Forecasting Rossmann Sales Figures

Chris Jee (cjee), Tejinder Singh (tejinder)

SCPD, Stanford University

cjee@stanford.edu

tdsingh@qti.qualcomm.com

CS 229 Project

1. Introduction

In any supply chain, an ability to accurately predict sales has a direct impact on its operating expenditure. Being able to accurately predict the sales validates understanding of the factors influencing it. A good understanding of these underling factors enable in taking "decisions" that can improve sales.

Rossmann operates over 3,000 drug stores in 7 European countries. Currently, Rossmann store managers are tasked with predicting their daily sales for up to six weeks in advance. Store sales are influenced by many factors, including promotions, competition, school and state holidays, seasonality, and locality. With thousands of individual managers predicting sales based on their unique circumstances, the accuracy of results can be quite varied.

2. Related Work

Time series prediction is a regression problem, for which there are several popular methods. Some of the methods include linear regression, softmax regression, and support vector regression.

A. Linear Regression

Linear regression uses ordinary least square optimization to fit a polynomial of degree n to the given data set to create a model. There are numerous literature discussing linear regression method with various techniques used for estimating the parameters, including gradient descent, newton's method, etc. One such literature is [1]. This model performs well for dataset that are mostly linear in nature. However, this method is not explored in this paper, as it is understood that this model will perform poorly for the given data set which has non-linear components.

B. Softmax Regression

Softmax regression was presented in CS229 lecture notes as a method of generalized linear models for multinomial distributions. In essence, it applies classification to regression data where the output labels is not a binomial but spans multiple values. If the label to be predicted has a known range of values with discrete set of values it can take or the discretization error is acceptable, then this method can be applied as regression model for continuous data. This method is further explored in later section of this paper.

C. Support Vector Regression

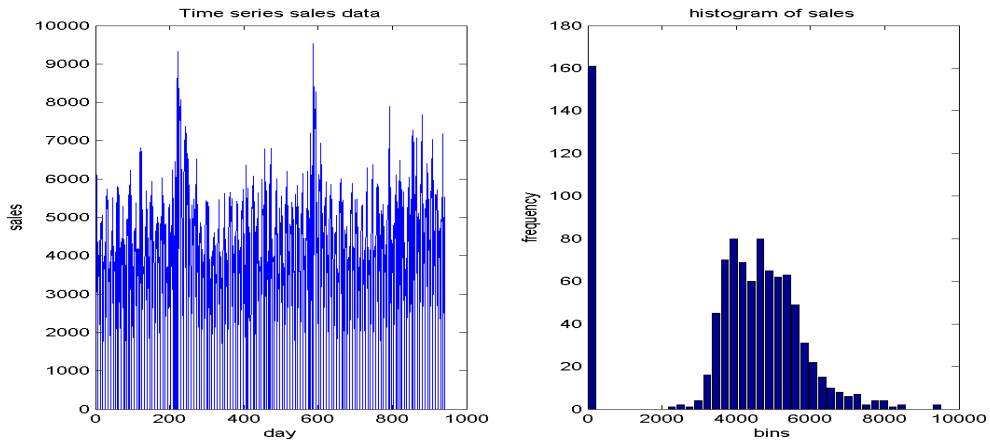
Support Vector Machine (SVM) concepts can easily applied to regression problems while keeping many of the benefits of SVM such as use of mapping functions to map seemingly nonlinear data into mostly linear data in higher dimensions, use of kernel tricks, individualizing hyperplanes, limiting the model parameters to support vectors that contribute, and maximizing the margins. Because the label is a real number, a margin of tolerance is set to allow an approximation to the SVM.
Different kernels and regularization parameters can be chosen to suit the particular data set to improve the results [3]. Further discussion of SVM and its application to the given problem is discussed in later section.

3. Dataset and Features

The data provided for training tags the stores into four different categories. Sale data, the "target variable" of every store is provided along with various features like the distance of the nearest competitor, promotional activities and number of customers to name a few. Data provided is a mixed set of continuous and discrete variables.

Training was done across the stores resulting in one prediction model for every store. This made us drop the features like competition distance and store category which remain constant for any given store. The four features chosen for training are day of the week, month, promotional activity and school holiday. Let $x^{(t)} \in R^4$ represent the feature vector at time t. As the number of features is small, feature reduction using PCA was not considered. All the approaches tried used the same feature subspace.

Sales data, the quantity to be predicted across time, is preprocessed. Preprocessing consists of two stages. In the first stage the minimum recorded sale of a given store, y_{min} is subtracted from the sale data making zero as the minimum value of result. The second stage quantizes the sales data into discrete quantization levels. Histogram of the sales data is computed to divide it into multiple bins, where the number of quantization levels k is controlled by adjusting the width of the bins Δ.



4. Methods

Softmax Algorithm

The preprocessed sales data y is quantized such that it can take any one of k values, so that $y \in \{1, \dots, k\}$. y is modelled as distributed according to multinomial distribution. Let the parameters $\emptyset_1, \dots, \emptyset_k$ specify the probability of each of the outcomes. Following is the hypothesis for softmax regression

$$p\left(y = i \middle| x^{(t)}; \theta\right) = \frac{e^{\theta_l^T x^{(t)}}}{\sum_{j=1}^{k} e^{\theta_j^T x^{(t)}}}$$

Where

$x^{(t)}$ is the feature vector at time t and $x^{(t)} \in R^n$

θ parameterizes the model and $\theta \in R^{n \times k}$

θ_l represents the l^{th} column of θ

k is the number of quantization levels

θ is obtained by maximizing the log likelihood function is defined as follows

$$l(\theta) = \sum_t \log \prod_{l=1}^{k} \left(\frac{e^{\theta_l^T x^{(t)}}}{\sum_{j=1}^{k} e^{\theta_j^T x^{(t)}}} \right)^{1\{y=l\}}$$

A value of θ is trained for every store, making one model per store. The predictor estimates the bin, $\hat{y}$ for every query point. The sale data is reconstructed using the following. 0.5 in the equation is used to align to the center of the bin.

$$sales = \ y_{min} + \ \Delta(\hat{y} - 0.5)$$

Support Vector Regression

Support Vector Regression is an application of Support Vector Machine principles applied to regression problems. The main objective function and the constraints are similar to that of Support Vector Machines, but it adds new parameter epsilon as the margin of tolerance and C for penalty factor for errors. The objective function and constrains are shown below [4]:

$$min_{\gamma,\omega,b} \frac{1}{2} ||\omega||^2 + C \sum_{i=1}^{m} (\xi_i - \xi_i^*) \ s.t.$$
$$y_i - \omega x_i - b \leq \varepsilon + \xi_i$$
$$\omega x_i + b - y_i \leq \epsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \geq 0$$

The kernel functions transform the data into higher dimensional feature space, to make it possible to perform the linear separation. Several different kernel functions are available, but two of the popular kernels are polynomial kernel and Gaussian radial basis function (RBF), which are shown below [4]:

$$polynomial:$$
$$k(x_i, x_j) = (x_i \cdot x_j)^d$$

$$Gaussian\ RBF:$$
$$k(x_i, x_j) = -\exp\left(\frac{||x_i - x_j||^2}{2\sigma^2} \right)$$

The main idea for Support Vector Regression remains the same as that of Support Vector Machines, which are to minimize error, individualize the separating hyperplane by mapping the input to higher dimension feature space while maximizing the margin. The introduction of the parameter epsilon and C allows control of how much error is tolerated and how much errors are penalized respectively.

5. Experiments Discussion and Results

Softmax Algorithm

Gradient ascent was employed to estimate the value of θ for every store. Initially for the ease of implementation gradient of log likelihood function was computed using first principles. The execution time for this was extremely slow to the point that the approach was prohibitive in spite of making a vectorized code. This lead to deriving and computing the following gradient in the closed form.

$$\frac{\partial l(\theta)}{\partial \theta_{pq}} = \sum_t f(t, p, q)$$

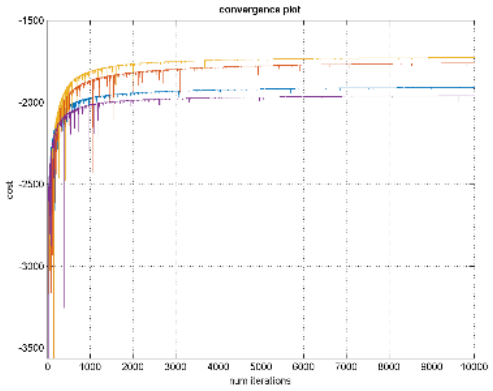
Where

$$f(t,q,p) = \begin{cases} x_p^{(t)}\left(1 - \dfrac{e^{\theta_q^T x^{(t)}}}{\sum_{j=1}^{k} e^{\theta_j^T x^{(t)}}}\right), & y^{(t)} = q \\ -\dfrac{x_p^{(t)} e^{\theta_q^T x^{(t)}}}{\sum_{j=1}^{k} e^{\theta_j^T x^{(t)}}}, & otherwise \end{cases}$$

Gradient ascent employing this closed form expression for gradient was approximately14x faster than the implementation using first principles. The implementation of closed form gradient was tested against the originally developed gradient using first principles. The Error Vector Magnitude (EVM) between the two implementations was observed to be close to -37dB.

Our initial approach was to try multiple fixed values of learning rate α but the results were not encouraging and the cost function to be maximized would start approaching $-\infty$ in a couple of iterations. This led us to using a heuristic that would adapt the learning rate in every iteration. Every step that resulted in a lower value of cost function than the current value was reverted and the value of α reduced. α was increased in the steps that showed an increase in the value of cost function. To make the learning rate biased towards decreasing the factor by which α is increased is chosen to be less than the factor that decreases it. This approach alleviates the burden of choosing α and convergence can be achieved by executing sufficient number of iterations.

The minimum number of bins and the width of the bins are the tuning parameters for this approach. Large execution times prevented us from sweeping across various values of these parameters. Minimum number of bins tried was 40 and 60, while the width of the bins was set to not greater than 300.



Support Vector Regression

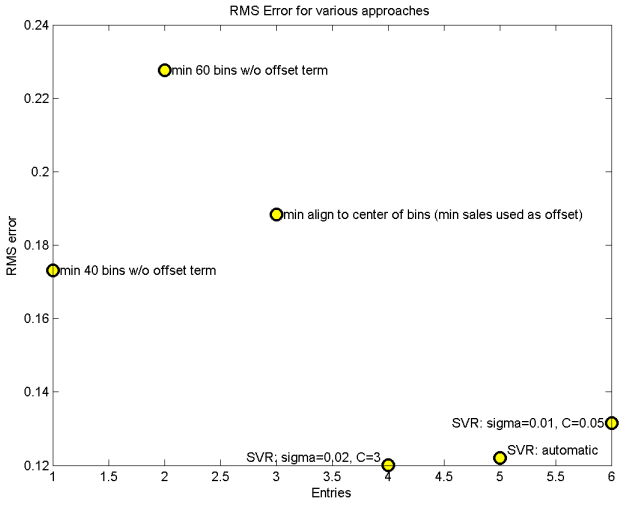
An important aspect of applying Support Vector Regression is the choice of parameters (σ, C). It is not immediate apparent what values should be chosen for these. A typical approach would be to do some form of tuning process to find the optimal value, such as grid search on different combinations of these values to minimize the root mean square norm error (RMSE) of the estimated label.

The SVR was tuned by sweeping over various values of C and σ and computing the RMS error of predictions. This is a compute intensive process.

σ	C=0.02	C=0.05	C=1	C=2
0.01	0.08880093	0.0898571	0.1321967	0.1561132
0.02	0.08926462	0.09032577	0.110581	0.1186986
0.05	0.08967524	0.0902758	0.09571032	0.09894424
0.1	0.08977889	0.0898571	0.09304627	0.09478889

Results

The following plot summarizes the results obtained by employing the various approaches described above.



6. Conclusion and Future Work

For the two methods tried on the problem, SVR algorithm performed significantly better than softmax algorithm. It is also worthwhile to note that the execution time for SVR was typically around 15 minutes while softmax algorithm took nearly 12 hours.

Given more time and resource, it may be worthwhile to fine tune SVR model further for the given problem. The areas to improve the SVR model would be:

- Group the stores by some combination of common features and choose different set of features for each group to train SVR model optimally. One simple way to group the stores would be to use k-means clustering algorithm.
- Given that dataset is time series distribution, introduce weighting factor that puts more emphasis on more recent results than the older results [6].
- Explore use of local support vector regression that allows SVR to automatically adjust the parameter **ε** [7].

7. References

[1] Björck, Åke (1996). "Numerical methods for least squares problems." Philadelphia: SIAM. ISBN 0-89871-360-9.

[2] Chih-Chung Chang and Chih-Jen Lin, "Training ν-Support Vector Regression: Theory and Algorithms", National Taiwan University

[3] Alex J. Smola*, Bernhard Scho¨lkopf, Klaus-Robert Mu¨ller, "The connection between regularization operators and support vector kernels, Neural Networks" 11 (1998) 637–649, GMD First, Rudower Chaussee 5, 12489 Berlin, Germany

[4] Alex J. Smola, Bernhard Scholkopf, "A Tutorial on Support Vector Regression", NeuroCOLT2 Technical Report Series NC2-TR-1998-030, Oct. 1998.

[5] Charles H Martin, “Kernels Part 1: What is an RBF Kernel? Really?”, https://charlesmartin14.wordpress.com/2012/02/06/kernels_part_1/
[6] Alex J. Smola, K.R. Muller, G. Ratsch, B. Scholkopf, J. Kohlmorgen, “Using Support Vector Machines for Times Series Prediction”, Ruower Chausee 5, 12489 Berlin, Germany
[7] Rodrigo Fernandez, “Predicting Time Series with a Local Support Vector Regression Machine”, LIPN Institute Galilee-Universite Paris 13