Drug Store Sales Forecast

Chenghao Wang, Yang Li

Background & Motivation

This problem was originally one of several Machine Learning problems on Kaggle (https://www.kaggle.com/c/rossmann-store-sales). 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 mean while provides a perfect real case to apply supervised learning algorithms.

Data Interpretation

Rossman uploaded the data online. The data comes in two sets

1. **Historical sales data for 1,115 Rossman stores from 2013/1/1 to 2015/6/30.** Store number, date, day of week, whether there's a promotion, whether it's a holiday and sales on that day are provided in each sample.

2. Stores' individual characteristics. This includes each store's store type, assortment level, nearest competitor's distance and date when the competitor opened, whether and when this store is on a consecutive promotion.

Although we don't have to collect, yet the data comes in a messy way, and the data is also very large. So it took us a while to preprocess the data.

70%/30% and k-fold cross validations are used in this problem for training and testing, and we used Root Mean Square Percentage Error (RMSPE) to measure accuracy, which is defined as below:

$$RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{y_i - \hat{y}_i}{y_i})^2}.$$

Single Store Prediction

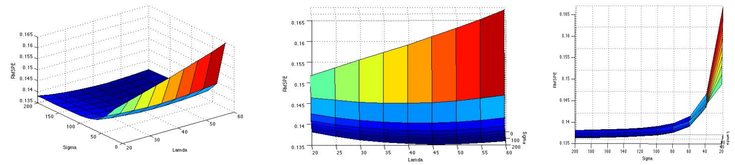
The first step for us is to analyze each store separately, regardless of each store's individual characteristics, so only consider the first dataset with historical sales data for 1,115 Rossman stores.

First we apply Linear Regression to Store 1, this gave us a RMSPE of 23.7% . Then we figured out that this problem can actually be kernelized (SVR). Here consider that case of applying MAP estimate for θ to avoid overfitting, which result in the following primal problem. $\theta = argmin||y - \theta^T X|| + \lambda||\theta||^2$

If we calculate α as $\alpha = (<X,X>+\lambda I)^{-1} y$. And define $H(X)=sum(\alpha_i<X,X_{(i)}>)$, we can see that this problem can actually be kernalized, thus we can apply the kernel trick.

We applied Polynomial Kernel and Gaussian Kernel to store 1 respectively. For Polynomial Kernel, interestingly for any d, where d means maximum degree of polynomial, the RMSPE remains the same value of 28.4%. So this problem may not conforms to a polynomial model.

For Gaussian Kernel, as RMSPE varies with λ and σ, we draw a figure to find the maximum value of λ and σ.



(a) (b) (c)

Figure 1. Effect of λ and σ on RMSPE for Gaussian Kernel.

3 different side views of this diagram is shown above. Figure (a) shows the overall shape of the surface, (b) shows that when λ =45 RMSPE reaches it's minimum, (c) shows that TMSPE decreases when σ, however when σ > 120, RMSPE becomes relatively stable, so we choose σ as 140. From the parameters above chosen, Gaussian gives us a RMSPE of 13.5%, which is better than linear regression.

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 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. We used k=10 folds cross validation for this step for RF.

In consideration of generality, we try to apply the algorithms to all stores. For Gaussian Kernel, we randomly chose several stores and tried to find their optimal λ and σ pair. We found that for all tested stores, λ =45 and σ = 140, may not be the best result, but provide a rather close result to optimal result. So we choose λ =45 and σ = 140 and apply Gaussian Kernel to all stores. The following histogram shows RSMPE of Gaussian Kernel and RF for each store.

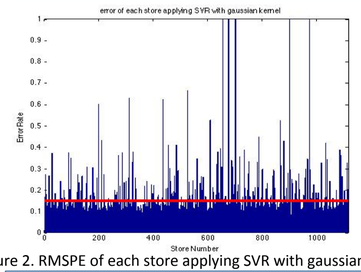


Figure 2. RMSPE of each store applying SVR with gaussian kernel

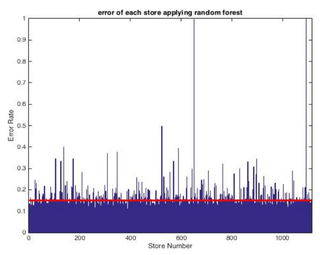


Figure 3. RMSPE of each store applying RF

From figure 2 we can see, although Gaussian works well on Store 1, it's no a robust model, which needs further improvement. If we disgard stores that has a RMSPE bigger than 1, the average of RSMPE of theses stores are 13.2%.

From figure 3 we can see, RF is also not a robust model. It has some outliers. If we remove those outliers, the average is %14.5.

Model For All Data

After trying models for each individual store, our following step is to build a common model for all stores. And of course, we need to take each individual store's characteristics into consideration, so consider both datasets.

After repeating the same method for Store 1, we try to apply SVR with Gaussian Kernel to all stores. A pair of Λ=5 and σ = 60 gives us a RSMPE of 22.4%, which is not good enough. The biggest issuse is that the data set is too large, that only a short time period could be operated on, and also it's impractical to try as many λ and σ pairs as possible.

We also applied RF after merging all data. We used dummy variables for categorical data. We also got a plot of RMSPE vs. number of trees as shown below. From this plot, we can choose the relatively smallest value, k=20 and result is 24.2%.

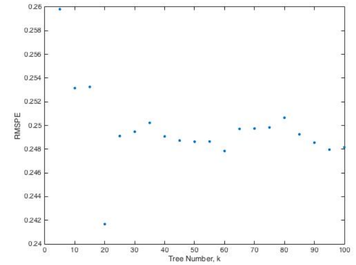


Figure 4. how RMSPE changes with the number of trees. We can choose the relatively smallest value, the tree number k=20 and the result is about 24.2%.

Comparision & Result

Model	Result	Note
Linear Regression for Store 1	23.7%	
SVR with Polynomial Kernel for Store 1	28.4%	for all d≥1 and all λ
SVR with Gaussian Kernel for Store 1	13.5%	λ =45 and σ = 140
SVR with Gaussian Kernel for all store s	22.4%	Λ=5 and σ = 60
Random Forest for all stores	24.2%	Number of trees: 20

Future Work

As you can see, linear regression, SVR with Gaussian/Polynomial Kernels and RF all have their own limitations and need further improvements. What we plan to do in the next few days is to 1. apply SVR with radial basis function kernel and probably more kernels; 2. combine SVR with RF and this may eliminate some outliers for using them separately. 3. remove some really noisy store and check the results.

Contact

Yang Li
MS @ Dept. Electrical Engineering
Stanford University
Email: yanglistanford@gmail.com

Chenghao Wang
MS @ Dept. Mechanical Engineering
Stanford University
Email: wchengh@umich.edu