Prediction of Bike Sharing Systems for Casual and Registered Users

Mahmood Alhusseini
mih@stanford.edu

CS229: Machine Learning

Abstract - In this project, two different approaches to predict Bike Sharing Demand are studied. The first approach tries to predict the exact number of bikes that will be rented using Support Vector Machines (SVM). The second approach tries to classify the demand into 5 different levels from 1 (lowest) to 5 (highest) using Softmax Regression and Support Vector Machines.

***Index Terms* –regression, classification, prediction SVM, Softmax**

I. Introduction

Bike sharing systems have been increasing in demand over the past two decades as a result of rapid advancements in technology (Figure 1). However, as seen in Figure 2, fluctuations in demand during the year are still present due to different factors such as temperature, time, etc. The goal of this project is to present a model for predicating fluctuations in this demand for both casual and registered consumers so that the service can be optimized for both system providers and consumers. Two approaches have been used: 1) continuous model to predict an exact demand, and 2) classification into 5 levels of demand. In the continuous model, SVM Regression was used to predict the data while SVM classification and Softmax Regression were used for approach 2.[1]

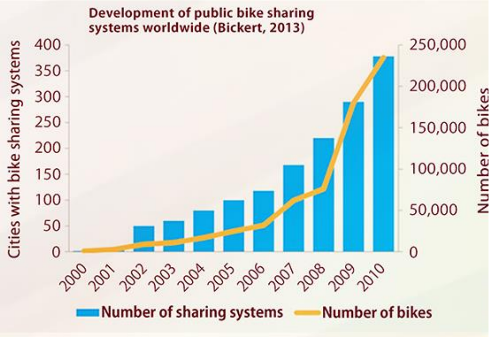

Figure 1: Constant demand increase for bike sharing systems from 2000 to 2010 worldwide.[2]

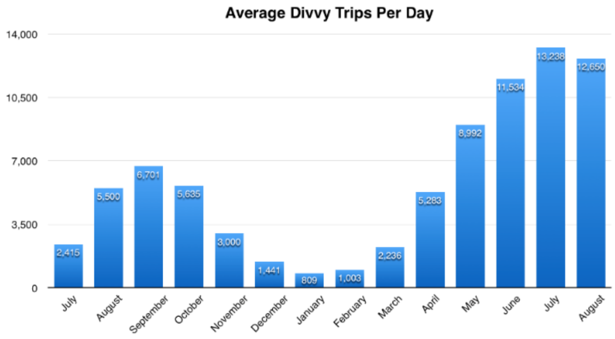

Figure 2: Fluctuations in demand for bike sharing systems over different months of the year.[3]

II. Data and Features

The data was obtained from an online website.[4] The number of sample points in the data is 10886 from which 8886 samples were used for training the model and 2000 used for testing the model. The original data contained 9 features and 3 labels classified as follows:

Feature/Labels	Numerical Value
Date and Time	Format is MM/DD/YYYY HH:MM
Season	Takes 4 values: 1 = spring, 2 = summer, 3 = fall, 4 = winter
Holiday	1 = yes, 0 = no
Working day	1 = yes, 0 = no
Weather	1: Clear, few clouds, partly cloudy 2: Mist + cloudy, mist + broken clouds, mist + few cloud 3: Light snow, light rain + thunderstorm + Scattered clouds, light rain + scattered clouds. 4: Heavy Rain + Ice Pallets+ thunderstorm + mist, snow + fog
Temp	Temperature in celcius
Atemp	“feels like” temperature
Humidity	Relative humidity
Wind speed	No units were given
Casual	Number of non-registered user rentals initiated
Registered	Number of Registered user rentals initiated
Count	Total number of rentals (Casual + Registered)

Table 1: Description of features and label as given in the website.[5]

As seen in Table 1 above, the date and time of each sample was given as one feature which made the feature difficult to interpret and incorporate in a model. Therefore, the date and time was split into 4 separate

[1] SVMs were implemented using the libsvm package available online. Approach I was done using the SVR option while approach II was done using the SVM option.

[2] Figure taken from http://www.kevinauyeung.com/cycleHireScheme.html, December 07, 2014.

[3] Taken from http://chi.streetsblog.org/wp-content/uploads/2014/09/Screenshot-2014-09-03-11.01.42.png, December 07, 2014.

[4] Data obtained from: https://www.kaggle.com/c/bike-sharing-demand/data, accessed December 10, 2014.

features: year, month, day, and time. This made the total number of features 13 - making it easier to integrate the features into the model.

Moreover, the features and labels on the data were also normalized according to: The labels for the data used was first normalized according to:

$$y' = \frac{y - \min(Y)}{\max(Y) - \min(Y)}$$

Where:

y': *normalized value*, y : *original value*, Y: *vector containing values* y

Because this project only deals with predicting the number of casual users and registered users independently (i.e. each label was treated as a separate problem having the same features), normalization was only performed for the number of casual users and the number of registered users; the total number of bikes used label was not considered.

III. Methods

a. Approach I: Continuous model prediction using SVM

Initially, Linear Regression was used to fit the data but the results obtained were very erroneous. Therefore, it was thought that using SVMs in the regression form would result in better result since different linear and nonlinear kernels can be used. A Gaussian kernel was used in the algorithm. The effectiveness of SVMs was then calculated using the Root Mean Squared Logarithmic Error (RMSLE) according to:

$$\sqrt{\frac{1}{m}\sum_{1}^{m}(\ln(y_i + 1) - \ln(y_i' + 1))^2}$$

Where:

m: # *of samples*, y_{i_i}: *predicted value*, y_i': *actually value*

The algorithms and calculations were performed for both casual and registered users independently. Feature analysis was performed to find which parameters contribute the most in accuracy improvement.

b. Approach II: Classification into five different classes

Turning the problem into a classification problem would allow for a better interpretation of how well the model is performing. At first, it was decided to turn the problem into ten different classes from 1 (lowest) to 10 (highest); however, this gave a maximum test accuracy of about 60%. In an attempt to simplify the problem and get better accuracy, the classification labels were reduced into five labels from 1 (lowest) to (five) highest – this way allowed for a larger error margin in the model.

Two learning algorithms were used in approach II. The first is Softmax regression, and the second is SVMs (with a Gaussian kernel). Different SVM settings were used to optimize the results. Similar to what was done in approach I, feature analysis was performed on the features to find which feature contributes the most in accuracy improvement. The performance of both algorithms is calculated using:

$$Accuracy = \frac{number\ of\ samples\ labled\ correctly}{total\ number\ of\ samples}$$

IV. Results and Discussion

a. Approach I: Continuous model prediction using SVM

The SVM algorithm was run many times under different options for c, g, kernel type, and kernel degree. The findings showed that increasing c resulted in better training data on the whole. On the other hand, the parameter g had to be adjusted to that we would avoid a case of under/over fitting the training data which translated in a bigger RMSLE for the testing data. Figure 3 and Figure 4 summarize the results.

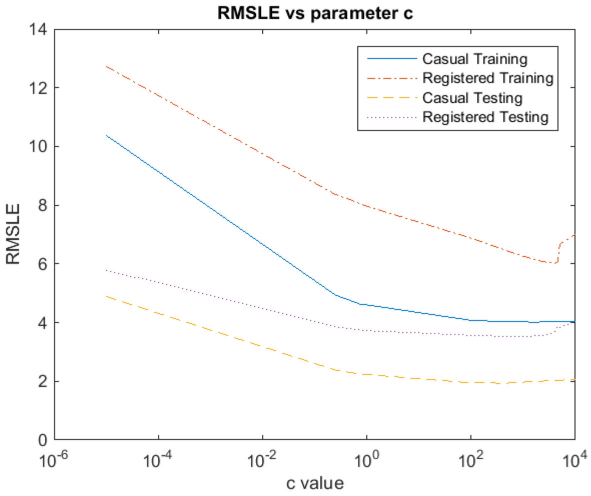

Figure 3: Best results for RMSLE error as a function of c in SVM

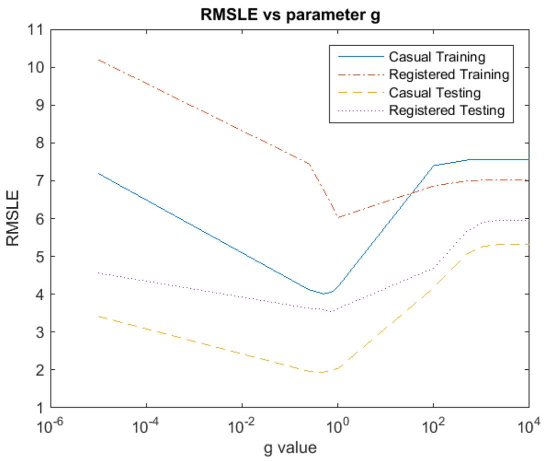

Figure 4: Best results for RMSLE error as a function of g in SVM

The best RMSLE results obtained gave an error of around 1.9 with c = 100 and g = 1. This result is still higher compared with results submitted on the online website where the data was obtained.[6] It is thought that the results can be still improved further by using different options for the SVM or even different machine learning algorithms such as Random Forests or Neural Networks.[7] However, given the time constraint of the project, these were the best solutions achieved given the long computational time it takes to run the program.

Feature selection was performed on the data to see which features contributed to the best results. Table 2 summarizes the results.

Feature Removed	Best Casual RMSLE	Best Registered RMSLE
Overall	1.93	3.92
Year	1.93	3.92
Month	1.93	3.92
Day	1.94	3.92
Time	1.94	4.52
Season	1.94	4.53
Holiday	1.94	4.72
Working day	2.03	4.83
Weather	2.21	5.07
Temperature	2.37	5.29
"Feels like" Temp.	3.92	6.22
Humidity	4.72	6.86

Table 2: Feature selection for SVM in approach 1

The results obtained follow the expected trend, less features resulted in a higher error. The two features that contributed to the highest error were humidity and "feels like" temperature, followed by weather, time, and holiday. The year, month, and day had very little or no contribution on the error.

b. Approach II: Classification Models

i. Softmax Regression Results

Softmax algorithm was run first in order to get a good initial estimate of the accuracy of the model. The accuracy for the training data was found to be 100% for both casual and registered users, while testing accuracy was found to be around 85% and 86% for casual and registered users, respectively.

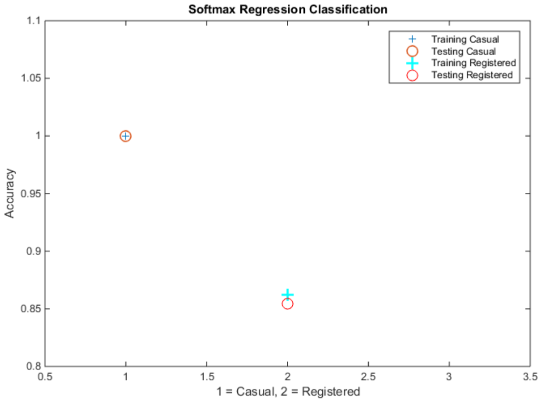

Figure 5: Softmax regression accuracy for classification

ii. Support Vector Machines

The second method used was support vector machines for classification. The algorithm was run for different values of c and g in SVM. The optimized solution was found at parameter c = 1 and g = 0.25. The training accuracy was around 99% for both casual

[6] As of Dec 12, 2014, results on Kaggle website range from 0.24976 to 4.76189. https://www.kaggle.com/c/bike-sharing-demand/leaderboard.

[7] A solution posted on kaggle website gives an RMSLE error of 0.70 using Random Forests. http://www.techdreams.org/programming/solving-kaggles-bike-sharing-demand-machine-learning-problem/9343-20140821

and registered users, while the testing accuracy for casual users was around 91% and 86% for casual users.

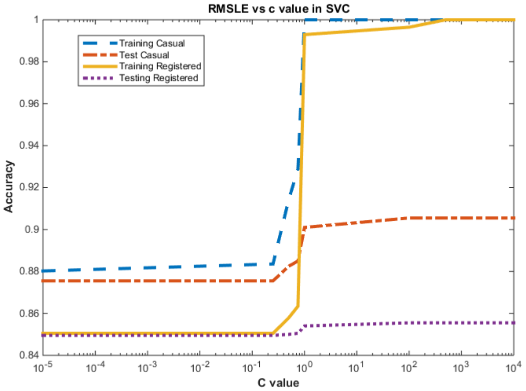

Figure 6: SVM accuracy as a function of parameter c

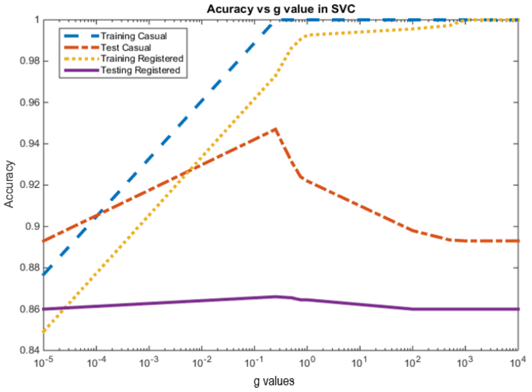

Figure 7: SVM accuracy as a function of parameter g

In order to understand which classes were harder than others to predict, a classification matrix was prepared. As seen in Table 3, most of the errors were in the classification of class 2 and 3 of the demand.

		Actual				
		1	2	3	4	5
Predicted	1	90%	9%	1%	0%	0%
	2	32%	39%	20%	8%	2%
	3	4%	23%	40%	28%	4%
	4	0%	0%	0%	67%	33%
	5	0%	0%	0%	0%	0%

Table 3: Classfication table for SVM regression

This result could be due to several reasons including the limited number of samples with labels 2 and 3 which, in turn, do not give our learning algorithm enough training examples to build a more solid model.

Similar to what performed in approach I, feature selection was implemented. The results for casual users were unexpected as accuracy increased to 100% as features were removed from the model. The most increase of around 5% came from removing the day feature. As for registered users, results were as expected – decreasing accuracy with less features. The most decrease in accuracy came from removing the day and working day features resulting in a decrease of about 2% in accuracy from each. Table 4 below shows a summary of the results.

Feature Removed	Casual Accur.	Registered Accur.
Overall	0.9395	0.8624
Year	0.9415	0.8599
Month	0.949	0.858
Day	0.9875	0.833
Time	0.9875	0.8345
Season	0.99	0.8215
Holiday	0.99	0.821
Working day	0.991	0.8055
Weather	0.991	0.813
Temperature	0.9945	0.8075
"Feels like" Temp.	1	0.8275
Humidity	1	0.8445

Table 4: Feature selection implemented in SVM algorithm

The results obtained for the increasing accuracy with less features for casual users was not expected. One reason could be that the data is very random and that the used featured don't capture the necessary attributes to predict how casual users choose to rent bikes. Such unused features include nationality, socioeconomic status, and length of visit. More data and features need to be used to understand the situation with casual users.

V. Conclusion and Future Work

Two approaches were taken to try and predict the bike sharing demand. In the first approach, the continuous model, initial results gave a high RMSLE of 7.65 error which were then lowered to 1.9 by changing the parameters of the system. The results contained a high variance which could be due to not having a large enough training samples or not selecting features that would optimize the problem. It is very possible that better results can be obtained if the SVM was optimized. In addition, other algorithms such as

Random Forest and Neural Networks could result in better results.

As for the second approach, demand classification, the accuracy for classifying the data into 5 groups gave good results of up to around 95%, with SVM outperforming Softmax regression. The results contained a high variance as well which could be due to the same reasons mentioned above. One interesting observation is the fact that decreasing the number of features increased accuracy for casual users while (as expected) it decreased the accuracy for registered users.

Future work aims at optimizing the SVM algorithm to produce a model that gives less testing error as well as using other algorithms (Random Forest and Neural Networks). Moreover, investigation into the feature selection effect on casual users in approach II is needed to understand how the algorithm produces unexpectedly better results with less features. In addition, for approach II, it would be beneficial to try and increase the classification labels into 10 (or more) different classes instead and still be able to get a high level of accuracy.

VI. References

[1]http://www.kevinauyeung.com/cycleHireScheme.html, December 07, 2014.

[2] http://chi.streetsblog.org/wp-content/uploads/2014/09/Screenshot-2014-09-03-11.01.42.png, December 07, 2014.

[3] https://www.kaggle.com/c/bike-sharing-demand. December 07, 2014. Note: Data was also taken from here.

[4] “Using Gradient Boosting Machines to Predict Bikesharing Station States,” Robert Regue and Will Recker, UC Irvine, TBR 2014, figure 2 pg. 11.

[5]http://beyondvalence.blogspot.com/2014/07/predicting-capital-bikeshare-demand-in_10.html, accessed Dec 07, 2014.

[6]http://beyondvalence.blogspot.com/2014/06/predicting-capital-bikeshare-demand-in.html accessed Dec 12,2014.

[7] http://brandonharris.io/kaggle-bike-sharing/ accessed Dec. 12, 2014.