Predicting Retail Sales with Weather

Jessica Jones
Computer Science, University of New Mexico
Albuquerque, NM, USA
e-mail: jjones203@unm.edu

***Abstract*—Retailers must forecast customer demand in order to stock the right products in the right quantities. Companies like Walmart analyze sales data to prevent "a retailer's twin nightmares: too much inventory, or not enough" [1]. For a previous Kaggle competition, Walmart released a dataset for 45 stores over 15 months [2]. The data included the daily units sold at each store of 111 items for which demand might vary with the weather, such as milk or umbrellas. Data from the weather station nearest each store was provided as well. Using a collaborative filtering approach, I sought to predict sales of each item at each store based on the daily average temperature. I found the most accurate results by assigning each temperature to a range, then using the range as the basis of the predictions.**

Keywords-sales forecasting, weather, collaborative filtering

I. Introduction

When ordering inventory, retailers walk a tightrope. If they order too much of a product, they will lose money on the unsold units; this is a particular concern for perishable or seasonal goods, which have a short shelf life. If they fail to order enough of a product, however, they lose potential sales and disappoint customers. They may even face penalties beyond their bottom line. As Itakura and Minazuki describe, Seven-Eleven Japan received a warning from the country's fair trade commission for "abandonment sales" of food products in 2009 [3]. Not only can analysis help companies avoid these pitfalls, but it can also help them discover opportunities. Using sales data from areas hit by Hurricane Charley in 2004, Walmart increased its inventory of Pop-Tarts and beer before Hurricane Frances reached Florida, knowing these items would sell quickly [1]. This example illustrates how retailers can benefit from factoring weather conditions into their sales forecasts.

II. Literature Review

Several researchers have applied machine learning techniques to sales forecasting, often including weather in their models. Traditionally, managers have started with a moving average or autoregressive moving average (ARIMA), then adjusted the results based on personal experience, upcoming holidays and sales, etc. Itakura and Minazuki [3] developed a computer program to help convenience store employees decide how much of a prepared food, such as soup, to put on display. The program takes a decision-tree approach: given the temperature, wind speed, and time of day, how much soup will likely be sold in the next three hours? Unfortunately, the store employees found the program difficult to use, so the study did not yield meaningful results.

Chen and Ou [4] also worked with a convenience store, attempting to predict the sales of refrigerated foods over the next several days. They tried various neural network models and noticed that "changes in temperature had a noticeable influence on cold chain logistics." Their ANN with lagging sales data and local temperature reduced the mean square error over 45% compared with the ARIMA model.

Sligro Food Group, a Dutch wholesaler and retailer, provided data for two studies. The first used ensemble learning with dynamic integration to predict weekly sales [5]. A pool of 24 heterogeneous classifiers, trained with 13, 26, and 52 weeks of data, was evaluated with Model Weighted Voting. The most accurate ensembles incorporated external features; the models typically selected temperatures and promotions from the external feature set.

The later study tried a context-aware prediction approach to forecasting Sligro's sales [6]. Each product was assigned to a category based on its sales pattern: flat, frequent, occasional, or seasonal. The researchers tried assigning the items with a decision-tree classifier and (slightly less accurately) by clustering the items around predefined centroids. They then developed a base predictor for each group, using both sales-related features and external features (such as weather or holidays). While more than 40% of the products were assigned to the correct group, the authors estimate they must achieve 85% accuracy in categorization to outperform the moving average in predictions. These diverse approaches to sales forecasting demonstrate both the difficulty of the problem and the interest in addressing it.

III. Data

The data was provided in three different files that needed to be cross referenced. The training set included the date, store number, item number, and number of units sold: e.g., (2012-04-21, 45, 6, 0). The item numbers were unique to each store; milk might be item 6 at one store and item 20 at another.

The weather data was indexed by station number and date. The weather-related fields included average temperature as well as precipitation, wind speed, and so on. A key gave the closest weather station to each store.

To use Spark's collaborative filtering information, the data must be turned into Rating objects. A rating is a triple consisting of a user number, item number, and rating, all integers. This constrained me from considering as many features as I would have liked. Initially, I created Ratings consisting of (StoreId + ProductId, Temperature, Sales). I then created functions of temperature and date while attempting to increase the model's accuracy.

The data was quite sparse. Some of the weather stations did not have the temperature data for all of the dates in the training set. I built over 4.4 million Rating triplets, but over 4.3 million of them had a value of 0 for sales. While collaborative filtering typically involves a sparse matrix, perhaps another approach could have better exploited the particular characteristics of this dataset in order to compensate for it (for example, looking for commonalities among sales of different products at a single store).

IV. Method

I used Spark's mllib package to create a recommender using collaborative filtering. Spark trains a matrix factorization model using regularized alternating least-squares. The algorithm alternates between adjusting the rows and columns of the user-item matrix until the error (the difference between users' actual ratings and the dot-product of the associated user and item vectors) converges. The model can then generate a predicted rating for a given user and item.

For the sake of comparison, I used the same parameters for each model: 12 iterations, rank of 4, and lambda of 0.1. I split the data into 80% for the training set and 20% for the testing set.

I tried various combinations of data as the “product” which was being rated by the sales number. I started with temperature alone. I also considered:

- Temperature range. I approximated temperature ranges using integer division; for example, product number = temperature/10. The temperatures in the data ranged from -16 to 100, so this would result in 12 ranges.
- Temperature range and week of the year, to account for seasonality. In this case, I multiplied the week by 1000 and added the temperature range. Thus days in the same week within the same temperature range would have the same product number.
- Temperature range, week of the year, and weekday vs. weekend. The product number was a linear combination of these factors, as described above.
- Average of multiple temperature ranges. I would have liked to have averaged multiple temperature ranges; for example, calculating the ranges for the stated temperature, the temperature – 3, and the temperature + 3. Unfortunately, it was difficult to achieve this using an RDD of Ratings as required by Spark.

V. Experiments and Results

Coalescing the temperature values into ranges increased accuracy. I tried a variety of possible divisors for creating the temperature ranges, attempting to minimize the RMSE (root mean squared error). Temperature and week of year, considered independently, yielded nearly identical results. Accuracy improved when both were used to make a prediction. Surprisingly, combining the temperature range, season, and weekend information did not appear to significantly change best-case accuracy. With a given approach held constant, changing the temperature range caused fluctuations in the RMSE. Table 1 shows the best observed results.

TABLE I.

Item	Collaborative Filtering: Item Choices and Results		
	RMSE	***RMSLE***	***MAE***
Temperature	6.23201	0.151221	0.397451
Range (Temp / 7)	5.72847	0.150829	0.396266
Week	6.21534	0.146986	0.387914
Week, Range (Temp / 20)	6.09759	0.145527	0.379715
Week, Weekday/End, Range (Temp / 4)	6.34212	0.146065	0.367966

The contest for which this data was intended evaluated entries using RMSLE (root mean squared log error). While RMSE penalizes large errors, RMSLE penalizes predictions that are too small. As mentioned above, on most days, stores sell zero units of the items in our dataset. So we might expect that errors of large magnitude might occur on the days when customers do buy multiple units, increasing the RMSE. Similarly, as a high-volume retailer, Walmart is likely more concerned about having too few units of a product than too many. If we want to prevent underestimating sales, RMSLE is a logical metric for our problem.

Mean average error (MAE) considers all errors equally. The MAE results were consistently small; again, this corresponds to a dataset with many zero values.

Although Spark presented challenges in terms of representing and analyzing the data, its performance was excellent. Training and testing the model takes less than five minutes on a single node. I used several of Spark's “lazy formations” to format the data as required by the model. In a different framework, this data processing would probably have taken much longer.

VI. Conclusion

As researchers and businesses have found, temperature alone cannot predict sales, but it can certainly influence them. Avenues for future work include content-based recommendation systems and decision trees. To utilize a content-based recommendation system for this problem, we would treat each day as a product. We could compare days based on their features: weather, time of year, etc. Using similarity measures (e.g., cosine similarity), we could estimate whether a given day's features would likely result in higher or lower than average sales. After analyzing the relative importance of the features, we could create a decision tree that would consider each in term. For example, perhaps the root node would represent temperature range. For the lower temperature ranges, the next node might determine whether the date fell before or after Christmas. Like a content-based recommendation system, this approach would allow us to apply our knowledge of the problem in order to refine the solution.

References

[1] C.L. Hays (2004, Nov. 14). “What Wal-Mart Knows About Customers' Habits”. New York Times [Online]. Available: http://www.nytimes.com/2004/11/14/business/yourmoney/what-walmart-knows-about-customers-habits.html. [Accessed: Dec. 11, 2015].

[2] Kaggle. “Walmart Recruiting II: Sales in Stormy Weather,” kaggle.com. [Online]. Available: https://www.kaggle.com/c/walmart-recruiting-sales-in-stormy-weather. [Accessed: Dec. 11, 2015].

[3] Y. Itakura and A. Minazuki, “A Study of the Support System for Displaying Food Products in Convenience Stores,” in *9th IEEE/ACIS International Conference on Computer and Information Science*, Yamagata, Japan, August 18-20, 2010, T. Matsuo, N. Ishii, R. Lee, Eds. Piscataway, NJ: The Institute of Electrical and Electronics Engineers, Inc., 2010. pp 421 – 426.

[4] F.L. Chen and T.Y. Ou, “A Neural-Network-Based Forecasting Method for Ordering Perishable Food in Convenience Stores,” in *Fourth International Conference on Natural Computation*, *2008*, Jinan, China, Oct. 18-20, 2008, M. Guo, L. Zhao, L. Wang, Eds. Piscataway, NJ: The Institute of Electrical and Electronics Engineers, Inc., 2008. pp 250 – 254.

[5] P. Meulstee and M. Pechenizkiy, “Food Sales Prediction: "If Only It Knew What We Know",” in *IEEE International Conference on Data Mining Workshops*, *2008*, Pisa, Italy, Dec. 15-19, 2008, F. Bonchi et al, Eds. Piscataway, NJ: The Institute of Electrical and Electronics Engineers, Inc., 2008. pp. 134 – 143.

[6] I. Zliobaite, J. Bakker, and M. Pechenizkiy, “Towards Context Aware Food Sales Prediction,” in *IEEE International Conference on Data Mining Workshops*, *2009*, Miami, FL, USA, Dec. 6, 2009, Y. Saygin et al, Eds. Piscataway, NJ: The Institute of Electrical and Electronics Engineers, Inc., 2009. pp 94 – 99.