Drug Store Sales Prediction

Chenghao Wang, Yang Li

***Abstract* -** In this paper we tried to apply machine learning algorithm into a real world problem – drug store sales forecasting. Given store information, and sales record we applied Linear Regression, Support Vector Regression(SVR) with Gaussian and Polynomial Kernels and Random Forest algorithm, and tried to predict sales for 1-3 weeks. Root Mean Square Percentage Error (RMSPE) is used to measure the accuracy. As it turned out, Random Forest outshined all other models and reached RMSPE of 12.3%, which is a reliable forecast that enables store managers allocate staff and stock up effectively.

1. INTRODUCTION

This problem is one of several Machine Learning problems on Kaggle[1] . The aim of this problem is to forecast future sales of 1,115 Rossman drug stores located across Germany based on their historical sales data. The practical meaning of solving this problem lies in that reliable sales forecasts enables store managers to create effective staff schedules that increase productivity and motivation. What's more, for the purpose of practicing what we learnt from the Machine Learning class, this problem saves us the trouble of collecting data, and in the meanwhile provides a perfect real case to apply supervised learning algorithms.

2. RELATED WORK

As a matter of fact, substantial effort has been put into sales prediction problems. Due to promising performance, artificial neural networks (ANNs) have been applied for sales forecasting in many scenarios. Thiesing, F.M. implemented a neural network forecasting system as a prototype to determine the expected sale figures[1]. What's more, R.J. Kuo utilized a fuzzy neural network with initial weights generated by genetic algorithm (GFNN) and further integrated GFNN with ANN forecast using the time series data and promotion length[2]. This is closely related to our problem because promotion has proved to be one of the most important features in our dataset. There are some interesting attempts too. For example, Xiaohui Yu tried to predict sales of products based on online reviews[3], and Michael Giering tried to correlate sales with customer demographics[4]. As for beginners to get started with sales prediction problem, Smola described a regression technique similar to SVM called Support Vector Regression (SVR)[5]. Breiman posed Random Forest algorithm[6] which is based on decision trees, but randomness is added. It performs very well compared to many other algorithms, including neural networks, discriminant analysis etc. and is robust against overfitting. SVR and Random Forest are both implemented in out project.

3. DATASET AND FEATURES

The dataset of this problem can be found online[2]. The data comes in two sets

[1] The link to this problem: https://www.kaggle.com/c/rossmann-store-sales

[2] The link to dataset: https://www.kaggle.com/c/rossmann-store-sales/data

1. Sales Dataset - Historical sales data for 1,115 Rossman stores from 2013/1/1 to 2015/7/31. Features include store number, date, day of week, whether there's a promotion, whether it's a school or state holiday and sales on that day.

2. Store Dataset - Stores' individual characteristics. Features include store type, assortment level, nearest competitor's distance and when the competitor was opened, and whether there's a consecutive promotion.

Throughout our trial, we've tried to take advantage of different subset of features. However, reducing number of features didn't increase accuracy for this problem. So all features are used for building models.

70%/30% and k-fold cross validations are used in this problem for training and testing. Root Mean Square Percentage Error (RMSPE) is used to measure accuracy, which is defined as: $RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{y_i - \hat{y}_i}{y_i})^2}$

4. METHODS

There are two methods to train the data. One is to train each store separately, which means forecasting sales of a single store based on its own sales record, regardless of store attributes. The other one is to train all stores together, considering store attributes as parameters.

To train each store separately, one straightforward idea is to apply linear regression. According to the normal equation, $\theta = (X^T X)^{-1} X^T \vec{y}$, we can easily predict sales by $H_\theta(x) = \theta^T x$

Further more, we figured that this problem can actually be kernelized. Here consider that case of applying MAP estimate for θ to avoid overfitting, which results in the following primal problem

$$\theta = argmin||y - \theta^T X|| + \lambda||\theta||^2$$

If we calculate α as $\alpha = (< X, X > + \lambda I)^{-1}\vec{y}$. And define $H(x) = \sum_{i=1}^{m} \alpha_i < x, x^{(i)} >$, we can see that this problem can actually be kernalized, thus we can apply the kernel trick. We tried Gaussian Kernel and Polynomial Kernel in this case, which is illustrated as following.

(a) Gaussian Kernel $K(x,z) = \exp\left(-\frac{||x-z||^2}{2\sigma^2}\right)$

(b) Polynomial Kernel $K(x,z) = (x * z + 1)^d$ (Polynomials of degree up to d)

Our next model for this project is Random Forest Regression. We tried this model because it's fast and can accommodate categorical data. RF first picked a certain amount of data from the dataset randomly (ie. bootstrap) and then picked a certain amount of features out of the total features randomly to build decision trees. The final result for each test data is average of results obtained by all these decision trees. Decision trees usually overfit the data; however randomness will average out the high variance

5. EXPERIMENTS AND RESULTS

Linear Regression

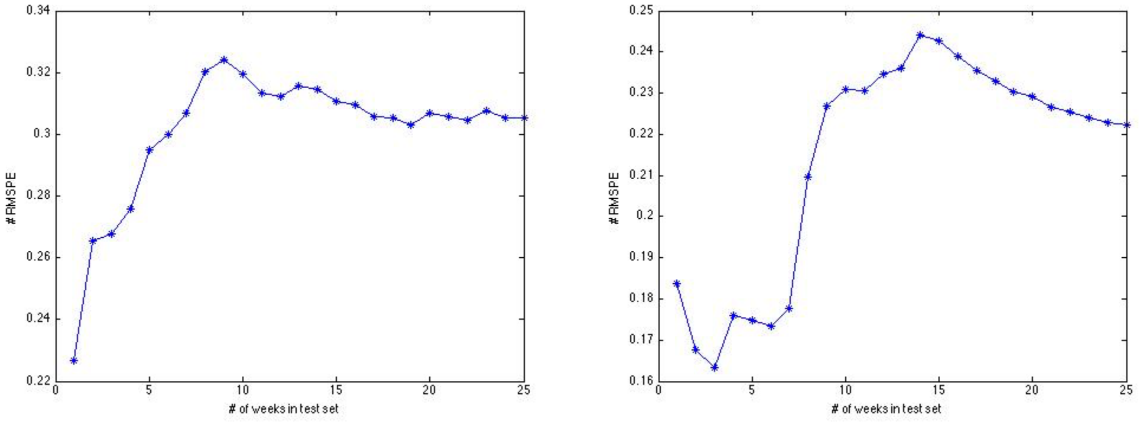
Linear regression is used as our baseline model. 70%/30% cross validation is used here to divide the data set into training set and test set. As it turns out, linear regression gives us a RMSPE of 52.8%.

Support Vector Regression

One thing special on the implementation of SVR is that, it need to build an m*m matrix, where m indicates the number of training samples. Since the size of our training set is ~700,000 , it's unrealistic to operate on the whole dataset. To take use of the abundant dataset practically, we build a SVR model for each store, and compute the mean of each store's RMSPE as our final error rate.

Firstly, we applied Polynomial Kernel and Gaussian Kernel for a single store, Store 1. By trying different pairs of λ & σ for Gaussian Kernel, and different pairs of λ & d for Polynomial Kernel, we found that when $\lambda = 140, \sigma = 45$, Gaussian Kernel gives the best RMSPE of 13.6%, when $\lambda = 0.1, d = 2$ Polynomial Kernel gives the best RMSPE of 12.8%. Two kernels are comparable in this scenario.

Secondly, using the method of finding optimal pairs of parameters discussed above, we applied Gaussian Kernel and Polynomial Kernel to all stores. As we dig deeper into the dataset, we found that accuracies vary on different time period of prediction. Below are figures of how RMSPE varies with different time period for prediction.

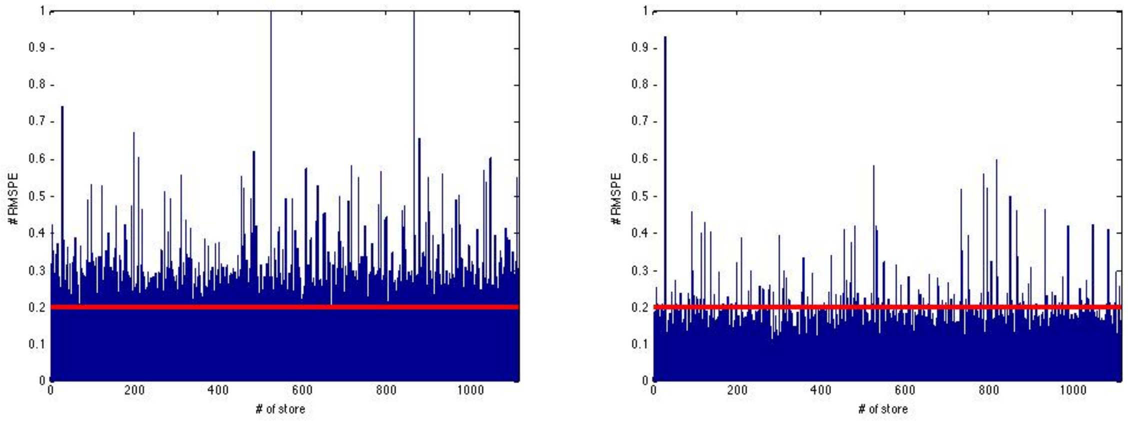


(a) Gaussian Kernel

(b) Polynomial Kernel

Figure 1. How RMSPE Varies with Different Time Period to Predict

As shown above, RMSPEs for both kernels first increase and then decrease as time period of prediction gets larger. For Gaussian Kernel, RMSPE reaches minimum when predicting for just one week, however, for Polynomial Kernel, RMSPE reaches minimum when predicting for 3 weeks. Given this result, we draw figures of RMSPE for all stores using Gaussian Kernel and Polynomial Kernel, predicting for 1 week and 3 weeks respectively.



(a) Gaussian Kernel(Predict for 1 week) (b) Polynomial Kernel(Predict for 3 weeks)

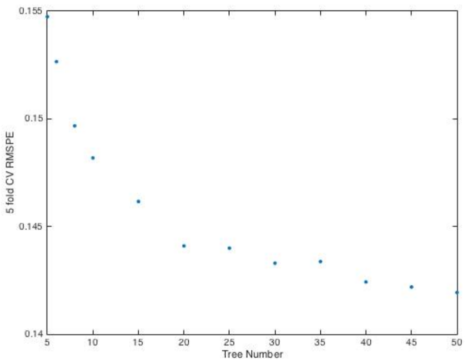
Figure 2. RMSPE for Each Store

In terms of average RMSPE, Polynomial Kernel(16.3%) beats Gaussian Kernel (26.8%) significantly. In the meantime Polynomial Kernel is also more robust than Gaussian Kernel, given that there are fewer outliers and no extreme outliers(RMSPE>1) in the figure of Polynomial Kernel. So overall, Polynomial Kernel suits the dataset better and provides more reliable results.

Random Forest

We applied Random Forest after merging all data including all the categorical data. We used scikit-learn package of python for implementing the algorithm[7].

The two main parameters we tuned for RF is the number of trees and the size of the random subsets of features to consider when splitting a node. We used 5 fold cross validation to get RMSPE while varying these parameters. Two plots are shown below. From these plots, we could see that RMSPE doesn't change too much after tree number reaches 30 and after feature number reaches 20.



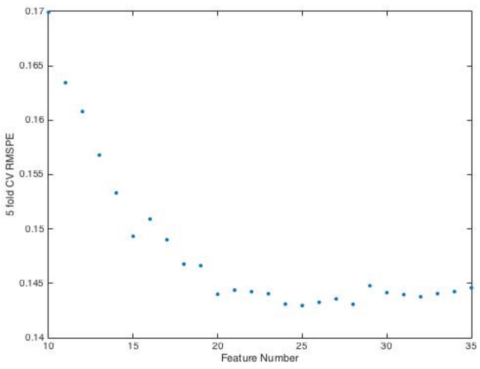


Fig. 3 How RMSPE changes with Feature number

Fig. 4 How RMSPE changes with Tree number

After tuning and fixing the optimal parameters, we tried to change the size of the training data to fit the test data better. We used the last two weeks 7/14/2015 – 7/31/2015 as our test data to get our final prediction RMSPE result. We got a plot RMSPE vs. Number of month before the test period shown below. We could see RMSPE almost doesn't change after month number reaches around 20. Our best result for RF is 12.3%. Figure 6 is the importance ranking bar plot for the most important 10 features shown below. Competitors and promotions prove to have the biggest impact on sales, whereas features. We also plotted how RMSPE changes as the duration for test data increases as shown below. We can see our RF model is still relatively accurate even for a long duration, up to 6 months.

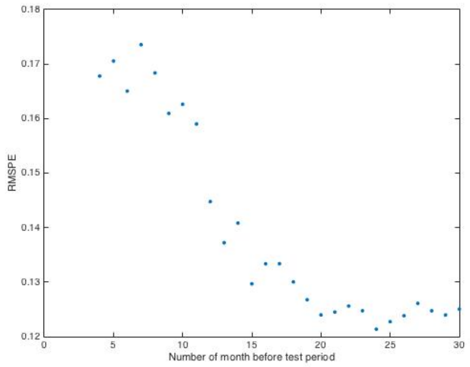


Fig. 5 How RMSPE changes with Number of month

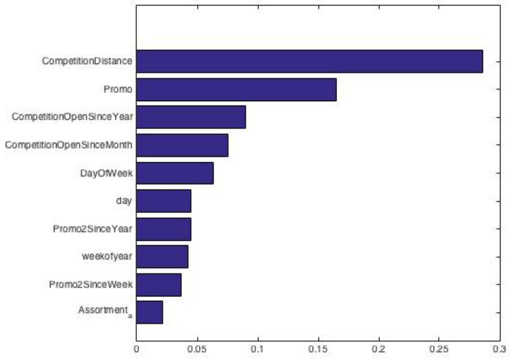


Fig. 6 Feature importance ranking (10 most important)

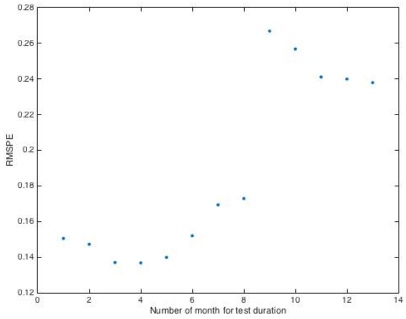


Fig. 7 How RMSPE changes with Number of month for test duration

6. CONCLUSION AND FUTURE WORK

The following table shows the results of our models.

Model	RMSPE	Remarks	Size of Test Set
Linear Regression	52.7%	For any λ	~300 days
SVR with Polynomial Kernel for Store 1	12.8%	$\lambda = 0.1\ d = 2$	~300 days
SVR with Gaussian Kernel for Store 1	13.6%	$\lambda = 140\ \sigma = 45$	~300 days
SVR with Gaussian Kernel for All Stores	Avg of 26.8%	Each store chooses its own optimal λ & σ	1 week
SVR with Polynomial Kernel for All Stores	Avg of 16.3%	Each store chooses its own optimal λ & d	3 weeks
Random Forest	12.3%	7/14-7/31/2015, 20 max features, 30 trees	2 weeks

As is shown in the result, among all models, Random Forest works the best, and provides a reliable prediction of the sales. Linear regression, SVR with Gaussian/Polynomial Kernels and RF all have their own strengths and limitations. By implementing these algorithms, we've studies the properties of the dataset and made reasonable predictions. In the future, we wish to use the fact that sales records are consecutive in time, and see how time series affect prediction result. Also there are still many effective machine learning algorithms worth trying, so we would like to try more algorithms in the future, such as Gradient Boosting and k-Nearest Neighbors algorithm.

7.REFERENCES

[1] Thiesing, Frank M., and Oliver Vornberger. "Sales forecasting using neural networks." Neural Networks, 1997.,

International Conference on. Vol. 4. IEEE, 1997.

[2] Kuo, R. J. "A sales forecasting system based on fuzzy neural network with initial weights generated by genetic algorithm." European Journal of Operational Research 129.3 (2001): 496-517.

[3] Yu, Xiaohui, et al. "A quality-aware model for sales prediction using reviews." Proceedings of the 19th international conference on World wide web. ACM, 2010.

[4] Giering, Michael. "Retail sales prediction and item recommendations using customer demographics at store level." ACM SIGKDD Explorations Newsletter 10.2 (2008): 84-89.

[5] Smola, Alex J., and Bernhard Schölkopf. "A tutorial on support vector regression." Statistics and computing 14.3 (2004): 199-222.

[6] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[7] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.