Discriminative vs Generative Classification:

A study on the performance of Logistic Regression and Naive Bayes on different types of datasets

Saurabh Katkar and Deric Pinto

Department of Computer Science

Illinois Institute of Technology

Chicago, IL 60616

{skatkar, dpinto}@hawk.iit.edu

Abstract

Comparison of generative and discriminative classifiers is an everlasting topic. As an important contribution to this topic, based on their theoretical and empirical comparisons between the Naïve Bayes classifier and linear logistic regression, Andrew Ng and Michael Jordan claimed that there exist two distinct regimes of performance between the generative and discriminative classifiers with regard to the training set size. In this paper we report on the empirical and simulation studies on different types of datasets to classify the data using Logistic Regression and Naïve Bayes to observe whether our studies corroborate to the aforementioned claim and the general criterion that although discriminative classifiers are almost always preferred, the generative classifier approaches its comparatively higher asymptotic error much faster than discriminative classifier.

Overview

Generative and discriminative learning are two of the major paradigms for solving prediction problems in machine learning, each offering important distinct advantages. These algorithm utilize a vastly different technique from each other in solving the classification problem and have their own advantages and drawbacks with respect to the other approach.

Generative Classifiers

Generative classifiers, such as Normal-based Discriminant Analysis and the Naive Bayes classifier, model the joint distribution P(x, y) of the measured features x and the class labels y factorized in the form P(x|y)P(y), and learn the model parameters through maximization of the likelihood given by P(x|y)P(y).

$$c_{map} = \arg\max_{c \in C}\left(P(c \mid d)\right) = \arg\max_{c \in C}\left(P(c)\prod_{1 \le k \le n_d} P(t_k \mid c)\right)$$

Fig1: *Formula for the Naïve Bayes Classifier*

In other words, a generative classifier tries to learn the model that generates the data behind the scenes by estimating the assumptions and distributions of the model. It then uses this to predict unseen data, because it assumes the model that was learned captures the real model.

Discriminative Classifiers

Discriminative classifiers, such as logistic regression, model the conditional distribution P(y|x) of the class labels given the features, and learn the model

parameters through maximizing the conditional likelihood based on P(y|x).

$$\theta_i = \frac{1}{1 + \exp\left[-\left(\beta_0 + \sum_{j=1}^{k} \beta_j x_{ij}\right)\right]}$$

Fig2: *Formula for the Logistic Classifier*

A discriminative classifier tries to model by just depending on the observed data. It makes fewer assumptions on the distributions but depends heavily on the quality of the data (For e.g. Is it representative? Is there a lot of data?).

For $j = 0, \ldots, d$
  $w_j \leftarrow$ rand(-0.01,0.01)
Repeat
  For $j = 0, \ldots, d$
    $\Delta w_j \leftarrow 0$
  For $t = 1, \ldots, N$
    $o \leftarrow 0$
    For $j = 0, \ldots, d$
      $o \leftarrow o + w_j x_j^t$
    $y \leftarrow \text{sigmoid}(o)$
    For $j = 0, \ldots, d$
      $\Delta w_j \leftarrow \Delta w_j + (r^t - y) x_j^t$
  For $j = 0, \ldots, d$
    $w_j \leftarrow w_j + \eta \Delta w_j$
Until convergence

Fig1: *Logistic discrimination algorithm implementing gradient descent for the single output case with two classes.*

Algorithm Analysis

Generative models allow you to make explicit claims about the process that underlies a dataset. For example, generative graphical models allow you to describe conditional dependencies between model parameters. If your model has a good fit to your data set, it strengthens your claim that your model accurately reflects the generative process that actually created the data that you are modeling.

Generative classifiers learn about the conditional probability indirectly, they can get the wrong assumptions of the data distribution. Quoting Vapnik from Statistical Learning Theory –

"One should solve the [classification] problem directly and never solve a more general problem as an intermediate step [such as modeling P(X|Y)]."

An important contribution to this topic is from Andrew Ng and Michael Jordan presenting some theoretical and empirical comparisons between linear logistic regression and the na¨ıve Bayes classifier. Their results suggested that, between the two classifiers, there were two distinct regimes of discriminant performance with respect to the training-set size. More precisely, they proposed that the discriminative classifier had lower asymptotic error rate while the generative classifier may approach its (higher) asymptotic error rate much faster.

In other words, the discriminative classifier performs better with larger training sets while the generative classifier does better with smaller training sets.

Objective

We compare the generative and discriminative classifiers by applying Logistic Regression and Naïve Bayes algorithm on different datasets and gathering inferences based on their performance with respect to time, precision, accuracy, error rate and various other factors on the data.

The goal of this project is to apply these algorithms on the several datasets, differentiating in size, features and the number of classes to identify the avenues of comparison with respect to performance as well as the empirical and theoretical advantages of each approach.

Setting Used for Comparison

The setting for the theoretical proof and empirical evidence includes a binary class label y, e.g, $\mathbf{y \in \{1, 2\}}$, a p-dimensional feature vector **x** and the assumption of conditional independence amongst **x|y**,

the features within a class. The Naive Bayes classifier, a generative classifier, assumes statistically independent features x within classes y and thus diagonal covariance matrices within classes. By contrast, linear logistic regression, a discriminative classifier, may not assume such conditional independence of the components of x. Both classifiers can be applied to discrete, continuous or mixed-valued features x.

In the case of discrete features x, each feature $\mathbf{x_i}$, where $\mathbf{i = 1,..,p}$, independently of other features within x, is assumed within a class to be a binomial variable such that its value $\mathbf{x_i \in \{0, 1\}}$. However, this may not guarantee the discriminant function $\mathbf{\lambda(\alpha) = \log\{p(y = 1|x)/p(y = 2|x)\}}$ of the Naive Bayes classifier, where α is a parameter vector, to be linear. As linear logistic regression uses a linear discriminant function, the Naive Bayes classifier may not be a partner of linear logistic regression as a generative-discriminative pair.

In the case of continuous features x, x|y is assumed to follow Gaussian distributions with equal covariance matrices across the two classes, i.e., $\mathbf{\Sigma_1 = \Sigma_2}$ and, in view of the conditional independence assumption, both covariance matrices are equal to a diagonal matrix. Thus under the assumption of a common diagonal covariance matrix for normally distributed data, the Naive Bayes method is equivalent to Linear Discriminant Analysis (LDA) and, under the assumption of unequal diagonal within-class covariance matrices, it is equivalent to Quadratic Discriminant Analysis(QDA).

Datasets Used

We are taking into consideration the following 6 datasets for assessing the performance of each method of which 3 are continuous datasets and 3 are discrete datasets , All the datasets are standard UCI Repository datasets as listed below –

Sonar – Continuous Dataset
This is the data set used by Gorman and Sejnowski in their study of the classification of sonar signals using a neural network. The task is to train a network to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock.

Ionosphere Dataset – Continuous Dataset
The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere.

Liver Disorders – Continuous Dataset
Classification of whether people do/do not have liver disorders by taking into account various factors such as alcohol consumption.

Wisconsin Breast Cancer dataset – Discrete Datasets
These are two of three domains provided by the Oncology Institute that has repeatedly appeared in the machine learning literature.

Voting Records – Discrete Dataset
This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the CQA.

Chess – Discrete Dataset
This dataset consists a collection of generated knight-pin Chess end-game samples. It consists of two classes, White-can-win ("won") and White-cannot-win ("nowin"). White is deemed to be unable to win if the Black pawn can safely advance.

Implementation of Algorithm
With the aim of reporting the experimental results and inferring on the data by observing the performance measures of the algorithm, the aforementioned datasets were used and the results were derived using the Logistic Regression and Naïve Bayes Algorithm.

For the continuous datasets, in the case of Logistic Regression, the features dealing with missing data were removed or converted to 0 and the y-values were scaled to [0,1]..
The dimensions of the datasets used are given the following table:

Dataset	Instances(m)	Features(N)
Sonar	208	61
Ionosphere	351	35
Liver	345	7
Breast Cancer	699	11

Voting	435	17
Chess	3196	37

Fig: *Dimensions of the datasets used*

We apply **gradient optimization** technique to find the local optima of the function in the algorithm. Taking an initial guess, the algorithm will go either to the negative or the positive direction of the gradient to optimize the gradient. This process is repeated till the algorithm converges

Repeat until convergence

{

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

}

Fig1: *Gradient optimization formula for classification.*

They are both first-order algorithms because they take only the first derivative of the function.

R already has built-in functions for performing Logistic Regression and Naïve Bayes classification on the provided datasets. Logistic regression is implemented by an R function glm from a standard package stats in R, and the Naïve Bayes classifier is implemented by an R function Naive Bayes from a contributed package e1071 for R. However, for the purpose of this experiment the classifiers were codified without the use of these classification packages and various performance measures such as Accuracy, Precision, Recall, and F-score were derived.

Experimental Observations

Upon executing the algorithm, the performance measures by which to gauge the efficiency of the algorithm are derived for comparing the asymptotic efficiency of Logistic Regression algorithm versus the Naïve Bayes Classifier.

For deriving the time taken by the classifiers, a built-in R function called ***system.time()*** is used. This function essentially times how fast R processes an expression. Using this function, the differences in speed of Logistic Regression and Naïve Bayes algorithm can be computed and compared and appropriate calculations can be made.

Logistic Regression Time Elapsed			
	user	**system**	**elapsed**
Ionosphere	25.41	0.08	25.64
Sonar	33.38	0.12	33.69
Breast Cancer	25.02	0.05	25.16
Voting	27.45	0.09	27.63
Liver	5.9	0.22	6.27
Chess	139.58	3.14	152.28

Fig: *Time taken for the execution of Logistic Regression Algorithm*

Generative Learning Time Elapsed			
	user	**system**	**elapsed**
Ionosphere	15.02	0.1	16.9
Sonar	14.09	0.09	14.35
Breast Cancer	14.78	0.07	14.97
Voting	7.09	0.14	9.94
Liver	2.4	0.3	3.2
Chess	87.11	0.08	87.52

Fig: *Time taken for the execution of Naïve Bayes Algorithm*

Training a Naive Bayes classifier, assuming you're given the feature vectors, is $O(n)$, where $n=\sum nnz(xi)$, where *nnz*(*xi*) is the number of nonzero features in element *i*. For logistic regression the optimization problem has to be solved, and this depends on the optimizer

Upon executing the algorithm it was observed that Naïve Bayes approaches its asymptotic error without the need for a large number of training examples, and it does so very quickly. Logistic regression, on the other hand, is capable of outperforming naive Bayes, given the number of training examples is large enough.

However, if the number of training examples is very large, then using Logistic regression is comparatively costly, since the parameters of the algorithm are needed to be optimized using gradient optimization. As a rule of thumb, Naive Bayes will almost surely outperform logistic regression if the number of training examples is small.

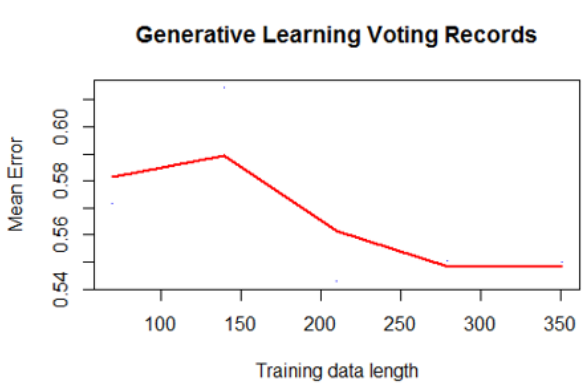
The efficiency of an algorithm can be characterized by its performance measures such as *Accuracy, Precision, Recall, F1 Score* etc. The different performance measures for Logistic Regression Algorithm were reported as follows:

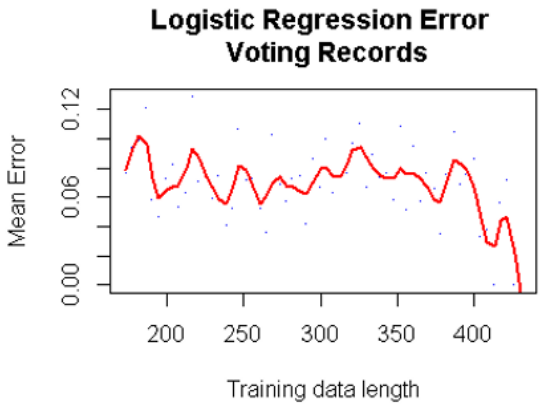
Logistic Regression Performance Measures							
Dataset	**Accuracy**	**Precision1**	**Precision2**	**Recall1**	**Recall2**	**F1**	**F2**
Sonar	0.731	0.817	0.948	0.964	0.753	0.884	0.839
Ionosphere	0.812	0.855	0.936	0.889	0.916	0.872	0.926
Liver	0.438	0.704	0.761	0.89	0.483	0.786	0.591
Breast Cancer	0.82	0.795	0.997	0.996	0.865	0.884	0.926
Voting	0.922	0.96	0.963	0.978	0.935	0.968	0.949
Chess	0.808	0.849	0.971	0.972	0.841	0.906	0.901

Fig: *Logistic Regression characterized by its performance measures.*

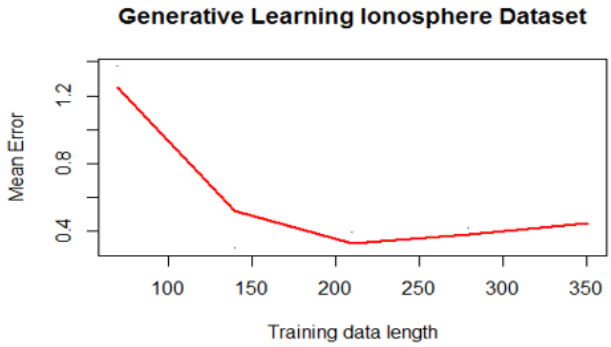
For calculating the mean error rate, we take into account the number of wrongly classified data against the number of training examples considered to fit the data.

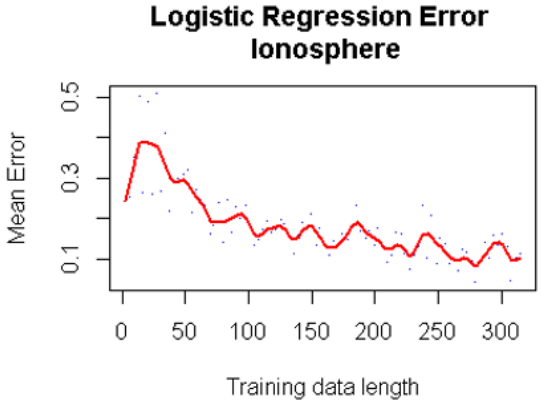
VOTING





IONOSPHERE





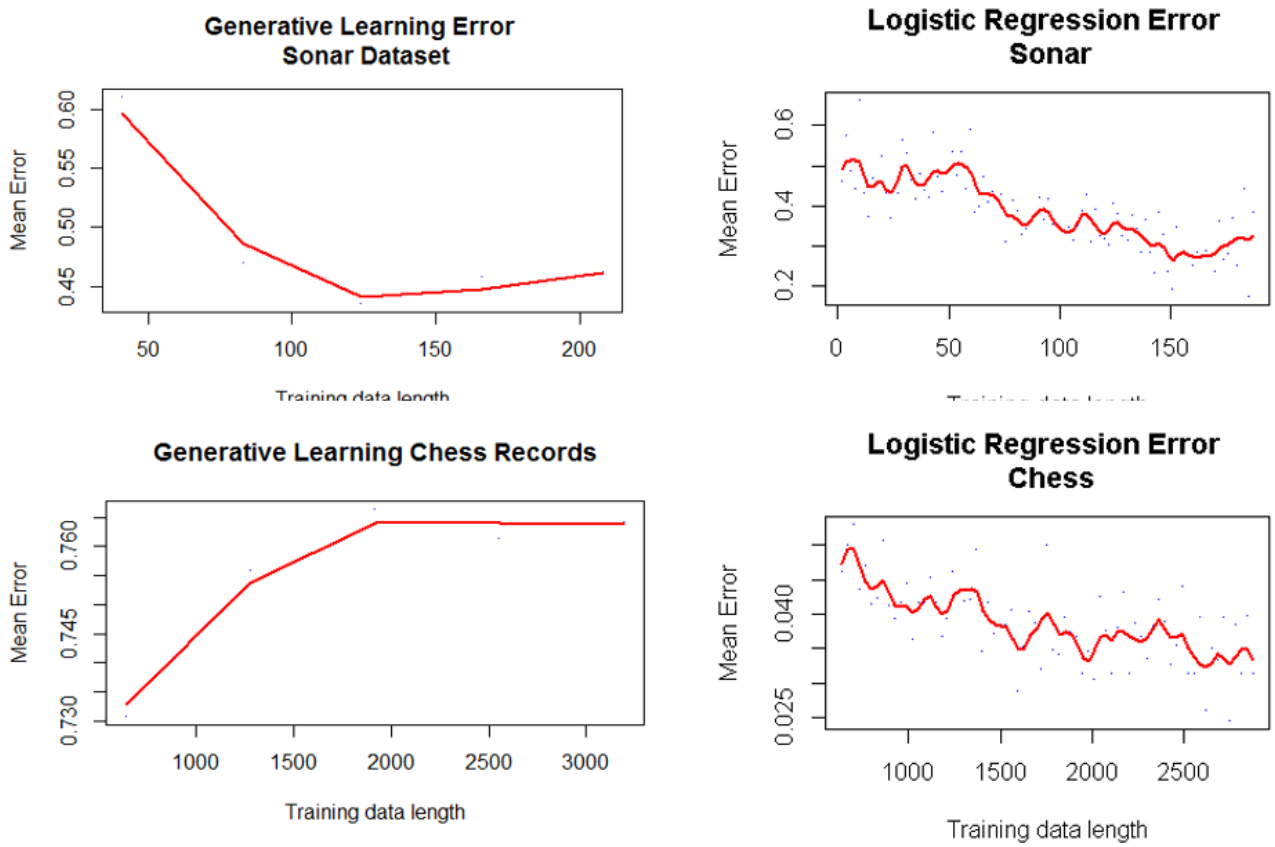


Fig3:.*Plots of misclassification error rate vs. training-set size m for the UCI datasets, with regards to Naïve Bayes and Logistic Regression.*

Inference

Upon executing the algorithm it was observed that Naïve Bayes approaches its asymptotic error without the need for a large number of training examples, and it does so very quickly. Logistic regression, on the other hand, is capable of outperforming naive Bayes, given the number of training examples is large enough.

However, if the number of training examples is very large, then using Logistic regression is comparatively costly, since the parameters of the algorithm are needed to be optimized using gradient optimization. As a rule of thumb, Naive Bayes will almost surely outperform logistic regression if the number of training examples is small. Further, it will reach its asymptotic error very quickly, making it much more desirable in such a scenario than logistic regression.

Thus on execution of both the algorithms on a training set, if Naïve Bayes is outperforming Logistic Regression by a large margin, then it is a safe bet to stick with Naive Bayes until a much larger amount of training examples are available. However, if it is observed that Naive Bayes is not outperforming logistic regression by much, then it can be deduced that Logistic Regression will outperform Naive Bayes within a relatively small amount of new training examples.

Conclusion

Naive Bayes and Logistic Regression are a "generative-discriminative pair," meaning they have the same model form (a linear classifier), but they estimate parameters in different ways.

For feature x and label y, **Naive Bayes** estimates a joint probability **p(x,y) = p(y)*p(x|y)** from the training data (that is, builds a model that could "generate" the data), and uses Bayes Rule to predict p(y|x) for new test instances. On the other hand, **logistic regression** estimates **p(y|x)** directly from the training data by minimizing an error function (which is more "discriminative").

These differences have implications for error rate:

When there are very few training instances, logistic regression might "overfit," because there isn't enough data to estimate p(y|x) reliably. Naive Bayes might do better because it models the entire joint distribution.

When the feature set is large (and sparse, like word features in text classification) naive Bayes might "double count" features that are correlated with each other, because it assumes that each p(x|y) event is independent, when they are not. Logistic regression can do a better job by naturally "splitting the difference" among these correlated features.

If the features really are (mostly) conditionally independent, both models might actually improve with more and more features, provided there are enough data instances. The problem comes when the training set size is small relative to the number of features. Priors on naive Bayes feature parameters, or regularization methods (like L1/Lasso or L2/Ridge) on logistic regression can help in these cases.

Future Work

On the empirical side, combinations of discriminative and generative methodologies have been explored in many fields such as natural language processing, speech recognition, and computer vision.

In particular, the recent "deep learning" revolution of neural networks relies heavily on a hybrid generative-discriminative approach: an unsupervised generative learning phase ("pre-training") is followed by discriminative fine-tuning. Given these recent trends, a workshop on the interplay of generative and discriminative learning seem especially relevant.

Hybrid generative-discriminative techniques face computational challenges. For some models, training these hybrids is akin to the discriminative training of generative models, which is a notoriously hard problem. Alternatively, the use of generative models in predictive settings has been be explored.

Examples of topics of interest that can be explored for further research in this field includes:

Techniques for combining generative and discriminative approaches

Successful applications of hybrids

Inclusion of prior knowledge in discriminative methods (semi-supervised approaches, generalized expectation criteria, posterior regularization, etc.)

Insights into the role of generative/discriminative interface for deep learning

Computational issues in discriminatively trained generative models/hybrid models

Bayesian approaches optimized for predictive performance

Comparison of model-free and model-based approaches in statistics or reinforcement learning

References

1. *On Discriminative vs. Generative classifiers: A comparison of Logistic Regression and Naive Bayes*, by Andrew Y. Ng , Michael I. Jordan, 2001, Neural Information Processing Systems Conference (NIPS)-14

2. *Discriminative vs. generative learning: which one is more efficient?* by Philip M. Long, Rocco Servedio and Hans Ulrich Simon, Information Processing Letters, 103 (4), 2007

3. *Comment on "On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes"* by Jing-Hao Xue and D. Michael Titterington, University College London Department of Statistical Science London WC1E 6BT UK
    a. Neural Processing Letters (Impact Factor: 1.24). 01/2008

4. *Discriminative versus Generative Parameter and Structure Learning of Bayesian Network Classifiers*, by Franz Pernkopf, Jeff Bilmes, ICML '05 Proceedings of the 22nd international conference on Machine learning, 2005

5. *A hybrid generative/discriminative Bayesian classifier*, by Changsung Kang , Jin Tian, In Proceedings of the 19th International FLAIRS Conference, 2006