BIOS 735 Project Report

Abby Foes, Zheng Lan, Justin Landis, Yu Liu, Alec Reinhardt

April 2025

1 Introduction

Data Description

We have identified a dataset of pasta sales from an Italian grocery store with observations made between January 2014 and December 2018 (Mancuso et al., 2021). This dataset contains the quantity of sales for 118 unique items across 1,798 equally-spaced time points (days). Along with this quantity of sales, we are provided with a brand identifier and a binary label of whether a given item was promoted on that day as well. This time series data is thus hierarchical, with four brands and up to 45 unique items in each brand.

Variable	Type	Description
DATE	Date	Date of sale (YYYY-MM-DD)
brand	Categorical	Brand identifier ("B1")
item	Integer	Item quantifier within brand
QTY	Integer	Quantity sold on that date
PROMO	Binary	Promotion indicator (0 = No, 1 = Yes)

Table 1: Pasta Sales - UCI Machine Learning Repository

Figure 1 provides a visualization of pasta sales on a log-scale with a gradient color label for the proportion of items in each brand on promotion at a given time. In general, we see that as more items are on a promotion, there are more sales, as witnessed by the darker lines at peaks in sales. We can also see that Brand 2 appears to have a variance in sales with a near constant promotion of at least one item across the brand.

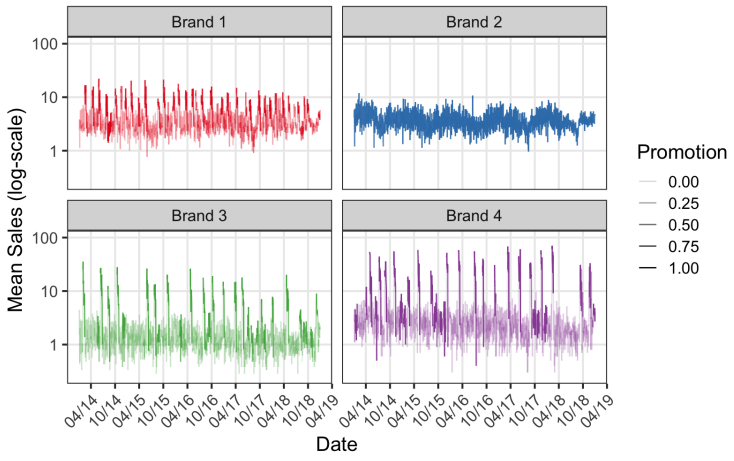


Figure 1: Mean sales by brand on a log-scale with promotion as a proportion of brand-level items on promotion.

We can also visualize correlation of items within each brand with Figure 2. Strong positive correlations suggest that promotional or temporal effects may extend across multiple items within a brand.

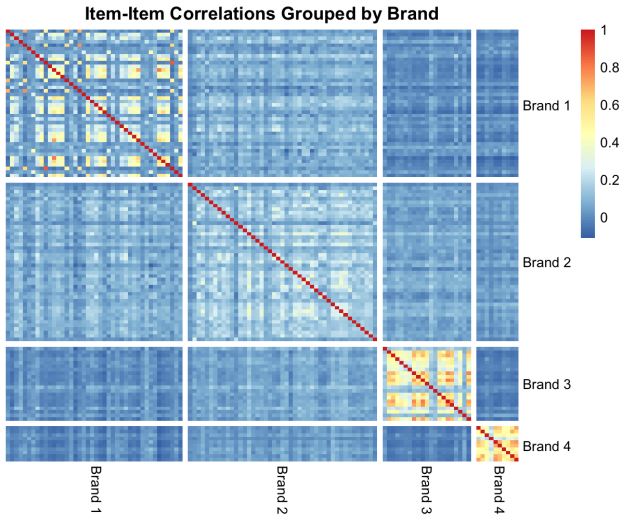


Figure 2: Heatmap displaying within-brand item-level correlations

Research Questions

The goal of this report is to investigate the following:

1. How do temporal trends and promotion-sales relationships vary across the four pasta brands?
2. Can we improve forecasting of sales by accounting for the hierarchical structure in the data compared to simply modeling each brand (and item) independently?

We compare likelihood-based models (BFGS), Bayesian inference (MCMC, Metropolis-Hastings), and machine learning benchmarks (Random Forests), using out-of-sample mean squared error (MSE) as the main evaluation metric.

2 Methods

2.1 Motivation

To address our research goals, we consider several variations of time series models. The key commonality behind these model is that they all use information from the previous history of sales as a predictor of the current sales for a given item. These types of models are known as autoregressive (AR) models, which have been widely-used across disciplines for time series forecasting (Hamilton, 2020). Additionally, our models all include the current promotion status for a given item as a predictive of the current sales for that item.

One issue that arises from this dataset is that the sales outcomes are discrete counts. Classical AR-based approaches, such as the Autoregressive Moving Average (ARMA), make normality assumptions about the current count outcomes given past history, but the model support is over the real line with a constant variance assumption. However, given the discrete, non-negative integer data, a more appropriate assumption in our case is that the sales are distributed according to a Poisson distribution. We can then incorporate the autoregressive structure into the Poisson mean parameter, leading to a Poisson autoregressive model (PAR), formulated below. To deal with the inherent correlations across items within each brand, we also formulate a variation of this PAR model that can jointly model sales across all items, after accounting for similarities by brand.

2.2 Autoregressive moving-average model (ARMA) for single item modeling

We first consider the classic ARMA model for time series data. Formally, an ARMA(p,q) model is given by

$$y_t = \alpha + \sum_{l=1}^{p} \beta_l y_{t-l} + \gamma x_t + \epsilon_t + \sum_{s=1}^{q} \epsilon_{t-s}\theta_s$$
$$\epsilon_t \overset{i.i.d.}{\sim} N(0, \sigma^2)$$

where y_t is the sales at timepoint t, x_t is the promotion indicator at time t, and ϵ_t is so-called i.i.d. white-noise.

2.3 Poisson autoregression (PAR)

For a given item, denote the sales at timepoint t by y_t, where $t = 1, \ldots, T$. Also let $\mathbf{x}_t$ be a control vector including promotion indicator at time t (and a 1 element for an intercept). Based on the work by Brandt and Williams (2001), we formulate the single-item Poisson autoregression model (PAR) as

$$y_t \sim \text{Poisson}(m_t) \tag{1}$$
$$m_t = \underbrace{\sum_{l=1}^{q} \beta_l y_{t-l}}_{\text{AR part}} + \underbrace{\left(1 - \sum_{l=1}^{q} \beta_l\right)}_{\text{mixing weight}} \cdot \underbrace{\exp(\mathbf{x}_t'\boldsymbol{\gamma})}_{\text{covariate part}}$$

where m_t is the mean Poisson parameter at time t, β_l is the autoregressive coefficient for lag l ($l = 1, \ldots, q$), and $\boldsymbol{\gamma}$ represents the effect of current covariates (including promotion) on sales. The time-varying Poisson mean can be interpreted as a weighted combination of previous observations of sales and the covariates, where the exponential term comes from the log link used for Poisson regression. We impose the constraints $\beta_l \geq 0$ and $\sum_{l=1}^{q} \beta_l < 1$ to ensure that the AR process is stationary (i.e. not diverging over time).

2.3.1 Likelihood and Gradient

Let $\boldsymbol{\theta}$ denote the full set of model parameters. The full likelihood across all items and times is given by

$$\mathcal{L}(\boldsymbol{\theta}) = \prod_{t=q+1}^{T} \frac{m_t^{y_t} e^{-m_t}}{y_t!}$$

The corresponding log-likelihood, used for optimization and posterior inference is

$$\log \mathcal{L}(\boldsymbol{\theta}) = \sum_{t=q+1}^{T} \left[y_t \log(m_t) - m_t - \log(y_t!)\right]$$

where m_t is defined above. For the gradient (i.e. score function), we define $a_t = \sum_{l=1}^{q} \beta_t y_{t-l}$, $c_t = \exp(x_t^T \boldsymbol{\gamma})$, $w = 1 - \sum \beta_t$, and $m_t = a_t + w \cdot c_t$. Then, the derivative of the log-likelihood with respect to γ_j is

$$\frac{\partial \log \mathcal{L}}{\partial \gamma_j} = \sum_{t=q+1}^{T} \left[\frac{y_t}{m_t} - 1\right] \cdot w \cdot c_t \cdot x_{tj}$$

and the derivative of the log-likelihood with respect to β_k is

$$\frac{\partial \log \mathcal{L}}{\partial \beta_k} = \sum_{t=q+1}^{T} \left[\frac{y_t}{m_t} - 1\right] [y_{t-k} - c_t]$$

2.4 Hierarchical PAR for multi-item modeling

We consider extending Model 1 to jointly model more than one item at a time. Suppose we have n items, T timepoints and B brands. Let $g_i \in \{1, \ldots, B\}$ denote the brand (group) to which item i belongs.

We propose a hierarchical Bayesian extension of the PAR model, which we refer to as the Poisson Vector Autoregression (PVAR) model. The sales y_{it} for item i and timepoint t are modeled similarly to the PAR case as

$$\begin{aligned} y_{it}|m_{it} &\sim \text{Poisson}(m_{it}) \\ m_{it} &= \sum_{l=1}^{q} \beta_{i,l} y_{i,t-l} + \left(1 - \sum_{l=1}^{q} \beta_{i,l}\right) \exp(\mathbf{x}_{it}' \boldsymbol{\gamma}_i) \end{aligned} \tag{2}$$

where each item is assumed to have its own autoregressive coefficients $\beta_{i,l}$ and covariate effects $\boldsymbol{\gamma}_i$. We use the following hierarchical priors on model parameters

$$\begin{aligned} &\boldsymbol{\gamma}_i \sim N(\boldsymbol{\mu}_{g_i}, \Sigma_{g_i}) \\ &\boldsymbol{\mu}_{g_i} \sim N(\boldsymbol{\mu}_0, \Sigma_0),\ \Sigma_{g_i} \sim \text{Inv-Wishart}(\nu, \Psi) \\ &\tilde{\boldsymbol{\beta}}_i|\tau_{g_i} \sim \text{Dirichlet}(\boldsymbol{\alpha}_{g_i}),\ \beta_{i,l} = \tau_{g_i}\tilde{\beta}_{i,l} \\ &\boldsymbol{\alpha}_{g_i} \sim \text{Dirichlet}(\alpha_0, \ldots, \alpha_0),\ \tau_{g_i} \sim \text{Beta}(a_\tau, b_\tau) \end{aligned}$$

where $\boldsymbol{\mu}_0$, Σ_0, ν, Ψ, α_0, a_τ, and b_τ are fixed hyperparameters. For our analysis, we choose $\boldsymbol{\mu}_0 = \mathbf{0}$, $\Sigma_0 = \Psi = \mathbf{I}$, $\nu = 2$ (1 covariate + intercept), $\alpha_0 = 1/q$ (q is fixed lag), and $a_\tau = b_\tau = 2$. Our structure of priors assume that the parameters for items within a given brand come from some common distribution. Effectively, this will pool information from within each brand when estimating the effects of previous sales and promotion for each item. We use Dirichlet and Beta distributions for the AR coefficients as a way to impose the constraint that $\sum_{l=1}^{q} \beta_{i,l} < 1$, which leads to stationarity in the Poisson AR processes for each item.

2.5 Broyden–Fletcher–Goldfarb–Shanno (BFGS)

To estimate model parameters, we minimized the negative log-likelihood function using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. BFGS is a quasi-Newton method that iteratively approximates the inverse Hessian matrix of the negative log-likelihood function. Optimization was initialized with θ zero values, and the inverse Hessian was set to the identity matrix. At each iteration, the step direction was determined by the approximate Hessian inverse, and a line search was performed to satisfy the Armijo condition for sufficient decrease. The algorithm was terminated once the difference in the magnitude of log-likelihood between iterations was below some tolerance value, typically 10^{-5} or when a maximum number of iterations was reached.

2.6 Expectation Maximization (EM)

Not every promotion time point yields an increase in sales. For this reason, we explore using Expectation Maximization (EM) to down-weight or silence promotion time points that would otherwise be regarded as failed promotions. Here we assume that the promotion covariates γ are confounded with some unknown effectiveness latent variable $z_i \sim \text{Bernoulli}(\pi_i)$, where π_i represents the probability that a certain pair of items from a certain brand has a successful promotion. The EM algorithm iteratively maximizes the expected complete-data log-likelihood, defined as

$$Q(\theta|\theta^{(k)}) = E_{z_k|\boldsymbol{y},\theta^{(k)}}[\log \mathcal{L}(\boldsymbol{y}, z_{k=\hat{k}}|\boldsymbol{\theta})]$$

For the E-step, we computed the conditional expectation of the complete-data log-likelihood given the observed data and current parameters. Within the M-step, we maximized the Q-function with respect to θ using the BFGS algorithm. Iterations of the EM algorithm were run until the relative change in log-likelihood was below some tolerance value.

2.7 Markov Chain Monte Carlo (MCMC)

We implemented a Metropolis-within-Gibbs MCMC algorithm to sample from the posterior distributions of model parameters for Model 1 and Model 2. The full conditional distributions for the autoregressive coefficients (β) and covariate effects (γ) did not have closed forms, necessitating the use of Metropolis-Hastings (MH) sampling within the Gibbs algorithm. Furthermore, the parameters associated with the AR coefficients were subject to simplex constraints. To apply the MH step for these terms, we transformed to the unconstrained space using a log-ratio transformation, generated a proposed value from a proposal distribution, then transformed the proposed value back into the constrained space. The corresponding acceptance ratio for the MH step accounted for the Jacobian associated with the inverse transformation. For all Metropolis-Hastings steps, we use a Normal proposal density with variance 0.05 and mean set to the current value of the (transformed) parameter. Gibbs steps were used for the $\boldsymbol{\mu}_{g_i}$ and Σ_{g_i}, which represent the mean and variance of covariate effects for items within brand g_i and have standard closed-form conditional posteriors.

2.8 Model Evaluation

To diagnose the performance of the MCMC sampling, we visually assessed traceplots for key model parameters, including the AR terms and the effect of promotion. We further stratified by brand to assess potential brand-specific issues with MCMC mixing. These traceplots are shown in Supplementary Materials.

For all forecasting schemes, we applied a 80%-20% training-test split, where the first 80% of timepoints for each item were used for model training. We then performed two types of forecasting on the test set using trained model estimates. The first type was *1-step forecasting*, where we compute the predicted value of y_{t+1} given the previous q time points of the observed sales data. In this sense, the 1-step forecasting still relies on training data to estimate model parameters, but makes use of the test set outcomes when generating predictions. The second type of forecasting we used was *H-step forecasting*, where H was set to the size of the test set. This involves predicting the entire span of time points in the test set, only using the training set information. To perform this step using the PAR model, for instance, we replace the autoregressive term $\sum_{l=1}^{q} \beta_l y_{t-l}$ with $\sum_{l=1}^{q} \beta_l \hat{m}_{t-l}$, where $\hat{m}_t$ is the estimate for the Poisson mean obtained only from the training data. After generating 1-step and H-step forecasts, we calculate the Mean Squared Error (MSE) and Poisson Deviance to compare model performance. This process was repeated across all items in the dataset.

2.9 Random Forest

We incorporate Random Forest to implement one-step and H-step prediction on the item level. The following elaboration is based on the data of first item of the Brand 1.

We first fit the one-step model. Since we are interested in the effect of promotion and brand on items' sales over time, except for the existing promotion information, we also created *lag1*(yesterday's sale), *lag7*(sales from a week ago), *TotalB1* (Total sales of all items within the same brand at that day), and *Roll7totalB1*(Mean sales of all items within same brand in the last a week) as the model features. In addition, we added some other features that might be informative, such as *Month* (In which month)and *dow* (which day of the week).

After the feature specification, We split the first 80 % as training data and remaining 20% as test data. The next step for one-step model is hyperparameter tuning. We search the grid based on out-of-bag RMSE metric to find the best combination of `mtry`(The number of randomly selected features considered for splitting at

each node) and `mini.node.size` (minimum number of observations for a tree to consider splitting.), and set the `num.tree` and `sample.fraction` to default values. The Final set-up is `mtry =4`, `mini.node.size =10`, `num.tree = 500` and `sample.fraction = 1`. We were then able to fit the one-step model, and do the prediction on the test set.

Based on the one-step prediction model, we can set up the H-step prediction. So instead of directly predicting the next timepoint based on current information, H-step prediction sequential prediction where the estimated $\hat{y}_t$ replaces the actual y_t for future steps. The features remain the same as in the one-step model. Take *lag1* as an example showing how we did this. Denote y_T as the sales on the last time point, T, of the training set, then we feed y_T back to the one-step model to predict $\hat{y}_{T+1}$, to predict $\hat{y}_{T+2}$, we now can only use $\hat{y}_{T+1}$ to feed the one step model.

Here predictions are generated from time $t+1$ to $t+H$, where H equals the number of observations in the test set.

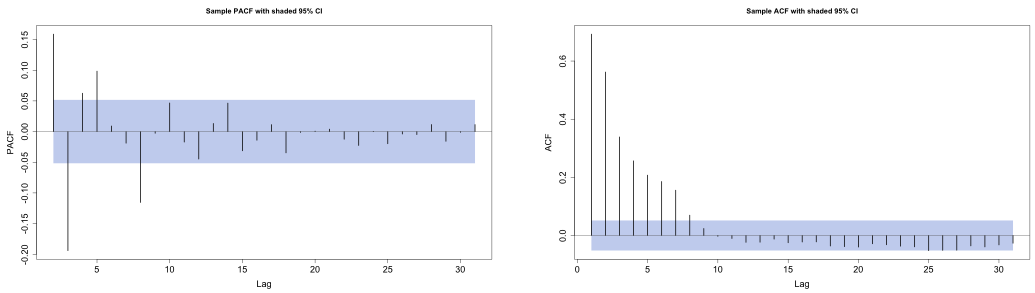
3 Results

3.1 Estimation

3.1.1 ARMA

We fit each brand item pair with ARMA model. We show here the results of item 1 in brand 1 for example.

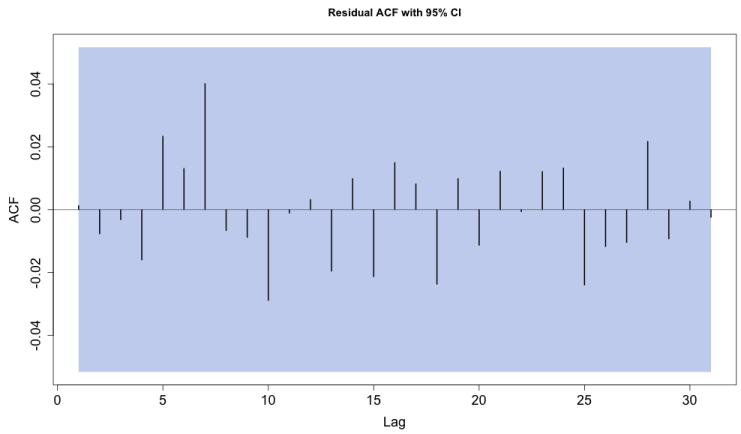
It is often critical to figure out the order of the ARMA model. We hence first use partial autocorrelation function (PACF) to determine the order in AR and use autocorrelation function (ACF) to determine the order in MA. Figure 3 is one example of sample PACF and ACF. We can see that the PACF on the left panel shows significant spikes at lags 1, 2, and 4, then cuts off, pointing us toward an AR(4) structure. Meanwhile, the left panel of ACF plot exhibits a slow exponential decay out to lag 8 or so, indicating a moving-average component of order at least about 4 or 5. Putting these clues together, we tentatively select an ARMA(4,5) specification.



PACF **ACF**

Figure 3: Sample PACF and ACF for item 1 in brand 1. The shaded blue areas are the 95% confidence band.

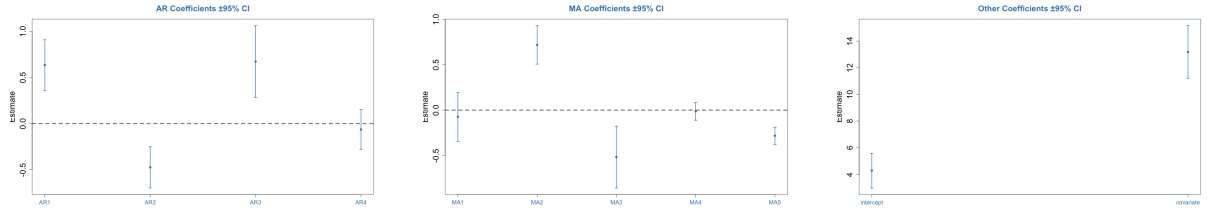
After fitting the ARMA(4,5) model, we check the residuals. We first check the autocorrelations again. Figure 4 shows the residual ACF with 95% confidence bands. Virtually, all autocorrelations lie within the bounds, indicating no remaining serial dependence. Moreover, the Ljung–Box test on the residuals yields a p-value of 0.519, so we cannot reject the null of no autocorrelation. We also run a Shapiro–Wilk test, however, the p-value ($2e-16$) is extremely small as expected, our response is count data. This highlights a limitation of the basic ARMA approach, but as a first approximation it still captures much of the serial pattern in pasta sales.



Residuals ACF

Figure 4: Residuals ACF for item 1 in brand 1 after fitting ARMA(4,5). The shaded blue areas are the 95% confidence band.

Furthermore, estimates for all the parameters except for σ^2 in the ARMA model are shown in figure 5 for item 1 in brand 1. We can see most of the coefficients are statistically significant, confirming our model structure. Especially, we see that γ is statistically significant and is also greater than 0, indicating a true positive correlation between promotion and sales of pasta.



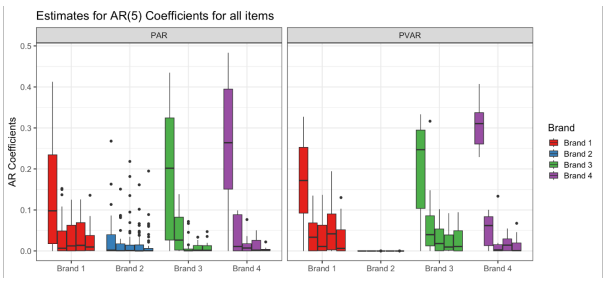
AR **MA** α and γ

Figure 5: Estimated coefficients with 95% confidence band for item 1 in brand 1.

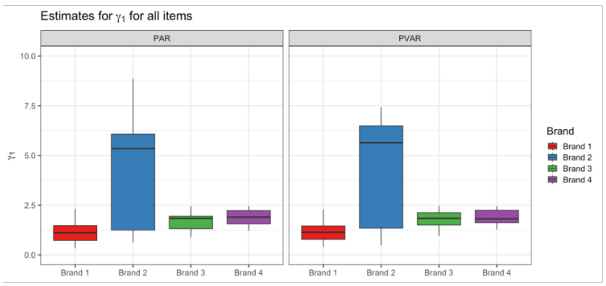
3.1.2 Bayesian PAR and PVAR

We performed MCMC sampling on all 118 items using both the item-by-item PAR Model (Model 1) and the multi-item PVAR Model (Model 2). Both sets of models included AR terms up to lag 5, an intercept term, and a promotion effect. Estimates for model parameters were obtained by averaging the post-burn-in MCMC samples (total number of MCMC iterations = $10,000$, burn-in set to $5,000$).

Estimates for the item-specific AR terms ($\boldsymbol{\beta}$), stratified by brand, are shown in Figure 6a. The leftmost box plots for each brand correspond to β_1 and the rightmost box plot corresponds to β_5. For both models, β_1 had the largest average estimate across items, with comparably low estimates for the other β_q. This result is consistent with the ACF shown in Figure 3, and is a common finding for time series data. Additionally, we find that the PVAR model led to smaller variance across items in estimated β values, which was due to information pooling within each brand. Notably, Brand 2 β estimates within the PVAR model are close to zero in comparison to the PAR model, signifying a minimal dependence on previous history of sales for that brand. For the remaining brands, we find visual evidence of differences in the distribution of autoregressive term estimates (e.g. brand 4 tends to have the largest AR-1 effect), but do not rigorously test for significant differences in this analysis.



(a) Item-level AR term estimates for 5 lag points for the PAR and PVAR models obtained using MCMC



(b) Item-level effects of promotion from PAR and PVAR models obtained using MCMC

Estimates for the promotion effects, (γ_1) are shown in Figure 6b for each item and model, again stratified by brand. Each box plot represents the estimate of promotion for a given item belonging to a certain brand. We find Brand 2 has the highest variance in parameter estimates in both the PAR and PVAR models compared to all other brands. We suspect this discrepancy is due to the average investment in an item for brand 2 is approximately 70%, whereas the average investment for the other brands was less than 20%. Investment here is calculated as the proportion of days a particular brand item was promoted over the course of the time series.

3.1.3 EM - Estimating Effective Promotions

Each brand item had parameters initially estimated through BFGS assuming a 5 time point lag and using an intercept and promotion as covariates to the model. The EM algorithm was run on each brand item until the relative change in log-likelihood was below 10^{-6}.

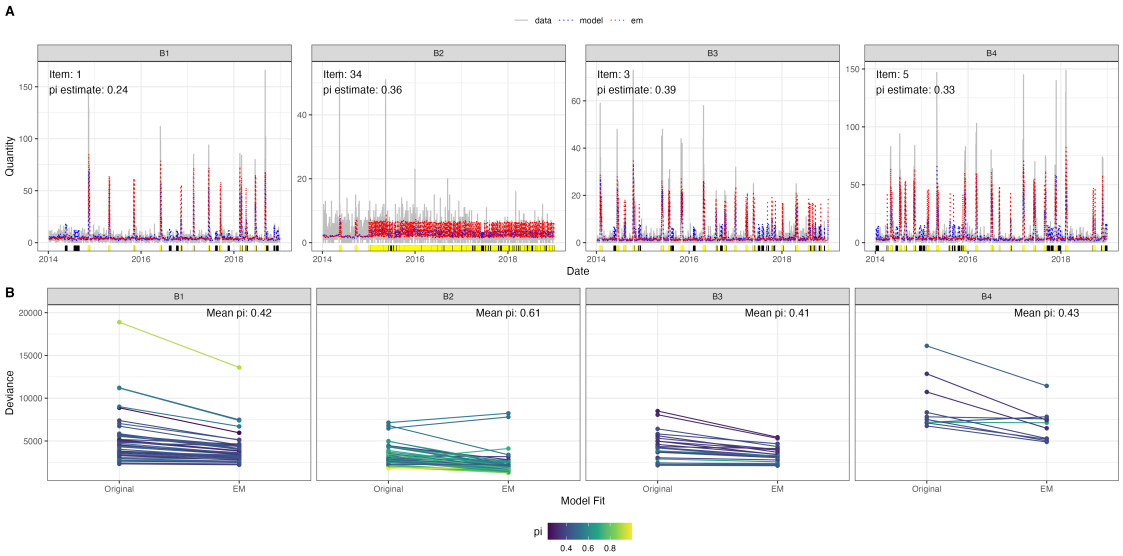


Figure 7: EM results of fitting each brand item using BFGS. Panel A: A time series of one representative example item from each brand. Grey lines indicate original observed data. Blue dotted lines indicate initial BFGS fit mean estimates. Red dotted lines indicate final mean fits after latent variable estimation. Rug marks indicate days in which a promotion occurred, while yellow marks indicate days that were effective. Panel B: deviance of the initial model estimates compared to the EM model estimates. Each brand item is colored by their estimated π_i from EM.

Representative samples from each brand are shown in Figure 7 Panel A. The Expectation-Maximization (EM) algorithm was generally effective in improving model fit relative to the initial parameter estimates

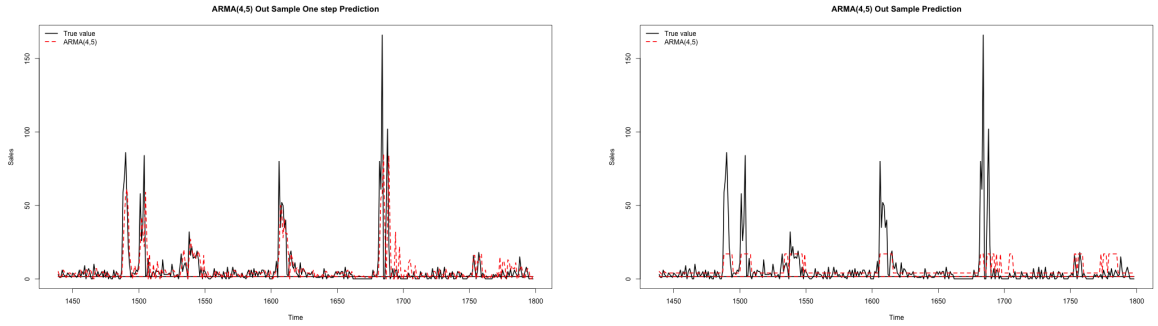
obtained via BFGS optimization. Across all brands, EM iterations consistently reduced the model deviance, indicating improved agreement between the fitted model and the observed data, with a few exceptions as seen in Figure 7 Panel B.

Analysis of the estimated latent variables z_i, representing the effectiveness of promotions on specific days, revealed that the inferred promotion success probabilities π_i were typically around 0.40. That is, on average, a given brand promotion was estimated to be effective approximately 40% of the time. An exception was observed for Brand 2, which exhibited a substantially higher promotion success rate of approximately 60%. Notably, Brand 2 also had a much higher overall frequency of promotions, with an average promotion rate exceeding that of other brands by more than a factor of three. This disparity may challenge the ability for the EM model to estimate days effected by promotion and those due to natural sale frequency.

3.2 Forecasting

3.2.1 ARMA

To illustrate the out-of-sample performance of ARMA(4,5) model, we evaluate the forecasts on our hold-out test set. To be specific, we split the data set into training set and test set, with the first 80% time points in the training set and the remaining 20% time points in the test set. Figure 8 shows the performance of prediction in detail. On the left is the one-step-ahead prediction: in most days, the forecast closely tracks the true sales spikes. On the right is the H-step forecast, we can see the performance degrades slightly as the horizon increases, but the model seems to still identify the timing of demand surges.



One-step **H-step**

Figure 8: Out-sample performance on the test set of ARMA(4,5) for item 1 in brand 1. The red dashed lines are predicted values, and the black solid lines are the true values.

Generally we can see the ARMA model performs well despite of some drawbacks, it can hence be treated as the baseline model.

3.2.2 Bayesian PAR and PVAR

The results from forecasting using the Bayesian PAR and PVAR models (Models 1 and 2) are shown in Figures 12-11. Figure 12 displays the 1-step and H-step forecasts for a single item, after running MCMC for the PAR and PVAR models on the training set. We visually see similar performance to the ARMA case, with relatively good performance using 1-step forecasting and moderate performance in the H-step case. The PVAR model appears to produce slightly lower forecasts around the peaks than the PAR model in this case, though the placement of those predicted peaks appears marginally better in the PVAR case.

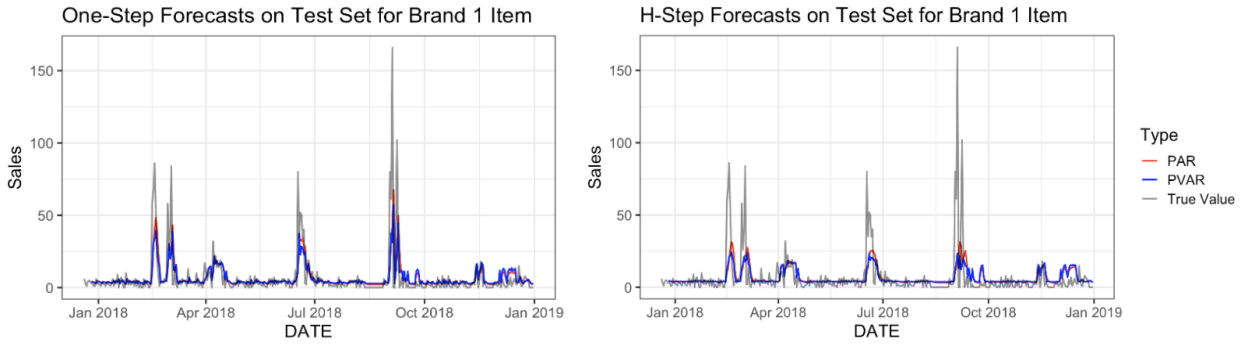


Figure 9: Out-of-sample performance on the test set using PAR and PVAR models for item 1 in brand 1.

We also display the brand-level forecasts from the PAR and PVAR model using 1-step forecasting in Figure 10. This indicate larger similar brand-level forecasting power between the PAR (non-hierarchical) and PVAR (hierarchical) models. However, we do note that the PVAR model allows for brand-level forecasting directly from model parameters, whereas the PAR model obtains these forecasts by first averaging the item-level estimates. Therefore, PVAR may be more appropriate for quantifying uncertainty about future trends at the brand-level.

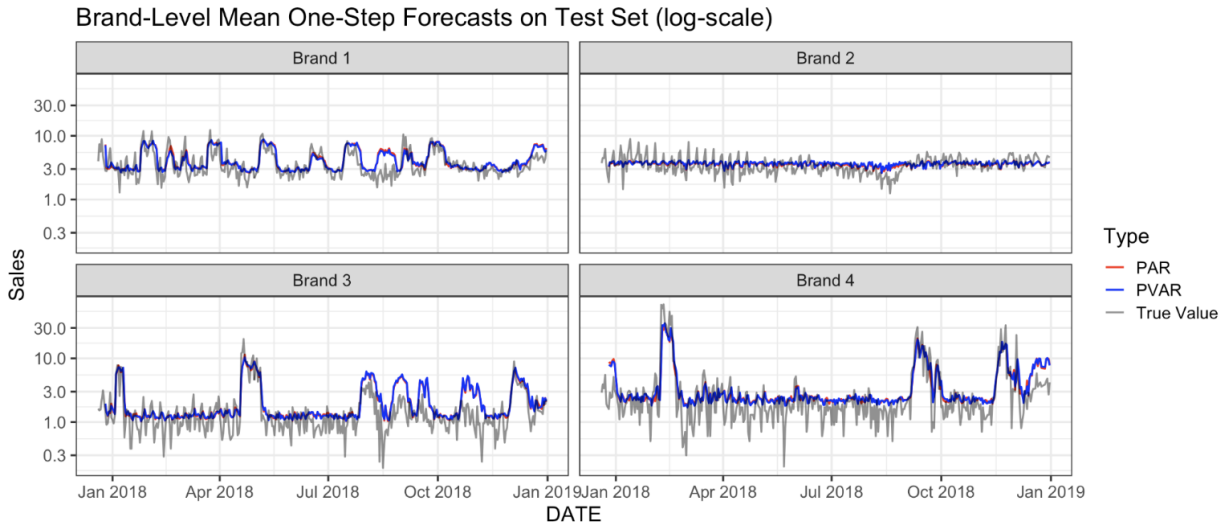


Figure 10: Out-of-sample performance on the test set using PAR and PVAR models for brand-level forecasts.

Lastly, we evaluated the 1-step forecasting performance for the PAR and PVAR models on all items. The resulting evaluation metrics, including Mean-Squared Error (MSE) and Poisson Deviance, are shown for all items in Figure 11. We find minimal differences in the forecasting performance of these two models, but do observe some additional outliers with poor predictive performance in the PAR case compared with the PVAR case.

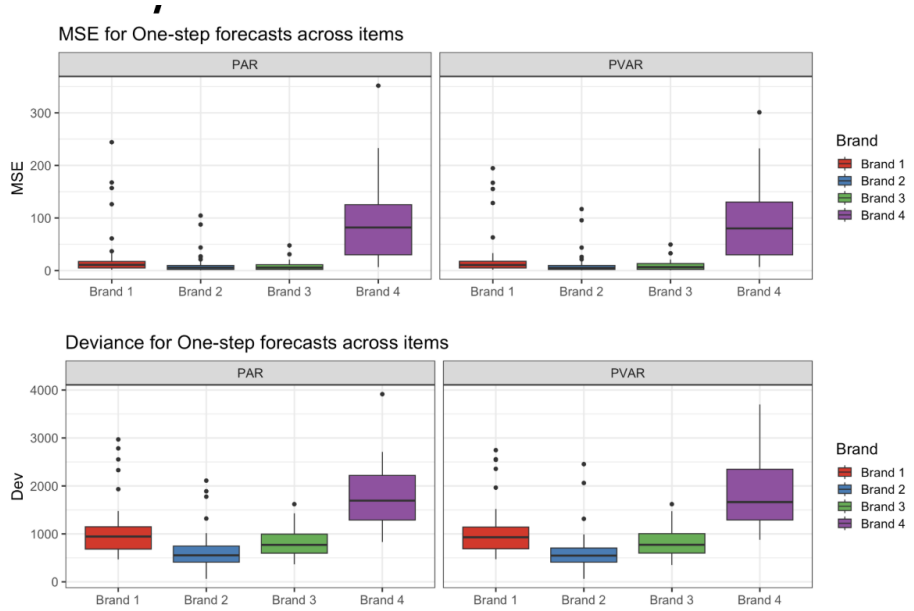


Figure 11: Item-level Mean Squared Errors (MSE) and Poisson Deviance from 1-step forecasting using PAR and PVAR models

3.2.3 Random Forest

The prediction results based on the first item of Brand 1 using the Random Forest one-step and H-step models are shown in Figure 12. We can observe that the one-step model performs well with MSE = 109.91, while the H-step model has a poor match with the actual value, and even misses several peaks, with a corresponding MSE = 217.49. The difference in performance between these two models makes sense, since the H-step model accumulates the prediction error during the model recursion. But since we just did the prediction for one item, we also need to consider the possibility that the one-step model has an overfitting issue.

Besides prediction, the Random Forest algorithm also provides the Gini Importance Index for every predictor. Here, the top 3 predictors are: *lag1*, *TotalB1* and *Promotion*. This suggests that the Random Forest-based models also identify Promotion and brand as predictors that have effect on the sales of one item. This conclusion, in some terms, coincides with the interpretation from the parametric models we fitted.

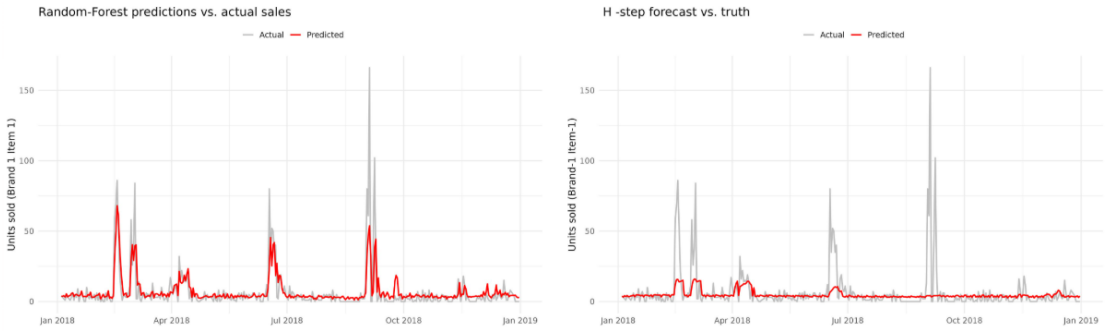


Figure 12: Out-of-sample performance using Random Forest for item 1 in brand 1.

4 Discussion

4.1 Takeaways

Revising our research questions, we found evidence that the patterns of pasta sales over time do vary across the four brands. Furthermore, the effect of promotion on current sales varies substantially across brands, with items in brand 1 showing the weakest effect of promotion, on average. Here, we also found that brand 2 items tended to show the largest effect of promotion, but are hesitant to trust these estimates given the imbalances in the dataset.

For forecasting, we find similar performance for all the various models we examined, with relatively good performance using 1-step prediction and weaker performance when low to moderate performance when predicting an entire year's worth of sales. We find some marginal evidence that the hierarchical model leads to better long-term forecasting than the non-hierarchical model for certain items, though this finding was certainly not definitive. That being said, the hierarchical model offers certain unique advantages for forecasting-related tasks, such as the ability to produce brand-specific forecasts directly from model parameters. Depending on specific forecasting goals, we recommend using some version of hierarchical model for this data, since it uses more available data and can be more comprehensive to other sorts of prediction tasks such as predicting sales for an item given information about the other items.

4.2 Limitations

While the models developed above provide some insight into modeling and forecasting pasta sales, there are many limitations that restrict our results and insights. One limitation of our forecasting approach is that one-step forecasts rely on previous observations, which may introduce information leakage from the test set and overstate predictive performance. Our H-step forecasting does not touch the test set, but performed worse than the one-step forecasting models, and failed to adequately predict the magnitude of spikes in the data (although we did largely identify where in time these spikes occurred). Another limitation was that fitting the PAR and PVAR models was computationally intensive, even with efficiency gains from `Rcpp`-based implementations, restricting our modeling capabilities. Lastly, the data sparsity, presence of large spikes, and the imbalance of promotion events across items complicate reliable estimation and may bias effect sizes. Specifically, brand 2 promoted almost all of their items at most times, restricting our ability to estimate model parameters from brand 2. Some of these limitations can be addressed in future studies.

4.3 Future Directions

One modification to our analysis that could make a significant difference in results is the distributional assumption. We have posited a Poisson distribution for our count data, where a negative binomial distribution may be more appropriate for accounting for overdispersion given the high peaks in our sales data. There are also smaller changes that could be made to improve our modeling and forecasting. This includes hyperparameter tuning for our Bayesian MCMC prior distribution. Due to time constraints, we only considered non-informative priors with a fixed set of hyperparameters. Tuning these hyperparameters, along with exploring more choices of lag, could improve predictive performance. Relatedly, we wish to explore other options for priors which could place less restrictive assumptions on the AR terms, potentially leading to better model fitting. Additionally, we would like to finish applying the single-item Machine Learning approaches to the rest of the items in all of the brands. A comparison of these parameter estimates and forecasts will provide more insight on the existence and significance of brand-level effects in our models.

References

Brandt, P. T. and Williams, J. T. (2001). A linear poisson autoregressive model: The poisson ar (p) model. *Political Analysis*, 9(2):164–184.

Hamilton, J. D. (2020). *Time series analysis*. Princeton university press.

Mancuso, P., Piccialli, V., and Sudoso, A. M. (2021). A machine learning approach for forecasting hierarchical time series. *Expert Systems with Applications*, 182:115102.

5 Supplementary Materials

5.1 MCMC Diagnostics

Figures 13-14 show some example traceplots corresponding to the autoregressive lag-1 term and promotion effect term under the PAR and PVAR models. Each plot consists of item-level terms across 2 randomly-chosen items in each of the 4 brands. We see that for both PAR and PVAR, the promotion effect terms appear largely convergent across items, except for the brand 2 item which shows evidence of poor mixing. This is consistent with the fact that brand 2 had highly imbalanced promotion data, which made estimation for that term difficult. Furthermore, the AR-1 terms show some evidence of poor mixing for certain items. This could be due to the high correlations between the AR-1 term and subsequent AR terms, i.e. the effect of higher lags may absorb some of the effect of lag-1 during sampling.

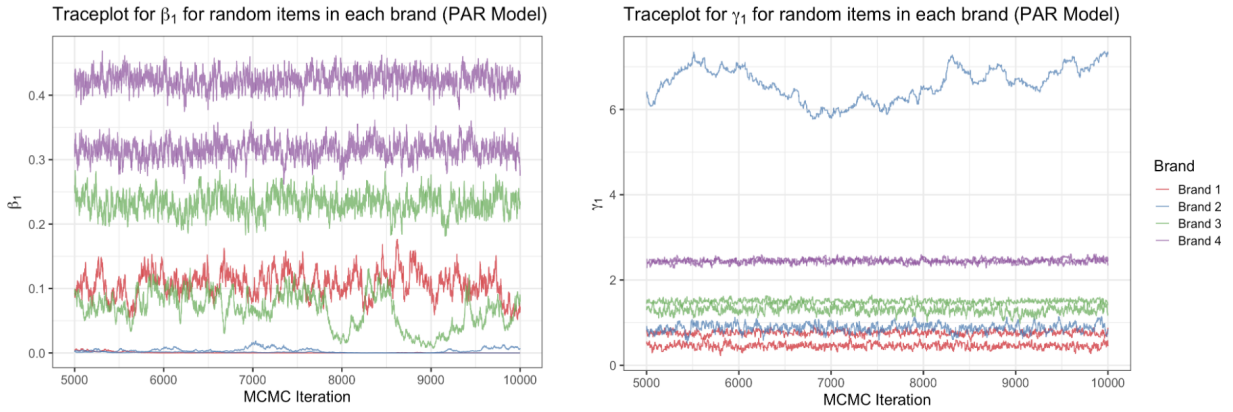


Figure 13: Traceplots for item-level AR-1 terms and promotion effect terms using the single-item PAR model.

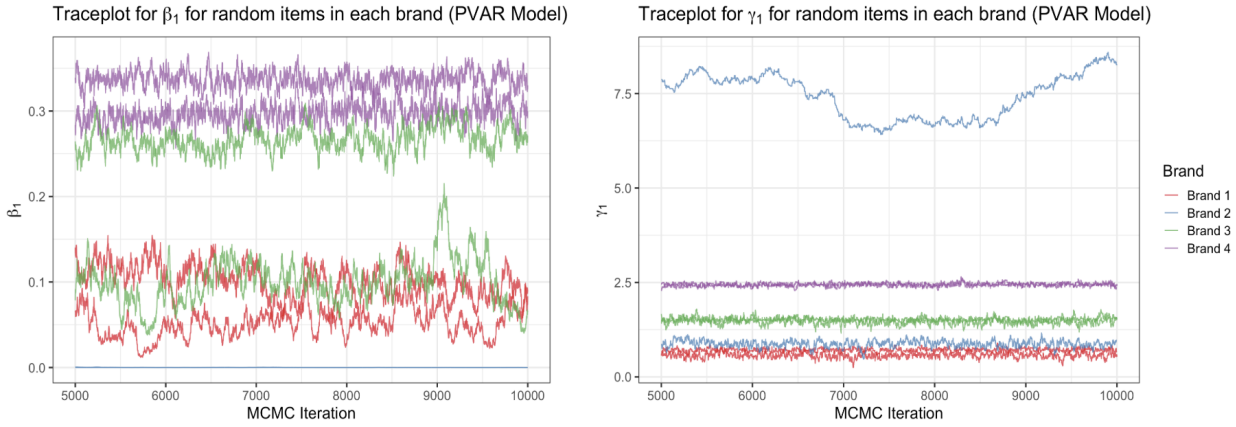


Figure 14: Traceplots for item-level AR-1 terms and promotion effect terms using the single-item PAR model.

5.2 Choice of Lag for PAR and PVAR Models

We did explore some alternative choices of lag for the PAR and PVAR models. The downstream forecasting performance using lags 1, 5, and 10 are summarized in Table 2. Performance metrics were first evaluated on each item, then we computed the mean and median metrics across all 118 items in the dataset, which

are presented in the table. We find that that for some outlier items, the PAR(1) model produces forecasts which are highly unrealistic, leading to highly inflated means. Comparing the mean evaluation metrics, we see justification of lag 5 for both the PAR and PVAR models, which is the choice we used for the current analysis. However, we also observe that higher las (i.e. lag 10) for the PVAR model may be more suitable for H-step forecasting on the test set.

Table 2: Summary of Forecast Performance (Mean / Median across Items)

Model	1-step MSE	H-step MSE	1-step Deviance	H-step Deviance
PAR(1)	3.989×10^{15} / 7.868	1.197×10^{20} / 9.416	3.542×10^{9} / 757.3	6.265×10^{11} / 834.9
PVAR(1)	24.78 / 7.694	29.84 / 8.001	941.3 / 759.3	1049.8 / 792.2
PAR(5)	25.21 / 7.489	30.51 / 7.923	940.2 / 748.2	1127 / 780.9
PVAR(5)	24.54 / 7.648	30.17 / 7.953	932.0 / 750.8	1052.0 / 786.2
PAR(10)	25.94 / 7.725	30.19 / 7.981	955.4 / 765.4	1077 / 820.9
PVAR(10)	26.46 / 7.994	29.95 / 7.972	963.2 / 758.0	1043.7 / 771.2