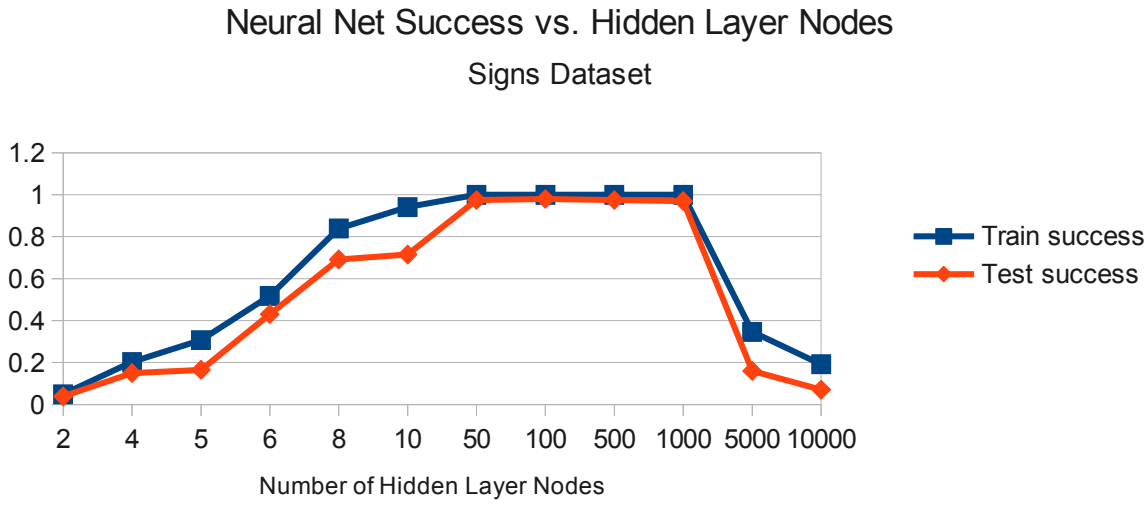
More information about the performance of the algorithms is available by looking at the resulting confusion matrices represented as heat maps figure 2. The heat map for the email dataset shows that most erroneous classifications occur with two classes, which are also the classes of most instances in the dataset. In contrast, the error is relatively spread out with the signs dataset. This unsurprisingly shows that if a dataset is heavy weights to have many instances having one or two classes instances are more likely to be classified with those classes with their actual classes mattering less than if the classes are uniformly spread out in the instances. Training times were below ten seconds for emails and close to a minute for signs, most likely since both datasets did not have a very large number of instances.

Neural Nets

With cross validation testing, Neural Nets with a single hidden layer was the best performing algorithm for the signs dataset and close to the worst for the email dataset. This is a noteworthy result, as it fit with my expectation that neural nets would offer an advantage to the numeric and sequential signs dataset but not the discrete and unordered email dataset. I believe that because there is no relation to capture between the features for the emails, the neural net can only attempt to replicate the logic learned by the decision tree and does so slightly worse. In contrast, since the features of the signs represent motions and are sequential neural nets perform better with them.

Overfitting was not observed, though interestingly with enough hidden layer nodes the performance started worsening for both the train and test data. I expect that this is because I did not increase the training epochs along with the number of hidden layer nodes, and so given the size of the net it was not trained as well in the same time. The time to train the biggest neural net was 34 minutes for the email dataset and an hour and nine minutes for the signs dataset, predictably showing the more features results in longer neural net training. Experiments were also performed with varying the number of epochs, but beyond an initial improvement in performance for the first three hundred epochs the performance stayed completely constant afterward, once again showing that there is no overfitting.





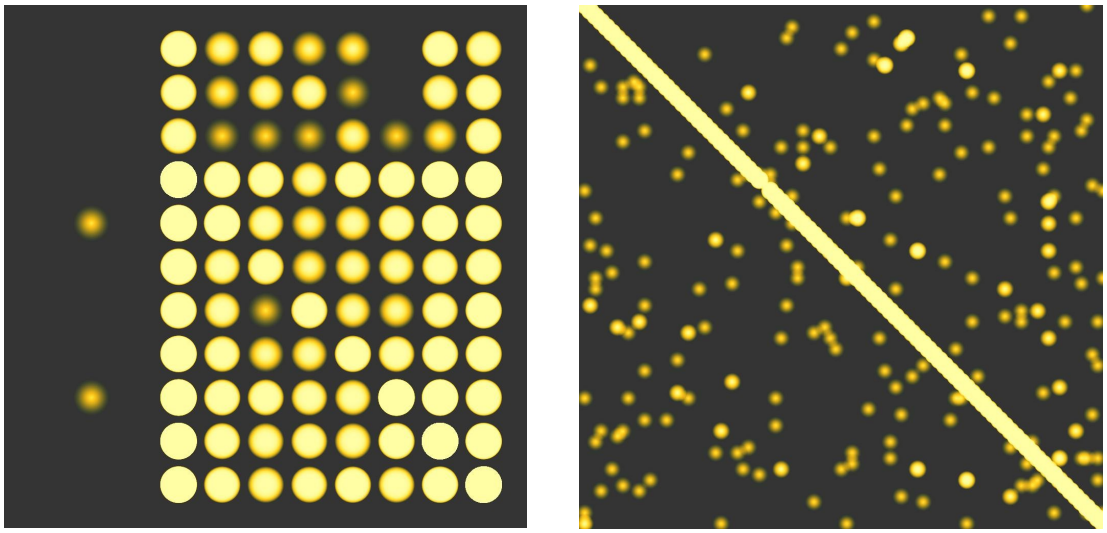


Figure 4. Data for Neural Nets, with the same heat maps as figure 2

A final detail to note is that the heatmap of the neural net for emails shows it to completely fail to classify 2 of the 11 classes. This indicates that the majority of instances having only 2 different classes affects it more than the decision trees, and is the reason for it performing worse. I did not expect a skewed class distribution to affect neural nets more than other algorithms, but that is what the results seem to suggest.

Support Vector Machines

Support vector machines were tested with two kernels, the Radial Basis Function kernel and Polynomial with a degree of 3. The gamma variable, which roughly acts to set how close the boundary should fit near the training examples, was altered to see how it behaves for these kernels. Overfitting was achieved for the signs dataset using a gamma several orders of magnitude larger than optimal, but otherwise as before the email dataset underfit and SVMs had good performance for signs that is inferior to neural nets. So, as expected it is possible to overfit for the signs dataset by learning the error but the data make it hard for that to happen. The other result is that RBF performs better than the polynomial kernel in both cases, which I expect is because it is a more general distance metric and not constrained by the shape of a third degree polynomial.

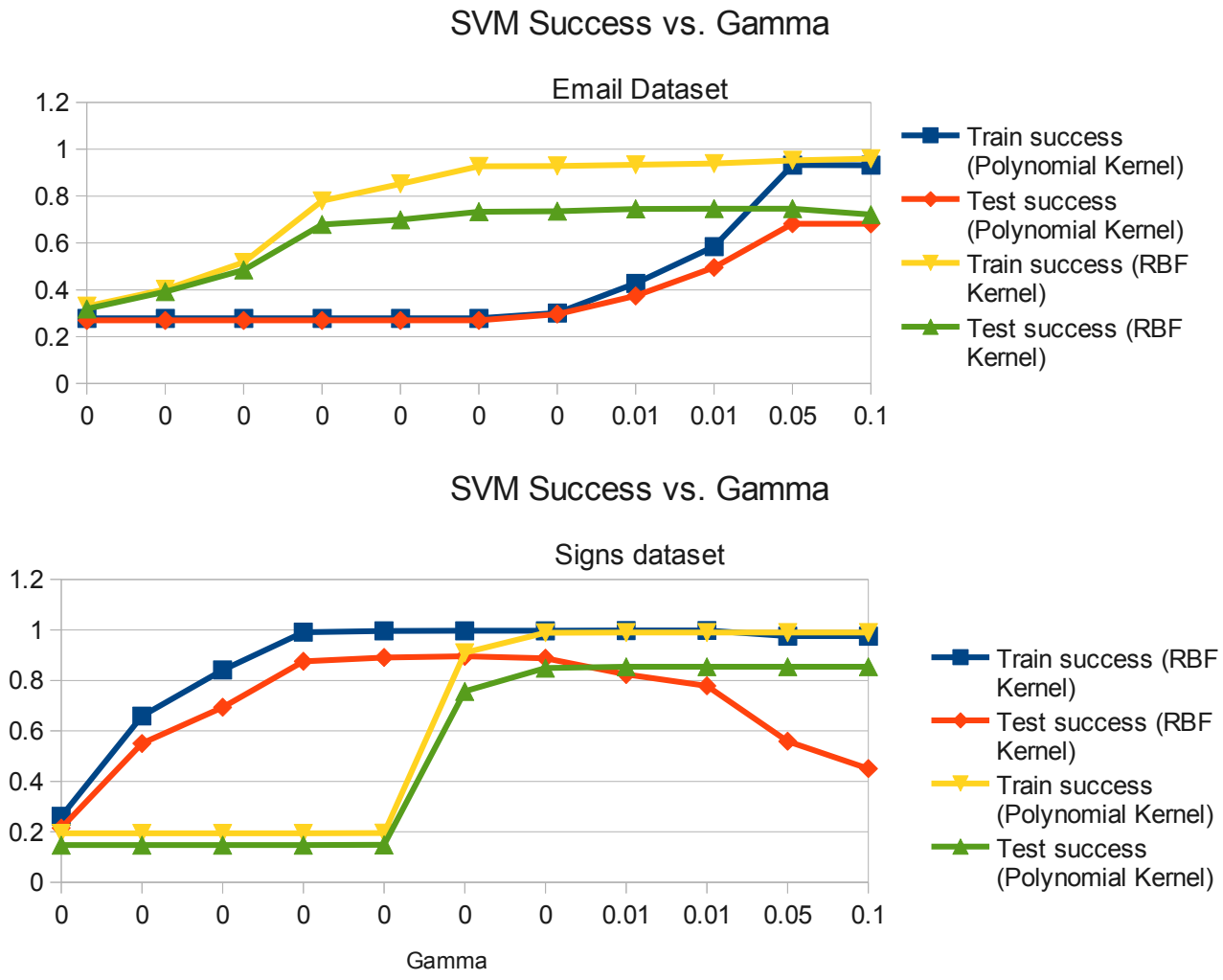


Figure 5. SVM results with different parameters. Heatmaps excluded due to no new information.

In terms of evaluating the results, I think they are consistent with the performance of other algorithms. An SVM can learn to represent the same information as a decision tree in the emails dataset, and so it achieves comparable but not better performance with values of gamma that roughly make it classify using all the features which is comparable to the decision tree having a large depth. For the signs dataset, I expect that the previously discussed advantage of neural nets does not translate with SVMs because while it can train to recognize the sequential motion it cannot handle offsets of it or other such variations.

Boosting

An unexpected result was obtained with boosting: altering the number of trained weak classifiers from 4 to 20 had absolutely no effect for both datasets. Furthermore, the success rate of boosting were almost exactly the same as those of the tree learners it was given, so trees with high pruning led to badly performing boosting classifiers and boosting with no pruning performed almost exactly the same as just training a decision tree would have. This result may be particular to the machine learning package I used, which was a python package called Orange. However, an explanation based on the nature of the data may be that with the numbers of instances being trained on no patterns of what a complicated or difficult classification emerges, so there is no benefit to weighing the individual instances. The result is that the success rates were only very slightly higher for unpruned trees. This is in agreement to an example result in Orange documentation for boosting learning, which shows that a boosted tree learner performed only 0.006 better than an unboosted tree learner. It is likely that boosting would have had more benefit if my datasets were more susceptible to overfitting.

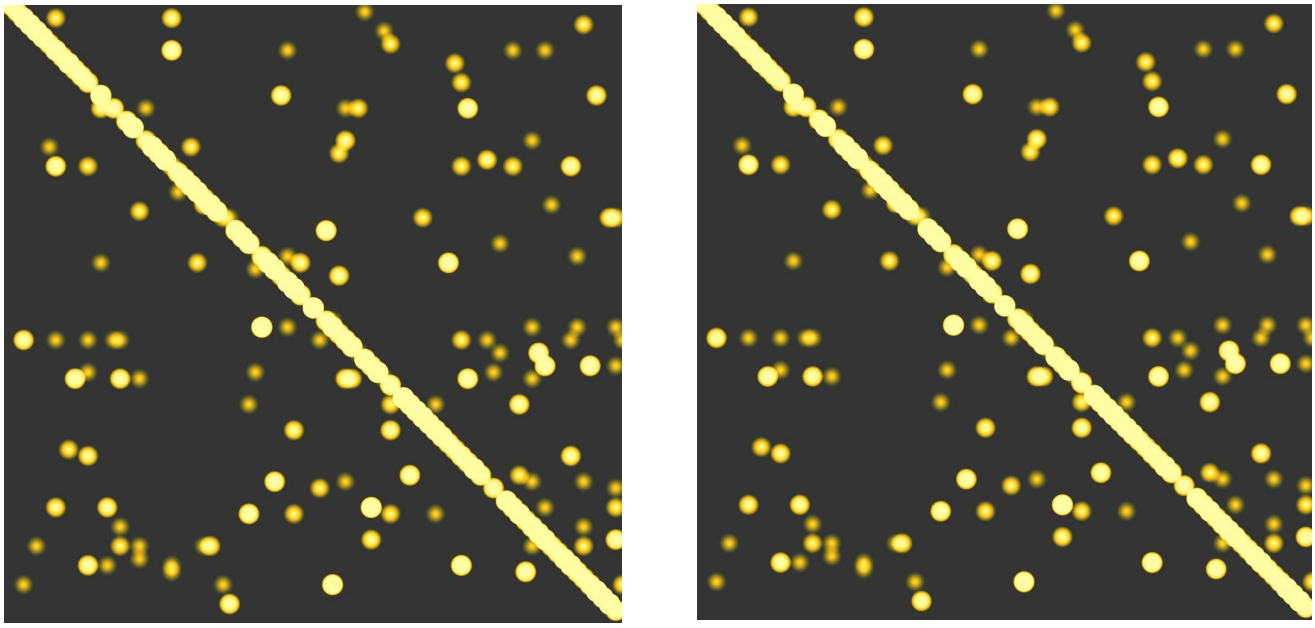
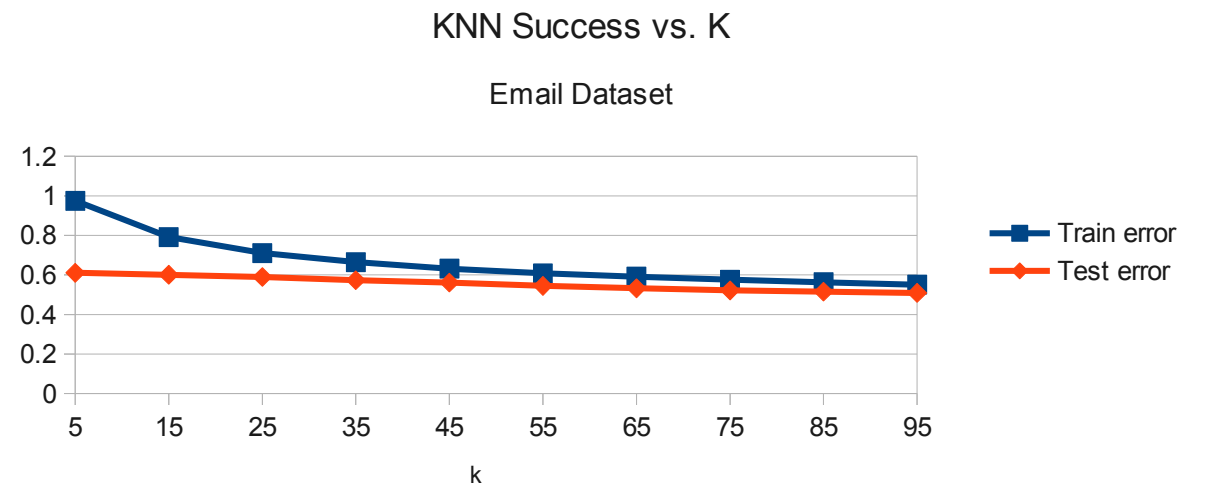
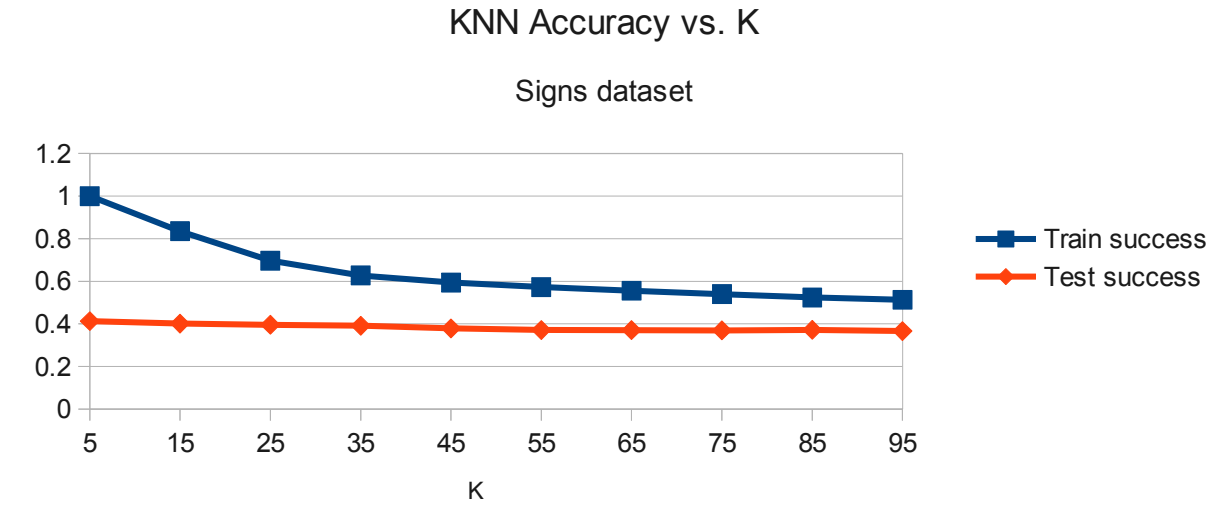


Figure 6. Tree confusion heat map(left) compared to boosting confusion heat map(right). Performance graphs not provided since they are constant in value and described in the text.

k-Nearest Neighbors

This was the worst performing algorithm for both datasets, though its performance was still only 0.2 worse than the best performing method and a whole 0.6 worse for the signs dataset. This agrees with my previous thought the the sequential and related nature of the features for signs make algorithms that don't take that into account perform badly. It is also consistent with email results, since Orange kNNs





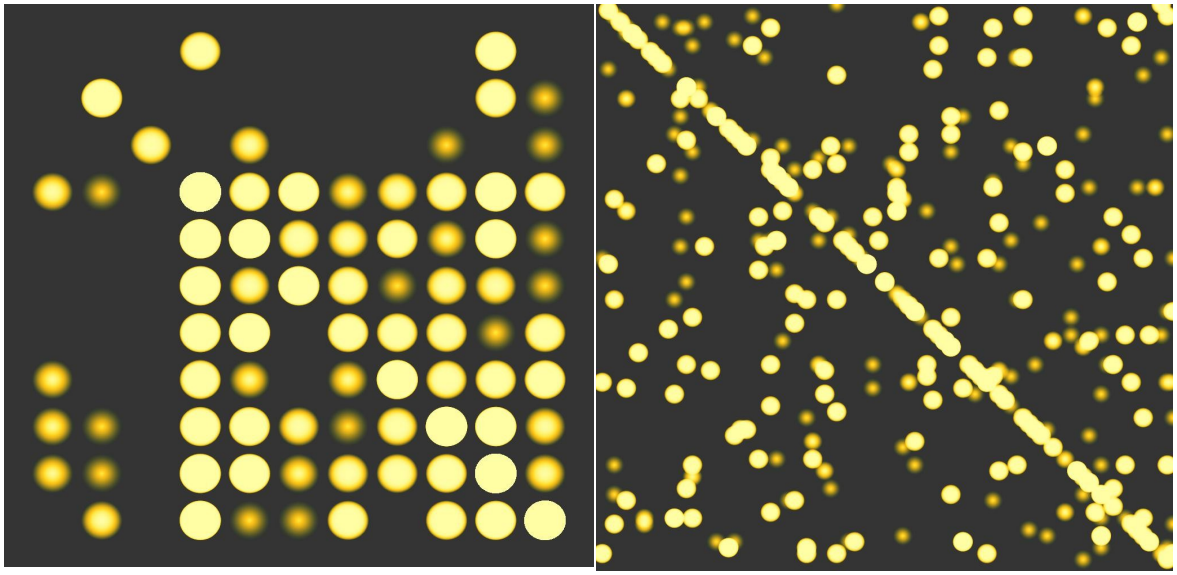


Figure 7. Data for kNNs for both datasets

only measure similarity of all features and are unable to capture the logic of certain features having precise values or being more important for a given class (the sender and domain features are significantly more important than the rest in other algorithms).

However, kNN still performs relatively well with emails, as do all the other algorithms. I believe this is because most of the features of the email dataset have values that do not relate to each other on any level beyond the statistical correlations of their appearance, and so a classifier could be made using simple statistical techniques rather than sophisticated machine learning. Although the logic of decision tree provides some benefit, kNNs performing so well implies that this is in fact the nature of the data since instances with similar feature values do tend to have the same class. This is as simple a machine learning result as one can get, so none of the algorithms fail to do at least as well as kNN and do not manage to do much better. In contrast, the signs dataset has features that are related by the fact that they are instances of a motion in 3D space, and so features are related to each other by much more than statistical correlation. More sophisticated machine learning techniques such as Neural Nets can learn this higher level relation, whereas kNNs do not.

Another interesting aspect of the result for kNNs is that the performance only decreases as k increases, even if it is small relative to the number of instances. This makes sense for the training set, since with a lesser k the same instance within the training set has a higher weight and leads to perfect classification, which is worsened with higher values of k. However, it is harder to explain why the testing sets are barely affected by the k value. The best explanation I have for this is that there are only a few instances that are very close to any given instance in the test set, and so they primarily determine the classification of that test instance and as k increases only instances that are farther are found and affect the result less.

Conclusion

As mentioned many times throughout the report, the most interesting result I obtained had to do with the effects of a skewed class distribution, of features that are either not related or strongly related to each other, and possible reasons why the two datasets I chose do not overfit easily. My ability to train a fairly well performing classifier for the email dataset also makes me consider applying the idea to a browser application that suggests how to classify new emails, which is a strong example of the sort of problem machine learning is good for.