TRAIN DELAY PREDICTOR

CSL341-Fundamentals of Machine Learning

PROJECT REPORT

11/16/2013

GROUP # 4

ABDUL HADI SHAKIR (2010CS50202)

ANUBHAV SINGH (2010CS10209)

BHARAT RATAN (2010CS50280)

RAHUL NAGAR (2010CS10236)

TABLE OF CONTENTS

1. OVERVIEW

Objective: The objective of this project is to predict the delay in arrival of train(s) given certain features like destination, day of week etc. These features are discussed in detail in the latter part of the report.

Motivation: In a country like India, where a vast majority of population depends on the Railway infrastructure (approx. 25 million passengers daily), coming up with an application that aids to their travelling experience would certainly be of great help. Trains in India get delayed frequently, and if we can predict this is in advance - it would be of great help for the passengers to plan their journey.

2. MODELLING THE INDIAN RAILWAY NETWORK

Given the short time to complete this project, our first aim in the project was to model the Indian Railway Network or IRN to something simpler and less complex in nature. To our rescue was the "*Statistical Analysis of the Indian Railway Network: A Complex Network Approach*" by Prof. Niloy Ganguly of IIT Kharagpur.

In order to make the analysis simpler we set out for identifying some of major station in IRN in terms of degree of station (connecting stations) and weight of station (traffic). Three key points were observed:

- Major stations are located in close vicinity to the metropolitan cities in India (e.g. Hazrat Nizamuddin near Delhi, Kalyan near Mumbai etc.).
- Major stations that are located in the central parts of the country or at the meeting points of railway lines connecting different zones.
- Most of traffic resided on tracks connecting these major stations.

OUR ZONES OF INTEREST

Thus we decided to use New Delhi and NCR regions to be the origin of station we analyse. We broadly divided the destinations to the four zones:

i) Southern UP/Madhya Pradesh Region
ii) Eastern UP/Bihar Region
iii) Rajasthan/Gujarat Region
iv) Chandigarh/Jammu Region

All the destinations were under radius of 1500 km from Delhi. This provided for a feasible yet reliable area of analysis. If required the model can be extended to other regions of India in a similar fashion.

3. FEATURES

The next step was to come up with features to be used for delay prediction. Feature extraction is the most important part of any machine learning task and so is the case with us. To build effective train delay predictor, our aim is to try to model attributes like quality of train, the duration of its journey, area of its journey etc.

At present, our model is using the following set of features:

a) Type of Train: It has often been noticed that certain train are given preference over other when it comes to passing over congested

network. Thus, we have divided the train into two categories. Type-0 train that include the best category trains of IRN like Rajdhani, Shatabdi etc. Type-1 includes the normal trains like express, superfast etc.

b) Travelling Distance: The trains travelling over long distance are often more vulnerable to delay. Thus we have also taken into account the distance that a train travels. It has been classified into two categories - Category-0: short-distance, less than 500km, and Category-1: long-distance, more than 500km.

c) Day of Week: We are expecting certain days of week to be more congested than others. For example, the weekends are expected to experience more rush than other days of week. The days are assigned label starting from Mon:1, Tue:2, …., Sun:7.

d) Region of Journey: Certain regions experience more railway traffic than others. For instance in the UP/Bihar region we have extensive cases of redundant chain-pulling, frequent track disorder, high concentration of passenger trains and the heavy volume of traffic. The regions are same as discussed above, and each of them have been assigned a label from 1 to 4.

4. DATA COLLECTION

Automated data collection was deployed using python cgi-based scripts. The primary sources of our data were websites that had information about trains between stations, real-time arrival/departure information. We took help from **www.runningstatus.in** and **www.etrain.info/in.** Due to limited number of requests served by these websites, we had to manually interfere in data collection.

For example, train no 12801, Puroshottam Express, from New Delhi to Kanpur (250 km approx.) on Wednesday was late by 1hr 30 min. The corresponding data entry would look like:

Train No	Train name	Type (0/1)	Distance (0/1)	Destination/path (1/2/3/4)	Day of week (1/2/…/7))	Delay
12801	Purushottam Express	1	0	4	3	1.5

We tried modelling the delays in classes rather than a continuous variable. This was done so that we can use Random Forest and SVM for classification. These models can be reused/implemented easily for classification as compared to regression. We created 5 bins for delays, which are as follows:

- 0-3 min = bin-1 (18% of data points)
- 3-10 min = bin-2 (23% of data points)
- 11-25 min = bin-3 (21% of data points)
- 25-60 min = bin-4 (21% of data points)
- > 60 min = bin-5 (17% of data points)

5. MODELS WE LOOKED UPON

i) SUPPORT VECTOR MACHINE (SVM)

We also used Support Vector Machine (SVM) for classifying the data and predict results. We used LibSvm tool from Matlab. We thought the result would be best for SVM, it was good for the train set, but for test set it highly underperformed (for train set error was very small but when we tried it for test set error was very high). Even increasing the cost penalty for delays did not helped for test set.

We also tried to classify the data in two classes (i.e. Binary viz. non-delayed (0-10 mins delay – 42% of data points) and delayed (>10 mins delay) – 58% of data points), so as to get an idea about which factors corresponds to delay. We used a Gaussian kernel (we tried various combinations of C and gamma). A weighted SVM does a slightly better job in predicting delays, but takes a much longer time to train.

We analysed for the problems which are causing this. One of the reasons we thought could be causing this is the variety and variance of parameters, and our inability to capture them in this limited time. Even if we reduce the feature space to make it more optimal we could not get even near to the desired results as the number of data points we had could not be enough to train it completely (it would take about tens of millions of data point for just one route), and it would make the calculations very slow.

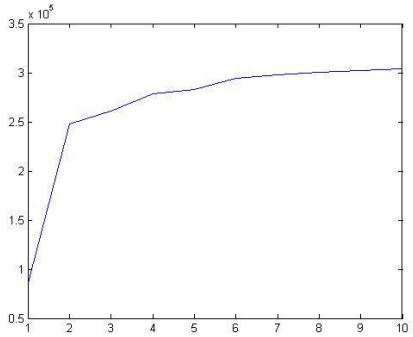
Reducing the number of feature space also had this disadvantage of being non-consistent with our original theme of giving at least some knowledge about train late pattern. Narrowing down the problem to, say, one station- one train- one day etc. will not result in generalized view of the problem.

The average training set accuracy was 87.16% and the test set error was 72.37%.

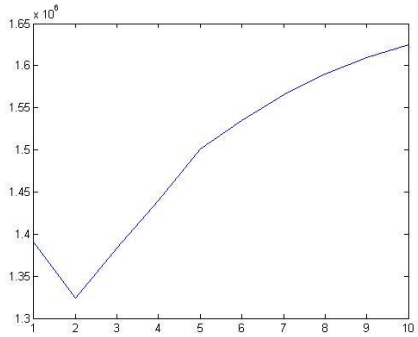
ii) POLYNOMIAL REGRESSION

We performed Linear Regression with Polynomial Basis Functions over the data. For this purpose data was randomly divided into training and test sets. Further we trained our model over the training set and calculated training as well as test error.

As we know, in order to apply this model we need to construct a polynomial basis function of degree d and using this function we can transform the original feature space to a new feature space over which simple linear regression can be applied. We varied d from 1 to10 and calculated the theta values. Using those values we calculated the errors over both training and testing data. Following is the graph of error vs d :



Test error



Training error

Observation:

We found that after d=3 the training error was unable to converge and hence we stopped minimizing it further after 1000 iterations. That is why the training error is initially decreasing but after that it's very high. So this model is unable to fit the data and hence not to be considered as a good predictor.

iii) RANDOM FOREST

Random Forests grows many classification trees. To classify a new object from an input vector, put the input vector down each of the trees in the forest. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

Each decision tree was trained on ⅔ of the data, called the 'bag', and tested on the other ⅓, called the out of bag data. Training data are selected with *replacement* from the original data set. The number of testing errors from each tree is then summed up and divided by the number of trees we specified, giving a metric known as the 'Out of bag error' (OOB). We used 50 trees for the majority of our analysis, and randomly chose 2 features to split the tree on.
The average training set accuracy was 84.21% and the test set error was 82.37%.

6. MODEL SELECTION

We used 5-fold cross validation for selecting the final model. Polynomial regression did not turn out well as it did not converge for squared error on training and the test data set. The following table summarise average training and test set accuracy for SVM and Random forest:

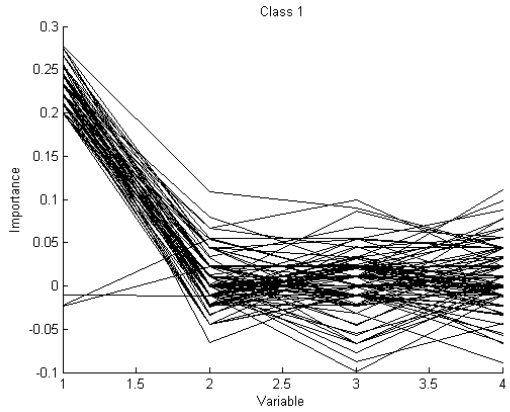
MODEL	TRAINING ACCURACY	TEST ACCURACY
SVM	87.16%	72.12%
RANDOM FOREST	84.21%	82.37%

As can be seen from the above table, SVM overfits the data and performs poorly on test set. Random forest perform almost equally well on both training and test set.

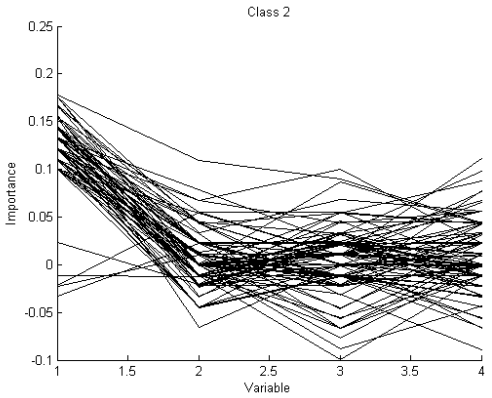
7. ANALYSIS USING *RANDOM FOREST*

i) VARIABLE IMPORTANCE

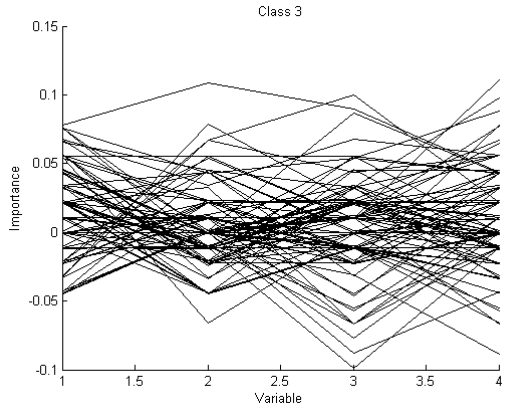
In every tree grown in the forest, we put down the oob cases and count the number of votes cast for the correct class. Now we randomly permute the values of variable m in the oob cases and put these cases down the tree. Subtract the number of votes for the correct class in the variable-m-permuted oob data from the number of votes for the correct class in the untouched oob data. The average of this number over all trees in the forest is the raw importance score for variable m. The importance plot for each class is shown below:



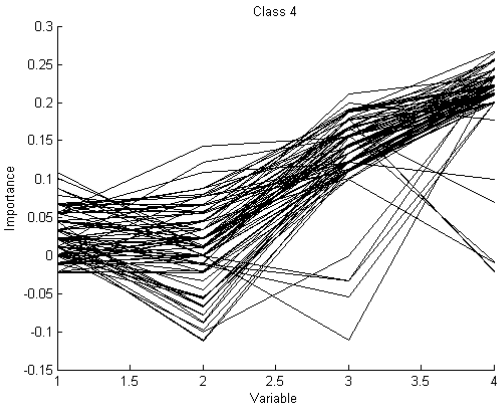
CLASS-1



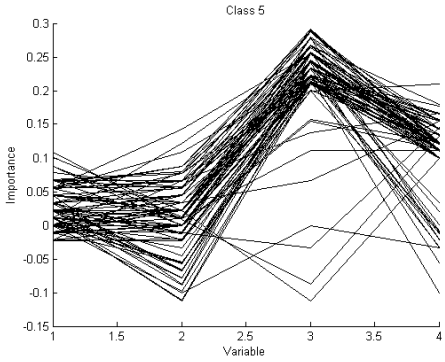
CLASS-2



CLASS-3



CLASS-4



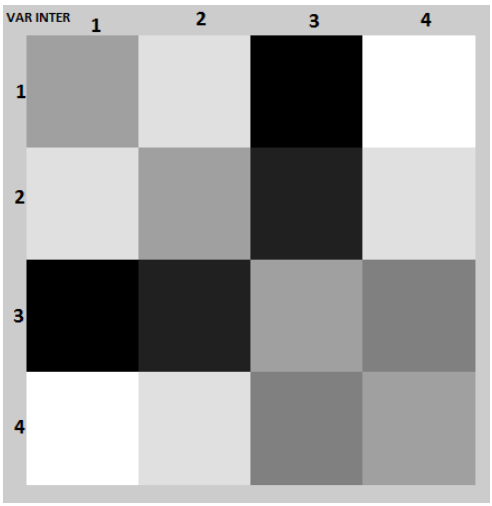
CLASS-5

We can observe that importance variable has peak for certain classes. Note the following points:

- Variable 1 is important for classifying class-1 and class-2.
- Variable 4 is important for classifying class-4.
- Variable 3 is important for classifying class-5.

ii) VARIABLE INTERACTION

The operating definition of interaction used is that variables m and k interact if a split on one variable, say m, in a tree makes a split on k either systematically less possible or more possible. We used the implementation provided by [1]. A large positive number implies that a split on one variable inhibits a split on the other and conversely. The following diagram shows the variable interaction matrix. Darker cells suggest interactions.



INTERACTION MATRIX

Observation: Variable 1 and 3 interact strongly. Variable 2 and 3 also interact significantly.

8. CONCLUSIONS

Based on the observations, we can conclude following things:

i. Since variable-1 (type of train) is important for class-1 (0-3 min delay) and class-2 (4-10 min delay) classification, certain type of trains almost never gets delayed or is delayed for very little time. These are the trains of high priority like Rajdhani, Shatabdi etc.

ii. Since variable-3 (region) is important for classifying both class-4(25-60 min delay) and class-5 (> 60 min delay), we can conclude that trains running in certain regions are almost every time delayed. This is particular to region-4, i.e. the UP/Bihar zone.

iii. Variable 1 (type of train) and variable 3 (region) interact strongly. Thus a split on one makes split on other almost improbable. Thus it can be concluded that a particular type of train running in any region is never delayed (like Rajdhani etc) or almost every-time delayed (like normal express trains running in UP/Bihar zone).

iv. Variable 2 (distance) and variable 3 (region) interact strongly. No matter how much the distance is to be travelled, trains in certain region are most probable of getting delayed (like trains in UP/Bihar zone).

9. FUTURE SCOPE

- **X-Factor**

  More often than not we hear about communal riots, strikes, railway vandalism and disasters that really affect the running schedule of the trains passing by concerned regions. We have pointed it as X-factor because if this is included in the model it would make the application more useful and convincing.

  To add such a feature we would need to keep a parameter X which will initially be just null. We will do data crawling on some known news websites of the country and look for any news that relates with factors that might cause trains to get delayed. This would not be trivial but some existing procedures can be used. If there are such news available related to a particular region we could set the parameter X

for that region and the model will now use this knowledge to predict delay.

- **Widening the scope of Regions/Stations**

  Right now we have only considered trains which commences from Delhi or its vicinity. We would like to increase the number of regions and concerned cities from which the trains depart. It is an essential extension to the model and it comes with inclusion of a lot more trains. To add the functionality we would add a parameter for the train departing region. We might be able get some interesting correlations between the departing and ending region.

- **Using Railway Infrastructure**

  It is very well known that number of track lanes differ a lot in different regions. Their number increases on busy routes and decreases on usually less-busy routes. We would like to incorporate this information in our current model. We have made a valid assumption that the routes which has less track lanes, encounters a train delay then the other trains which are going through the route might get some drift. In case of the routes which have a good number of track lanes available this assumption might not help but it will impact to a certain extent.

- **Let's look at it from other side**

  We can use the predicted/real delays to tell the problems with the railways infrastructure which could be come out to be a good tool for the infrastructure engineers. We can certainly find correlations between parameters and suggest them where do they need to take care or add something to the infrastructure.

10. REFERENCES

1. *Statistical Analysis of the Indian Railway Network: A Complex Network Approach* by Saptarshi Ghosh[y], Avishek Banerjee, Naveen Sharma, Sanket Agarwal, Niloy Ganguly.

2. *Hsu, Chih-Wei; and Lin, Chih-Jen (2002). "A Comparison of Methods for Multiclass Support Vector Machines". IEEE Transactions on Neural Networks.*

3. *Predicting flight delays* by Dieterich Lawson and William Castillo – Stanford University

4. *Using Random Forest to Learn Imbalance Data* by Chao Chen, Andy Liaw and Leo Breiman, July 2004.

5. http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

6. www.runningstatus.in