Diagnosing Malignant versus Benign Breast Tumors via Machine Learning Techniques in High Dimensions

Danielle C. Maddix
CS 229 Machine Learning
Final Writeup

December 12, 2014

1 Introduction and Predicting

Machine learning applications are vast; one such particular application to be investigated is in regards to classifying whether a breast tumor is malignant or benign. Even though there are predictive medical procedures that are available for diagnosis, one test may not be definitive enough and there can be error margins of either false positives or false negatives. An algorithm which could take into account many test diagnostics and make a prediction has potential to have a broad impact in the medical field. In fact, the medical literature is already becoming rich in such methods, with the potential goal of submitting patients to fewer extensive testing. In terms of machine learning, this is a binary classification problem with continuous *input* feature vectors x_i that can be solved via supervised learning, since the corresponding labels y_i, 0 for negative benign or 1 for positive malignant, are also given, forming m training examples (x_i, y_i). The *output* is either a negative label 0 for benign or positive label 1 for malignant. It is clear that some features are very predictive, such as the size of the tumor, but only considering this feature cannot give definitive results. The following approach considers the effect of a larger number of features or dimensions, n.

2 Data and Features

The comprehensive dataset utilized is available from the Breast Cancer Wisconsin (Diagnostic) Dataset on the UC Irvine Machine Learning Repository. The dataset is fairly rich in examples, considering $m = 569$ patients. It consists of a matrix with 32 columns, where the first such column is the patient ID and so ignored in this study and the second column is the label $M = 1$ for malignant and $B = 0$ for benign. The remaining 30 columns form the vector x_i in the training example (x_i, y_i). There are ten distinct continuous features measured, namely the radius, texture, perimeter, area, smoothness, compactness, concave points, concavity, symmetry and fractal dimension. For each of these, the average, standard error and the worst case measurements are reported. The class distribution is given by 357 benign samples($\sim$62.7%) and 212 malignant samples($\sim$37.3%). Note that this is fairly representative of the positive learning malignant samples, which can be difficult in medical based datasets, such as with AIDs diagnoses. Another dataset to be explored for future research focuses other physical and also biological features, such as clump thickness, since cancer cells tend to form multilayers, uniformity of cell size and shape, epithelial cell size, bare nuclei, bland chromatin and mitoses. A comparison of the results from training on this dataset to the prior one could help distinguish the characteristics that are most relevant in the diagnosis process.

3 Models

The approach to this binary classification problem was to implement several supervised learning algorithms and compare and contrast their results and error properties. The first such algorithm was logistic regression,

which is a discriminative learning algorithm directly modeling the conditional probability, $p(y|x)$. The fitting parameters $\theta \in \mathbb{R}^{n+1}$, including the intercept terms are computed via the maximum likelihood estimators and then an optimization algorithm is used to find the optimal θ. Both the second order Newton's method and gradient ascent were explored. Newton's Method was preferred to both stochastic and batch gradient ascent. Even though the implementation of the gradient ascent is simpler and each iteration is cheaper, since it only requires calculating the gradient rather the Hessian, it took far more iterations to converge. Newton's Method already had error, as measured by the norm of the gradient, of approximately machine precision ϵ after 9-11 iterations, in comparison to the hundreds for both gradient ascents. Another challenge of gradient descent is choosing the proper step length α to expedite convergence. This could be done via parameter fitting or by choosing an adaptive α via a bisection method. However, a downside to Newton's Method is that it is more subject to round-off errors. The Hessian must stay negative semi-definite and not be poorly-conditioned, since it is being inverted in the algorithm. In order to avoid these poor numerical properties, the feature data in the design matrix $X \in \mathbb{R}^{mx(n+1)}$ was normalized for logistic regression, since the original feature data given had very different scales of magnitude. Note that this normalization was not necessary the Gaussian Discriminant Analysis (GDA) algorithm and so the implementation of GDA was fairly straightforward in such that no modifications needed to be made.

GDA is a generative learning algorithm, which contrary to discriminative algorithms, first builds a model for $p(x|y = 1)$, the positive class of malignant tumors and also builds a model for $p(x|y = 0)$, the negative class of benign tumors. It then learns $p(y|x)$ using Bayes' Rule:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x|y = 0)p(y = 0) + p(x|y = 1)p(y = 1)} \tag{1}$$

In linear GDA, the posterior densities are assumed to be Multivariate Gaussians with means μ_0 and μ_1, respectively and same covariance matrix Σ and the prior density $p(y)$ is assumed to be Bernoulli distributed. In other words, $x|y = 0 \sim \mathcal{N}(\mu_0, \Sigma)$, $x|y = 1 \sim \mathcal{N}(\mu_1, \Sigma)$ and $y \sim$ Bernouilli(ϕ). In the quadratic case, the means and prior remain the same, but there are two distinct covariance matrices, Σ_0 and Σ_1 such that $x|y = 0 \sim \mathcal{N}(\mu_0, \Sigma_0)$ and $x|y = 1 \sim \mathcal{N}(\mu_1, \Sigma_1)$. Note that these parameters were computed, using their corresponding maximum likelihood estimates (MLE). An advantage of the quadratic case is that it allows the decision boundary to be nonlinear, which can help in models with high bias to increase the dimension of the hypothesis space, $\mathcal{H}$.

The last supervised learning algorithm implemented was Support Vector Machine (SVM). This algorithm is designed to maximize the functional and geometric margins and so it is known as the optimal margin classifier. This is a desired property to prevent the dataset points from clustering around the decision boundary, where the margin for misclassification is the highest. Thus, it solves an optimization problem to maximize the distance between the points and decision boundary, as the primal is displayed below:

$$\min_{\gamma,w,b} \frac{1}{2}\|w\|^2 \tag{2}$$

$$\text{s.t. } y^{(i)}(w^T x^{(i)} + b) \geq 1, \text{ i} = 1, \text{ . . ., m} \tag{3}$$

The *train* and *predict* functions within liblinear-1.94 were used for implementation purposes in Matlab. Note that all of the other above described algorithms were implemented from scratch in Matlab.

4 Results and Analysis

Below are the error tables comparing the results from the various algorithms, namely linear GDA, quadratic GDA, logistic regression and SVM. Note that the error measurements reported for logistic regression are using the second-order Newton's Method, rather than stochastic or batch gradient ascent, since it produced more accurate results with a fewer number of iterations. To explore the usage of the best number of features, the errors were tabulated for a various number of features.

Table 1: Results: Various tests versus various number of features for GDA

Linear GDA	1	9	19	29
Hold-out CV	6.47%	5.88%	2.94%	2.35%
k-fold CV	12.50%	7.32%	6.96%	4.46%
Recall	85.0%	94.87%	94.87%	92.31%
Precision	87.18%	82.22%	92.50%	97.30%

Table 2: Results: Various tests versus various number of features for GDA

Quadratic GDA	1	9	19	29
Hold-out CV	7.65%	2.35%	2.35%	3.53%
k-fold CV	17.32%	15.89%	15.54%	12.32%
Recall	74.36%	92.31%	89.74%	84.62%
Precision	90.62%	97.30%	100.0%	100.0%

Table 3: Results: Various tests versus various number of features for Newton Logistic Regression

Logistic Regression	1	9	15	19
Hold-out CV	14.12%	9.41%	6.47%	5.29%
Number of Iterations	8	11	12	12
k-fold CV	13.39%	6.79%	5.89%	5.36%
Recall	89.74%	94.87%	97.44%	97.44%
Precision	63.64%	72.55 %	79.17%	82.61%

Table 4: Results: Various tests versus various number of features for SVM

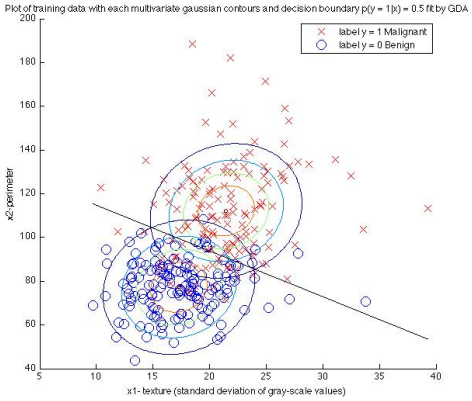
SVM	15	25	30
Hold-out CV	7.65%	4.71%	4.12%
Optimization Accuracy	92.3529%	95.2941%	95.8824%
k-fold CV	11.07%	10.71%	7.32%
Recall	69.23%	89.74%	92.31%
Precision	96.43%	89.74 %	90.00%

The hold-out cross validation approximation to the generalization error was computed by training on 70% of the data, namely 399 samples and testing on the remaining 170. The recall, the ratio of true positives to actual positives, as a measure of the lack of false negatives, and the precision, the ratio of true positives to labeled positives, as a measure of the lack of false positives were also computed on the above data sampling. It is clear from the above tables that GDA produces higher precision than logistic regression and SVM, whereas logistic regression produces highest recall. In this problem in particular, the higher recall may be more valuable, since a false negative could be more dangerous to the care of a patient, who then may not be treated, whereas with a false positive, the patient would most likely undergo more testing before treatment. Furthermore, we also note that in all cases, the hold-out CV error decreases as the number of features increase, which is indicative of a high bias problem. This error for GDA is generally lower, which can be explained by GDA's property to use data more efficiently, since it can learn more quickly on smaller datasets.

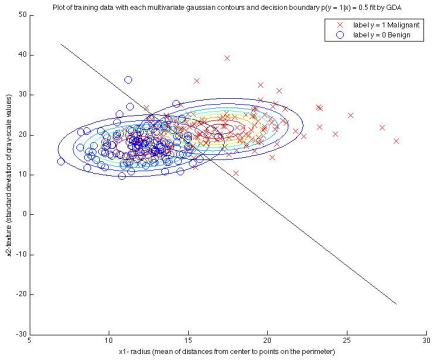
For another error metric, the k-fold CV was also calculated, which only holds out m/k, for $k = 10$, of the training data each time for testing and trains on the remaining and takes the average of these k errors. As expected, logistic regression has lower error in the 9 and 19 features case, since it is generally asymptotically more efficient and robust with larger data. We note that the k-fold CV error is higher than the hold-out

CV error and it also decreases as the number of features increase. For more features than 20, the Hessian becomes ill-conditioned due to the dependency of the features and so that error is not reported.

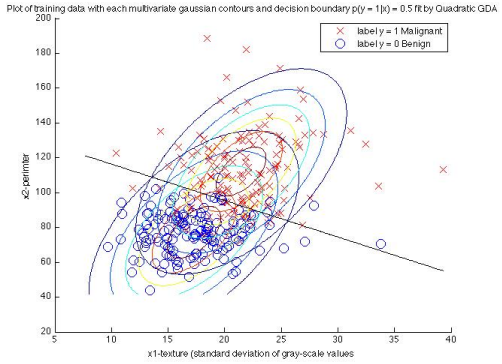
To visualize the results, a comparison of two two-dimensional cases for the algorithms trained on the first 300 of the training examples are displayed below, along with their training errors and in addition for logistic regression the number of iterations it took Newton's Method to converge. The plots on the left are perimeter versus texture and the plots on the right are texture versus radius. Note the decision boundaries and for both versions of GDA and the contours of each multivariate gaussian modeling the positive and negative classes.



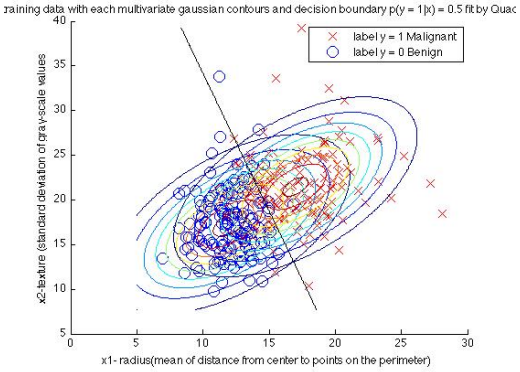
(a) Linear GDA: training error = 12.33%



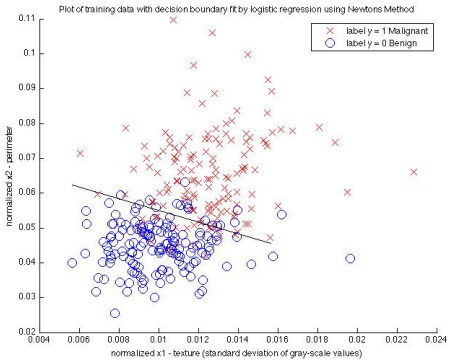
(b) Linear GDA: training error = 12.33%



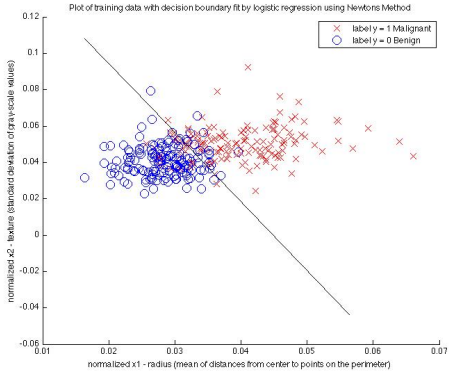
(c) Quadratic GDA: training error =12.33%



(d) Quadratic GDA: training error = 13.33%



(e) Logistic Regression: training error =12.67%, niter = 9



(f) Logistic Regression: training error = 12.33%, niter = 9

5 Discussion

The GDA algorithm seems to product the best overall results, which are even better than logistic regression and SVM. At first, this may seem surprising, since in general the latter two algorithms make less assumptions and are more robust. A possible explanation for this is that the posterior data is actually distributed as a multivariate Gaussian. This is collaborated by the alignment of the contours in the plotting section. If this is the case, then GDA will clearly outperform both algorithms. Moreover, to compare linear vs quadratic GDA, we see that the assumption linear GDA makes a good approximation to data, namely that both distributions share the same covariance matrix Σ. We observe that Σ_0 and Σ_1 are very close in values of elements and so quadratic GDA appears as linear, since the quadratic term $\frac{1}{2}x^T(\Sigma_0^{-1} - \Sigma_1^{-1})x$ is very small in terms of machine precision and linear term dominates. Furthermore, generally for all models, there is decrease in the error measurements with increased dimensional feature space. This demonstrates that solving this binary classification problem in higher dimensional feature spaces positively affects the results. Note that SVM specifically does better in higher dimensional feature space and so these are the results included in the SVM table.

6 Future Work and Conclusions

It is evident from the error decrease with increased number of features that the linear models suffer from high bias. Moreover, the small gap between the training error and approximate generalization error indicates high bias. This implies that the hypothesis class of linear separators is not rich enough. Future work includes investigating using a nonlinear decision boundary with higher dimensional polynomials.

Alternate approaches to consider are regularization with Bayesian logistic regression or in terms of generative learning algorithms a multinomial Naive Bayes model with Laplace smoothing, where the features are discretized within certain ranges. Moreover, it is key to find the optimal number of features relevant to tumor diagnosis to solve the problem in a smaller subspace and so feature selection should be implemented.

Lastly, investigating the other two datasets available in the repository would be interesting. As stated in the data section, a comparison of this more physical based dataset with measurements from medical images to the dataset with 10 more biological based features could be used to indicate which of these types of characteristics is more highly correlated in diagnosis. Another available dataset can be used to solve a slightly different binary classification in supervised learning of whether a tumor is more likely to recur of non-recur.

It is clear that there are promising results in the area of applying supervised learning algorithms into the realm of cancer diagnosis for potential use in collaboration with the established medical tests and to help avoid evasive diagnostic tests on patients.

References

[1] UCI Machine Learning Repository: Center for Machine Learning and Intelligent Systems. Breast cancer wisconsin (diagnostic) data set: wdbc.data. `http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wpbc.data`, 1996.

[2] UCI Machine Learning Repository: Center for Machine Learning and Intelligent Systems. Breast cancer wisconsin (diagnostic) data set: wdbc.names. `http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.names`, 1996.

[3] Andrew Ng. Lecture notes 1-5. `http://cs229.stanford.edu/materials.html`, 2014.