Predictive Validity of a Robotic Surgery Simulator

Anirudh Pasupuleti **SUID:** 05833435 **SCPD#:** X120939

Introduction

The primary goal of this project is to answer the question: does performance/training with this simulator transfer to improvements in clinical robotic-surgery on real patients?
Briefly, the role of surgical simulation is to train/evaluate Residents (medical students) before stepping into the shoes of robotic-surgeons. So, this project focuses most on evaluating the simulator that evaluates doctors. In sense, we are trying to predict the validity of a simulator (given) by collecting the data it provides after a group of experts (robotic-surgery) perform on it. So we will be predicting the R-score (which is the governing score for graduating a Resident into an operation theatre), from a given collection of R-scores, which is in turn dependent on a set of governing performance related metrics, e.g. Time taken, Camera Usage, Clutch Usage, Left pinches, Right pinches, Cauterizer Usage and etc.

Data and Features

The data is available at my current employer's database. I work for Simulated Surgical Systems. We build the RoSS (Robotic Surgery Simulator). It is a start-up company in NY and this simulator is designed for training doctors for Intuitive Surgical's Da Vinci Surgical System. So, we already have about 20 simulators spread around USA and a couple in Europe. So, we have collected a huge set of data of experts practicing on it and teaching their residents using the RoSS.

Features	Examples
Time taken (sec)	100
Camera Usage (count)	12
Clutch Usage (count)	5
Left grasps (count)	7
Right grasps (count)	9
Distance moved by left tool (mm)	1840
Distance moved by right tool (mm)	2014

I have spent time on selecting Features and generating them. This helped me in consolidating the useful data from unwanted features. In our model, the predicting problem is converted into multiclass classification problem. We will Naïve Bayes as our prototype model initially for training and testing data. We will compare the performance of the technique by employing another technique, the Logistic Regression, and then will try to achieve the final feature and model.

Sentiment Polarity	Sentiment Score	Examples
Strong Positive	+2	Expert
Weak Positive	+1	Challenger

Neutral	0	Competitor
Weak Negative	-1	Trainee
Strong Negative	-2	Novice

In our experiments, the Time taken is the strongest individual indicator of a user's R-Score. In the final report, I will attach the plot for the X-Y scatter graph of Time taken and R-Score, which helps us in finding their correlation, hence using it as the baseline result. In sense, all reasonable models should be able to achieve at least this correlation.

Model 1: Linear Regression

In our first model, we used a standard least gradient descent, which we implemented in C for efficiency. Once we had trained a set of feature weights, we could then generate the R-Score prediction

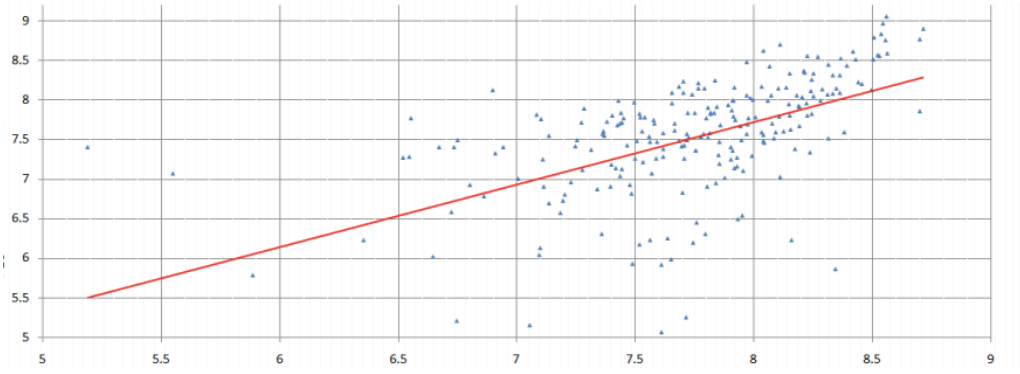
R-Score = $\theta 0$ + $\theta 1$ * F1 + $\theta 2 * F2$ + … + $\theta n * Fn$, where θi are the weights, Fi are the features and n is the number of features

To measure the "goodness" of our results, we looked at the correlation between our predicted metric values and the actual metric values, as was done in other papers [1][2]. As alternative measures to interpret our results, we also considered other metrics such as mean absolute percentage error (MAPE) and symmetric mean absolute percentage error (SMAPE).

MAPE is not the best metric for our work because the error is unbounded metric value (Time) of 1000 sec for a given surgery exercise that was performed in 100 sec, then error would be 900%, skewing any average we would take over all test examples. In compensation to this, we also tried SMAPE, which returns error values from 0 to 100%. However neither metric gave consistent, explainable results.
Below are the correlation results we found for each of our feature sets, using 90% of our data in training

In our first model, we used a standard least-squares linear regression. To do this, we used stochastic gradient descent, which we implemented in C for efficiency. Once we had trained a set of feature weights, we could then generate R-Score predictions as follows:

	Simple Features	Complex Features	Sentiment Features
Test Data Set	0.7198 correlation	0.7428 correlation	0.7506 correlation
Training Data Set	0.7291 correlation	0.8120 correlation	0.8139 correlation



Predicted and Actual R-Score (LogX-LogY Scatter)

Model 2: Classification by Logistic Regression

As a second model, we also tried classification by standard L1-regularized logistic regression. We chose this method because it generated a multi-class model with linear weights, most directly comparable to the feature weights given by linear regression. To define our classes, we drew a histogram of surgery exercise to create 5 different buckets for prediction as shown below:

The first bucket includes the lowest 20% of the R-Score distribution and the last bucket includes the highest 20%.

Buckets	Bucket 1	Bucket 2	Bucket 3	Bucket 4	Bucket 5
R-Score	0 to 1.4	1.4 to 11.1	11.2 to 29.2	29.2 to 69.6	69.6

The procedure for using logistic regression was fairly similar to that of linear regression; the difference being that we now use labeled buckets as our y-values (instead of real-valued R-scores) and pass the data to liblinear to build the model for classification. This model gave the following accuracy results on our 10% test set:

In general, none of these accuracy figures were as high as we had hoped, indicating that this kind of classification was not the right approach to the problem.

	Simple Features	Complex Features	Sentiment Features
Test Data Set	49.05%	50.14%	49.40%

Final Comparison:

Having developed these different models, we need some way of comparing these results. For this project, I implemented two methods, we map the results from linear regression into the five bucket classes from logistic regression. So, in order to do this, we take the real-valued outputs from our linear regression model, assign labels to them according to the buckets into which they fall, and check these corresponding values in the same bucket as those of the actual R-Score.

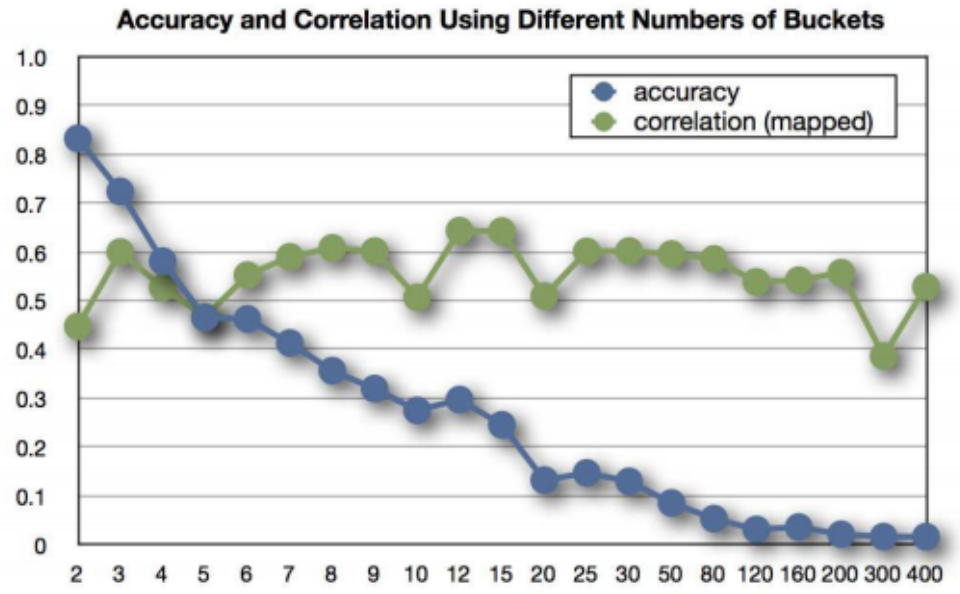
	Simple Features	Complex Features	Sentiment Features
Test Data Set	49.56%	45.14%	41.04%

These numbers decrease with additional features, likely because of increased variance (that is, some over-fitting on the training set-- further discussion of this in the following section). However, all are roughly comparable to the 49% accuracy achieved by logistic regression on the classification problem, showing that linear regression is almost as good at the task of classification as logistic regression, the algorithm dedicated to classification.

In the second method, instead of mapping from real values to buckets, we map from our five buckets to real values. To do this, we first find the average R-scores classified into each bucket in logistic regression. Then, instead of generating class labels from logistic regression, we can use the corresponding averages instead, giving us real-valued output from the classifier. Thus the resulting correlation scores are as follows:

	Simple Features	Complex Features	Sentiment Features
Test Data Set	0.5217 correlation	0.5675 correlation	0.5601 correlation

None of these approach the 0.75 range of correlation seen with linear regression, much less the 0.63 baseline correlation using a simulation exercise's time taken alone. Much of this has to do with the fact that logistic regression can only generate one of five distinct values, so we thought we might experiment with different number of buckets. Our expectation was that accuracy would consistently decrease as number of buckets increased, but that correlation would have some optimal point where the (positive) granularity of having smaller-ranged buckets balanced with the (negative) trend toward fewer training examples per bucket. We were surprised to find that while accuracy decreased, correlation remained fairly constant, as shown in the following figure:



It appears that, in terms of the correlation measure, having fewer training examples per bucket was evenly offset by the greater granularity of having smaller-ranged buckets.

Conclusion

We framed this problem as both a regression and classification problem because we were not sure which would provide a better result; as such, we implemented both and devised methods to compare them. In general, we found that linear regression works almost as well as logistic regression for classification on our data, while having a much better correlation with the actual gross revenues. In general, we found that the features we used (simple numeric, text, and sentiment features) were insufficient to make strong predictions of R-Score. For future work, besides using different feature sets, we might consider using better regularization on linear regression in order to provide a more rigorous safeguard against high-variance models, as we consistently observed decreases in linear regression's test accuracy with increasing numbers of features.

Another, fundamentally different, data set that might be useful in predicting a simulator's validity would be social graph data: using such data, we could analyze the characteristics of how a trainee's performance propagates through real surgery, as well as characteristics of the propagation tree, such as its speed and extent over time. The propagation speed of a trainee would represent performance expectation to see that a simulation exercise, which we expect will be directly related to its R-score.

Moreover, we looked at our feature weights for text-based features, and from them extracted the highest- and lowest-weighted features in right hand movement, left hand movement, and clutch pedal usage, camera usage.

References

1) Stegemann AP, Kesavadas T, Rehman S, Sharif M, Rao A, DuPont N, Shi Y, Wilding,G, Hassett J, and Guru K, "Development, Implementation and Validation of a Simulation-Based Curriculum for Robot-Assisted Surgery" Presented at the AUA poster session, May 19-23, 2012 Atlanta, GA.
2) Kesavadas T, Stegemann A, Sathyaseelan, G, Chowriappa A, Srimathveeravalli G, Seixas-Mikelus S, Chandrasekhar R, Wilding G, and Guru K. "Validation of Robotic Surgery Simulator (RoSS)", Stud Health Technol Inform. 2011;163:274-6.
3) Zorn K, and Gautam G, "Training, Credentialing, and Hospital Privileging for Robotic Urological Surgery" In Robotics in Genitourinary Surgery 2011, Part 2, PP 169-181.
4) Su D, and Barone J "Initial Experience with the ROSS Robotic Simulator in Residency Training" moderated poster, AUA 2011
5) Seixas-Mikelus SA, Stegemann AP, Kesavadas T, Srimathveeravalli G, Sathyaseelan G, Chandrasekhar R, Wilding GE, Peabody JO, and Guru KA. "Content validation of a novel robotic surgical simulator", British Journal of Urology, 2011 2011 Apr;107(7):1130-5