Modeling Promotional Pasta Sales with Hierarchical Poisson Autoregression

Group 2

Contents

1 Introduction

We analyze pasta sales data from an Italian grocery store, consisting of 116 items across four brands, observed daily from 2014 to 2018. Our goal is to model item-level sales with dependence on past observations, promotional status, and latent brand effects.

We compare likelihood-based models (EM, Newton-Raphson), Bayesian inference (MCMC), and machine learning benchmarks (Random Forests, Hidden Markov Models), using predictive accuracy as the main evaluation metric.

1.1 Background

1.2 Goals

2 Data Description

```r
data("data_set_tidy")
glimpse(data_set_tidy)
```

```
## Rows: 212,164
## Columns: 5
## $ DATE  <date> 2014-01-02, 2014-01-02, 2014-01-02, 2014-01-02, 2014-01-02, 201~
## $ brand <chr> "B1", "B1", "B1", "B1", "B1", "B1", "B1", "B1", "B1", "B1", "B1"~
## $ item  <chr> "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "~
## $ QTY   <dbl> 7, 3, 0, 2, 3, 1, 0, 4, 0, 0, 0, 0, 7, 2, 5, 2, 0, 7, 1, 0, 0, 4~
## $ PROMO <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

Each row corresponds to daily sales for a specific item, along with brand and promotion indicators.

3 Model Derivation

3.1 Extended Poisson Autoregressive Model

Let the sales outcome for item i at time t be denoted by y_{it}, where:

$$i \in \{1, \ldots, n\}, \quad t \in \{1, \ldots, T\}, \quad g_i \in \{1, \ldots, B\}$$

Here, g_i denotes the brand (group) to which item i belongs.

We define the model as:

$$y_t \sim \text{Poisson}(m_t)$$

$$m_t = \sum_{l=1}^{q} \beta_l y_{t-l} + \left(1 - \sum_{l=1}^{q} \beta_l\right) \cdot \exp(\mathbf{x}_t^\top \boldsymbol{\gamma})$$

Where:

- β_l: AR coefficients (constrained so sum ≤ 1)
- γ: covariate effect vector

3.2 Likelihood Function

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

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

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,

$$m_t = \underbrace{\sum_{l=1}^{q} \beta_l y_{t-l}}_{\text{AR part}} + \underbrace{\left(1 - \sum l = 1^q \beta_l\right)}_{\text{mixing weight}} \cdot \underbrace{\exp(\mathbf{x}t^\top \boldsymbol{\gamma})}_{\text{covariate part}}$$

3.3 Gradient (Score Function)

Define the following:

- $a_t = \sum_{l=q}^{q} \beta_t y_{t-l}$
- $c_t = \exp(x_t^T \gamma)$
- $w = 1 - \sum \beta_t$
- $m_t = a_t + w \cdot c_t$

Then, the derivative of the log-likelihood with respect to γ_j is:

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

And the derivative of the log-likelihood with respect to β_k is:

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

4 Estimation Methods

4.1 EM Algorithm

- E-step: Applicable if we add new random intercepts (either item or brand level)
- M-step: Maximize the expected log-likelihood with respect to model parameters γ, β, and τ

4.2 Newton-Raphson

- Use gradients and Hessians of the log-likelihood to iteratively update parameters

4.3 Bayesian Inference (MCMC)

- Fit the independent Poisson Autoregression (PAR) model for each item with MCMC (currently implemented in STAN)
- Place priors directly on model parameters
- τ: mixing weight for AR component, $\beta = \tau \tilde{\boldsymbol{\beta}}$

Priors:

- $\boldsymbol{\gamma} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma_\gamma)$
- $\Sigma_\gamma \sim$ Inv-Wishart(ν, Ψ) — (approximated as fixed in Stan)
- $\tilde{\boldsymbol{\beta}} \sim$ Dirichlet(α)
- $\tau \sim$ Beta(a, b)

5 Model Evaluation and Predictive Performance

We evaluate models using holdout-based prediction: the final 10–20% of each item's time series is reserved for testing.

```
# Example: results <- evaluate_holdout(...)
# knitr::kable(results)
```

** Evaluation Metrics ** - Root Mean Squared Error (RMSE) - Mean Absolute Error (MAE) - Log Predictive Density (for Bayesian models) - Posterior predictive interval coverage (for MCMC)

6 Machine Learning Benchmarks

6.1 Random Forest

```
# rf_fit <- fit_random_forest(...)
```

6.2 Hidden Markov Model

```
# hmm_fit <- fit_hmm(...)
```

7 Summary and Discussion

- Which methods give the most accurate predictions for sparse, autocorrelated data?
- What is the trade-off between hierarchical shrinkage and item-specific flexibility?
- Are machine learning models (e.g., random forests) better suited for forecasting than structured generative models?