Drugs store sales forecast using Machine Learning

Hongyu Xiong (hxiong2), Xi Wu (wuxi), Jingying Yue (jingying)

1 Introduction

Nowadays medical-related sales prediction is of great interest; with reliable sales prediction, medical companies could allocate their resources more wisely and make better profits. We join a Kaggle competition to predict the everyday drug sale for each store based on the store, promotion and competitor data. We aim to apply different machine learning techniques to tune the model and make predictions on drug sales using time series analysis.

2 Data

The Kaggle website provides us training data of 1115 Rossmann stores' daily sales dated back to 2013, with 1,017,209 entries in total. The training data includes features of promotion and competitors' information. Since we are not able to access to the real sales amount for testing during Kaggle competition, we decide to use 70% of the contest given training data as the training set for our model, the rest 30% as test set for cross validation. For now, the cross validation will be our estimated test error.

3 Method and Results

Since the problem involves time series data, we intend to use time series analysis model to deal with it in the first place. We establish auto-regression (AR) model and train it using the data we have to get the parameters. We test AR models with different order numbers, and calculate the test errors.

In the time series analysis part, we predicted each store's sale based on just its past data. Now we want to see how stores are different according to their different features. First, we establish a time-independent model by averaging the daily sales per store and collapsing the time dimension; in this frame we use Random Forest to select features and then use Support Vector Regression to fit different features (such as assortment type and competitors) to each store's mean sales.

Finally, we manage to generalize the time series model to predict several stores at a time, instead of one by one in the first section.

3.1 Time Series Analysis

In order to get a big picture, we first plot several stores daily sales with respect to time evolution. From the plots, we recognize that it's a time series data and for a certain store, its daily sales evolve periodically with a period of a year. For example, according to Figure 2, we can see that there will be sales peaks at certain dates around January and May, but in general the sales keep at a constant level.

3.1.1 Auto-Regression (AR) Model

Since the prediction for the daily sale for each store from past data seems to be a time series problem, we manage to solve this problem by building a time series analysis model. The basic method we are using is auto-regression (AR) model.

$$X_t = \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$$

We looked online and discovered a package called "Forecast" in R, which does auto-regression. We made some revision so the program could do what we want. To use this model, we pre-process the data to make weekly sales as a time unit. The first reason we want to do that is because that "zero Sunday sales" makes it hard to use daily sales as a time unit, and monthly sale would kill all the detail patterns; the second reason is that we plot sales at different days in a week through time, and discover that they basically follow the same pattern (Figure 1). Since we condense the daily sales to weekly sales at the beginning, after we predicted the weekly sales we still have to regenerate sales on each day in that week. We assume a normal distribution $N(\mu, \sigma^2)$ with μ equal to the weekly sales and variance σ^2 to be trained. We compare AR models with order p = 1, 2, 5, 10, 20 to see which order number fit best.

Figure 1 Sales of store1 on different days of the week

3.1.2 Cross-validation

For each order p, we train parameters φ, ε, μ, and σ with 70% of the data and cross validation with the remain 30% to get the test error. The cost function we are using is the sum of the square of the difference between predicted and real daily sales of store1. The table below is the comparison among test errors using different order number.

Order p	1	2	5	10	20
Test Error (10^10)	1.4447	0.47354	2.4489	2.1463	1.9401

As we can see order number p = 2 yields the lowest test errors. We plot the predicted sales of store1 (Figure 3). If we compare Fig 2 and Fig 3, we can see the model capture the general pattern of the store1's daily sales.

Specifically speaking, for sales on Tuesday, Wednesday and Thursday, our predictions are within 10% of the real daily sales. However, it gives conservative predictions for Mondays and Fridays, when sales are apparently higher compared to other days in a week.

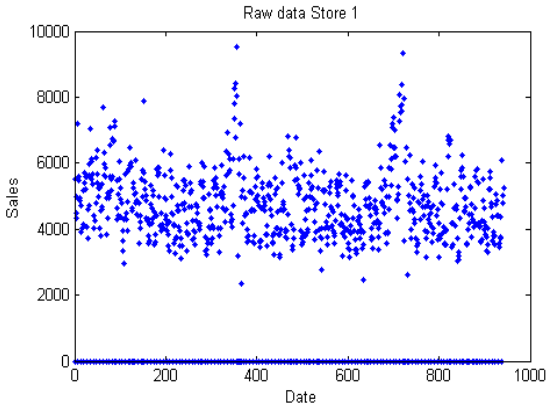

Figure 2 Sales of store1 at daily base

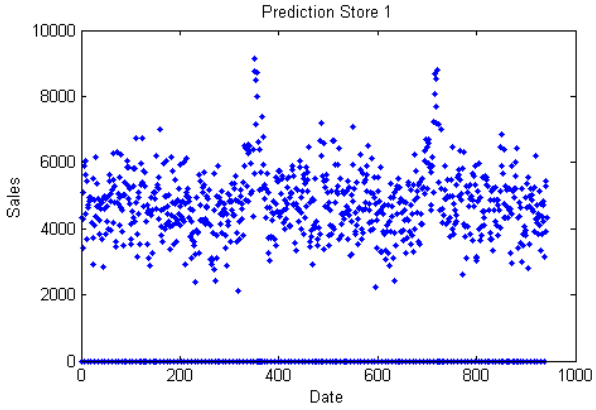

Figure 3 Predicted daily sales of store1

3.2 Random Forest

Since there are so many factors influencing the sales, such as store types, promotions, competitors, and even holidays, we are trying to identify the most important features that influence the sales. We use random forest to do that; in order to decrease the amount of data to test the model and the program, we average the daily sales for each store, so the amount of data decrease from a million to a thousand.

In the raw data provided, five parameters have missing data points. We first combine *CompetitionSinceMonth* and *CompetitionSinceYear* to a new variable *CompetitionExistMonth*. Similarly, we create a new variable *PromoExistMonth* from four promotion variables. We also scale the variable *CompetitionDistance* by 100.

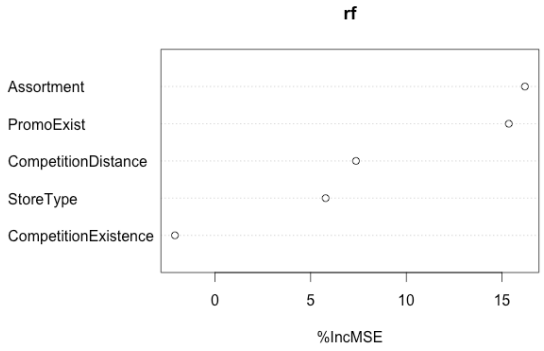

Figure 4 Feature selection through random forest

We built a model using the "random Forest" package in R. Our model was built on 500 trees and 8 variables in the "store.csv" and the mean of sales for each store as the response variable. We used na.omit for the missing values and generated the variable importance plot for the feature selection. The out-of-bag errors were traced for the model.

It seems that the out-of-bag errors are pretty small. But we will have to dig more into the data and model fitting to find out if it's overfitting. The variable importance plot is as following:

As you can see from the plot, the feature *Assortment* is the most important variable because including it could make the biggest reduction in the MSE. Also notice that *CompetitionExistence* is the least important one among all variables in the store characteristics.

3.3 Support Vector Regression (SVR)

We used Support Vector Regression (SVR) method to find relation between each store's mean sales and different kind of features. SVR is a variation of Support Vector Machine (SVM). The essence of SVM is to find a classification boundary with maximum margins to the feature points; by similar token, SVR is to find a regression curve with minimum margins to the feature points. We used the standard liblinear2.1 package in MATLAB to implement SVR.

$$\min_{w,b} \quad \frac{1}{2}\|\omega\|^2$$

$$s.t. \quad \begin{cases} y^{(i)} - \omega^T x^{(i)} - b \le \varepsilon \\ \omega^T x^{(i)} + b - y^{(i)} \le \varepsilon \end{cases} \qquad \text{for i = 1, 2, ... , m}$$

According to the data, there are four different store types (a, b, c, d) and three different assortment types (a, b, c), in total 8 combinations (aa, ac, ba, bb, ca, cc, da, dc). We ran SVR algorithm with linear kernel according to different combinations.

Test Error (10^8)	(a, a)	(a, c)	(b, a)	(b, b)	(c, a)	(c, c)	(d, a)	(d, c)
Linear	2.63	1.60	2.61	0.00412	0.110	0.378	0.263	1.09
Linear + Parabola	2.62	1.57	933	0.00497	0.107	0.385	0.305	1.09
Sqrt + Parabola	2.49	1.57	1.15*10 ^7	0.00938	0.106	0.378	0.228	1.09

3.4 Extension for Time Series Model

In this model we used a method similar to seasonalized times series regression, in which prediction of sales $Y = T \times SI$. We adjusted this equation to $Y = T \times DI$, in which Y means the daily sale prediction, T means the sale trend, and DI means daily sale index. To obtain T, we linearly fitted of all historical sales data of a single store with equation $y = a + bt$, in which $a = \frac{\sum y}{n}$, and $b = \frac{\sum ty}{\sum t^2}$. Then to predict sale at time t_0, $T(t_0) = a + bt_0$. DI was obtained from historical data, and the 365 days in a year each has a different DI. For example, if we wish to get DI for January 2nd, we got historical sales on January 2nd, say s_1, s_2, s_3, at time point t_1, t_2, t_3, respectively, and linearly fitted sales y_1, y_2, y_3, then $DI = \frac{1}{3}\left(\frac{s_1}{y_1} + \frac{s_2}{y_2} + \frac{s_3}{y_3}\right)$. DI reflects the sales fluctuations at different periods within a year. We take considerations for zero sales cases, and if $s_1 = 0$, we just ignores it when calculating DI. Instead in the final prediction we assigned sales on Sundays to be zero, based on observations of historical data, and holiday dates with all historical sales data equal to zero, like January 1st, were also predicted to be zero. We used 76% historical data for training, and 24% for testing, and the cost functions for 10 stores were listed.

	Store1	Store2	Store3	Store4	Store5
Test Error (10^8)	1.3154	4.0130	6.8025	5.9619	5.3891
	Store6	**Store7**	**Store8**	**Store9**	**Store10**
Test Error (10^8)	2.6773	1.0367	7.1013	5.0954	2.6067

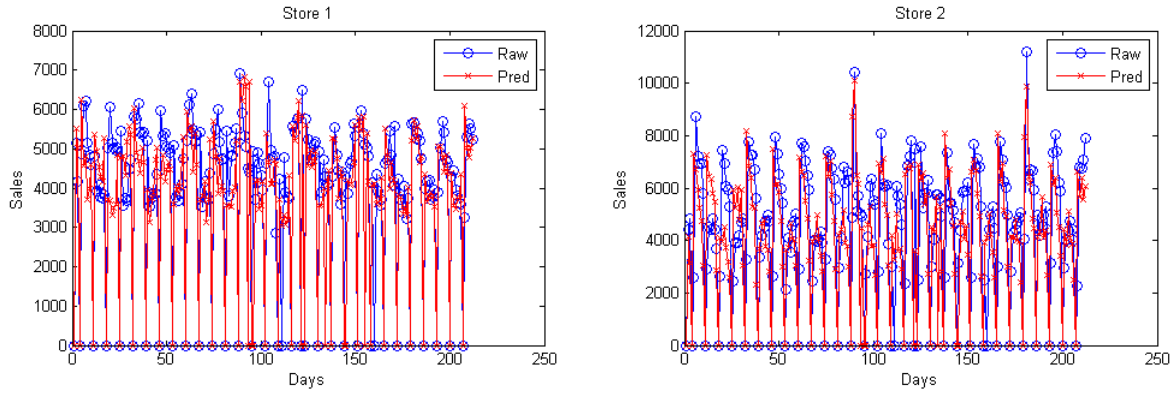

Figure 5 Comparison between raw and predicted sales of store 1 & 2.

4 Conclusion and Prospective

We use AR model to predict the sales with small discrepancy to the test data, and we use RF and SVR to find relations between store mean sales and other features. There are certainly rooms for improvements. We can make further predictions on daily sales using SVR. Even though we found the relations between the features, and make fairly good predictions on average sales for each store. We think next we could try to use SVR see how the parameters set in AR change according to features. By doing so, we could automate the process of making predictions on daily sales for all the stores in the time series model.

Reference:

[1] Wensen Dai et al. “A Clustering-based Sales Forecasting Scheme Using Support Vector Regression for Computer Server.” Procedia Manufacturing 2 (2015) 82 – 86.

[2] Marco Hulsmann et al. “General Sales Forecast Models for Automobile Markets and their Analysis.” Transactions on Machine Learning and Data Mining Vol. 5, No. 2 (2012) 65-86.