A Comparison of Generative and Discriminative Classifiers

Mohamed Kleit
Department of Computer Science
University of Montreal
Montreal QC H3C 3J7
mohamed.kleit@umontreal.ca

Annabelle Harvey
Department of Computer Science
University of Montreal
Montreal QC H3C 3J7
annabelle.harvey@umontreal.ca

Jimmy Ho
Department of Computer Science
University of Montreal
Montreal QC H3C 3J7
jimmy.ho.1@umontreal.ca

Abstract

We compare discriminative and generative learning classifiers using random forest, support vector machine (SVM) and naive Bayes. We test the performance of the classifiers on three datasets of different sizes. As the training set size is increased, we show that discriminative classifiers have lower asymptotic error but generative classifiers approach their asymptotic error faster [1].

1 Introduction

Generative classifiers assume some functional form for the joint probability p(x,y) and estimate the joint likelihood p(x|y) and prior probability p(y) directly from the training data. Using those parameters, generative classifiers make predictions by using Bayes rule to calculate p(y|x). Discriminative classifiers assume some functional form for the posterior p(y|x) and estimate it directly from the training data. Another way would be to learn a direct map from inputs x to the class labels. In summary, generative classifiers model the distribution of individual classes when discriminative classifiers learn the (hard or soft) boundary between two classes. In this paper, we study empirically to what extent the hypotheses we have stated previously are true: (a) The generative classifier has a higher asymptotic error with an increasing number of training examples than the discriminative classifiers (b) The generative classifier approaches its asymptotic error faster than the discriminative classifiers.

2 Datasets

Pump it Up: Data Mining the Water Table The data [2] relates to potentially faulty waterpoints in Tanzania and consists of 59,401 labeled training instances with 39 attributes describing them. This is a classification dataset with three labels: functional, functional needs repair and non functional.

Census Income The data [3] concerns a binary classification problem of prediction whether or not a person makes over 50k/ year based on information spread in 14 different attributes taken from the census. Since imbalanced, we artificially reduced the size of the dataset to 23,374 training examples with an equal amount of examples for each label.

Breast Cancer Wisconsin The data [4] concerns a binary classification problem of predicting the type of cancer, malignant or benign. The dataset consists of 569 training examples each described by 10 different attributes.

3 Pre-processing

Data Cleaning Fill missing data with an appropriate value based on the feature (mean, mode etc.) or drop features that are uninformative or with too many missing values that we can't fill with a reasonable value.

Feature Selection In some datasets we have features that are either uninformative or need to be dropped to reduce the dimension. For numerical features we can use a correlation for comparison, for nominal features we can use cramer's v association. For features that are highly correlated/associated or duplicates, we iteratively choose one and verify the impact on the quality of prediction keeping the feature that best impacts prediction.

Ordinal vs. Nominal Features Ordinal features are naturally ranked, whereas nominal features are not. Numerical features are naturally ordinal. Some categorical features are ranked as well: ie in the wells dataset we have status of the well 'non-functional', 'functional needs repair', and 'functional'.

Onehot Encoding Nominal features need to be onehot encoded, otherwise if they are encoded with integers the learning algorithm will assume they are ordinally ranked. Depending on the feature, this can make a large difference in the quality of predictions.

Ordinal Encoding Ordinal features get mapped in corresponding order to integers.

Scaling SVM is sensitive to the relative scale of the features, so if feature 1 takes values in range (0,100) and feature 2 takes values in range (0,1) the algorithm will more heavily weight feature 1. In order to avoid this bias on the predictions, we need to scale all the features of our data to be within a standard range. We chose to do this using sklearn MinMaxScaler() and range (0,1).

4 Analysis

4.1 Classifiers

Naive Bayes Generative probabilistic classifier that makes classifications using the maximum a posteriori decision rule in a Bayesian setting. It first learns training examples in an a priori probability when given unseen examples. It is called "Naive" because it assumes the features are independent.

Support Vector Machine Discriminative classifier defined by a hyper-plane. Useful model for learning linear and non-linear predictors in high dimensional feature spaces which raise sample and computational complexity challenges.

Random Forest Discriminative classifier that consists of a collection of decision trees where each tree is constructed by applying a different algorithm on randomly selected data samples. The prediction of the random forest is a majority vote over the predictions of the individual trees.

4.2 Experiments

We plot the learning curves for our three classifiers for different sizes of the training set and observe the generalization error trend for each.

4.2.1 Pump it Up: Data Mining the Water Table

In Figure 1, we see the generalization error graphs for random forest and Support Vector Machine follow the same trend and considerably decrease - going from 28% error rate to 23% for SVM and

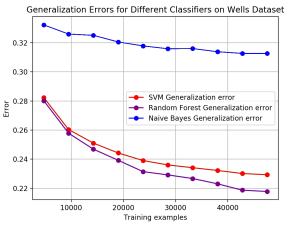

Figure 1: Generalization errors for Wells dataset

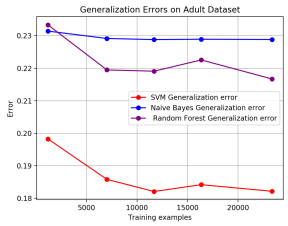

Figure 2: Generalization errors for Census Income dataset

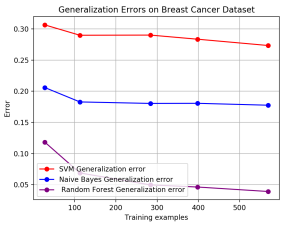

Figure 3: Generalization errors for Breast Cancer dataset

from 28% to less than 22% for random forest - as the number of training examples increases. Adding more training data does seem to reduce variance as the generalization error keeps decreasing.In the case of naive Bayes, the generalization error remains considerably higher than the discriminative classifiers regardless of the number of training examples. However, naive Bayes seems to approach its higher asymptotic error much faster. Indeed, the generalization error starts around 33% to reach an error rate slightly lower than 32% for a training set size of 25,000 and more. Adding more training data does not seem to reduce the generalization error anymore.

4.2.2 Census Income

We notice in Figure 2 that both discriminative classifiers have a (much, in the case of SVM) lower asymptotic error than our generative classifier. SVM shows an error rate around 20% for a dataset of less than 200 examples. It the decreases as the number of examples increases to reach an error rate close to 18%. Random forest shows an error rate slightly higher than naive Bayes for a very small training set size. Naive Bayes however reaches its higher asymptotic error (23%) very quickly (around 7,000 examples) before stabilizing. On the other hand, random forest decreases considerably as the number of training examples increases (reached 21.5% error rate with 30,000 training examples).

4.2.3 Breast Cancer Wisconsin

The results on the breast cancer dataset, shown in Figure 3, give us a slightly different insight than the first two datasets, as it is a much smaller dataset than the previous two (569 training examples). We notice here that the naive Bayes shows a much lower asymptotic error (18%) than SVM (27%). Random forest, on the other hand shows the lowest asymptotic error (less than 5%) amongst all three classifiers. These results are very interesting as they show a different behaviour for our classifiers for very small datasets.

4.3 Discussion

The results of the previous section show that even though random forest and SVM classifiers tend to output much lower asymptotic errors than naive Bayes, the latter, which is generative, tends to converge more quickly to its asymptotic error. Thus, as the number of training examples increases, discriminative classifiers tend to improve their accuracy considerably when the generative classifier performance does not seem to respond to that increase in training examples.
This can be explained by the nature of our generative classifier. Indeed, naive Bayes ignores the interaction between features as it assumes conditional independence between them. Its hypothesis function is thus relatively simple and can't properly model the interactions between the features thus leading to a higher asymptotic error - synonym of high bias - but lower variance.
SVM and random forest have a much richer hypothesis function than naive Bayes and do not assume independence between the features. Their asymptotic error is thus lower, which is a sign of lower bias. We also notice that adding more training examples does improve the discriminative classifiers performance.
The third dataset we have exploited, on breast cancer, has 569 training examples. This is much smaller than the other two datasets: 59,401 for Wells and 23,374 for the Census Income dataset. We observe that naive Bayes performs much better than SVM (but poorer than random forest) on the breast cancer dataset. This suggests that generative classifiers can perform better than discriminative

classifiers on datasets of relatively small size. Discriminative classifiers, since more complex, tend to overfit the data when the number of training examples is too small and thus provide a much poorer generalization error. Generative classifiers such as naive Bayes don't overfit as much as they don't learn the spurious correlations that occur in the training data. Appendix A shows more detailed learning curves for each dataset as it shows the training errors and generalization errors on the same graph.
Finally, the results of the experiments shown in Figure 1, Figure 2 and Figure 3 confirm our initial assumptions that discriminative classifiers tend to have much lower asymptotic error but generative classifiers approach theirs more quickly. Moreover, we observed there can be two distinct regimes of performance as the number of training examples increases. Indeed, on very small datasets, generative classifiers can outperform discriminative classifiers as shown in Figure 3.

5 Results

To compare the classifiers, we decided to compare the most suitable metrics for each dataset.

For the broken wells dataset, there are three types of labels: functional, non-functional and functional but need repair. It makes more sense to focus not on the functional wells, but on those that are not functional or those that need repair. We base ourselves on a goal, which is to obtain, in the end, a maximum number of functional wells, by identifying those that are not. Thus, it is more important to predict a functional well as non-functional than to predict a non-functional well as being functional. It is then necessary to give more importance to false positives by giving them a certain weight, and this by calculating the precision of each classifier. The graph 5 shows the results obtained. We notice that the best precisions and accuracies are given by a discriminative classifiers (random forest, or linear SVM). This is due to the fact that here, the number of different categories is small (3). Discriminative classifiers give better accuracies for a large number of classes.

For the cancer dataset, it is more important here to distinguish false negatives. Indeed, it is important to have a good prediction on cancerous subjects noted as non-cancerous. Thus, the recall has been computed.

Table 1: Wells results

Train Set	Naive Bayes	Random Forest	SVM
80%	0.70	0.80	0.77
50%	0.69	0.79	0.76
20%	0.68	0.77	0.74
Precision	0.69	0.78	0.75
Recall	0.69	0.79	0.76

Table 2: Census results

Train Set	Naive Bayes	Random Forest	SVM
80%	0.77	0.86	0.86
50%	0.76	0.85	0.86
20%	0.76	0.84	0.86
Precision	0.81	0.84	0.85
Recall	0.76	0.85	0.85

Table 3: Breast cancer results

Train Set	Naive Bayes	Random Forest	SVM
80%	0.87	0.97	0.70
50%	0.84	0.95	0.74
20%	0.82	0.92	0.66
Precision	0.84	0.95	0.79
Recall	0.84	0.94	0.70

Acknowledgments

Each member contributed equally to the project. Annabelle worked on the pre-processing and feature selection of the datasets. Mohamed conducted experiments, especially by plotting the learning curves for each classifier on every dataset and thus discussing the results in order to evaluate them against the initial hypotheses stated in the paper. Finally, Jimmy evaluated the performance of the classifiers using different metrics in order to suggest which classifier would be best suited for each dataset.

6 References

[1] Ng, A., Jordan, M.: On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In: NIPS (2002)

[2] DrivenData. (n.d.). Pump it Up: Data Mining the Water Table. Retrieved November 13, 2019, from https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/

[3] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [https://archive.ics.uci.edu/ml/datasets/Adult]. Irvine, CA: University of California, School of Information and Computer Science.

[4] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29]. Irvine, CA: University of California, School of Information and Computer Science.

7 Appendix A

We show here the detailed learning curves for each classifier on each dataset.

7.1 Pump it Up: Data Mining the Water Table

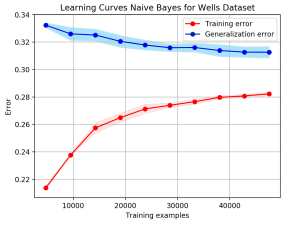

Figure 4: Learning curves for naive Bayes

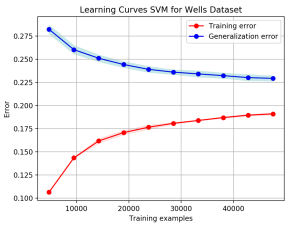

Figure 5: Learning curves for SVM

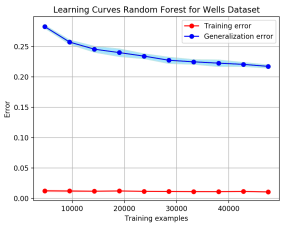

Figure 6: Learning curves for random forest

7.2 Census Income

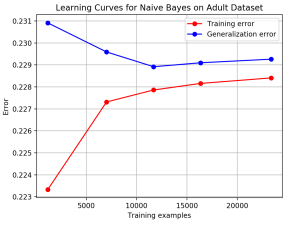

Figure 7: Learning curves for naive Bayes

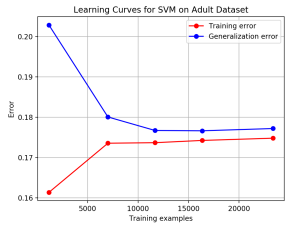

Figure 8: Learning curves for SVM

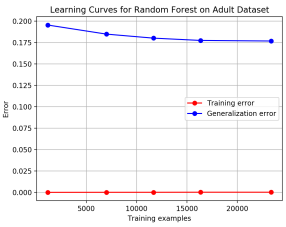

Figure 9: Learning curves for random forest

7.3 Breast Cancer Wisconsin

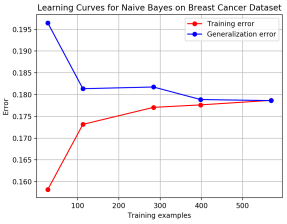

Figure 10: Learning curves for naive Bayes

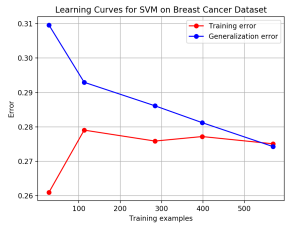

Figure 11: Learning curves for SVM

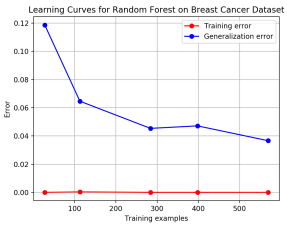

Figure 12: Learning curves for random forest