#MoralMachines: Developing a Crowdsourced Moral Framework for Autonomous Vehicle Decisions

Tara Balakrishnan
taragb@stanford.edu

Jenny Chen
jchen15@stanford.edu

Tulsee Doshi
tdoshi@stanford.edu

***Abstract*— With the advent of artificial intelligence algorithms, computer systems are capable of making decisions independent of human control. It is therefore important for computers to have a moral framework that dictates their response to a particular action or situation. In this paper, we investigate the following questions: Given an ethical dilemma, can we build a crowdsourced moral framework? Can our trained model provide a response that mimics human behavior and judgment with high accuracy? What features impact an individuals moral decision when faced with an ethical dilemma? We address these questions via a modified Trolley Problem (a common ethical thought experiment), placed in the context of autonomous vehicles. We survey approximately 315 students to determine the most statistically significant course of action for each of the 203 posed scenarios. We then utilize feature selection to determine the most relevant features, and model selection techniques to find the most accurate classifier for the given survey data. We are able to accurately reflect popular moral judgment in 80-83% of Trolley Problem scenarios using, in best case, an SVM classifier. We thus determine that the framework developed by our model is successful and is, infact, a hybrid of the standard Utilitarian and Care Ethics frameworks.**

I. Background

A. Motivation

As we begin to create machines that have the ability to make decisions, it is imperative to consider the ramifications. Especially when a human life is at risk, giving a machine responsibility also requires giving it an ethical framework within which it can act. No such models currently exist, because of the difficulties inherent in generalizing a concept as difficult to define as morality. As such, we decided to crowdsource a moral framework by polling responses from a large audience, and using statistical learning to teach a model to generalize a populations ethical standards.

Previous research in moral decision-making notes the difficulties in accruing accurate ethical statements from individuals, often because such decisions can be motivated by a number of hidden factors. The research therefore highlights the success of simple standard ethical experiments (Singer-Clark). We chose to use the Trolley Problem, a famous thought experiment, as our basis for studying ethical dilemmas. The basic form of the problem forces individuals to choose between allowing a rogue trolley to kill five people standing in its path, or pulling lever to divert the trolley onto a side path, thereby killing only one person who was standing in the alternate path.

B. Our Problem

We converted the Trolley Problem into one that is more relatable to the modern day dilemmas surrounding autonomous vehicle ethics. In this modified problem, we pose individuals with the choice of saving their own car or saving the other car, in the case of an on-coming accident between two vehicles. Though accidents are often complex, we simplify the problem to the binary case of being able to definitively save only one car. In the question, we provide subjects with a number of facts or demographic information regarding the occupants of each car. The demographic information provided was selected based upon their relevance in the context of deontological, care ethics, and utilitarian moral frameworks. These facts include the number of people in the cars, their ages, occupations, stages in life, and relationships (or lack thereof) with the individual making the choice. Based on this information, the individual must determine what the moral decision should be.We limit the infinite scopes of the facts in the problem by allowing the people in each car to have only one of 6 occupations, 6 stages in life, and 7 possible relationships to the individual.

C. Past Work

There has been much discussion about machine morality, yet very little successful and concrete implementation of ideas, especially with regards to machine learning. Previous similar studies have discussed operational morality - purely rule-based systems (Wallach). Statistical implementations are sparse, with a few studies regarding medical ethics and euthanasia showcasing promising results for medical robots (Wallach).

There are many previous works that describe crowdsourced machine learning techniques and how to handle the presence of noisy labels. These works have determined nuanced methods to use maximum likelihood to eliminate biased voters and therefore better select the Gold Standard response from a number of survey responses. They also discuss majority voting, selecting the response with the greatest number of votes as correct, as a viable baseline method when faced with a limited number of voters (or survey responders) (Donmez).

II. Data Set

A. The Survey

As previously mentioned, our goal was to use crowdsourced data to statistically learn a moral framework. Because we created a novel question and set of inputs and outputs, we were required to self-generate the data in the form of an online survey.

We created a corpus of 203 questions, each with at least one value for each of the features described. An example survey question is below:

Your car contains: you and your best friend, who is also your age The other car contains: Nine 10-year olds, one of whom is your brother, and their 30 year old teacher who you have met before What should your car do: Save your car Save the other car

In order to constrain our data to a reasonable pool, the survey was limited to the context of Stanford, and was distributed to 315 students. These students covered a wide spectrum of diversity with regard to class year, major, ethnicity, and religion. Each student was randomly presented 20 of the 203 questions. Thus, each question received approximately 10-30 responses.

B. Features

In order to convert the survey questions into feature vectors, we mapped the occupancy of each car to a set of features. For each question, we created a 40-dimensional feature space with the first 20 features corresponding to the first car, and the second 20 to the second. The set of features included the following values:

1. Number of people in each car
2. Minimum, average, and maximum ages of people in each car

and binary values (0 or 1) for the presence of each type of the following:

3. Occupations of people in each car (President, Teacher, Terrorist, Student, Retired, or None)
4. Stages of life of people in each car: (Child, Adult, Parent, Single Parent, Elderly, or None)
5. Connections to decision-maker of people in each car: (Self, Child-of, Immediate Family, Extended Family, Friend, Acquaintance, or None)

The majority response from students for each survey question has been selected as the gold standard response, and used as the correct value. In other words, for a particular feature vector, the y-value is represented as 0 if the majority of respondents deemed that the other car should be saved, and a 1 if a majority of respondents deemed that their cars should be saved.

III. Methods

A. Cross Validation

We selected the models to optimise further by finding the baseline models with the lowest test errors using cross validation. To apply cross validation techniques we split the training data into 7 sets of 29 random examples. We first determined a single generalization error by training the model on 6 sets and then testing on the remaining single hold-out set. Because our data set is incredibly small (203 examples), we let our test error equal the average generalization error for each baseline model after running cross-validation across all training sets and hold-out set permutations.

B. Model Selection

The first step of building a moral framework entailed selecting the most appropriate model (ie. that with the lowest training error). We note that our problem is one of binary classification, and are feature vectors are high-dimensional and sparse, with mostly binary features. Thus, we selected the following three techniques to attempt classification: L1-regularized SVM, Logistic Regression, and Bernoulli Naive Bayes. All models were implemented using sci-kit.

In addition, we tried each model both on the data as it was collected and after each feature vector was standardized to be Gaussian with zero mean and unit variance. Because most of our features were binary, the features for age and number of passengers in the car could have been a larger role in weighting the eventual decisions, simply because of their larger size. Normalizing the vector had the potential to reduce this bias.

1) The SVM: The SVM is a standard choice for classification problems, because it is known to be one of the best models for supervised learning. We chose to penalize the SVM with the L1-norm in order to account for the high likelihood of having non-linearly separable data, given that the data was crowdsourced and potentially erroneous.

Furthermore, with potentially non-linearly separable data in mind, we start with a baseline Radial Basis Function (RBF) kernel. This is because the RBF kernel, unlike the Linear kernel, maps the features to a non-linear higher-dimensional space. The RBF kernel references the following equation:

$$K(x, x') = exp(\gamma \left\| x - x' \right\|^2)$$

With our RBF kernel, we started with the recommended baseline C value of 1.0 and Gamma value of

$$\left(\frac{1}{number\ of\ features} \right) = 0.025$$

(sci-kit).

Because the two parameters that must be experimented with are C and Gamma, we implemented the SVM while varying each of these parameters in order to fine-tune and find the most accurate choice.

2) Logistic Regression: Logistic regression is the most common discriminative algorithm choice for supervised learning classification because of its ease to implement and reasonable accuracy. Logistic regression allows us to take a non-linear problem and classify it in a linear form. For logistic regression, we experimented with both L1 and L2 penalization, and for each, with the C (cost) value. We note that the difference between the two penalizations can be seen with regard to the vector w. The baseline selected was L1 penalization with C = 1.0 (sci-kit). L1 penalization:

$$min_{w,c} \left\| w \right\| + C \sum_{i=1}^{n} (log(exp(-y_i(X_i^T w + c)) + 1))$$

vs. L2 penalization:

$$min_{w,c} w^T w + C \sum_{i=1}^{n} (log(exp(-y_i(X_i^T w + c)) + 1))$$

3) Bernoulli Naive Bayes: Because 36 of our 40 features are binary, we also attempt a Bernoulli Naive Bayes model in which the features that are not binary are given the value of 1. This model could theoretically lead to accurate results in cases where the average age or number of individuals in the car hold little relevance. We note that Bernoulli Naive Bayes is a generative algorithm, and therefore utilizes the probability that a particular feature will be 1 or 0 to conduct classification:

$$P(x_i|y) = P(i|y)x_i + (1 - P(i|y))(1 - x_i)$$

(sci-kit).

C. Feature Selection

In order to minimize the risk of overfitting, it is necessary for the number of training examples to be ¿¿ than the number of features. However, our feature vector is of size 40 whereas we are training on sets of only 175 examples, which is not significantly greater than 40. Therefore in order to improve the performance of our models on the hold-out or test sets, we implemented recursive feature elimination using our baseline SVM model to reduce the size of our feature vector. Recursive Feature Elimination (RFE) uses backward search to recursively consider smaller and smaller sets of features until a desired number of features is reached. The SVM model used in RFE assigns weights to features, which RFE then consults with to recursively prune the features with the smallest weights, from the desired feature set (sci-kit). The features with smaller weights are ones that are generally redundant, conflicting, or lead to overfitting. To determine the desired number of features, we iteratively implemented RFE over all possible desired numbers of features, 1 to 40, and computed the number of features which returned the lowest error.

D. Continuous Regression

Because we used majority-voting to determine our gold-standard, our labels were inherently noisy and didnt take into account the distribution of people who chose a particular answer. In order to account for the fact that not all users chose a single label on each example, we trained a baseline SVM continuous regression model to predict the likelihood that humans choose a particular answer. This model is helpful in conducting error analysis on the questions which the binary models predicted incorrectly.

IV. Results and Discussion

A. Baseline

For the three tested models, we present a table and graph (Fig. 1) with the averages of the generalization errors across each held-out set of 29 examples before and after normalization. As can be seen below, all three models achieve an accuracy that is <40% and therefore better than random. Logistic regression, post-normalization, achieves the lowest error of .21. In fact, Logistic Regression appears to work better than SVM both before and after normalization. We hypothesize that this is the case because while SVM creates a bound, Logistic Regression applies discriminative analysis. Thus, with 40 features, the SVM creates a bound that is strict and potential overfitting, logistic regression is more likely to achieve success. We also note that normalization positively benefits both the SVM and Logistic Regression models, but it has enormous effect on SVM! This is because the SVM model is not inherently scale-invariant. Thus, modeling the feature vector around the gaussian served as an equalizer among the features.

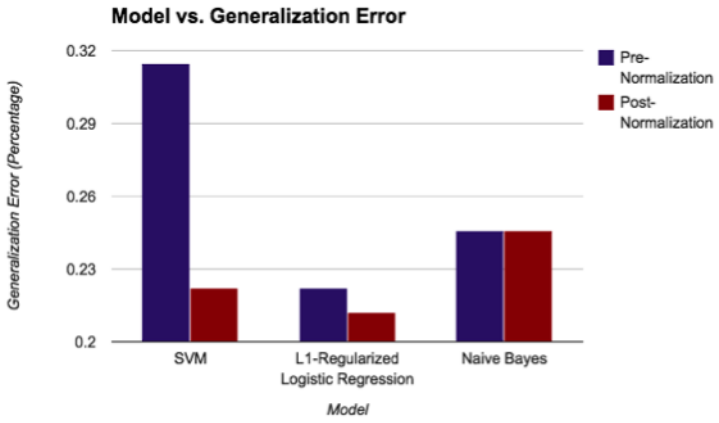

Fig. 1. Model versus Generalization Error for baseline parameters

B. Modfications to the Model

As mentioned in the methods, in order to improve our models, we tested our training data with various variations of C and Gamma for the SVM model, and penalizations and C values for logistic regression. Because Naive Bayes performed significantly worse than the other two models, we decided to forego further testing on it. The figures below (Fig. 2, Fig. 3) showcase the effects of various combinations of C and Gamma and C and Penalty Norm on the models in question. We see that C = 1 and Gamma = 0.01 lead to the lowest training error for an SVM with dimension 40. With logistic regression, we see that L1-regularization and a C = 0.4 lead to the largest result. Thus, L1-regularization provides the right amount of penalty, and a small C is necessary so that there exists flexibility in the bound.

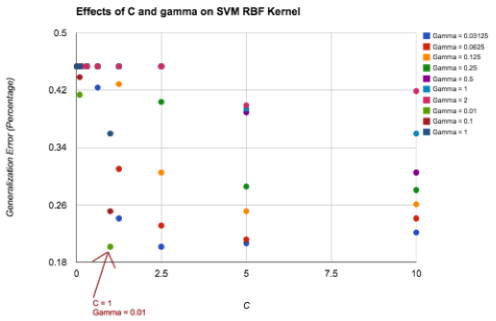

Fig. 2. Effects of C and gamma on SVM RBF Kernel

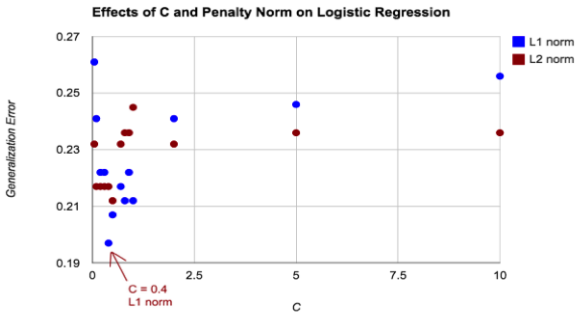

Fig. 3. Effects of C and penalty norm on Logistic Regression

C. Feature Selection

Also as discussed in the methods, to reduce overfitting and improve our models, we enabled Recursive Feature Elimination. This process further served to help us understand the features that held the most importance in our model and, therefore, were the most valuable when making ethical decisions.

The following graph (Fig. 4) showcases how reducing the number of features affects the generalization error for each of the 3 models we attempted. We see that, as predicted, while an extremely low number of features leads to underfitting, less than 40 features does, in all 3 cases, produce a lower error. For the SVM, the lowest error occurs with 17 features. With Logistic Regression, it occurs with 33.

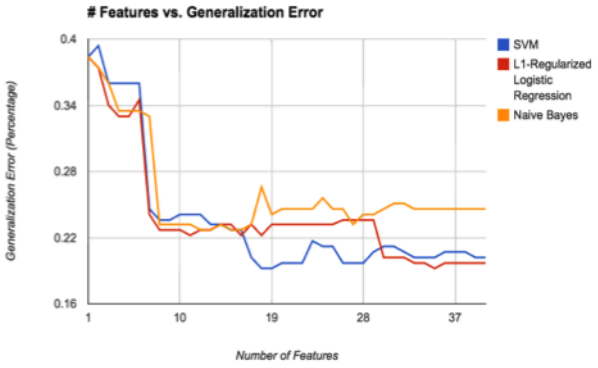

Fig. 4. Number of Features versus Generalization Error

We took this understanding further, and realized that an SVM tested and trained with 40 features would potentially require a lower C value (ie. that of 1.0). Thus, it would be possible that a smaller feature vector and a stricter C could actually train a more accurate model that would have been previously missed because it overfit on 40 features. Trial and error showcased that this hypothesis was, in fact, correct. Training an SVM with C = 3.0 and Gamma = 0.125 with an 8 feature vector leads to the overall lowest training error of .17. These selected 8 features, listed to the right, reflect the importance of family values and social good. Of highest importance are ones immediate family and children. At the same time, retired individuals and terrorists, despite their family relation, are downweighted while the president is upweighted, when considering the greater good.

TABLE I
SELECTED FEATURES

Features - Car 1	Features - Car2
One's Child	One's Child
Immediate Family	Immediate Family
Acquaintance	Terrorist
Retired Individual	President

D. Final Results

Our final results, shown in the graph below (Fig. 5), showcase the success rates found for the most optimized SVM and Logistic Regression models, with the ideal number of features. We see that both examples have success rates greater than 80%, with the SVM performing the best. Logistic Regression has an 80.8% success rate, while the SVM has an 82.8% success rate.

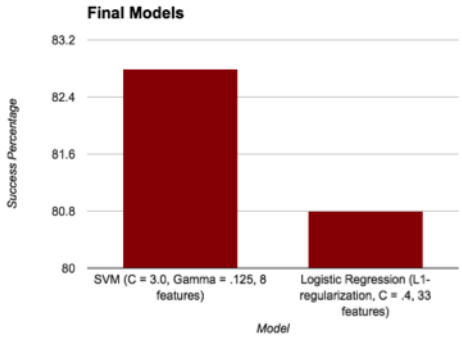

Fig. 5. Final Models

E. Error Analysis

The vast majority of errors came from examples with noisy labels where there was an equal, or close to equal, number of people for and against saving your car. If more people had answered that example question then it is possible that the majority label would have been the opposite of our gold standard. Since humans themselves were unable to come to a consensus regarding these questions, it is likely that the model would have also had trouble determining an answer. Either the example contained conflicting features or no features that correspond to the feature vector. Conflicting features refer to the presence of multiple features which have the opposite effect on classifying the example. For example, if you have your sister in your car, your car is more likely to be saved, but if a terrorist is also in your car, then your car is less likely to be saved than before. An example with no features that occur in the recursively determined feature vector is rare, but it exists due to the

fact that we have a small data set. For example, if your car contains one adult who is a stranger to you and the other car contains five adults who are strangers to you, neither car has feature which correspond to the feature vector, and the model essentially makes a random guess in the binary classification problem. Using the SVM continuous regression model, we attempted to classify a decimal answer rather than a strict binary 1 or 0, where a 1 represents saving your car and a 0 represents saving the other car. The result of the continuous regression model gives us a rough estimate of the certainty with which the classifier chose to save either your car or the other car. The distance of the classifiers result from 0.5 gives us a certainty value; if the result is very close to 0.5 then the model was extremely unsure about its choice. The figure to the left graphs the certainty values for all examples which were predicted incorrectly by our optimised SVM binary classifier. All values are under 0.5 which proposes that our incorrect results are partially due to low certainty and increased noise in the labels themselves.

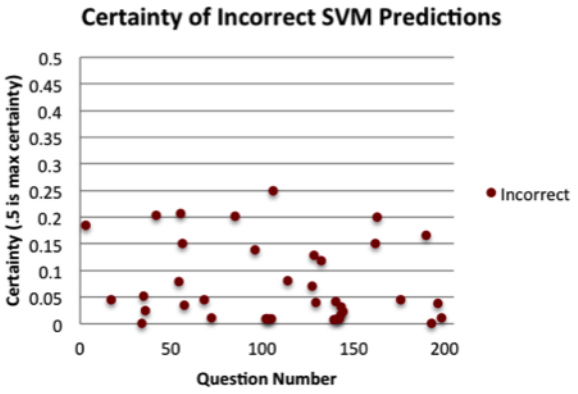

Fig. 6. Certainty of Incorrect SVM Predictions

V. Conclusions

Overall, we see that while both the SVM and L1-Regularized Logistic Regression models are effective, an SVM with 8 features, C = 3.0, and Gamma = 0.125 leads to the highest accuracy with respect to predicting the majority human choice for ethical action. Use of continuous regression also shows that the answers classified incorrectly by the model are those with a low confidence-score, indicating that majority-voting may be the culprit for an amount of inaccuracy.

Perhaps even more exciting, in conclusion, that crowd-sourcing responses for a simple ethical question can indeed build a stable, accurate, well-predicting moral framework. In fact, the moral framework is a hybrid of Utilitarianism and Care Ethics frameworks, a hybrid that makes sense because it emphasizes a desire to protect ones immediate social circle while still retaining caveats with respect to the greater good. This duality can be seen in both feature selection and the answers that the model provides to each individual question.

This result paves the way for considering crowdsourced statistical learning as a feasible method for research in machine ethics. Further, going forward, we should perhaps consider adopting a similar hybrid model as a basis for decision-making when considering human-owned autonomous

VI. Future Work

Though we were happy with the level of success of our model, we would work to improve this success by accruing more survey results. With a larger set of voters on each question, we would apply an EM-algorithm (see: Raykar 2010) in order to select a more appropriate answer as the gold standard. Especially in cases where the votes were close to 50-50 (the majority of our incorrectly classified answers), Raykar et. als research indicates that utilizing a maximum-likelihood approach to eliminate certain voter bias leads to significantly less noisy labeling, and a more accurate overall model.

Furthermore, we chose to simplify our problem by creating a virtual world in which we self-determined a select list of possible feature values. In extending this project further, we would make the problem more realistic by allowing for a wide range of possible feature values and using text-processing to classify the values into particular groups. This would allow for a more dynamic set of questions that better capture real-life scenarios. In fact, many of these questions could be pulled directly from a database of accident scenarios.

With the data we currently have, conditioning on ethnicity, major, or religion could lead to interesting skews in the moral framework learned, as well as the accuracy of the framework. We would condition on these values to better understand the differences in morality that accompany such social groups.

Lastly, it would be interesting to apply the learned model to other binary ethical dilemmas, such as those in medical ethics with regards to patient life or death. It would also be interesting to crowdsource data about other theoretical problems, learn frameworks, and similarly compare them.

References

[1] Donmez, Pinar, Jaime Carbonell, Jeff Schneider, et al. Efficiently Learning the Accuracy of Labeling Sources for Selective Sampling. New York, NY: ACM, 2009.Carnegie Mellon University. ACM, 2009. Web. Dec. 2014.

[2] Raykar, Vikas, Shipeng Yu, Linda Zhao, et al. "Learning From Crowds."Learning From Crowds (2010): n. pag. Journal of Machine Learning Research. Journal of Machine Learning Research, Apr. 2010. Web. Dec. 2014.

[3] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[4] Singer-Clark, Tyler, and 20 June 5. "Morality Metrics On Iterated Prisoners Dilemma Players." (n.d.): n. pag. Web.

[5] Wallach, Wendell, and Colin Allen. Moral Machine: Teaching Right from Wrong. New York: Oxford UP, 2009. 2009. Web. 11 Dec. 2014.