

Improving Positron Emission Tomography Imaging with Machine Learning

David Fan-Chung Hsu (fcdh@stanford.edu), CS 229 Fall 2014-15

1. Introduction and Motivation

High-resolution Positron Emission Tomography (PET) systems can image the specific biological activity of malignant tumors, making it useful for cancer screening. In PET imaging, the patient is first injected with a radioactively labeled molecule that serves as a contrast agent. The most common agent is ^{18}F -Fluorodeoxyglucose (FDG), an analog of the sugar glucose, which is rapidly taken in by cancer cells. Every ^{18}F atom that undergoes positron decay in the cell creates two high-energy photons simultaneously, that travel in opposite directions. When both of these photons in any given decay event are picked up by the system's detectors, the positron decay is assumed to have occurred on the line joining these two detectors, called the line-of-response (LOR). Over time, the lines that join the detected high-energy photon pairs accumulate in image space, and, by solving a mathematical inverse problem, an image representing the biodistribution of the contrast agent is reconstructed. The intersection of the lines, which show up as bright spots in the image, help doctors determine where the tumors are likely located in the body. We call these photon pairs "true pairs" for the purpose of this project. The illustration on the right shows an example of a "true pair".

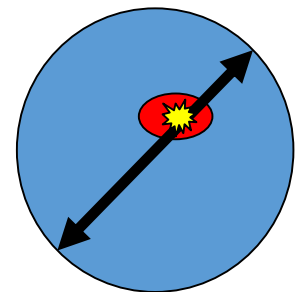


Figure 1: True Pair

A problem arises when photon pairs are misidentified, leading to the event being placed on an incorrect detector response line. The largest cause of misidentification is the low detector efficiency in the system, combined with high decay rate, meaning that often 2 non-correlated photons that originated from different decays are detected. Misidentification increases the background in the images, leading to loss of tumor contrast and errors in its intensity seen in the image. We call all misidentified pairings as "false pairs" for this project. The illustration on the right shows a "false pair", where the tumor is thought to be located along the dotted line, because of 2 decays which occurred simultaneously.

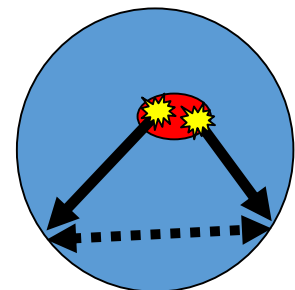


Figure 2: False Pair

Conventional algorithms do not have an easy way of distinguishing false pairings from true pairings in PET imaging. The primary goal of this project is to use machine learning to find structures that are embedded in the data, which may act as predictors for false pairings versus true pairings. In imaging statistics, the contrast-to-noise ratio (CNR) of the image is generally considered as the primary metric which determines the detectability of a tumor when examined by a radiologist. In terms of machine learning, contrast-to-noise ratio is represented by the proportion of false-positives in the final dataset used to reconstruct the image. By using logistic regression, naïve bayes, and support vector machines, I aim to classify true pairings and false pairings with a higher accuracy than current algorithms, in order to improve the CNR of the final image set.

2. Simulation Data Acquisition and Pre-Processing

The problem addressed by this project can be tackled by supervised learning. To achieve this, we perform a simulation of our system using GATE, the most accurate and comprehensive simulation program for tomography-based imaging systems. The simulation outputs photon pairings and the ground truth of whether or not these are true or false pairings. With the ground truth, we can use supervised learning to train our model, and perform testing using our model. In order to introduce hardware-related uncertainties (such as jitter or energy blurring into the data), we pre-process the simulation dataset with various programs that are intended to add in uncertainties in order to better mimic real-world data taken with our system. I ended up with 25000 pair samples, and the breakdown is quantified in the table below. We see that the ratio of false to true pairs is very small.

Table 1: Summary of Simulation Data

	Training	Testing
Number of Samples	12000	13000
Number of True Pairs	11210	12133
Number of False Pairs	790	867

3. Features

The raw features used in the supervised learning originate from the GATE simulation program. There are 8 main features that are generated by the simulation with each pairing: 6 cartesian coordinates X1, X2, Y1, Y2, Z1, Z2, as well as two energy parameters E1 and E2. The positional information is relevant in learning because typical PET scanners have sensitivity and efficiency profiles that are not uniform over the entire spatial imaging space. This means that there is some dependency of the position of the pairings. The energy of the pair is relevant, because “false pairs” can be correlated with some loss of energy, during scattering events. Testing using these raw features, both continuous and discretized, did not yield good results, so a new set of position features were generated by re-mapping the positions, in order to better approximate the spatial position of the pair over the field of view. The position coordinates are converted from 6 features into 3, representing $(X1+X2)/2$, $(Y1+Y2)/2$, and $(Z1+Z2)/2$, which are the centroids of X, Y, and Z positions of the pair of coordinates. Compared to the situation when a linear mapping of cartesian coordinates of both endpoints of the are used, a centroid location gives much more intuitive information regarding where the response-line that is drawn between the two points of a pair will pass through in image space. The horizontal and vertical angular information of the pair is also derived, using $\text{Tan}^{-1}[(Y2-Y1)/(X2-X1)]$ and $\text{Tan}^{-1}[(Z2-Z1)/(X2-X1)]$. In continuous-space, there are a total of 7 generated features used in our learning. Discretization of the feature space is performed to turn these 7 features into 1252 binary-valued bins for testing.

Table 2: Summary of the Number of Features

	Raw Features	Discretized Raw	Generated Features	Discretized Generated
Position	6	552	5	456
Energy	2	700	2	700

4. Learning Process and Results

We chose to use logistic regression and support vector machines with both continuous-valued and discretized features, while we used naïve Bayes, using a multinomial event model, with the discretized features only. Logistic regression and naïve Bayes are both implemented in MATLAB using code I wrote myself, while SVM was performed using the LibLinear library provided at the link listed in the reference. Since the algorithms we used are all unmodified from what was covered in lecture, I will leave out the equations in this report as they are unnecessary.

One very important point to note for this project is that the usual “training error” and “testing error” equations do not apply in a straightforward fashion for this project. For radiology images, absolute magnitude in the image pixels do not convey any information, but rather it is the **contrast** in the images which provides the information regarding tumor locations. Therefore, the magnitude of the error does not matter, but rather it is the ratio of errors (or “false positives”) to true positives that is important. I define below the equation for **Contrast-to-Noise Ratio (CNR)**, which is typically used as the primary figure-of-merit in evaluating imaging performance, to evaluate the performance of various algorithms in this project. It should be noted that the CNR takes into account all 4 main categories of classified data: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Therefore, an algorithm with a low FP rate that throws out a lot of TPs by categorizing them as FNs, will have a worse CNR than one which has a higher FP rate but keeps many more of the TPs.

$$\text{CNR} = \frac{(\text{True Positives}) - (\text{False Positives})}{(\text{True Positives})}$$

The baseline for all comparisons is the “conventional” algorithm: the “majority rules” algorithm which assumes that all pairs are true pairs. The extreme skew in the ratio of true pairs to false pairs means that “majority rules” already yields a high percentage of accuracy in predicting true pairs. The error percentage is calculated in each test case, which is the ratio of false positives to true positives. Looking at table 1, we can see that we can already achieve a baseline CNR of 92.8% with majority rules. The percentage of improvement in CNR is defined as:

$$\% \text{ CNR Improvement} = \frac{(\text{CNR}_{\text{algorithm}} - \text{CNR}_{\text{baseline}})}{\text{CNR}_{\text{baseline}}}$$

4.1 Using Raw Features

The first step I tried was to simply use the learning algorithms on the raw features, both continuous and discretized. The results are summarized in Table 4.1 below. We can see the performance is marginally better than baseline, but not by much.

4.2 Using Generated Features

Using the generated features defined in section 3, we observe much better results, which are summarized in the table 4.2 below. In all situations, we used 30 iterations for logistic regression. We see that logistic regression of the discretized generated features offers the best improvement in CNR.

Table 4.1: Raw Features Learning Summary

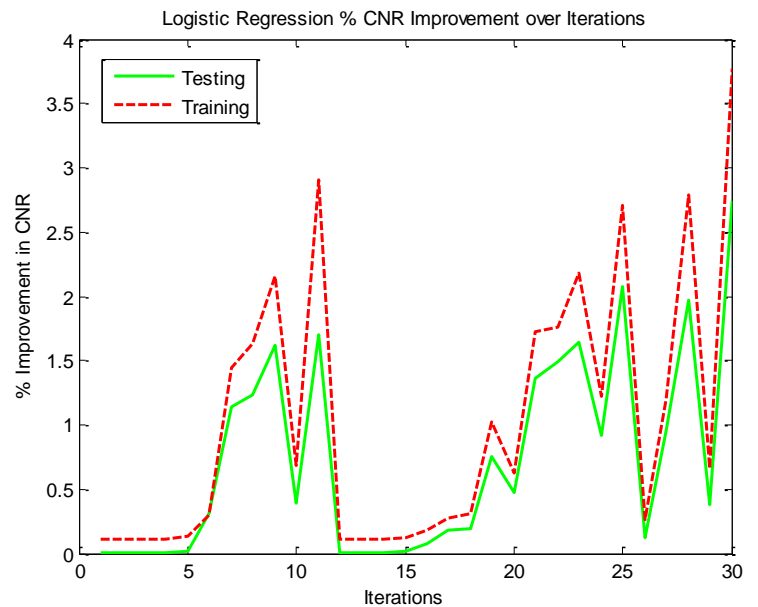
	Continuous		Discretized		
	Logistic Reg.	SVM	Logistic Reg.	Naïve Bayes	SVM
True Positives	6146	12133	8405	12005	12123
False Positive	432	861	550	848	865
Error %	7.0%	7.1%	6.6%	7.1%	7.1%
% CNR Improve	+0.2%	+0.1%	+0.7%	+0.1%	+0.1%

Table 4.2: Generated Features Learning Summary

	Continuous		Discretized		
	Logistic Reg.	SVM	Logistic Reg.	Naïve Bayes	SVM
True Positives	8641	12133	9023	12029	12113
False Positive	544	867	415	699	742
Error %	6.3%	7.2%	4.6%	5.8%	6.1%
% CNR Improv.	+1.0%	+0%	+2.8%	+1.5%	+1.2%

5. Discussion

To put the numbers in perspective, the best result achieved, +2.8% in CNR, means the CNR increased from 92.8% to 95.4%. This means that the learning algorithm was able to reduce the proportion of false pairs over all data by almost half, from 7.2% down to 4.6%. I plotted the learning curve of the best-case logistic regression (highlighted in green above in the table), and the learning curve is shown on the right. We can see that in all cases, the training curve remains above the testing curve, meaning it performs better (larger improvement). We also observe a characteristic “zig-zagging” during training of the logistic regression model, which I have not been able to explain fully. While the general trend of the learning curve is for the improvement to slowly increase with the number of iterations, it seems that the contrast-to-noise ratio does not increase monotonically with the number of iterations. Another interesting observation is that, in all cases, it seems that SVMs perform the worse. I believe this might be due to the high imbalance between true positives and true negatives (where the true negatives constitute less than 8% of true positives).



The improvement offered by the re-mapping of raw features to generate a new set of features was very clear. I believe this improvement comes about because of the intrinsic nature of PET imaging, in that the locations of the lines drawn through imaging space are more important than the positions of the actual detectors. For example, one detector, by itself, offers no information whatsoever regarding the position of the cancerous tumor, because PET requires two end points in order to draw a line through space. Therefore, it makes logical sense that, during learning processes, we can re-map the two end points to new features which represent the line connecting the two end points. In my implementation, I chose to re-map the end points to a location centroid and angles representing the direction of the line, but there might be even better ways to map the end points.

6. Conclusion and Future Work

In conclusion, the work in this project resulted in an improvement of +2.8% in the contrast-to-noise ratio in the PET dataset, equivalent to a 36% reduction of identification error, from 7.2% from 4.6%. We find that, using a generated feature set composing of energy, positional centroids, and angles between detectors, along with appropriate discretization of feature space, we are able to achieve the best results.

The next steps that I am planning on taking with this project is to extend the learning to both unsupervised learning, as well as deep learning. I have written a grant application, using ideas derived from the work presented here, to use deep learning on PET data. The fact that logistic regression performed so well on this dataset is encouraging, since it points to the potential success of using logistic activation functions in the deep learning algorithm. Unsupervised learning would allow us to learn from actual measurement data. These data contains all the effects of various uncertainties in the system, but does not offer ground-truth labeling of data. A successful work in unsupervised learning would have the ability to boost the capabilities of PET systems all around the world without requiring hardware upgrades.

7. Acknowledgements

The author would like to acknowledge the help of Wenbo Zhang in explaining the use of GATE to simulate PET systems.

8. References

GATE Simulation Software:

<http://www.opengatecollaboration.org/>

LibLinear Library for SVM:

<http://www.csie.ntu.edu.tw/~cjlin/liblinear/>