Classification of High Grade vs Low Grade GBM Tumors

Vincent-Pierre Berges, Victor Storchan, Kevin Luo

December 2015

Abstract

This paper address Glioblastoma (GBM) disease, which is a very frequent brain cancer in adults. Classifying these tumors between two clusters of high-grades and low grades tumors by noninvasive methods would improve the accuracy of targeted therapy and personalized treatment. We dealt with a partially processed data set of 274 patients, consisting of 220 patients affected by high grade tumors (very dangerous) and 54 affected by low grade tumors (less dangerous). We performed classification using the logistic regression algorithm. After cross-validation, the resulting overall error is about 25%. The Confusion matrix and Receiver operating characteristic (ROC curve) are provided.

1 Introduction and Related Work

Traditionally, medical imaging has been approached as a qualitative science. New advances in medical imaging acquisition and analysis enable the extraction of imaging features to estimate the differences between biological tissues. Glioblastoma (GBM) is a notorious brain cancer with a high rate of death for adult patients. According to P.Y. Wen and Santosh Kesari (see [1]), malignant Glioblastoma accounts for approximately 70% of the 22,500 new cases of malignant brain tumors that are found in adults in the United States each year. As this disease is very common and performing invasive methods such as biopsies can be harmful for the patient, it is of high importance to be able to recognize when immediate treatment is necessary. We are thinking of using noninvasive methods such as imaging and machine learning to help radiologists determine how dangerous a GBM tumor is. This approach is related to an expanding and promising field called Radiomics. Although some papers dealt with multi-modality imaging in cancer classification, they based their analysis on huge balanced data sets (see [**?**]). What if someone only had access to a smaller data set, or even to an unbalanced data set between low grade and high grade tumors? Our work tries to provide insight to this question.

2 Data processing and Features Extraction

2.1 Analysis of the Data Set

Decoding tumor phenotype by noninvasive methods is an emerging field and literature provides the first attempts of converting imaging data into a high dimensional workable feature space (see [2], [3]) . We used data from MICCAI BRATS provided by Professor Olivier Gevaert and his students Darvin Yi and Mu Zhou from the Stanford Center for Biomedical Informatics. We have 274 patients consisting of 220 patients affected by high grade tumors (very dangerous) and 54 affected by low grade tumors (less dangerous). That data has already been partially processed to identify the location, size, and segments of the tumor. Within a tumor, four regions can be emphasized:

Necrosis • Edema • Non Enhancing Tumor • Enhancing Tumor

The matrix of predictors has to keep track of their specificities. Each patient has five volumetric images (3D scans): four concern different modalities of the brain (T1: longitudinal relaxation time; T1c: longitudinal relaxation time after administration of contrast agent; T2 : transverse relaxation time; Flair: Fluid attenuated inversion recovery MRI) and one just focuses on the tumor. Note that we do not follow a patient overtime through his therapy. Even though it does not reflect the exact reality, we count each data as being related to a different patient.

2.2 Features Extraction

The data takes the form of 2 lists (one high grade and one low grade) of folders of patients each containing 5 tomographic images. The function **readMHA.m** uses the library ReadData3D and allows us to extract 3D matrices of pixel intensities from .mha files. From the volume computed tomography images, we are extracting histograms of the pixel intensities for each patient, which have to be preprocessed to be comparable. Indeed, because of the improvement of software overtime, our data set contains images extracted by different devices so the mean contrast and intensity are not comparable. To deal with this issue, we converted the intensities to their z-scores on the whole brain. This is the role of **normalizeVolume.m**. Then we extracted histograms for each of the four regions and four modalities and used them to generate the features. There are some brain scans which do not contain certain regions and cannot produce the histograms. One of the challenges of this project was to determine how to deal with lack of information on a small set of training examples (see the next subsection for the resolution of the issue). The function **patientIterator.m** iterates over all the folders and images and calls **getPredictors.m** on each patient to obtain the vector of predictors. When **patientIterator.m** terminates, it generates and saves a matrix ID containing the names of the patients, a matrix of predictors X and a vector of output Y that can be used later for the classification.

2.3 Choice of the Predictors

For this project, our first step was to use n = 36 features: from the first four images of the brain, the histograms of the pixel intensities were computed on the four available regions, and we extracted both the mean and the standard deviation from them. This yields 32 features. From the tumor image, we can extract the size of the different regions of the tumor. This produces four more features. After training our algorithm in a first attempt, this naive approach led to complete separation of the set. We obtained the following Warning in MATLAB:

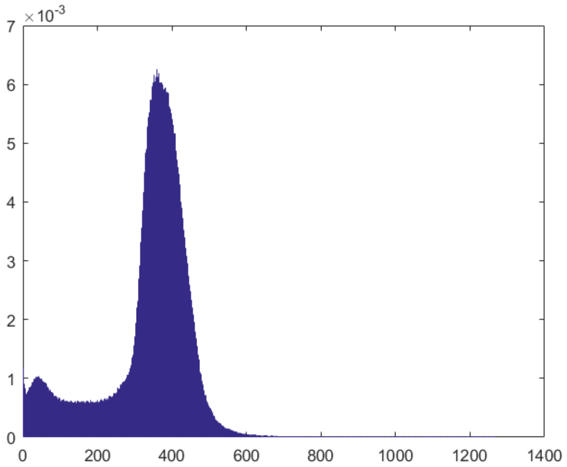
```
Warning: Iteration limit reached.
Warning: The estimated coefficients perfectly separate failures from successes. This
    means the theoretical best estimates are not finite.
```

Actually, the 36 features separated entirely our subset of training low grades tumors (approximately 70% of 54). We concluded that the number of predictors was too big compared to the number of low grades of our training sets.

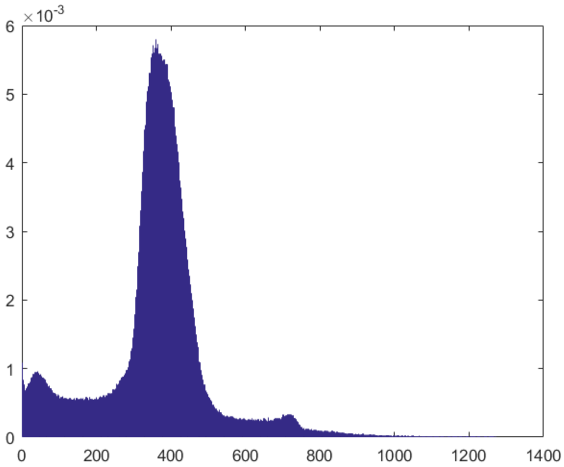
For this reason, we decided to reduce the number of features: we tried to focus on getting regions Necrosis (1) and Enhancing Tumor (4) most accurately. Non Enhancing Tumor (3) is important for low grade tumors, but due to its scarcity, this should lead to lots of 'NaN' in the predictors. Edema (2) represents swelling. As a consequence, it may not be extremely clinically relevant. Therefore we reduced to a set of 24 features: the size (pixel count) of the different regions were removed, and the information (mean and variance) related to edema on each image was removed too. With only 24 features, there was no longer a perfect separation of low and high grades (we reduced the flexibility of the model). At this point, one can produce a matrix X of predictors, filled with floats and 'NaN' values at the places were there was no information provided by the volumetric images. The idea is then to replace the ‘NaN’ values by a combination of known values of the same kind. This is the role of the script **cleanData.m**:

- ‘NaN’ of the standard deviation are replaced by 0
- ‘NaN’ of the mean of the histograms are replaced by the mean value of the predictor

On the picture (a) below, one can see that the histogram extracted from the healthy tissue does not have the tiny bump existing on picture (b). One need to investigate on this bump and to try to extract information from it. The path we followed was to consider the characteristics of the 4 regions of the tumor. The information is extracted by Counting the frequency of the colors which appear on the volumetric images of the regions.



(a) A histogram from a non tumor slice



(b) A histogram of a slice of the tumor

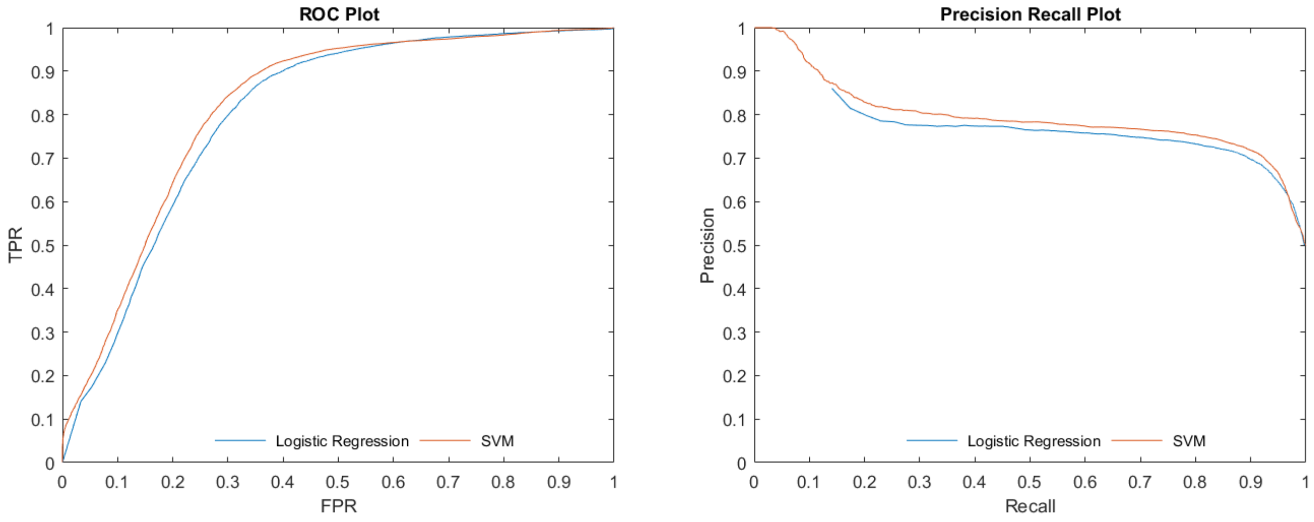
3 Validation of the Model and Consistency of the Results

We tried two approaches: logistic regression and SVM. We compare their performances in this part.

3.1 The Hold out Cross-Validation Implementation

We used the hold out cross validation to predict what is the best threshold to use with our logistic regression. To do so, we separated the matrix of predictors into 1000 training sets and associated test sets for each value of the threshold (the threshold ranges from 0 to 1) and used the ‘**glmfit()**’ set to logistic regression and ‘**glmval()**’ functions of MATLAB. We then averaged the values found for the true positive, true negative, false positive and false negative to obtain the hold out cross validation values. We designed our test sets to be picked at random but we wanted them to contain the same number of high and low grade. For this reason, out of the 274 patients, each test set contains 16 low and 16 high. This means that a purely random prediction would predict with 50% error.

To compare different methods, and their efficiency, we ran a SVM implementation on our data set with MATLAB toolbox ‘liblinear-2.1’. In this SVM approach, we carry out the same experiment except that the tuning parameter is the "decision parameter" (or "decision value"). In MATLAB, it takes a value in [-5,5] and corresponds to the distance of the separating hyperplane to the origin.



(a) The ROC curves for both regressions

(b) Precision Recall curves for both regressions

3.2 The ROC

With different values of threshold (or decision parameter), we were able to compute the receiver operating characteristic curve and the area under it:

- The area under the ROC curve for logistic regression is **0.7997**.
- The area under the ROC curve for SVM is **0.8180**.

We can notice that SVM gives better results than the logistic regression.

3.3 The Precision Recall

With different values of threshold (or decision parameter), we were able to compute the precision recall curve and the area under it:

- The area under the precision recall curve for logistic regression is **0.6420**.
- The area under the precision recall curve for SVM is **0.7924**.

Once again, SVM gives better results than logistic regression.

3.4 The Confusion Matrix

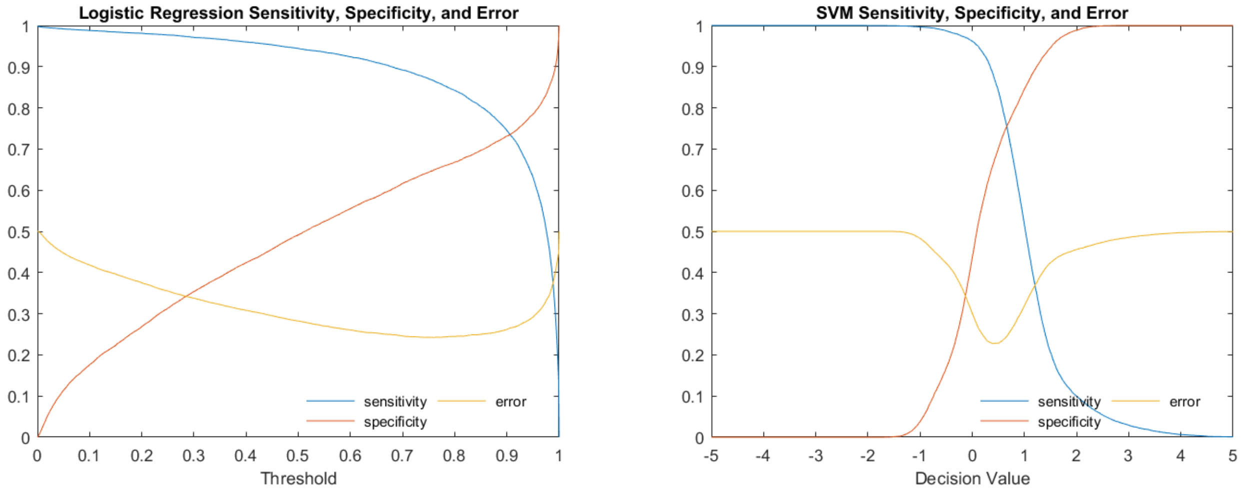
After computing the threshold that minimizes the hold out cross validation, we can fit a logistic regression using this threshold. In our case, the optimal threshold is around 0.75 and gives the following confusion matrix:

	real false	**real true**
predicted false	10.35	2.11
predicted true	5.65	13.89

Using this threshold, the test error is 24.24% (random guessing is 50%).
There is another way to fix the threshold. We could use the threshold that equates specificity and sensitivity. In our case, determining the low grade patients is as important as determining the high grade ones. For this reason, this method seems appropriate for our project. Looking at the sensitivity vs specificity curve, we see they cross for a threshold value of 0.90. The confusion matrix obtained is:

	real false	**real true**
predicted false	11.77	4.24
predicted true	4.23	11.76

As expected, the confusion matrix is now more symmetric, with more balanced false positive and false negatives. The error increased only moderately to 26.50% which makes this method for choosing the threshold very encouraging.



(a) Error, Specificity and Sensitivity analysis for Logistic

(b) Error, Specificity and Sensitivity analysis for SVM

Comparing these results with the SVM approach, the decision value that minimizes the error is 0.40. The error is 22.75%. For this value, the confusion matrix is:

	real false	**real true**
predicted false	10.56	1.84
predicted true	5.44	14.16

Similarly, when we equate specificity and sensitivity, the decision value is 0.66 and the error only increased to reach 24.5%. The resulting confusion matrix is:

	real false	**real true**
predicted false	12.07	3.91
predicted true	3.93	12.09

We notice that the SVM prediction is better, once again than the logistic regression.

4 Conclusion

To conclude, after carefully cleaning the data set, then studying it both in terms of understanding the relevant features, and in terms of extracting these features we handled two classification algorithms. The results showed that even with an unbalanced data set which provide few information on the low grade tumor, the learning phase generates predictions which out-perform the naïve approach of 30%. Although the point of view adopted in this work stays at the level of phenotypic information, other approaches could be handled. For instance, Radiogenomics, tries to establish links between image features and gene expression. Olivier Gevaert published innovative research on this field (see [4], [5]).

Note that all the source code is available online at https://github.com/cs229classif.

5 References

[1] Patrick Y. Wen, M.D., and Santosh Kesari, M.D., Ph.D. N Engl J Med 2008; 359:492-507 July 31, 2008DOI: 10.1056/NEJMra0708126

[2] Aerts, H. J., Velazquez, E. R., Leijenaar, R. T., Parmar, C., Grossmann, P., Cavalho, S., ... & Lambin, P. (2014). Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. Nature communications, 5.

[3] Lambin, P., Rios-Velazquez, E., Leijenaar, R., Carvalho, S., van Stiphout, R. G., Granton, P., ... & Aerts, H. J. (2012). Radiomics: extracting more information from medical images using advanced feature analysis. European Journal of Cancer, 48(4), 441-446.

[4] Itakura, H., Achrol, A. S., Mitchell, L. A., Loya, J. J., Liu, T., Westbroek, E. M., ... & Gevaert, O. (2015). Magnetic resonance image features identify glioblastoma phenotypic subtypes with distinct molecular pathway activities. Science translational medicine, 7(303), 303ra138-303ra138.

[5] Gevaert, O., Mitchell, L. A., Achrol, A. S., Xu, J., Echegaray, S., Steinberg, G. K., ... & Plevritis, S. K. (2014). Glioblastoma multiforme: exploratory radiogenomic analysis by using quantitative image features. Radiology, 273(1), 168-174.