SVM for Epilepsy Patients Classification

Connor Baumler and Noah Crowley

1 Introduction

By gathering data from electroencephalogram (EEG) readings from patients with epilepsy during both period of seizure and periods of non-seizure, we are able to use statistical processing and optimization procedures to train classifiers that will be able to quickly identify when a patient is having a seizure. In this case study, we have investigated using three different types of classifiers – a simple classifier, a linear support-vector machine (SVM), and a nonlinear SVM.

We follow Firpi et. al.[1] and use data that has been pre-processed so as to remove noise and improve the potential accuracy of our classifiers. Each sample in our data is an average of 4096 consecutive EEG samples, or one epoch, which is then broken into 20 spectral sub-bands (0-1Hz, 1-2Hz, ..., 19-20Hz). We had 50 labelled samples of both seizure and non-seizure data available for training, as well as another 50 samples for each case that were put into a separate test set.

2 Classifiers

We developed and trained three separate classifiers for this case study. They are listed and explained in increasing order of complexity.

2.1 Simple Classifier

For our simple classifier, we took the average point in each class. To classify a test point, we compared the euclidean distance between the two points and placed it in the class corresponding to the average point it lied closer to. This classifier is essentially creating a non-overlapping binary cluster classifier that we can expect to give less than ideal results.

2.2 Linear SVM

For the linear SVM, we trained on a quadratic linear program and used the built-in MATLAB `quadprog` function as our solver. This allowed us to generate a n dimensional vector of weights, $\mathbf{w} \in \mathbb{R}^n$, as well as a bias term, $b \in \mathbb{R}$, that together are able to fully represent our classifier. Then given any sample $\mathbf{x} \in \mathbb{R}^n$ we can classify the sample as 0 or 1 using the following formula:

$$\frac{sign(\mathbf{w}^\intercal \mathbf{x} - b) + 1}{2}$$

2.3 Nonlinear SVM

For the nonlinear SVM, we chose to use the kernel function $\Phi(x) = x^2$. We used MATLAB's built-in `fmincon` function as our solver. This gave us a set of weights, $\mathbf{w} \in \mathbb{R}^n$, as well as a bias term, $b \in \mathbb{R}$, that together are able to fully represent our classifier. Then given any sample $\mathbf{x} \in \mathbb{R}^n$ we can classify the sample as 0 or 1 using the following formula:

$$\frac{sign((\mathbf{w}^\intercal \mathbf{x})^2 - b) + 1}{2}$$

3 Methods

We used two methods for testing our classifiers. The first is a straight-forward method using the datasets as explained above, and the second involves a more nuanced approach that produces more informative results.

3.1 Simple training

As stated previously, the data was separated into three sets – two labelled sets of 50 samples each, one for each label, and one set of 100 non-labelled samples that contains 50 samples of each label. In the simple training stage, the classifiers were each trained using the two sets of 50 labelled examples and tested on the remaining 100 samples (which we had the ground-truth labels for stored separately).

3.2 Cross-validation training

In order to further protect against overfitting and to allow us to do statistical tests to compare our classifiers we also train and test the classifiers using 5-fold stratified cross-validation (CV). To do this, we split all the data into 5 folds that have approximately equal class membership ratio. Then, we train and test our models 5 times. Each time, we choose one fold to be left out for testing and use the rest for training. For this, we included all 200 of our samples and also added in the pre-existing labels for the 100 test samples so that we could accurately divide the folds and test against any subset of samples. This generated folds of 40 elements each, which resulted in each training step being 160 samples and each testing step being 40 samples.

4 Results

We recorded the results using both simple training (Table 1) and CV (Table 2). Additionally, used these results to compare each classifier pairwise with another classifier, giving us three sets of pairs.

Table 1: Results on given training set and test set

Classifier	Accuracy	True Positive Rate	False Positive Rate
Simple Classifier	0.9300	1.0000	0.1228
Linear SVM	0.9100	0.8727	0.0444
Nonlinear SVM	0.9960	0.9259	0.0000

Table 2: Results with with 5-fold stratified CV

Classifier	Accuracy	True Positive Rate	False Positive Rate
Simple Classifier	0.9150	0.9889	0.1382
Linear SVM	1.0000	1.0000	0.0000
Nonlinear SVM	0.9650	0.9359	0.0000

4.1 Simple Classifier vs Linear SVM

h_0 : The accuracies of the simple classifier and linear SVM are equal on our epilepsy dataset.
h_a : The accuracies of the simple classifier and linear SVM are not equal on our epilepsy dataset.

$\delta = -0.085$ is the average difference in accuracy across all folds
$s = \sqrt{\frac{\sum_{i=1}^{n}(\delta_i - \delta)^2}{n(n-1)}} = 0.01$
$\delta \pm t_{c,n-1} = [-0.10632, -0.06368]$
$0 \notin [-0.10632, -0.06368]$

Since 0 is not in our 95% confidence interval, we reject the null hypothesis and conclude that the accuracies of the simple classifier and linear SVM are not equal on our epilepsy dataset.

4.2 Simple Classifier vs Nonlinear SVM

h_0 : The accuracies of the simple classifier and nonlinear SVM are equal on our epilepsy dataset.
h_a : The accuracies of the simple classifier and nonlinear SVM are not equal on our epilepsy dataset.

$\delta = -0.05$
$s = 0.0.0158$
$0 \notin [-0.08371, -0.01629]$

Since 0 is not in our 95% confidence interval, we reject the null hypothesis and conclude that the accuracies of the simple classifier and nonlinear SVM are not equal on our epilepsy dataset.

4.3 Linear SVM vs Nonlinear SVM

h_0 : The accuracies of the linear SVM and nonlinear SVM are equal on our epilepsy dataset.
h_a : The accuracies of the linear SVM and nonlinear SVM are not equal on our epilepsy dataset.

$\delta = 0.035$
$s = 0.01$
$0 \notin [0.01368, 0.05632]$

Since 0 is not in our 95% confidence interval, we reject the null hypothesis and conclude that the accuracies of the linear SVM and nonlinear SVM are not equal on our epilepsy dataset.

5 Conclusion

Ultimately, we found that the linear SVM performed best when running in our CV tests, with the non-linear SVM close behind. In the simple tests, the non-linear SVM produced the most accurate results.

One point of concern could be noted in that the linear SVM achieved perfect accuracy in all five folds of the CV testing, while it performed the worst of all classifiers during the simple training. This seems to indicate that the linear SVM needed slightly more than just 100 training samples in order to achieve its best results, which the CV tests allowed it to do.

In conclusion, we determined that the linear SVM is likely to produce the best results so long as it is given a sufficient amount of data.

References

[1] H. Firpi, E. Goodman, and J. Echauz, "Epileptic seizure detection by means of genetically programmed artificial features," in *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, GECCO '05, (New York, NY, USA), pp. 461–466, ACM, 2005.