Identifying Epilepsy Using MRIs of the Hippocampus

Andrew Duchi

12/12/08

Background:

Temporal lobe epilepsy is a form of epilepsy rooted in problems in the temporal region of the brain – specifically the amygdala and hippocampus. Among these cases, approximately 65% manifest Mesial Temporal Sclerosis, a loss of neuron cells in the hippocampus. Evidence of this can often be observed by the shrinking of different regions of the hippocampus and changing of intensity of regions as viewed in MRIs.

The goal of this project will be to create an algorithm that can identify pathological and non-pathological hippocampi by analyzing coronal MRIs.

The Model:

In this study we will use a "bag of words" model, which previous studies have shown to be a simple, yet effective, approach to scene and object classification[1][2]. The basics of this method are a decomposition of an image into a certain set of visual features, which we will call "keywords." These "keywords" can be used to create a codebook or dictionary, just as in text analysis. With such a codebook, images can be classified by analyzing which keywords, and how many of each, are included in a given image.

This study will apply this approach to see if it can make finer differentiations than overall object or scene identification. Specifically, we will attempt to use the "bag of words" approach to differentiate a given object type (hippocampus) into two classes (healthy and unhealthy).

The Data:

In this study we had 85 coronal MRIs, 52 from patients with temporal lobe epilepsy and 33 "normal" control MRIs. First, all 24 slices given by each MRI were reviewed and the slice in which the body of the two hippocampi had the greatest contrast to the rest of the brain tissue was selected. Next, all 85 of these slices were cropped into two 100x100 pixel images – a square around the right hippocampus and a square around the left hippocampus. Initial results with this data were poor and it was noted that the hippocampi generally occupied only a small portion of the 100x100 image. Thus, all images were recropped to a size of 40x70 pixels (a smaller size that still included the entirety of each hippocampus) to reduce the amount of noise contributed by information taken from non-hippocampal regions of the brain.

Features and Keywords:

The first set of feature used were generated using the Scale Invariant Feature Transform algorithm (SIFT). The implementation used was created by Andrea Vedaldi of Oxford University. These features were used because they have been effective in previous studies for object identification[4]. Additionally, these local features are relatively resilient against changes in illumination, minor viewpoint change, and random noise[3].

The SIFT algorithm was run on each of the right and left hippocampus images, generating a total of 2742 features for the left hippocampi and 2863 features for the right hippocampi. These features were then translated into keywords by running the k-means

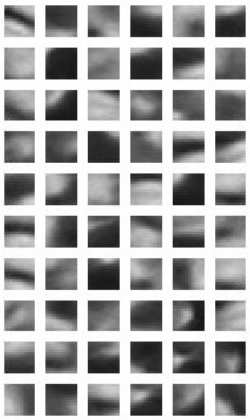
algorithm on the data for clusterings of 10, 20, 30, 40, 80, and 120 keywords for each of the right and left. Each detected feature was then assigned the number of the closest keyword and the number of times each keyword occurred for a given image was recorded to give the features describing the image.

Unfortunately, preliminary analysis suggested that the SIFT features did not results in good performance for classification. Thus, another set of features were selected, specifically, local patches of the images, as were used in a study by L. Fei Fei and P. Perona[2]. The sizes used were 6-by-6 pixel, 9-by-9 pixel, and 12-by-12 pixel patches. These patches were gathered in two ways:

1. Evenly spaced patches of each size were gathered from all images
2. 20 Randomly located patches of each size were selected from each image

These two methods were used in an attempt to ensure that the entirety of each image was considered (evenly spaced), while not unintentionally creating patterns among the patches or leaving out a significant pattern that might fall across a border of patches in the grid (random).

These patches gave 7310 6x6 pixel patches, 4080 9x9 pixel patches, and 2975 12x12 pixel patches for the right hippocampi and for the left hippocampi (14620, 8160, and 5950 total). To eliminate noise cause by differences in brightness (as some MRIs were generally lighter or darker than others) the patches were greyscale normalized to values from 0 to 255. Following this, sets of keywords were extracted from each patch set using the k-means algorithm, grouping the features into clusterings of 20, 40, 60... to 180 keywords. An example of one such codebook is shown at the right – it is the codebook of 60 keywords for 12x12 patches.



The Algorithm:

A standard Naïve Bayes algorithm using Laplace smoothing was trained on the data and used for classification. Given the many different potential codebook sizes, a derivation of the *k*-fold cross validation was used to select the best codebook for training. Specifically, the positive and negative training examples were each split into three approximately equal sized groups. The algorithm was then trained on the 9 possible permutations in which a group from each of the positive and negative examples was withheld for testing. Testing was done in this manner to maintain the ratio of positive to negative examples. The codebook size with the best average estimated generalization error was then chosen and used for training on the entire training set.

After the Naïve Bayes algorithm had been trained on this training set, it was tested on 30% of the data, which was withheld from the training set. As with the k-folds cross validation, the ratio of positive to negative examples was maintained between the training set and test set.

Classifying an unknown image:

To classify an unknown image, one of two approaches is taken depending on the features we are using to classify:

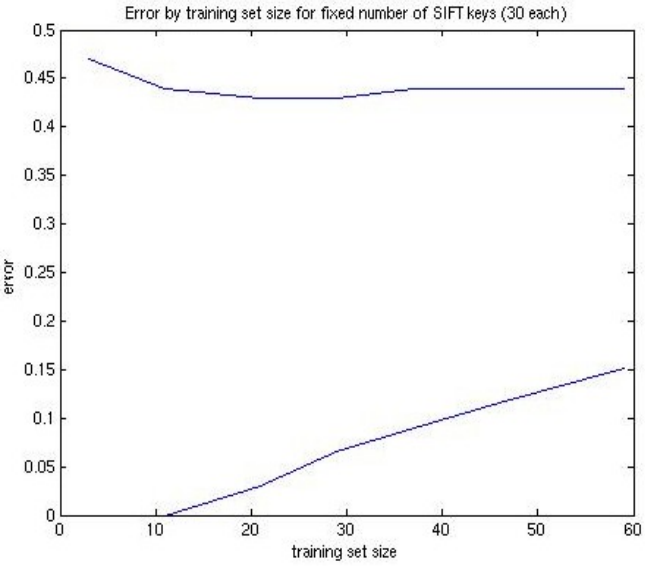
- SIFT: SIFT features are extracted from the image and each feature is matched to the keyword in the codebook that best approximates that feature
- Patches: We iterate pixel by pixel over every possible patch of the appropriate size in the

image. Each patch is then matched to the best approximating keyword from the codebook and thus counts for the keywords in a given image are realized.
The trained Naïve Bayes classifier is then used to classify the unknown image.

Results:

SIFT Feature Results:
Average error (250 random iterations)

- Average training set error: 8.95%
- Average generalization error: 43.77%



To the right is a graph showing the change in training and test set error for the Naïve Bayes algorithm trained on the codebook of 30 test features (for each side). All codebooks had similar error graphs for the SIFT features.

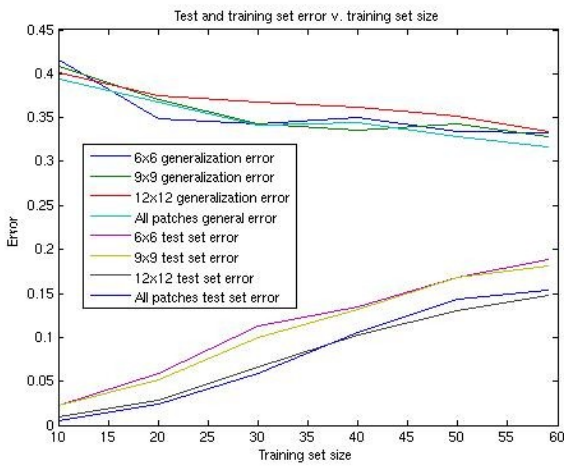
Discussion of SIFT:

The training set error was relatively low for the SIFT features, but the generalization error was unacceptably high. In fact, if we were simply to classify the training set according to which of the classes was more frequent, we would have achieved only 39% expected generalization error. Thus, the SIFT features appear to actually be worse than random.

Combining this with the fact that the generalization error did not decrease with increases of training set size after 30 training examples, while the training set error continued to increase, seems to indicate that the SIFT features were not giving a signal indicating whether or not the hippocampus had MTS.

Patches Results:
Average error (250 random iterations)

Type of descriptor	Predicted generalization error	Training set error
6x6 patches	.3328	.1892
9x9 patches	.3282	.1816
12x12 patches	.3349	.1479
All patches	.3163	.1531



Discussion of patches results:

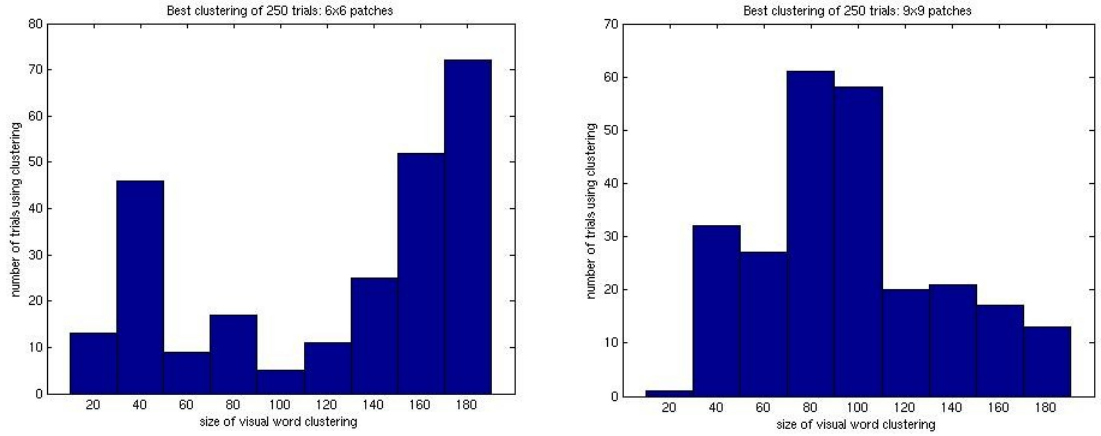
The average estimated generalization error for any of the patch features was better than the SIFT features. Additionally, it was better than the 38.9% error boundary which would have been attained simply by always guessing that the hippocampus was from a patient with TLE.

Unfortunately, the estimated general error was rather high (at best 31.6%) as was the error on the training set for each set of patches. First, let us address the high generalization error.

Graphs of the expected generalization (as in the results section) seem to indicate that we had a case of relatively high variance. This could be helped by addition of more training examples, as the generalization error does not seem to have converged (unfortunately, attaining more training examples was not possible in this study). Another possibility would be to use a smaller set of features, this will be addressed later.

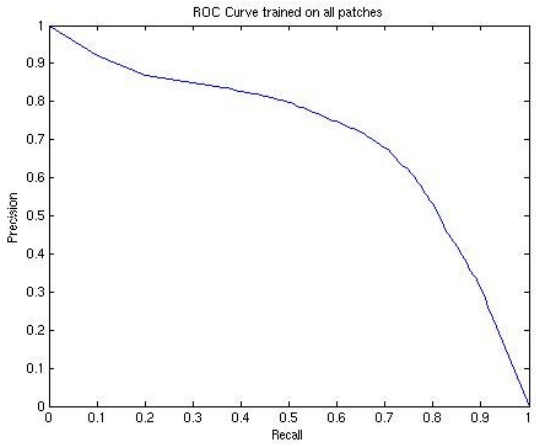
We have an idea that increasing the number of training examples would improve the classifier; however, given how high the training set error was, we might consider this a case of high bias were we to have more training examples. This would imply two solutions: a larger feature set and/or using different features.

The concept of a different sized feature set was addressed by investigating how often each size of codebook was used for classification with each set of patches. Below are the histograms for the 6x6 (left) and 9x9 patches (right):



The histogram for the 6x6 patches seems to indicate that a larger training set may be needed; however, preliminary testing with 200 cluster and 220 cluster training sets actually gave slightly worse expected generalization errors (.3423 and .3511) with only moderately better training set error (.1843 and .1817) over 50 random trials. The 9x9 patches on the other hand seem to be well represented by the given cluster sizes, and the 12x12 patches had even fewer examples than the 9x9 case and also had a median in the middle of our choices for clusterings.

Thus, it seems that in order to get better classification of hippocampi would require a different set of features. This makes sense as one major feature completely ignored by this algorithm is spatial relationships within images, which may be important as the signs of TLE-MTS are primarily manifested in the CA1 and CA3 areas of a hippocampus, and these two areas do have a consistent anatomical spatial relationship that might help to identify a hippocampus as pathological or not.



Finally, an ROC Curve is given (left) for the algorithm trained on the combination of all patches. It was generated by running the algorithm on 250 randomly generated 70-30 splits of the data and testing with different classification thresholds. The area under the ROC Curve was .7126, which was better than the area under the ROC curve for any of the individual patch sizes, which had areas .7124 for 6x6, .6997 for 9x9 and .6918 for 12x12.

Conclusion:

The SIFT features were uninformative for deciding whether a hippocampus was suffering from TLE-MTS. The patches on the other hand did enable us to make a better than random guess about the state of the hippocampus. Additionally, the best decision was reached by using the information from a combination of all patch types.

This indicates that a "bag of words" approach is viable for this type of classification; however, new features considering other aspects of the image, perhaps emphasizing spatial relationships, would be needed to attain acceptable levels of predicted generalization error. Additionally, the training set used in this study was not large enough for the algorithm to fully converge.

References:

[1] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In ECCV Workshop on Statistical Learning in Computer Vision, 2004.

[2] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages 524–531, 2005.

[3] K. Mikolajczyk and C. Schmid. A Performance evaluation of local descriptors. In IEEE Transactions on Pattern Analysis and Machine Learning, volume 27, pages 1615-1630, 2005.

[4] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. International Journal of Computer Vision, volume 73, pages 213-238.

Acknowledgements

I would like to thank Geremy Heitz of the Stanford Computer Science Department for his advising and direction throughout this project. I would also like to thank Ian Cheong and Susanne Mueller of the University of California San Francisco for providing the data set.