Comparison of Linear, Quadratic, and Poisson Discriminant Analysis

Wednesday, December 17th, 2014

Introduction

Linear and quadratic discriminant analysis (LDA, QDA) are two popular statistical methods for classification. In both cases, we assume that the predictors are normally distributed. However, such an assumption may not be appropriate in all situations. For example, count data may be better modeled using a Poisson distribution.

Two methods using the Poisson assumption are explored. The first uses the so-called "naive bayes" assumption that all features are *iid* univariate Poisson and estimates each λ_j individually using maximum likelihood. The second, suggested by Witten [2], also uses the "naive bayes" assumption but estimates a decomposed and normalized λ_j and is designed for gene sequencing data. I refer to the first as Poisson (log-linear) discriminant analysis (PDA) and the second as sequence Poisson discriminant analysis (SDA).

The goal of this report is to compare the performance of LDA, QDA, PDA, and SDA on data sets where the Poisson assumption may be more appropriate than the Gaussian assumption. Ideally, models that make the Poisson assumption should perform better in these cases. However, the results of the simulations and data applications in this report indicate that LDA and QDA perform about as well as or better than PDA and SDA on count data.

Methods

I will assume the typical classification framework, in which there are K discrete classes, a N by P data matrix X, and a length N response vector $y_i \in \{1, ..., K\}$.

The densities for each row of X are calculated using our assumption regarding the predictors. In the cases of LDA and QDA, they are normally distributed with and without a common covariance, respectively. For PDA and SDA, each feature is considered independently as a univariate Poisson. Our predictor assumptions are listed below:

$$\begin{array}{ll}
\text{LDA} & X_i \sim MVN(\mu_k, \Sigma) \\
\text{QDA} & X_i \sim MVN(\mu_k, \Sigma_k) \\
\text{PDA} & X_{ij} \sim Poisson(\lambda) \\
\text{SDA} & X_{ij} \sim Poisson(N_{ij} d_{kj})
\end{array}$$

For each method, we first calculate the MLE parameter estimates for the given underlying distribution. In all of the above cases, analytic solutions for the estimates exist. In the case of

SDA, λ is decomposed into N_{ij}, the number of counts per sample multiplied by the number of counts per feature, and d_{kj}, which attempts to capture the condition under which the observation was observed. The MLE estimates are as follows :

$$\widehat{N_{ij}} = \frac{X_{i.} * X_{.j}}{X_{..}} \qquad \widehat{d_{kj}} = \frac{X_{C_k j}}{\sum_{i \in C_k} \hat{N}_{ij}}$$

After estimates have been obtained, the posteriors for each class for each observation are calculated by using bayes rule, multiplying each density with the class prior probabilities. The maximum a posterior estimate for each observation is the class with the largest posterior. Predicting on new data only involves calculating the new posteriors using the old training parameter estimates.

Software

All the software for this report is contained in the `plda` **R** package, which is not available on CRAN but can be installed using `devtools`:

```
install.packages("devtools")
devtools::install_github("mattdelhey/plda")
```

As an example, one can run QDA on the iris data set:

```
library(plda)
data(iris)
X <- as.matrix(iris[, 1:4])
y <- as.factor(iris[, 5])
fit <- plda(X, y, type = "linear")
fit
```

```
X(150 x 4) with K = 3
Type: **linear** discriminant analysis
Dist: predictors have a **normal** distribution

pi.hat:
      setosa versicolor virginica
[1,] 0.33333    0.33333    0.3333

mu.hat:
           Sepal.Length Sepal.Width Petal.Length Petal.Width
setosa            5.006       3.428        1.462       0.246
versicolor        5.936       2.770        4.260       1.326
virginica         6.588       2.974        5.552       2.026
```

```
sigma.hat:
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length     0.265008    0.092721     0.167514      0.0384
Sepal.Width      0.092721    0.115388     0.055244      0.0327
Petal.Length     0.167514    0.055244     0.185188      0.0427
Petal.Width      0.038401    0.032710     0.042665      0.0419
```

The simulations, data applications, and the sequencing data set can be found in the `plda` package directory which can be located by running the following command in **R**:

```
system.file(package = "plda")
```

Simulations

Two simulated data sets were used to compare the four models. In each simulation, N_k *iid* observations are generated for each of the K classes. Each of the P features are generated from independent univariate Poisson with a unique lambda for each class. The parameters for each of the simulations are as follows:

Simulation Parameters

	N_k	P	K
Small Simulation	50	2	3
Large Simulation	100	5	5

The small simulation is for visualizing the decision boundaries of the different methods, whereas the large simulation will be be compared using error rates. Therefore, in the first case all data is used to train the classifiers whereas in the latter half of the available data will be reserved for testing.

Small simulation

In this simulation, both QDA and PDA successfully classify all observations and LDA misclassified two observations. SDA performs very poorly, and I am not sure why. The simulation is limited to two dimensions so that the decision boundaries can visualized in the original feature space, seen in Figure 1. The Poisson approaches, like LDA, allow only for linear boundaries between classes. On the other hand, QDA allows for elliptical decision boundaries between classes.

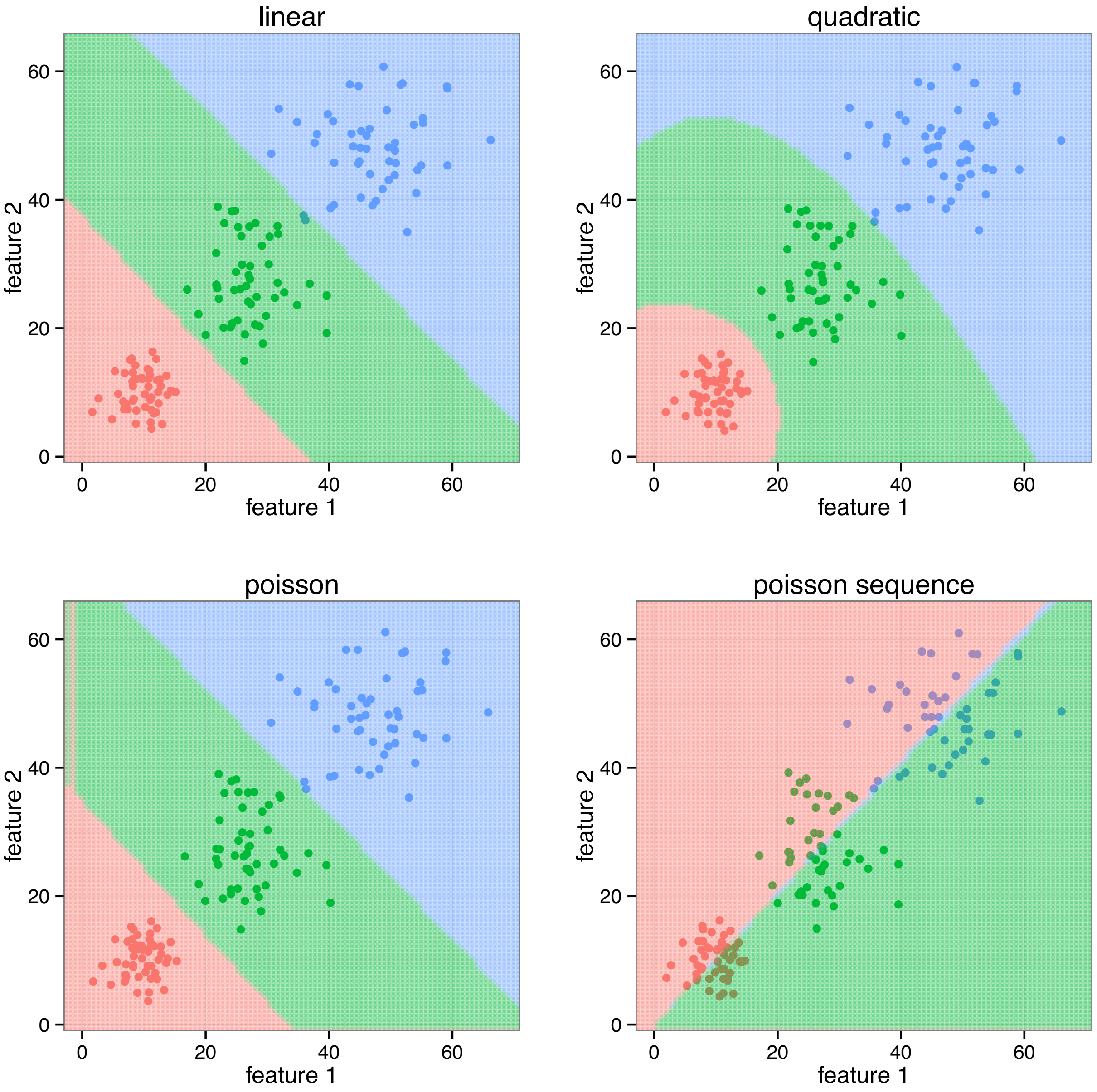

Figure 1: Decision boundaries for the small simulation. The color of each point is its actual class, and the color of the background grid is the predicted class.

Large simulation

In this simulation, half of the available observations are used are reserved and testing error rates are compared. Error rates were collected for 100 simulated data sets, and the standard errors are for the error rates.

Error Rates

	Mean	SE
LDA	4.96	1.37
QDA	4.60	1.54
PDA	4.80	1.38
SDA	69.48	3.09

LDA, QDA, and PDA all perform similarly, successfully classifying about 95% of the data. This conflicts with the notion that PDA might outperform LDA and QDA given that the data is generated directly from the PDA model assumptions (independent univariate Poisson distributions).

Data sets

Iris

The iris data set contains 150 observations of sepal and petal measurements for three species of iris flowers. The predictors in this data set are best modeled using a normality assumption as opposed to a Poisson assumption. The goal of this data set is to see how the Poisson methods perform when the data is not likely drawn from a Poisson distributions.

Each model was fit 100 times on the data set, with each replication randomly splitting the data into training (60%) and testing (40%) sets. Reported below are the mean misclassification rates and their standard errors.

Error Rates

	Mean	SE
LDA	2.20	1.57
QDA	2.00	1.54
PDA	4.32	2.03
SDA	3.18	1.71

As expected, LDA and QDA outperform PDA and SDA. SDA performs much better on real data than on the simulated data.

Sequence data

The sequencing data set was obtained from the UCI Machine Learning Repository [1] and contains 3190 primate splice-junction gene sequences.[1] Each observation contains a DNA sequence of 60 nucleobases that is classified either as an IE "acceptor", EI "donor", or neither.

The data is then transformed into counts of so-called DNA "words" or oligomers of size two, in which we count the number of appearance of each combination of two nucleobases. This results in $P = 4^2 = 16$ features for each observation.

We then split the data into training (60%) and testing (40%) and calculate the misclassification rate of all four methods. This process is replicated 100 times, and the standard errors are of the misclassification rates.

Error Rates

	Mean	SE
LDA	42.27	0.79
QDA	44.13	2.38
PDA	47.19	0.74
SDA	47.30	0.80

All methods perform rather similarly and rather poorly, as the original release of the dataset in 1992 included methods that achieved misclassification rates of 10% or lower. In relative comparison, LDA and QDA performed slightly better than PDA and SDA, with LDA performing the best overall. Once again, SDA performs the worst overall despite its attempts to exploit the counting nature of the data. QDA has a noticeably higher standard error than the other models due to the fact that it must estimate the most parameters.

Conclusions

In conclusion, LDA and QDA seem to perform as well as or better than PDA and SDA, even on data sets that appear to be modelable by the Poisson distribution. Given the prevalence of LDA and QDA routines, the extra effort required to run PDA or SDA does not appear to be worth it in the typical case. However, PDA and SDA have their advantage in the $P >> N$ scenario, where LDA and QDA are likely to fail, as they must take the inverse of $\hat{\Sigma}$, which is potentially ill-conditioned.

It may be possible to improve upon the Poisson models by adopted them to the Negative Binomial distribution. Additionally, improvements may be achieved by using a zero-inflated Poisson model. Various transformations to Poisson-like data that aid in Poisson modeling have also been suggested. Despite their moderate performance on the above data applications, Poisson discriminant models mays still be a useful statistical method in the $P >> N$ case where normal models are unavailable.

[1] From the data set information: "Splice junctions are points on a DNA sequence at which "superfluous" DNA is removed during the process of protein creation in higher organisms."

References

[1] Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[2] Witten, D. (2012). Classification and clustering of sequencing data using a poisson model. The Annals of Applied Statistics, 5(4):2493–2518.