EECS 349 Titanic – Machine Learning From Disaster

Xiaodong Yang
Northwestern University

Abstract

In this project, we see how we can use machine-learning techniques to predict survivors of the Titanic. With a dataset of 891 individuals containing features like sex, age, and class, we attempt to predict the survivors of a small test group of 418. In particular, compare different machine learning techniques like Decision Tree, SVM, and Random Forest analysis.

1. Introduction

Using data provided by www.kaggle.com, our goal is to apply machine-learning techniques to successfully predict which passengers survived the sinking of the Titanic. Features like ticket price, age, sex, and class will be used to make the predictions.

I take several approaches to this problem in order to compare and contrast the different machine learning techniques. By looking at the results of each technique we can make some insights about the problem. The methods used in the project include Simple Class Model, Random Forest, SVM, and decision tree. Using these methods, we try to predict the survival of passengers using different combinations of features. The challenge boils down to a classification problem given a set of features. One way to make predictions would be to use Decision Tree. Another would be to use SVM to map the features to a higher dimensional space. My approach will be to first use Decision Tree as a baseline measure of what is achievable and replace missing values. Once this is complete, we use SVM on our data to see if we can achieve better results. Lastly we use random forest analysis.

2. Data Sets

The data we used for our project was provided on the Kaggle website. We were given 891 passenger samples for our training set and their associated labels of whether or not the passenger survived. For each passenger, we were given his/her passenger class, name, sex, age, number of siblings/spouses aboard, number of parents/children aboard, ticket number, fare, cabin embarked, and port of embarkation. For the test data, we had 418 samples in the same format. The dataset is not complete, meaning that for several samples, one or many of fields were not available and marked empty (especially in the latter fields – age, fare, cabin, and port). However, all sample points contained at least information about gender and passenger class. In order to replace missing values, I take 2 steps to finish this task.

3. Data preparation

In order to prepare our data for training in our classifier, we have to take a simple look at the data set. At first, let us start with a principle we all know: save children and women first in the disaster. So let us take a look at the Sex and Age variables to see if any patterns are evident. We'll start with the gender of the passengers.

	0	1
Female	0.258	0.742
Male	0.811	0.189

We can see that the majority of females abroad survived, and a very low percentage of males did. Now, we can dig into the age variables:

Min	1st Qu	Median	Mean	3rd Qu	Max	NA's
0.42	20.12	28.00	29.70	38.00	80.00	177

Table 1: The age distribution of passengers in training data

With this table, we can define the passengers who under 18 years old are belong to "children", otherwise belongs to adults. We can get a form about age and sex pattern:

	Child	Sex	Survived
1	No	Female	0.7529
2	Yes	Female	0.6909
3	No	Male	0.1657
4	Yes	Male	0.3965

Table 2: The Age and Sex distribution of passengers in training data

It seems that if the passenger is female most survive, and if they were male most don't, regardless of whether they were children or not. Now, let's look at a couple of other potentially interesting variables to see if we can find anything more.: the class they were riding in, and what they paid for their tickets.

Table 1 Gender Class Model

	Fare2	class	Sex	Survived
1	20-30	1	female	0.8333
2	30+	1	female	0.9772
3	10-20	2	female	0.9142
4	20-30	2	female	0.9000
5	30+	2	female	1.0000
6	<10	3	female	0.5937
7	10-20	3	female	0.5813
8	20-30	3	female	0.3333
9	30+	3	female	0.1250
10	<10	1	male	0.0000
11	20-30	1	male	0.4000
12	30+	1	male	0.3837
13	<10	2	male	0.0000
14	10-20	2	male	0.1587
15	20-30	2	male	0.1600
16	30+	2	male	0.2142
17	<10	3	male	0.1115
18	10-20	3	male	0.2368
19	20-30	3	male	0.1250
20	30+	3	male	0.2400

Table 3: The class and Fare distribution of passengers in training data

Interestingly, we can simply refer some new hypotheses: people in the first class have more chances survived than the lower classes. People who paid more will get more chances of surviving.

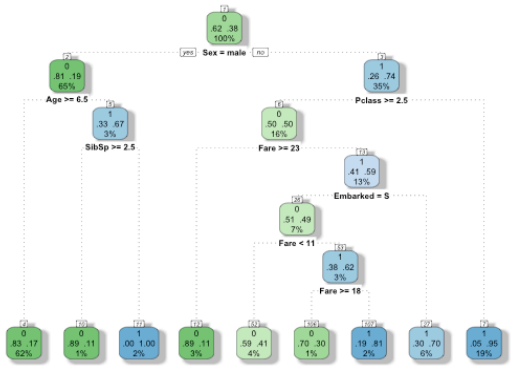
4. Modeling

Decision_Tree:

We built our decision starting with gender model. We split all the train dataset into male and female. Because it was most correlated with the chance of survival. From just using a single feature, we achieved an accuracy of 74.79%. Then we split both males and females into passenger classes. Even after splitting the data into passenger class, males in each class are more likely to die, and passengers in each class, other than class 3, are more likely to survive. If we choose the hard decision that female passengers in class 3 all survive, it will still produce an accuracy of 76.79% because the classifier hasn't changed from the earlier process of labeling all males as died and females as survived. However, if we choose the hard decision that all females in class 3 will die, our accuracy improves to 75.27% on the test data.

Next, we look at the feature age. Since the domain of age is continuous, we have to find a good decision boundary to split our data. After plotting the age and survival of passengers in each gender and passenger class, we decided to use a binary decision because in most cases, older passengers were more likely to die than younger ones. Instead of using the same age boundary for each gender and passenger class, we considered each gender and passenger class, case by case and found different boundary thresholds for each. To find our boundary threshold, we tried to minimize the classification error on our training set. This means that we chose the age boundary for each gender and passenger class such that if we classify all samples below the age boundary as survived and all above as died, we minimize the classification error

on the training set. After including again the decision tree, we achieve a classification error of 76.39%. After that, I use the function of Rpart in R library, which will generate custom features automatically of dataset and output a decision tree model.



the accuracy of this model will be 79.469% which is greater than previous result.

To compare our results classification, we used support vector machines. We considered the following features: 1) passenger class, 2) sex, 3) age, 4) number of siblings, 5) patriarchal status, 6) fare, and 7) place of embarkation. We used a Gaussian radial basis function as our kernel and set the tolerance to ε =0 .001.

Iterating through all possible feature combinations, we were able to achieve an accuracy rate of 77.99% on the test data set using only three features. The three features that achieve this rate were class, sex and place of embarkation. Using age, fare, and place of embarkation resulted in the worst accuracy of 58.13%. It is interesting to note that this accuracy would be less if we had just guessed that all test points died (accuracy of 63.23%). This suggests that perhaps class and sex are strong indicators of survival whereas age and fare are weaker indicators of survival.

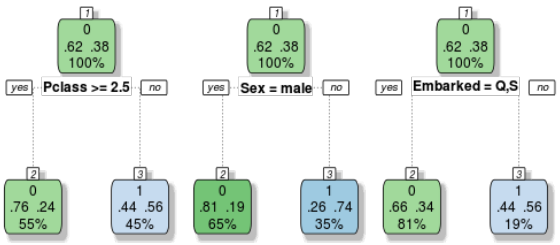
SVM

Table 2. SVM Accuracy – Using Different Features

Pclass	Sex	Age	Sibsp	Parch	Fare	Embarked	Accuracy
Yes	Yes	No	No	No	No	Yes	77.99%
No	Yes	Yes	Yes	Yes	No	No	74.88%
No	Yes	No	Yes	No	Yes	No	73.21%
No	No	No	No	Yes	Yes	Yes	67.70%
No	No	Yes	No	Yes	No	Yes	64.83%
No	Yes	Yes	Yes	No	Yes	Yes	61.96%
No	No	Yes	No	No	Yes	Yes	58.13%

Random Forest:

Previously, we found that decision tree has over-fitting sometimes when dealing with terrible parameters. But if we grow a whole lot of them and have them vote on the outcome, we can get passed this limitation. That's why we use random forests. At first, we build a very small ensemble of three simple decision trees to illustrate:



Each of these trees make their classification decisions based on different variables. So let's imagine a female passenger from Southampton who rode in first class. Tree one and two would vote that she survived, but tree three votes that she perishes. If we take a vote, it's 2 to 1 in favor of her survival, so we would classify this passenger as a survivor.

Random Forest models grow trees much deeper than the decision stumps above, in fact the default behavior is to grow each tree out as far as possible, But since the formulas for building a single decision tree are the same every time, some source of randomness is required to make these trees different from one another. Random Forests do this in two ways.

The first way is to use bagging, for bootstrap aggregating. Bagging takes a randomized sample of

the rows in the training set, with replacement. This is easy to simulate in R using the sample function. Let's say we wanted to perform bagging on a training set with 10 rows.

```
> sample(1:10, replace = TRUE)
 [1]  3  1  9  1  7 10 10  2  2  9
```

In this simulation, we would still have 10 rows to work with, but rows 1, 2, 9 and 10 are each repeated twice, while rows 4, 5, 6 and 8 are excluded. On average, around 37% of the rows will be left out of the bootstrapped sample. With these repeated and omitted rows, each decision tree grown with bagging would evolve slightly differently. If we have very strong features such as gender in our example though, that variable will probably still dominate the first decision in most of our trees.

The second source of randomness gets past this limitation though. Instead of looking at the entire pool of available variables, Random Forests take only a subset of them, typically the square root of the number available. In our case we have 10 variables, so using a subset of three variables would be reasonable. The selection of available variables is changed for each and every node in the decision trees. This way, many of the trees won't even have the gender variable available at the first split, and might not even see it until several nodes deep.

Through these two sources of randomness, the ensemble contains a collection of totally unique trees which all make their classifications differently. As with our simple example, each tree is called to make a classification for a given passenger, the votes are tallied (with perhaps many hundreds, or thousands of trees) and the majority decision is chosen. Since each tree is grown out fully, they each over-fit, but in different ways. Thus the mistakes one makes will be averaged out over them all.

One of the toughest things is replacing missing value. We can use a decision tree to fill in those values instead. We should pick up where we left off last lesson, and take a look at the combined data frame's age variable to see what we're up against:

```
->summary(combi$Age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   0.17   21.00   28.00   29.88   39.00   80.00     263
```

263 values out of 1309 were missing this whole time, that's a whopping 20%! A few new pieces of syntax to use. Instead of subsetting by boolean logic, we can use the R function is.na(), and it's reciprocal !is.na() (the bang symbol represents 'not'). This subsets on whether a value is missing or not. We now also want to use the method="anova" version of our decision tree, as we are not trying to predict a category any more, but a continuous variable. So we grow a tree on the subset of the data with the age values available, and then replace those that are missing:

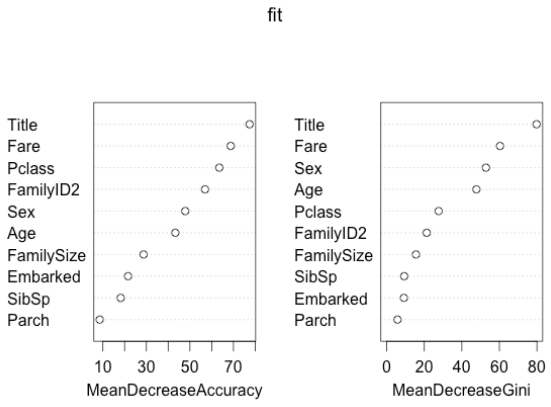
```
Agefit <- rpart(Age ~ Pclass + Sex + SibSp + Parch + Fare + Embarked + Title + FamilySize, data=combi[!is.na(combi$Age),], method="anova")

combi$Age[is.na(combi$Age)] <- predict(Agefit, combi[is.na(combi$Age),])
```

After this, all missing values are predicted by decision tree model we use in the first section.At this time, we can use the random forest tools to solve the problem.

```
fit <- randomForest(as.factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked + Title +FamilySize+FamilyID2,data=train, importance=TRUE, ntree=2000)
```

The result is as following.



There's two types of importance measures shown above. The accuracy one tests to see how worse the model performs without each variable, so a high decrease in accuracy would be expected for very predictive variables. The Gini one digs into the mathematics behind decision trees, but essentially measures how pure the nodes are at the end of the tree. Again it tests to see the result if each variable is taken out and a high score means the variable was important. Unsurprisingly, our Title variable was at the top for both measures. We should be pretty happy to see that the remaining engineered variables are doing quite nicely too. Finally, this model get 81.342% accuracy!!!

5. Conclusion

After implementing three methods, we got some conclusions: Of all of three methods, SVM performs worst with 77.99% but Random Forest performs best with 81.34% accuracy. So only the difference is nearly 3 percent. This is probably because there was one feature that was strongly correlated with whether a passenger survives. Decision Tree and SVM only combine features but didn't give them specific weights and correlations. So this shows that assuming that features are independent is not necessarily a bad assumption for our problem. Table 3 offers a summary of the achievable accuracy using Decision Tree, SVM, and Random Forest analysis.

Table 3 Comparison of Performance

Decision Tree	79.46%
SVM	77.99%
Random Forest	81.34%

As for the future work, I suggest that I can pay attention to find the internal correlations between other attributes and optimize the parameter of Random Forest.

In terms of the question that who will survive from the disaster, there are some inferences:

1. Woman and Children have the best chance of surviving because human always have to protect the vulnerable groups first. Most adult males have to survive by their own faith ,physical conditions or luck.

2. Wealth and upper –class people may be more possible to get survived mainly because they paid expensive tickets, which contributes to the result that they might live closer to safeboat than people who live under the deck.

3. People who are despicable and shameless may survive from disaster partially because they can obey their core values in order to get a seat in safe boat even those who comes from upper class and well-educated family.

I believe there are more interesting conclusions to be found when I go further and collect more data about sinking ship events.