Jason DiMasi
CSCI E-87 Graduate Research Project Paper

Literature Search

The aim of this project is to investigate approaches to estimating Q_{max}, the maximum possible quality of a predictive model on target value(s) Y given a dataset X. In other words, this means we want to investigate the minimum possible error of said predictive model. It turns out that the theoretical minimum error rate for a classification model is known as the Bayes error rate (BER). From previous work on estimating through nearest neighbor methods, BER has well-established lower and upper bounds as follows: $\varepsilon \leq R_{knn} \leq \varepsilon(2\text{-}m\varepsilon/(m\text{-}1))$ where ε is BER, R_{knn} is the kNN error rate, and m is the number of classes in the dataset [1]. However, R_{knn} is strongly influenced by the number of neighbors parameter k used for the kNN algorithm, so choosing an suboptimal value for k will drastically change the values of the BER bounds. Chen et al. (2025) sought to find a consistent and unbiased BER estimator, and did so in a 2023 study called BN-BER. However, in this new study they proposed a new BER estimator called EF-BER that uses hierarchical k-means clustering and kNN classification. This new estimator ends up being slightly more computationally efficient than BN-BER. The details are provided in the article. They then demonstrated the effectiveness of EF-BER compared to other existing methods by computing the error rate after training and predicting on 13 benchmark datasets and 6 synthetic datasets. EF-BER was able to achieve lower average error rates than the existing methods on 11 of the benchmark datasets and all 6 of the synthetic datasets. They found that the limitations of this method were reduced effectiveness on smaller datasets with high dimensions, as kNN in general suffers from the curse of dimensionality. Sekeh et al. (2020) derives a new set of bounds on BER using a highly mathematical derivation. They then use these formulas to estimate the upper and lower bounds for the MNIST and CIFAR datasets, both of which are commonly used as benchmarks in high level image classification. They specifically pass the datasets through dimensionality reduction via principal component analysis and 1-layer autoencoding, and compare the bounds to error rates generated by kNN, linear SVM, and random forest classification. They found that after just 20 principal components or latent dimensions in their dimensionality reduction method, their upper and lower bounds for BER were minimized. Additionally, PCA generally achieved a smaller lower bond for BER than autoencoding. They believe that the BER bounds for autoencoding could be reduced by either adding more layers to the autoencoding or feeding the dataset through a convolutional neural network. Wheat et al. (2025) provide a more approachable paper, focusing on the classification methods rather than the BER bounds. They performed Monte Carlo simulations with a number of synthetic datasets using kNN, Generalized Henze-Penrose divergence, and Kernel Density Estimation which are all BER estimation techniques. In their simulations, kNN proved to be the most accurate predictor of BER. Even at higher dimensionalities, they found that of these predictors, the upper bound and the midpoint of the lower and upper bounds predicted by kNN tended to be the most accurate estimations of BER. Additionally, they found that estimators tend to have less accurate

BER bounds when there is high variance and bias. They recommend for duplicate research to have thousands of simulations per round in order to accurately estimate the BER bounds.

Reference List

[1] Chen, Q., Cao, F., Xing, Y., & Liang, J. (2025). An efficient Bayes error rate estimation method. *Machine Learning*, *114*(6), 134. https://doi.org/10.1007/s10994-025-06761-w

[2] Sekeh, S. Y., Oselio, B., & Hero, A. O. (2020). Learning to Bound the Multi-Class Bayes Error. *IEEE Transactions on Signal Processing*, *68*, 3793–3807. https://doi.org/10.1109/TSP.2020.2994807

[3] Wheat, L., Mohrenschildt, M. V., & Habibi, S. (2025). Testing Bayes Error Rate Estimators in Difficult Situations Using Monte Carlo Simulations. *IEEE Access*, *13*, 165810–165829. https://doi.org/10.1109/ACCESS.2025.3609630

Personal Ideas

My goal is to perform a basic level Monte Carlo simulation on four well-known, standard classification datasets to see how effective different classification algorithms are on a variety of dataset sizes. The five methods to be compared are kNN, random forest, SVC, Gaussian Naive Bayes, and AdaBoost with decision tree base. The first dataset is “iris”, a very small dataset with only 150 samples, 4 parameters, and 3 species of irises based on petal and sepal dimensions. The second dataset is “digits”, a significantly larger dataset with 1797 samples, 64 parameters, and 10 classes representing the 10 digits. The parameters represent the color value of a single pixel in an 8x8 image. The third dataset is “breast cancer Wisconsin”, a relatively medium-sized dataset with 569 samples, 30 parameters, and 2 classes representing either malignant or benign cancer growths based on various metrics. The final dataset is “wine”, another small dataset with 178 samples, 13 features, and 3 classes representing types of wine grown in Italy based on chemical composition. All four datasets are built into the scikit-learn Python library. While it won’t result in accurate representation of the BER estimates of the various algorithms, I will reduce the number of simulations per dataset per algorithm to 10 to save runtime. Hopefully, this results at least in a rough representation of how these methods perform in estimating BER. My code is provided below. As shown, I begin by tuning the parameters of the various methods via GridSearch cross-validation. Then, I perform the Monte Carlo simulations on each dataset using each model. The primary output gives the maximum individual simulation accuracy score and the mean accuracy score of all simulations for each model, separated by dataset. I built this code in Jupyter Notebook, so I apologize if the code converted to .py does not run properly. This code still produces interesting results. All algorithms seem to be very effective for the “iris” dataset. This makes sense because it is a small dataset with only 4 features. Compare this to the “wine” dataset which has 13 features, and we can see that kNN and SVC perform significantly worse. To

overcome the curse of dimensionality, we need to either use different methods or increase our sample size. This can be verified by the “breast cancer Wisconsin” and “wine” datasets. kNN performs much better on these datasets because of relatively large sample sizes, despite the large numbers of features. Random forest appears to be the most consistently good classifier across all datasets. With all that said, to more accurately obtain the accuracy and BER predictions, the number of simulations needs to be massively increased, especially on these datasets that are much smaller than those used in the formal research papers.

Conclusion

The idea of the theoretical maximum prediction quality Q_{max} is directly related to the Bayes error rate, the theoretical minimum possible test error. BER has historically been estimated using kNN, but new estimation methods are frequently being formulated. The challenge of these estimators is being consistent and unbiased. kNN is still frequently used as the most accurate BER estimator. The general effectiveness of kNN classifiers comes when the curse of dimensionality is overcome by large sample sizes. When the curse of dimensionality is relevant, low BER estimates can still be obtained using other standard classifiers such as random forest that are resistant to overfitting. It can be implied that methods like random forest can also be superior to kNN when non-linear data is present. A basic representation of these concepts is provided via the code.