A Machine Learning Approach to CT Perfusion Imaging

Lauren Fromm
Computer Science
University of California, Los Angeles
laurenfromm@gmail.com

Arianna Groetsema
Mathematics of Computation
University of California, Los Angeles
agroetsema@ucla.edu

Michelle Lam
Computer Science
University of California, Los Angeles
michelleklam@ucla.edu

Abstract—Computed Tomographic Perfusion (CTP) imaging aids the diagnosis and treatment of AIS patients by providing insight into their cerebral hemodynamics. In this project we utilized existing methods of analyzing MRI images with machine learning and applied it to CT's.

Index Terms—CT scan; machine learning; stroke; perfusion

1 Introduction

Acute ischemic strokes (AIS) account for approximately 85% of all strokes and is characterized by the loss of blood circulation in the brain resulting in a loss of the structural integrity of brain tissue [1]. As one of the leading causes of death, the accurate diagnosis, prompt evaluation, and treatment of AIS patients is crucial for preventing long term disabilities and saving lives [2]. Each hour increase in time from the onset of a stroke to treatment reduces the odds of favorable clinical outcomes by 38% [3]. Thus, improving diagnostic metrics to reduce the time to treatment is crucial for successful intervention and clinical results.

An important diagnostic tool in evaluating cerebral hemodynamics is Computed Tomographic Perfusion (CTP). CTP imaging is important for evaluating stroke patients as it allows the differentiation between irreversibly ischemic infarcted tissue (the infarct core) and potentially salvageable ischemic brain tissue (penumbra). This information is crucial for assessing the kind of treatment a patient needs based on the size and severity of ischemia, generally clot retrieval or thrombolysis [4]. For example, a patient with a small infarct core and large penumbra is more likely to respond positively to reperfusion surgery. CTP imaging is considered a favorable diagnostic technique due to the speed of image acquisition, widespread accessibility, and relative low cost. However, CTP's are commonly interpreted through visual inspection and these observations are prone to user interpretation. We will look to identify and qualify perfusion parameters to identify core infarct and penumbra to bypass our current reliance on subjective visual analysis.

Important parameters commonly discussed in examining CTP scans are: Cerebral Blood Volume (CBV) or the volume of blood within an image's voxel, Cerebral Blood Flow (CBF) or the volume of blood moving through an image voxel in a given time frame, Mean Transit Time (MTT) or the average time it takes for the contrast medium to travel to a given part of the brain, and Time to Peak (TTP) which is defined as the time from the injection of the contrast agent to maximal enhancement. These four parameters are extrapolated by method of deconvolution analysis of CTP data [5].

There has already been research towards using a machine learning approach to perfusion imaging using MRI scans [6]. Instead, we are proposing a model that works with CT scans. CT scans are better for blood vessels and bony material, while MRIs are better for tissues, and since the topic we are interested in involves perfusion, CT scans might prove better because we want to focus our attention on blood vessels. Additionally, CT scans are quicker than MRIs, which is helpful in context, since we want to inspect a scan as quickly as possible after the onset of a stroke. Another advantage of CT scans is patient comfort in that patients are not required to endure the process of an MRI scan which can be uncomfortable, especially for those who are claustrophobic.

A CT scan uses X-ray devices in order to acquire detailed images of internal organs, and can help show features not usually seen with normal X-rays. The donut shaped instrument surrounds the patient and the X-ray spins around, grabbing multiple images from many different angles. The CT also differs from a regular X-ray because it uses a focused higher radiation dose, and this allows much more information to be obtained. The many images that are acquired from a CT scan gives us the ability to see many thin slices of the brain in order to acquire a large set of data. [7].

CT perfusion imaging in particular helps show where blood flows within the brain. The scans are not only painless and noninvasive, but also very fast and accurate, making it very useful when grabbing images to measure blood flow after a stroke. Using machine learning to evaluate CT perfusion scans can be immensely important and valuable to quickly and accurately identify salvageable tissue in a brain after a patient has a stroke [8].

Machine Learning can be applied to the CT perfusion scans to identify if there is tissue at risk and how much tissue is at risk. The perfusion values, CBF, TTP, CBV, and MTT at certain points of a scan can be mapped to tissue on the CT. CT scans show the perfusion of blood over time by taking multiple scans of the brain for a period of time. The machine learning algorithms will learn the perfusion values from the pattern of perfusion at a point in the brain over time. Learning the correlation between the perfusion over time and the perfusion values will predict a result from various models. Models can be built from several different machine learning methods.

Classification of the tissue can be done through models such as nearest neighbors, support vector machines, and ensemble methods. Nearest neighbors predicts a label based on the labels of the points that are closest to it from provided data. A support vector machine separates categories in high dimensions, dividing the categories by a gap that would predict the category based on which side the data point falls on. Ensemble methods include random forests, and Adaboost. Random forests are the creation of decision trees which choose their split at random, and when aggregated together predict with high bias. Adaboost is the aggregation of weak learners, such as small decision trees. It takes the prediction from the weak learners, and applies weights based on samples it needs to produce better predictions. By aggregating these results, it produces a much stronger prediction [9]. Results from various models will vary, validation and tests will show which model performs the best.

2 Methods

The purpose of this project is to determine if we can compute perfusion parameters, specifically rBV, TTP, MTT and rBF, through regression analysis.

2.1 Data

The data used in this project was obtained from Professor Scalzo's team. The data given to us came from 54 patients who had a diagnosis of acute ischemic stroke and had a last known well time within 24 hours of when the CT images were obtained. This data set consists of 5 categories of images: Perfusion, rBV, TTP, MTT and rBF images.

The perfusion images show CT scans as they were taken while blood perfusion was occurring in the brain of the patient. Multiple scans are taken at various times, over various locations of the brain, making each scan 4D. As the blood perfusion is shown in the images, the values at a point change depending on the amount of blood in the section of the brain. To obtain the pattern of perfusion over time, one must observe the value at a specific location of the brain over time. Therefore to get the perfusion over time, one must take a pixel value from a specific point in the scan, in a specific slice location, and the time the scan was taken. The pixel values must be organized in order of time that the CTs were taken to obtain the values of blood flow in the brain over time.

The rBV, TTP, MTT, and rBF images have values corresponding to the location in the perfusion image. Each of these outputs are 3D since they correspond to x, y and z locations in the brain. Our goal was to take the 4D perfusion input, resample the time curve using interpolation, and map the input array to an output value. We then would pass these inptus and outputs to different training models using the scikit-learn tool in Python.

2.2 Preparing the Dataset

Our first goal was to randomly choose points at (x, y, z) to sample from every patient, and then populate an array with our input Perfusion values of form of 100 interpolated points based on the values of perfusion over time in an array as well as four output arrays filled with values from the TPP, rBF, rBV, and MTT data.

One of our main tasks was using interpolation to resample the time curve of each image. Since different timing increments were used while capturing the perfusion data of every patient, we wanted to use interpolation to make timing consistent so we could have more reliable data. In order to obtain information such as timing about each image, we used pydicom.dcmread which helped extract the meta-information that we cared about for each image. We used the 'Aquisition Time' value in order to acquire details needed for interpolation to get regular time increments for each patient.

To get the random input points, we used the randint() function and kept track of which points we use, so we never used the same point twice while looking at one patient. We used the 'Slice Location' parameter from the meta-information to give us a z value, since the slice location shows where the scan is on the z axis in the brain. Once we had a random (x, y, z) point in a patient, we find the TPP, rBF, rBV, and MTT values at this point and record these values in seperate arrays.

After collecting all of the input and output values in the 54 patients, it was time to train and test our model. We decided to use about 60% of the data for training and 40% for testing. Since we obtained the data randomly, we just split our large data array of all points collected from each patient into two arrays, 34/54 of the array for training, and 20/54 of the array for testing. We did this for each array we had, which included the one input array and the 4 output arrays holding values for TPP, rBF, rBV and MTT. Next, we needaed to use machine learning models to train and test the data.

2.3 Training the Model

We used five machine learning models: K-Nearest-Neighbors, SVM using a linear model, SVM using a radial basis function model, Adaboost, and Random Forests Decision Trees. Group leave one out cross validation is used on the models to determine the best parameters to use on the data. It is necessary to group the data based on the patient to avoid allowing the data from one patient to be in both the training and test data, decreasing the bias. Leave one out cross validation allows us to use the training data to see which parameters produce the best results to be used on the test set.

2.3.1 K-Nearest-Neighbors (KNN)

K-Nearest-Neighbors uses instance based learning by storing the instances of the training data. The classification is based on majority vote of the k nearest neighbors in the training set data. Cross validation was used to choose the best k, number of neighbors to use for the best performance. The values of k that were tested depended on the number of values in the set.

2.3.2 Support Vector Machines (SVM) using Linear Model

SVM finds the optimal hyperplane that categorizes data such that misclassification rate is minimized and the sum of the distances of sample points from the hyperplane are maximized. Thus we must find α such that:

$$argmin_{\alpha} \quad \frac{1}{2}\alpha^T Q\alpha - \vec{1}^T \tag{1}$$

where $y^T\alpha = 0$ and $0 \leq \alpha_i \leq C$ for $i = 1, ..., n$. C is a constant that minimizes the penalty on error during the minimization of the miscalculation rate, $\vec{1}$ is a vector of ones, and Q is the matrix defined by: $Q_{ij} = y_i y_j K(x_i, x_j)$. $K(x_i, x_j)$ is the linear kernel defined as:

$$K(x_i, x_j) =< x_i, x_j > \tag{2}$$

2.3.3 Support Vector Machines (SVM) using Radial Basis Function Model

For SVM with the rbf Model, the matrix Q_{ij} is defined the same as above except $K(x_i, x_j)$ is the rbf kernel defined as:

$$K(x_i, x_j) = exp(-\frac{||x_i - x_j||^2}{2\sigma^2}) \tag{3}$$

where $||x_i - xj||^2$ is the squared Euclidean distance between the points x_i and x_j and σ is the standard deviation of the kernel. This is equivalent to:

$$K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2) \tag{4}$$

where $\gamma = \frac{1}{2\sigma^2} > 0$.

Parameters C and γ were tested with cross validation for the rbf kernel. C was tested at 0.01, 0.1, 1, and 10. γ was tested at 0.15, 0.1, 0.05, and 0.01.

2.3.4 Adaboost

Adaboost uses a collection of weak learners that are determined using the training set. These weak learners include small decision trees. The predictions from these weak learners are combined using weighted majority vote, to determine what the test data should be predicted to. The weights of the learners are determined when fitting to the training data, re-weighting the learners depending on what areas need more help. The n-estimators parameter is determined with cross validation, the values tested are 1, 7, 13, 19, 25.

2.3.5 Random Forests

The model is built using trees that are created from samples of the training set. The random forests model does not calculate which is the best split, but splits the tree randomly. By splitting randomly, the bias of the model is increased, but because of averaging the variance decreases. The increase in bias generally results in a better model overall.

To test the variability of the accuracy of the machine learning models to see how much they depend on the amount of data, the models were tested on the 54 folders selecting 50 and 100 random points.

2.3.6 Metrics of Accuracy

We calculated the accuracy of our models using normalized root-mean-square error (NRMSE). The numerator of NRSME is defined by the root-mean-squared-error (RSME):

$$RSME = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \tag{5}$$

In this, n is the number of samples of data, y_i is the true value of the point, and $\hat{y}_i$ is the value that was predicted by the machine learning algorithm. The denominator for

NRSME is $y_{max} - y_{min}$. Where y_{max} is the maximum value in the true values, and y_{min} is the minimum value in the true values.

$$NRSME = \frac{RSME}{y_{max} - y_{min}} \tag{6}$$

3 Results

The various machine learning algorithms were run twice. The first time was on a data set that was created using 50 random points from the perfusion images and the corresponding points from the perfusion values. The second data set was created using 100 random points from the perfusion images and the corresponding points from the perfusion values. A 200 random point data set was also intended to be run, but the leave on out cross validation proved to take too long. The accuracy score was calculated by sklearn metrics package, as well as the mean-squared-error, which was used in part to calculate the NRSME. Accuracy was used to choose the best hyperparameters to use, and after the hyperparameters were chosen for each machine learning algorithm, the NRSME was measured to show the error of the machine learning model.

The cross validation on the K-Nearest-Neighbors model chose 43 for the number of k-neighbors as the best hyperparameter in the 50 randomly chosen point data set, and 86 neighbors in the 100 randomly chosen data set. The NRSME is about the same for both data sets for TTP, but rBF, rBV and MTT all have lower NRSME in the 100 randomly chosen point data set. Overall, our model exemplfies accuracy in a low NRMSE of about 0.002.

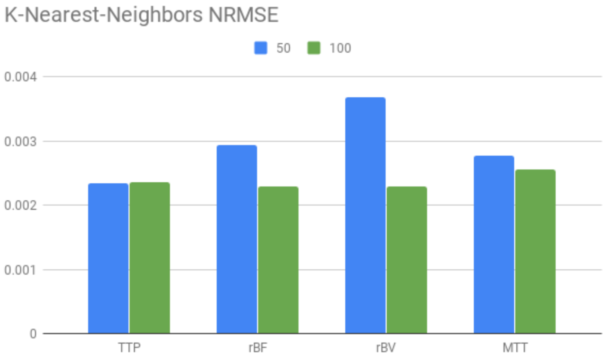


Fig. 1. Results of the normalized root mean squared error for the K-Nearest-Neighbors model

The SVM linear model did not have any hyperparameters to be set and therefore, there was no cross validation run on it. The error on the 100 randomly chosen point data set was lower in all four cases in comparison to the 50 randomly chosen point data set. An NRSME of about 0.0025 is observed for the SVM Linear model in the 100 randomly chosen point data set.

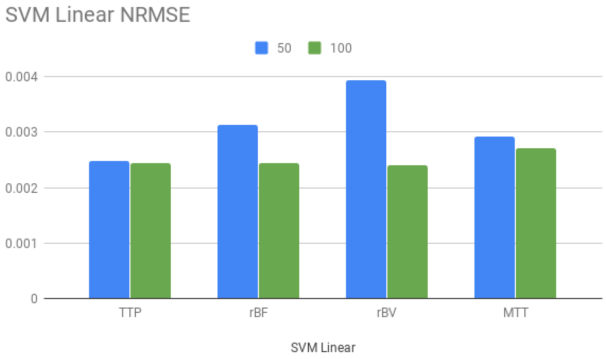


Fig. 2. Results of the normalized root mean squared error for the SVM Linear model

The SVM RBF model had two hyperparameters to be chosen, the constant C and gamma. The hyperparameters chosen were $C = 0.01$ and $\gamma = 0.15$ in both data sets. The NRSME was compared to the automatically chosen values generated by sklearn, which yielded the same NRSME, showing that either the same or a very similar model was used. The 100 randomly chosen point data set had lower NRSME in all the perfusion values predicted. The NRSME among all values were about 0.0023 for the 100 randomly chosen point data set.

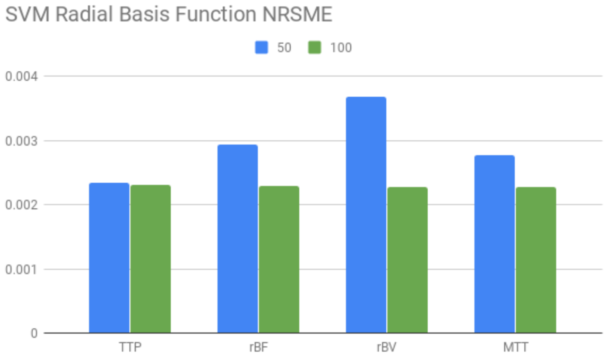


Fig. 3. Results of the normalized root mean squared error for the SVM Radial Basis Function model

The hyperparameter to be chosen in the Adaboost model was the n-estimators. The number chosen though cross validation for both data sets was 1 estimator. The NRSME for Adaboost was about 0.0023 for the 100 randomly chosen point data set.

The cross validation was only performed on the 50 randomly chosen point data set which chose 17 estimators for its n-estimators. The 100 randomly chosen point

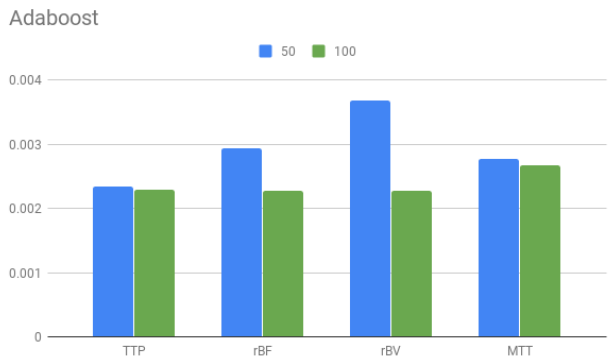
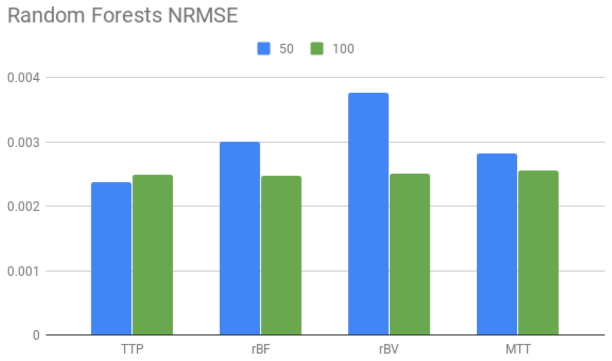


Fig. 4. Results of the normalized root mean squared error for the Adaboost model

data set yielded a memory error, and could not fit for n greater than 5. Therefore, the n-estimator tested was 3. The NRSME for Random Forest was abour 0.0025 for the 100 randomly chosen point data set.



z

Fig. 5. Results of the normalized root mean squared error for the Random Forests model

4 Discussion

Present methods of treatment for acute ischemic strokes involves the reperfusion of ischemic tissue. This method relies on neuroimaging techniques and has led to many improvements in the ability for doctors to treat patients. However, imaging techniques have a number of challenges some of them being: 1) limited availability of medical imaging, 2) time it takes to interpret an image and make a diagnosis , and 3) observation error. In order to address these issues, we have generated machine learning models that predict the perfusion parameters to help better understand a patients conditions.

The Support Vector Machine using Radial Basis Function model seemed to perform the most consistently with the lowest NRSME. This can be seen in the graph below. This shows that from the machine learning algorithms we performed, it is the most accurate.

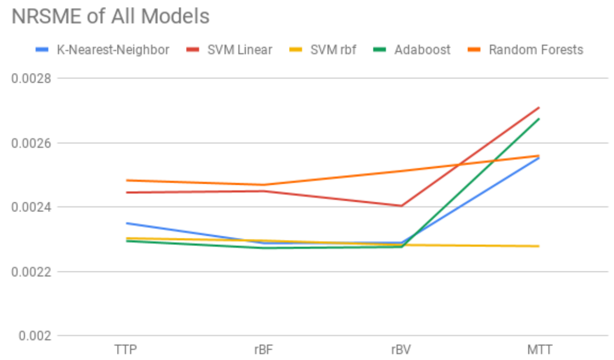


Fig. 6. Results of the normalized root mean squared error for all machine learning models on the 100 randomly chosen point data set.

The results show that in almost all cases, the error is lower in the 100 random point data set compared to the 50 point data set. Therefore, it is believed that with more training data, the model would be able to better predict the perfusion values.

As the values of error were fairly low among all models, it shows that it is possible to predict the perfusion values of TTP, rBF, rBV, and MTT using machine learning models using CT scans. As the training set size increases, the accuracy should increase as the error has shown to decrease.

References

[1] S. S. Prabal Deb and K. Hassan, “Pathophysiologic mechanisms of acute ischemic stroke: An overview with emphasis on therapeutic significance beyond thrombolysis,” *Pathophysiology*, vol. 17, no. 3, p. 197218, June 2010. [Online]. Available: https://www.pathophysiologyjournal.com/article/S0928-4680(09)00136-9/pdf

[2] K. K. Murphy SL, Xu JQ and A. E., “Mortality in the united states, 2017,” *NCHS Data Brief*, no. 328, November 2018. [Online]. Available: https://www.cdc.gov/nchs/data/databriefs/db328-h.pdf

[3] B. K. Menon, M. A. Almekhlafi, V. M. Pereira, J. Gralla, A. Bonafe, A. Davalos, R. Chapot, and M. Goyal, “Optimal workflow and process-based performance measures for endovascular therapy in acute ischemic stroke,” *Stroke*, vol. 45, no. 7, pp. 2024–2029, 2014. [Online]. Available: https://www.ahajournals.org/doi/abs/10.1161/STROKEAHA.114.005050

[4] M. Goyal, B. K. Menon, and C. P. Derdeyn, “Perfusion imaging in acute ischemic stroke: Let us improve the science before changing clinical practice,” *Radiology*, vol. 266, no. 1, pp. 16–21, 2013, pMID: 23264523. [Online]. Available: https://doi.org/10.1148/radiol.12112134

[5] Y. W. L. Andrew Mark Allmendinger, Elizabeth R. Tang and V. Spektor, “Imaging of stroke: Part 1, perfusion ctoverview of imaging technique, interpretation pearls, and common pitfalls,” *American Journal of Roentgenology*, vol. 198, pp. 52–62, January 2012. [Online]. Available: https://doi.org/10.2214/AJR.10.7255

[6] R. W. D. L. Richard McKinley, Fan Hung and F. Scalzo, “A machine learning approach to perfusion imaging with dynamic susceptibility contrast mr,” *Frontiers in Neurology*, vol. 9, no. 717, September 2018. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fneur.2018.00717/full

[7] C. Nordqvist, “Ct scan or cat scan: How does it work?” *Medical News Today*, vol. 12, July 2018. [Online]. Available: https://www.medicalnewstoday.com/articles/153201.php

[8] S. C. Wing and H. S. Markus, “Interpreting ct perfusion in stroke,” *Practical Neurology*, vol. 19, no. 2, pp. 136–142, 2019. [Online]. Available: https://pn.bmj.com/content/19/2/136

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.