Fig. 1 Time series sales data used in this work

Sales Prediction with Time Series Modeling

Venkateswarlu.S

I. Introduction

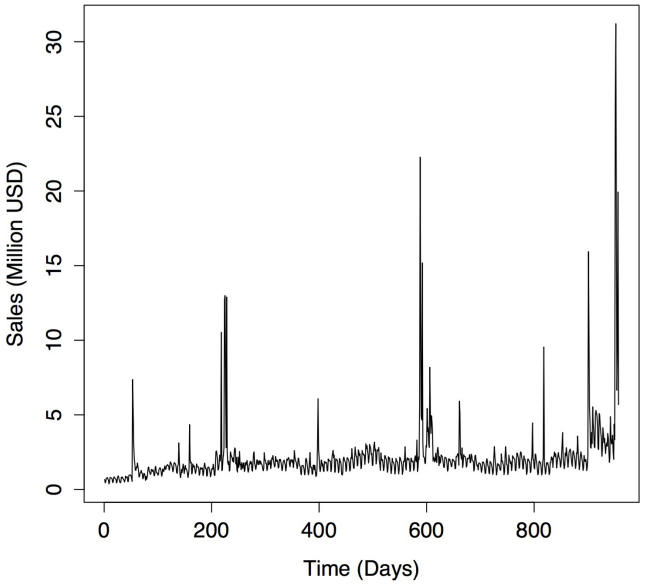
Predicting sales-related time series quantities like number of transactions, page views, and revenues is important for retail companies. Our work focuses on the revenue data for a US-based online retail company (Digital River, Inc.) that is responsible for the ecommerce platform of its clients. As online sales are increasing at a massive rate, accurate prediction of sales allows the company to properly prepare for handling the shocks to product stock, website traffic, and customer support.

During the 3 biggest days – Thanksgiving, Black Friday and Cyber Monday – the company earns about 10% of the revenue of the whole year, so it is considered a leading indicator of overall sales. Predicting revenue on special days such as this is especially challenging, as those are usually large spikes (i.e. anomalies) compared to normal days. For example, gross sales on Black Friday are usually more than 10 times of the median sales of the year. Our data was limited to only 2-3 years of Black Friday, Cyber Monday, and holiday season sales data so building a robust model is difficult because these special incidents have only a few data points.

II. Data and Prior Work

Time series forecasting grew out of econometrics and involves parameter fitting using data to predict future values of some quantity. The input data consists of pairs (r_t, t) of some quantity r at time t. Unlike the i.i.d. observations prevalent in most of machine learning, time series data points are emphatically not independent, and in fact we rely on their autocorrelation structure to forecast the future.

Traditional forecasting tools are based on autoregression (AR) and moving averages (MA), which are described below. In addition to these, we will use feed-forward artificial neural networks with time series inputs to see how they perform. In principle, neural nets have greater freedom to handle nonlinearities, but they are also non-convex black boxes while AR and MA have the advantage of being interpretable and parsimonious because they are specific to time series. Neural nets were popular for time series forecasting in the 1990's, but interest died down due to mixed results relative to AR and MA models [1][2]. They have been used specifically for sales forecasting with some success [3][4].



The data we will use for forecasting has been taken for one large client of Digital River from April 2013 until the present. For this data set itself, prior predictions by the company have been carried out by moving averages, which have low accuracy. Naïve seasonal methods have also been used, which gives decent accuracy but no dynamism since it's merely repeating the prior year's observations. We will attempt to use learning and prediction to forecast this series, with particular attention to the distinctive spikes in sales on holidays.

III. Methods and Features

Autoregressive integrated moving average (ARIMA)

The baseline time series modeling methods are [5]:

1. Autoregression (AR): the output at time *t* is a linear combination of past outputs

$$r_t = c + \epsilon_t + \sum_{i=1}^{p} \phi_i r_{t-i}$$

2. Moving average (MA): the output at time t is a linear combination of past shocks (noise terms)

$$r_t = \mu + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}$$

AR and MA can be combined. They can also be applied to the differenced series (i.e. an approximation to the derivative) of the desired order and integrated to retrieve the original series. Putting these three features together yields the more general Autoregressive Integrated Moving Average (ARIMA) model:

$$d_t = c + \epsilon_t + \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}$$

The order (*p,d,q*) of the ARIMA model specifies the number of autoregression lags, order of differencing, number of moving average lags, respectively. Our model chose these parameters based on the Akaike information criterion (AIC), which trades off the likelihood of the model against the number of parameters. It is therefore a regularized maximum likelihood estimate and theoretically minimizes 1-step MSE.

Because we have multiple seasonality components and discrete spikes in the data to deal with, we have used ARIMA with additional regressors. One set of regressors are the first 3 to 15 largest Fourier components for weekly and yearly seasonality. The other set are indicator vectors that are 0 throughout the year except for 1's on certain special days like Thanksgiving, Black Friday, and Christmas. The latter is necessary (rather than simply using the input $r_{t+\Delta t}$) because some of these days change dates every year (i.e. always on a Friday) and because of leap years.

Seasonal and Trend decomposition using Loess (STL)

STL Decomposition is a useful tool for accounting for seasonal effects. In sales data, it is common to have repeating patterns every 24 hours, or every 7 days, or every 365 days due to work/sleep cycles and holidays. It is beneficial to remove these consistent fluctuations so that the model parameters have greater freedom to fit any underlying trends, i.e. broad changes, and then add the periodic pattern back in as a post-processing step. Both our ARIMA and neural net models made use of STL decomposition.

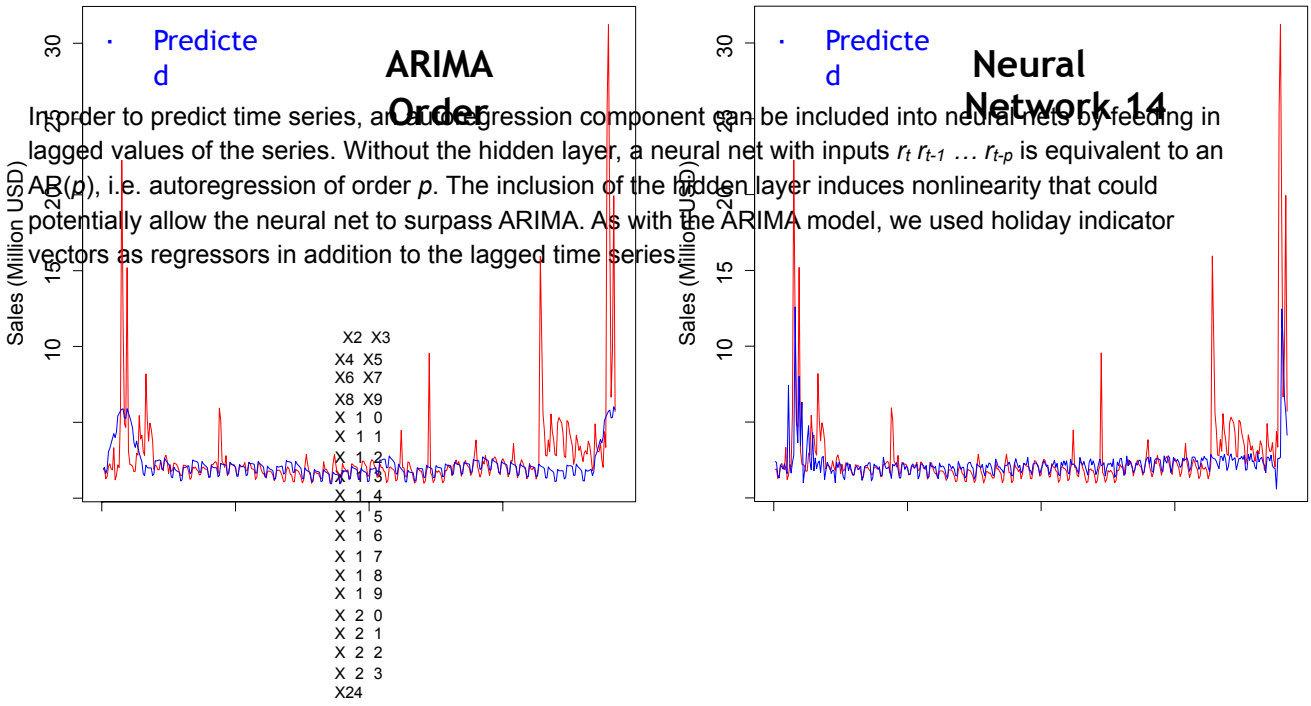
Feed-forward Neural Networks (FFNN)

A feed-forward neural network weights its inputs and feeds them into hidden layers that apply some function to these internal inputs. The input y_j into the j^{th} hidden layer is:

$$y_j = b_j + \sum_{i=1}^{n} w_{ij} x_i$$

After which y_j is transformed using a sigmoidal function for categorical or probability outputs or a linear function for regression outputs, which is applicable to our case. The parameters b_j and w_{ij} are learned from the data and used to make future predictions. Fig. 2 shows a visualization of one of the actual neural networks used in this work, with the line thicknesses encoding the trained weight parameter values.

Fig. 3 Sales forecasts using ARIMA (left) and neural nets (right)



In order to predict time series, an autoregression component can be included into neural nets by feeding in lagged values of the series. Without the hidden layer, a neural net with inputs r_t r_{t-1} … r_{t-p} is equivalent to an AR(p), i.e. autoregression of order p. The inclusion of the hidden layer induces nonlinearity that could potentially allow the neural net to surpass ARIMA. As with the ARIMA model, we used holiday indicator vectors as regressors in addition to the lagged time series.

IV. Results

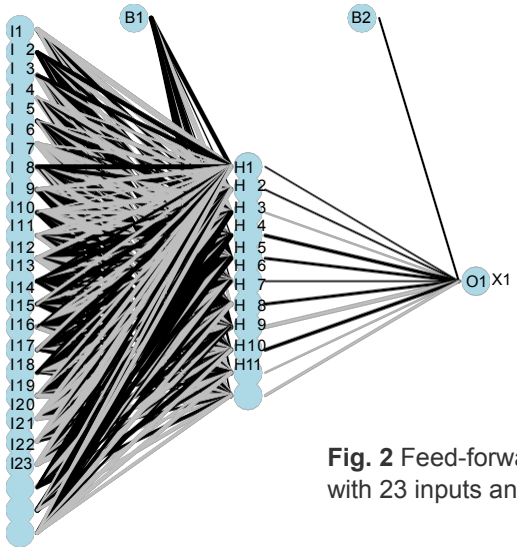


Fig. 2 Feed-forward neural network with 23 inputs and 11 hidden layers

0 100 200 300
Time
(Days)

0 100 200 300
Time
(Days)

Our 390-day sales forecast is shown above for an ARIMA(5,2,0) model and a neural net with 10 autoregression lagged inputs, 5 indicator vectors for special days, and 14 hidden nodes. The ARIMA model had a higher mean square error (MSE) of 59 to the neural net's 49, but their failures are similar.

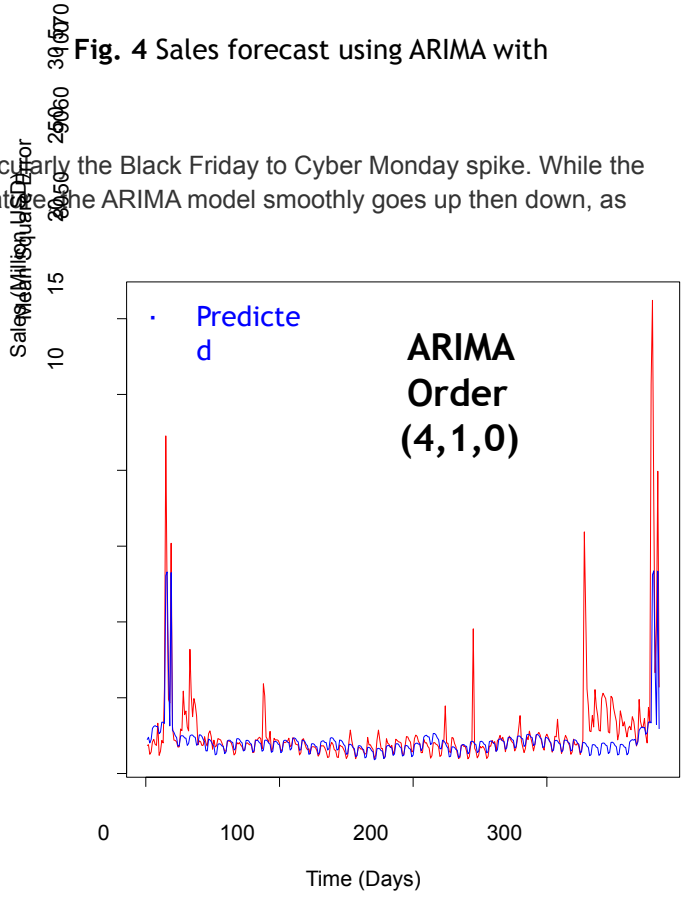


Fig. 4 Sales forecast using ARIMA with

Both severely underpredict holiday sales, particularly the Black Friday to Cyber Monday spike. While the neural net successfully captures its discrete nature, the ARIMA model smoothly goes up then down, as expected from its equations.

The notable spike at around day 320 highlights the difficulty of this prediction task: this particular jump in sales was induced by a hyped-up new product launch and thus could not have been predicted using any of our input features. A hypothetical feature vector to handle this would need to be trained on previous product launches and needs to somehow encode the event into a representative number.

Since the neural net can qualitatively capture the holiday spike, it would be interesting to combine linear regression with holiday indicator variables into the ARIMA equation to capture the same effect. Fig. 4 shows such a model, where the optimal order has now changed to (4,1,0) per the AIC criterion and the MSE decreased further to 34.

Parameter Fitting in Neural Nets

The neural net optimization problem is non-convex when there are one or more hidden layers, so the solution lands in a local optimum almost every time and differs for different initial parameters. We used 100 runs of the neural net initialized randomly using different seed values and averaged the predictions.

The number of hidden nodes was chosen by test error minimization, but the caveat is that the actual MSE was close in magnitude to similar models and all of them are ensemble local optima solutions so it's possible the true global minimum is not at our chosen value of 14.

The order of autoregression was similarly chosen by minimizing test MSE, but we found that beyond 5 terms made little additional difference. The same caveat applies, although the anticipated effect is small since MSE values did not differ much.

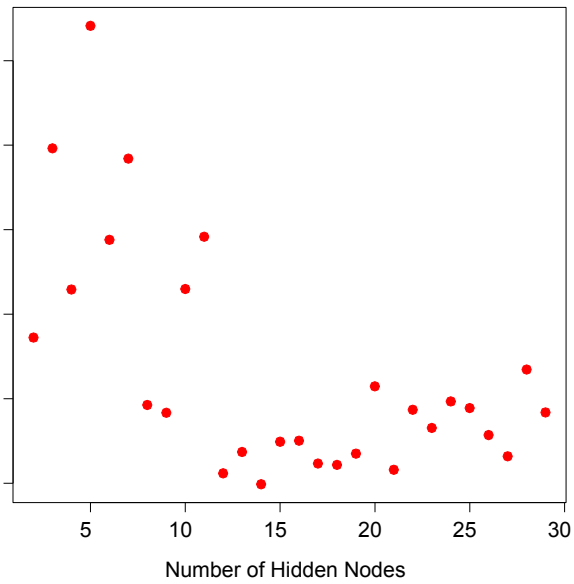
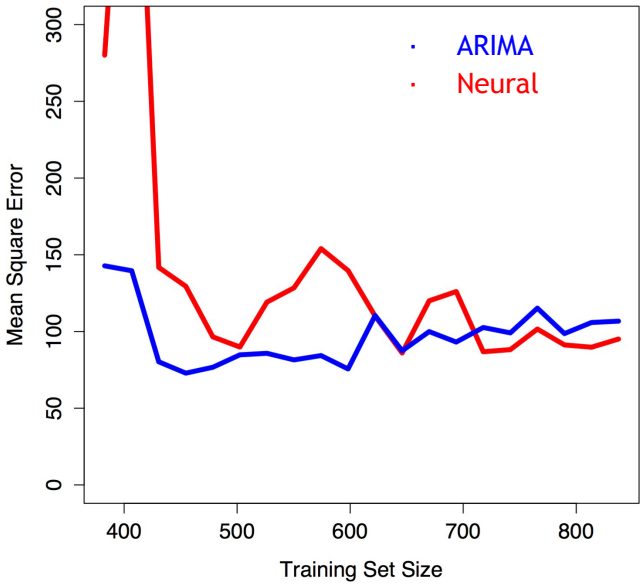


Fig. 5 Test MSE against hidden node count



The learning curve for our time series data is shown in Fig. 6. The curves are non-monotonic since different set sizes entail different forecast intervals and some parts of the series are more difficult to capture. Nevertheless, we can observe that neural nets have much higher generalization error for low training set sizes but become roughly comparable as the set size increases.

Discussion and Future Work

Pure black box time series prediction can only do well under limited circumstances, such as highly seasonal data with no long-term changes. The greatest improvement to our models could come from the use of domain knowledge to construct highly relevant regressors that can account for discrete events (e.g. a product release) or long-term changes (e.g. company or economy is on the upswing). We demonstrated this with the use of indicator vectors for special days like Thanksgiving but much more could be done given greater domain knowledge and data.

The issue of whether neural nets can outperform conventional time series models remains open. We showed that they are comparable, but much greater effort needs to be put into the neural net to achieve even that. In principle, neural nets would have an advantage if interacting regressors were useful for prediction because the hidden layers can capture that cross-term. Another improvement that could be made to the neural net is to input differenced time series as in ARIMA and integrate after prediction in order to capture the non-stationarity that is typical in sales data. Another path to explore is the decomposition demonstrated in [6] demonstrated some success in using ARIMA to model the linear component of the data and neural nets to model the residual nonlinear component