Pedro Sanchez
COEN 140

Final Project Report

Problem

Our species' is the most successful or most destructive on this planet, depending on what perspective you take. Our production and intellectual capability is second to none, but these achievements are a result of the exploitation of our world's natural resources. I am Peruvian, and as a result of my heritage I understand the greediness of humans when it comes to resources like Gold and Wood, as these can be sold as raw materials or forged into more complex products, also to be sold. It is this globally shared greed that currently accounts for the overall loss in forest cover in the USA from 2000-2005, according to Mongabay. (https://rainforests.mongabay.com/deforestation.html) These statistics are the result of analyzing the different types and amounts of forest cover types as time goes on. Gathering and maintaining such statistics is environmentally-motivated, but it can hopefully be solved (or optimized) with the application of machine-learning techniques.

Dataset

So, I picked the "Covertype" dataset off the UCI Machine Learning repository. It has 581,012 instances and 54 (+1) real, integer and binarized independent features such as elevation, slope, distance to hydrography, hillshade index, wilderness area and present soil types. It also includes 1-7 labels to be used as ground truths for model training using supervised learning.

I split my data similarly to Crain's work, using 15,120 instances for training and 565,892 for testing. My training set is composed of his training + cross-validation sets, and our training sets are equivalent. I did this in order to be able to effectively compare results with a control.

Motivation

This dataset could be used to train a model to analyze how our species' deforestation practices affect our world's air quality as time goes on, and although these are important concerns, they do not particularly motivate me. I was motivated to take on this dataset to see if I could reduce the man-hours involved in generating usefulness from this data by using unsupervised learning techniques for classification.

Besides the labels, all data found in this dataset was gathered (or could be gathered) autonomously. For this exact dataset, it is an amalgamation of both the US Forest Services' RIS data and from the 1999 US Geological Survey. These are routine data collection endeavors, but what is not is the production of the given labels. Producing these requires human activity proportional to N samples, since one must classify each data instance. For this dataset alone, that is 581K samples to be classified. Clearly, this is the bottleneck in being able to analyze this data meaningfully over time.

Existing Techniques/Related Work

People have worked on this dataset and the problem it presents in past. Notably, Blackard et. al in 1999 and Crain & Davis for their Stanford machine learning project in 2014. I hoped to replicate their work to at least the same accuracy, but also build on their conclusions and report my own contributions.

Blackard et al. used a variety of techniques for classification. Originally, they applied LDA, QDA and an Artificial Neural Network (ANN). However, they found that their QDA model "became unstable when qualitative or discrete variables were considered." They did not find similar problems with their LDA model, so their QDA model was removed in favor of further testing using LDA. With respect to the ANN, they chose their architecture and parameters via experimentation, and chose only those whose parameters produced the best model. Using these, they generated an additional 30 networks with the same parameters, but different initial weights. The goal was to find a variety of different local optimal points, and by doing so increase their chances of finding the global optimum. Finally, they also ran a variety of dimensionality reduction techniques, including joining all the binary variables, and getting rid of the wildfire variable altogether. They reported an accuracy of 58.38% for their LDA and 70.58% for their ANN.

Crain et al. used similar classification techniques: Multi-class Support Vector Machine, K-Means clustering, and PCA on both of these for dimensionality reduction. Interestingly, their PCA was able to keep 98.35% of the original variance across the first 3 principal components, which allowed for visualization of their data. Furthermore, with PCA applied, the training/testing error were nearly equivalent across all runs, although when applied specifically to the binary variables, overfitting was reduced at more significant cost to accuracy. Their best results using their SVM was 78.24% without any PCA applied. They do not provide their methodology for measuring error, but they claim that their accuracy was ~60% when k = ~65.

My Technique

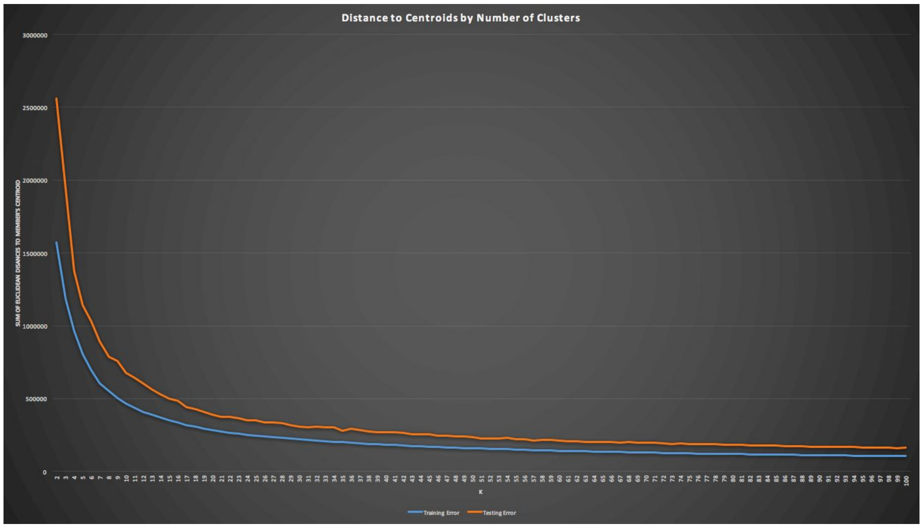
I decided to replicate a subset of the union of the techniques presented in both papers. I would use LDA, QDA and K-means clustering for classification, with the goal of minimizing the human effort required in data collection/interpretation. I considered using PCA for dimensionality reduction, but opted against it since their best results ultimately came from no PCA, which is to be expected. I used Sublime Text's built-in Python interpreter for my IDE, NumPy and the standard library for data parsing, and finally sk-learn's LDA, QDA, and K-Means libraries for the implementation of the machine learning models.

QDA quickly showed similar failure as it did in Blackard's study so I quickly dropped it after one run. LDA looked promising, despite the exclusion of the cross-validation phase and hyper-parameter.

When implementing K-Means, I quickly gained some key intuition. For my first attempt, I naively tried to set k = #num labels, expecting one label to map to one cluster. I did this by taking the cluster's label mapping to be the most frequent label present among its members. However, this was quickly shown not to be the case, since I had multiple clusters mapping to one label and no clusters mapping to other labels. Clearly, there needed to be more clusters to account for the high-dimensional shape these classifications took, as well as the noise/variance they had. This set me back, since I was hoping that a human could run k-means with k=7, take a look at the resulting clusters, and easily determine the label it represented.

However, it wasn't as bad of a problem as I thought. I realized that increasing the required human interaction from 7 to 50, or even 100, was nowhere near as bad as ~500K. So, I decided to run the same K-means algorithm and analyze the sum of euclidean distances as K increased to find an elbow point for K and thus the optimal amount of human interaction needed in this problem. Note, I always used k-means++'s "smart" intialization and kept the centroid that minimized overall member distance across 10 runs, since there are mutiple local solutions.

Experimental Results



..	Training (%)	Testing (%)
LDA	65.0198	58.1252
QDA	43.1018	8.0458

Conclusion

QDA is a bad model for this problem. Blackard's paper mentions that this was due to the inclusion of the discrete/qualitative variables, but my results also show heavy overfitting (bad generalization) when applied to the testing data.

LDA can provide cheap (in terms of setup cost, runtime, and computational resources) predictions with acceptable accuracy and generalization. Furthermore, in Crain's study, he shows that he is able to achieve a marginal .25% increase in testing accuracy over my model with the use of 10-fold cross validation. This is a classic trade-off between runtime and accuracy, and depending on the use case the extra accuracy may be worth it to you or not, since on one hand, LDA runs pretty well for the simplicity of the model, and the accuracy increase isn't very large. On the other hand, LDA runs the fastest across all tested models, which would remain the case even with the CV optimization.

K-Means was the most interesting one. I found the elbow point, somewhere roughly between 10 and 30, depending on how stringent your accuracy requirement is and how stingy you are about runtime increases. Note that each increment of k was a fairly significant increase in the model's runtime.

That being said, it needs significantly less human interaction, since the labels found in the dataset were withheld from it, and according to Crain's work, at k = ~65, the accuracy for classification was ~60% which is better than LDA's. With respect to my original goal of reducing the human interaction required in this problem, this would go a long way, since a human would only need to classify 65 clusters instead of >500K samples. However, the accuracies presented in both paper's as well as my results remain between 58-78%, nowhere near the desired 90%+ accuracies reported in other machine learning problems. So, both the classification techniques (and the unimplemented ANN technique) have accuracy issues that hinder them with respect to their accuracy, but provide huge improvements with respect to the amount of human activity required for the problem.

Future Work

Moving forward, it would be interesting to apply PCA for data visualization and for the runtime improvements, but I am not as interested since the main problem is the accuracy of the models while still reducing the human activity involved. PCA would likely not help with this task.

To improve the accuracy, I can think of two approaches. First, a modern implementation of a neural network, with a focus on high accuracy despite costly runtime. Our class hasn't gone into the details of such a network, so I cannot speak to the details. Alternatively, one could increase k to be sufficiently large to increase the accuracy from an acceptable level to a good level. As long as k remains small relative to 500K instances, I would consider my problem solved. However, due to the nature of k-means and diminishing returns, it is unlikely this will happen for a small k relative to 500K.

Finally, the most prominent task at hand in my k-means clustering results is solving the problem of finding the mapping between clusters and labels. While this will probably require some labeled data for the training set, developing a technique to find this mapping is crucial in order to reduce the human element.

References

1. Blackard, Jock A., Dean, Denis J. "Comparative accuracies of artifical neural networks and discriminant analysis in predicting forest cover types from cartogrpahic variables". Computers and Electronics in Agriculture (1999).
2. Crain, Kevin, and Graham Davis. "Classifying Forest Type Using Cartographic Features." Stanford, Stanford, Dec. 2014, cs229.stanford.edu/proj2014/Kevin%20Crain,%20Graham%20Davis,%20Classifying%20Forest%20Cover%20Type%20using%20Cartographic%20Features.pdf.