Presentation Speeches

Introduction

This project intends to using machine learning techniques to find best model to predict the popularity of online news. Our data comes from UCI (University of California, Irvine) Machine Learning Repository. Our project can help newspaper media to predict the popularity of the article before publication.

(Next page - Content)
Our presentation consists of four parts: data & features; EDA; Models; Comparison

(Next Page - 具体介绍下)
We make use of a dataset with over 39,000 observations and 61 variables (58 predictive attributes, 2 non-predictive, 1 goal field).
Five Aspects : ... + One response variable (Y)

(Next Page - EDA- 1. Data Distribution)

EDA and variables selection

Used EDA to modify data and did a basic variable selection.

First of all, plot all the variable data by histogram to check the distributions. (Only to limit space, only shows part of the histogram.)

Second, review all the histograms, and identify some problems. – Skewed data. Most of the variables, including Y (shares) are heavily skewed.

(Next Page - 2. Data Transformation)

So use log and square root to transform those variables. (For those variables with all values bigger than 0 , we use log ,and those have value = 0, we use square root. because log(0) = inf)

(Next Page- 3. Data Distribution)

After transforming the response variable Y(shares), Y is almost normally distributed. Other variables look better.

(Next Page -4. Check interesting Variables)

Checking some interesting variables: Check whether days of week affect the shares and Check whether channel affect the shares. Publishing day didn’t show much influence on shares neither. So I will get rid of all the indicators but leave “is_weekend” because I do see some difference between weekdays and weekend data. Channel do have an effect on shares, so keep them.

(Next Page - 5. EDA Continues)

Other Modifciations: 19th, 21st, 23rd, 25th Variables contains negative values that cannot be explained by information available, so they will be removed. (nature language processing, should not have negative values for those variables)
19: kw_min_min: Worst keyword (min. shares)
21: kw_avg_min: Worst keyword (avg. shares)
23: kw_max_max: Best keyword (max. shares)
25: kw_min_avg: Avg. keyword (min. shares)
outlier in variable “n_unique_tokens”, “n_non_stop_words”, and “n_non_stop_unique_tokens”, may due to typing error as they are the ratios and should have a value between 0 and 1. We will remove that observation.
The number of total variables left is 49 + 1 response variable (去掉4 + 5 +2 non predictive variable).

(Next Page - Features Selection)

Features Selection

In order to improve the efficiency of our models, we do features selection first. We apply two methods to do that.

- Remove redundant variables
  The absolute values of pair-wise correlations among predictor variables are considered. If two variables have a high correlation (their correlation > 0.5), then we look at the mean absolute correlation of each variable and removes the variable with the largest mean absolute correlation. (It chooses one of the two variables based on how correlated it is with all the other variables.)

(Turn Page- Correlation figure)

In total, we remove 14 variables that are highly correlated other variables.

(Turn Page)

- Rank variables by importance
  We use Pearson's correlation as the indicator to measures the linear relationship between continuous variables and the response variable. Then rank them from the highest correlated to the lowest correlated.

(Turn Page - rank figure)

We rank the remaining variables by their importance, and selected top 16 variables and plus the remaining 4 categorical variables.

(Turn Page- variables table)

Here is the total variables we have selected.

Linear Regression

First we check the assumption of linear regression which are Linearity/Non-constancy of error variance/Non-Normality/Correlated error. Our data pass the assumption test, so we can use linear regression model for our data. Here is our result of linear regression model. Most our estimators are significant. Then we use our linear model to predict number of shares and compare our result to actual value. MSE for linear regression is 0.7841.

Variables:

- num_hrefs: Number of links
- num_self_hrefs: Number of links to other articles published by Mashable
- num_videos: Number of videos
- num_keywords: Number of keywords in the metadata
- data_channel_is_lifestyle: Is data channel 'Lifestyle'?
- data_channel_is_entertainment: Is data channel 'Entertainment'?
- data_channel_is_bus: Is data channel 'Business'?
- data_channel_is_socmed: Is data channel 'Social Media'?
- kw_max_min: Worst keyword (max. shares)
- kw_avg_max: Best keyword (avg. shares)
- kw_max_avg: Avg. keyword (max. shares)
- kw_avg_avg: Avg. keyword (avg. shares)
- self_reference_min_shares: Min. shares of referenced articles in Mashable
- self_reference_max_shares: Max. shares of referenced articles in Mashable
- self_reference_avg_sharess: Avg. shares of referenced articles in Mashable
- global_subjectivity: Text subjectivity
- avg_positive_polarity: Avg. polarity of positive words
- title_subjectivity: Title subjectivity
- title_sentiment_polarity: Title polarity
- abs_title_sentiment_polarity: Absolute polarity level
- shares: Number of shares (target)

Support Vector Machine

We would like to try Support vector machine model because it performs effectively in high dimensional space.
We start with linear kernel. Then we use more complex kernels, i.e. Gaussian radial. In order to improve our model, we use grid search with cross validation to select best parameters. Totally, we have three parameters: epsilon which defines a margin of tolerance where no penalty is associated in the training loss function with points predicted within a distance epsilon from the actual value; cost is a parameter allows you to trade off between training error and model complexity; sigma determines the width of Gaussian distribution.
For linear kernel, our parameters are...; for radial kernel, our parameters are....
We compute Mean Squared Error and training time for comparison of different models

Regression Trees
第一个slide
The reason why we use regression tree in this project, is because regression tree building method allows input variables to be a mixture of continuous and categorical variables. Considering we have both numeric and categorical variables in our data set, and our response variable shares is continuous numeric variable.

There are different method to build regression trees, and we decide to use rpart.

rpart is Recursive partitioning for classification, regression and survival trees.
In our project, we use A rpart model with a continuous response based on anova model. The reason why use rpart, is because the rpart programs build regression models in a very general structure with two stage procedure; and the resulting models can be represented as binary trees.
This is our final tree plots, and we have 12 splits, with 53 nodes.
Each node shows
-the predicted value
-the percentage of observations in the node.

翻页：
This slide shows how we split trees based on CP value.
1. cp is It is the amount by which *splitting* that node improved the relative error.

2. The relative error is (1 − R^2) , similar to linear regression.

3. The xerror is related to the PRESS statistic. (**PRESS statistic** means the predicted residual error sum of squares.)

from the left plot, we can see that The first split and second appears to improve the fit the most

from right plot: we can see the tree should be pruned to include only 1 or 2 splits

翻页：variable importance

Here are all the important variables selected by rpart

翻页： we calculated and visulized our predicted value based on rpart regression tree model.
Prediction: test error = 0.87
which means fitted Mean Squared of Error is 0.87
For the plot of residuals v.s predicted value, we can see that This plot shows the residuals of shares versus the predicted shares based on the nodes/leaves. There appears to be more

variability in node 11 than in some of the other leaves.

翻页：random forest
Since rpart has such a good performance, why we still need a random forest?
because too many trees can lead to overfitting in rpart.
Moreover, from a famous schalor，不要念名字，直接念下面的话

> “Given its performance, random forest and variable selection using random forest should probably become part of the standard tool-box of methods for the analysis of microarray data.” **This is why we choose Random Forest.**
>
> ------ from a famous scholar: Ramón Díaz-Uriarte (2010)

翻页： this is variable seletion based on random forest. Random forest calculated variable importance based on >RSS(residual sum of squares) for regression tree, and Gini-Index for classification tree.

翻页： from the plot on this slide, we can see that

Random forest Improve on rpart with respect to: accuracy and stability

• *Accuracy*–Random Forests
test error = 0.7569 smaller than rpart, which is 0.8108

• *stability* – if we change the data a little, random forest is relatively stable because it is a combination of many trees.

Gradient Boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

1. n.trees – Number of trees (the number of gradient boosting iteration) --150
2. interaction.depth (Maximum nodes per tree) - number of splits it has to perform on a tree (starting from a single node). --3
3. Shrinkage (Learning Rate) – It is considered as a learning rate. --0.1
(In the context of GBMs, shrinkage is used for reducing, or shrinking, the impact of each additional fitted base-learner (tree). It reduces the size of incremental steps and thus penalizes the importance of each consecutive iteration. The intuition behind this technique is that it is better to improve a model by taking many small steps than by

taking fewer large steps. If one of the boosting iterations turns out to be erroneous, its negative impact can be easily corrected in subsequent steps.)
This default uses very slow learn rates for small data sets and **uses 0.1 for all data sets with more than 10,000 records.**
One typically chooses the shrinkage parameter beforehand and varies the number of iterations (trees) N with respect to the chosen shrinkage.
4. n.minobsinnode - the minimum number of observations in trees' terminal nodes.
--10
Set n.minobsinnode = 10. When working with small training samples it may be vital to lower this setting to five or even three.

Comparison/ Result
Compare time and MSE, RF is the best.