Bayesian Modeling of Networks in Complex Business Intelligence Problems
MTH 422 Group project by:

Aditya V ()
Dwija Kakkad(210367)
Saee Kamat(231080070)

1   Introduction

This paper introduces a Bayesian hierarchical model to address complex business intelligence problems, specifically for optimizing cross-selling strategies. The authors utilize data on customer product choices and co-subscription networks across multiple agencies. Their model clusters agencies with similar customer purchasing behaviors and then, within each cluster, employs a mixture of latent eigenmodels to understand customer preferences. This framework allows for the development of targeted cross-sell strategies and the evaluation of their potential effectiveness. The paper cites several other approaches in the literature for cross-selling strategies and related business intelligence problems as follows:

1. Latent Trait Model:Kamakura et al. (1991)developed a latent trait model to estimate a customer's propensity towards a particular product based on their ownership of other products. This was an initial effort to address cross-selling based on shared acquisition patterns

2. Improved Latent Trait Model: Kamakura (2008)later improved their latent trait model by combining information from customer databases with survey data

3. Multivariate Probit Model: Verhoef and Donkers (2001) focused on predicting the potential value of a current customer using a multivariate probit model. They also proposed a two-by-two segmentation to improve targeting for cross-selling

4. Multivariate Credibility Method: Thuring et al. (2012) developed a multivariate credibility method to identify a profitable set of customers for cross-selling. This method estimates a latent risk profile for each customer by exploiting information on claims. Kaishev et al. (2013) are also mentioned for recently developed cross-sell strategies and overviews of available methodologies

5. Approaches Utilizing Diverse Data Sources: Previous procedures often exploit different sources of information, including customer demographics and survey data, to estimate co-subscription probabilities for individual customers within a single agency.

The authors of the current paper differentiate their approach by noting that they do not observe customer demographic data for a single agency. Instead, they monitor mono-product customer choices along with co-subscription networks across multiple agencies.

2 Methodology

The paper proposes a Bayesian hierarchical model to cluster agencies based on their mono-product customer choices and co-subscription networks, with the goal of designing targeted cross-sell strategies. The model integrates two key data types first mono-products choices, i.e. categorical data representing customer subscriptions to single insurance products and second co-subscription networks i.e. binary adjacency matrices encoding multi-product purchasing behavior.

The goal is to design an effective cross selling campaign on the basis of customers' behaviour.

2.1 The Model

Monoproduct customers are the customers $n_{i\nu}$ subscribed to one product ν with $\nu \in V = \{1, \ldots 15\}$. The paper considers data $y_{is} \in \{1, \ldots, V\}$ as the product chosen by mono-product customers $s = 1, \ldots n_i$ in agency i, where n_i denotes the number of customers in agency i, $n = 130$ denotes the number of agencies and $V = 15$ represent the available insurance products. Further let A_i be a $V \times V$ symmetric adjacency matrix for agency i, where $A_{i[vu]} = 1$ if products v and u are co-subscribed by at least 10% of the multi-product customers subscribed to atleast one of the two, and $A_{i[v\nu]} = A_{i[u\nu]} = 0$ otherwise for $\nu = 2, \ldots V$ and $u = 1, \ldots, \nu - 1$. The presence of an edge between two products suggests a preference of customers in agency i for that specific pair. A strategy is defined as the best product $u \neq \nu$ to offer to a mono-product customer subscribed to ν. The best product is defined as

$$u_{iv} = \mathrm{argmax}_u \{\Pr(A_{i[vu]} = 1) : u \neq \nu\}, \tag{1}$$

where $A_{i[vu]}$ is the random variable defined above. Note that this leads to V different cross-sell strategies per agency i. Each strategy u_{iv} is associated with a performance indicator defined as

$$e_{iv} = p_{iv} \cdot \max \{\Pr(A_i[vu] = 1) : u \neq v\},$$

for each $v = 1, \ldots, V$ and $i = 1, \ldots, n$. Strategies with a high e_{iv} value aim to target a sizable proportion of the available monoproduct customers in agency i by advertising a new product that is likely to appeal to them. When developing an evaluating cross-sell strategies in this setup, the authors discuss two main problems. First, the estimation of key strategy components such as decision rules u_{iv} and success indicators e_{iv} are subject to statistical error, particularly due to limited data, which complicates the estimation of probabilities like $\Pr(A_i[vu] = 1)$. Second, administrative efficiency can be improved by applying the same strategy across agencies with similar customer profiles. Doing so reduces overhead without sacrificing strategic effectiveness. So, the authors cluster n agencies into $K < n$ agencies via Chinese Restaurant Process (CRP) prior:

$$\mathbf{C} = (C_1, \ldots, C_n), \quad C_i \in \{1, \ldots, K\} \tag{2}$$

where K is inferred from the data. The CRP encourages a small number of clusters while allowing flexibility. For each cluster k, mono-product choices follow a categorical distribution:

$$\mathbf{p}_k = (p_{k1}, \ldots, p_{kV}), \tag{3}$$

where $p_{k\nu}$ is the probability that a monoproduct customer in an agency within cluster k subscribes to product ν, with likelihood for agency i in cluster k:

$$\mathrm{pr}(\mathbf{y}_i | C_i = k, \mathbf{p}_k) = \prod_{v=1}^{V} p_{kv}^{n_{iv}} \tag{4}$$

where n_{iv} counts mono-product customers choosing product v in agency i, and $\mathscr{Y}_k$ is categorical random variable. To reduce dimension while maintaining flexibility, the authors model each $\mathscr{L}(\mathscr{A}_k)$ via a cluster-dependent mixture of latent eigenmodels:

$$\mathrm{pr}\{\mathscr{L}(A_k) = \mathscr{L}(A_i)\} = \sum_{h=1}^{H} \nu_{hk} \prod_{l=1}^{V(V-1)/2} \left\{\pi_l^{(h)}\right\}^{\mathscr{L}(A_i)_l} \left\{1 - \pi_l^{(h)}\right\}^{1-\mathscr{L}(A_i)_l} \tag{5}$$

where ν_{hk} are cluster-specific mixture weights and edge probabilities $\pi_l^{(h)}$ follow a logistic model:

$$\pi_l^{(h)} = \left[1 + \exp\left\{-Z_l - D_l^{(h)}\right\}\right]^{-1}, \quad \mathbf{D}^{(h)} = \mathscr{L}(X^{(h)} \Lambda^{(h)} X^{(h)\top}) \tag{6}$$

Note that here $X^{(h)}$ contains latent product coordinates in $\mathbb{R}^R$ and $\Lambda^{(h)} = \mathrm{diag}(\lambda_1^{(h)}, \ldots, \lambda_R^{(h)})$ weights latent dimensions. We now can explain the hierarchical representation of the model. The model introduces latent class indices $G_i \in \{1, \ldots, H\}$ for each agency:

$$\mathscr{L}(A_i)_l | \pi_{il} \sim \mathrm{Bern}(\pi_{il}) \tag{7}$$

$$\pi_i | G_i = h = \pi^{(h)} \tag{8}$$

$$\mathrm{pr}(G_i = h | C_i = k) = \nu_{hk} \tag{9}$$

This enables information sharing across clusters while maintaining cluster-specific characteristics. The key innovations of this model are Flexible network representation via mixture of eigenmodels, Nonparametric clustering via CRP prior, joint modelling of categorical and network data and interpretable outputs for business decision making.

2.2 Cross-Sell Strategy Derivation and prior specification

For each cluster k, the optimal cross-sell strategy for product v is:

$$u_{kv} = \arg \max_{u \neq v} \mathrm{pr}(A_{k[vu]} = 1) \tag{10}$$

with performance measured by:

$$e_{kv} = p_{kv} \cdot \max_{u \neq v} \mathrm{pr}(A_{k[vu]} = 1) \tag{11}$$

where $\mathrm{pr}(A_{k[vu]} = 1) = \bar{\pi}_{kl}$ is the posterior mean edge probability. The model employs the following hierarchical priors:

1. Cluster Assignment Prior Agencies are clustered using a Chinese Restaurant Process (CRP):

$$\mathbf{C} = (C_1, \ldots, C_n) \sim \mathrm{CRP}(\alpha_c) \tag{12}$$

with conditional probabilities:

$$\mathrm{pr}(C_i = k | C_{-i}) = \begin{cases} \frac{n_{k,-i}}{n-1+\alpha_c} & \text{for } k = 1, \ldots, K_{-i} \\ \frac{\alpha_c}{n-1+\alpha_c} & \text{for } k = K_{-i} + 1 \end{cases} \tag{13}$$

where $n_{k,-i}$ counts agencies in cluster k excluding i, and $\alpha_c > 0$ controls the expected number of occupied clusters $E(K) = O\{\alpha_c log(n)\}$. A higher value of α_C favours more clusters a priori, with $E(K)$ increasing as sample size n increases.

2. **Mono-Product Choice Prior** Cluster-specific probability vectors follow Dirichlet distributions:

$$\mathbf{p}_k \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_V), \quad k = 1, \ldots, K \tag{14}$$

They authors chose this large support prior to avoid to avoid ruling out any apriori generative mechanism.

3. **Network Model Priors**

   - **Shared similarity vector**:

     $$Z_l \sim \mathcal{N}(\mu_l, \sigma_l^2), \quad l = 1, \ldots, V(V-1)/2 \tag{15}$$

   - **Latent coordinates**:

     $$X_{vr}^{(h)} \sim \mathcal{N}(0,1), \quad v = 1, \ldots, V,\ r = 1, \ldots, R \tag{16}$$

   - **Mixing weights**:

     $$\boldsymbol{\nu}_k \sim \text{Dirichlet}(1/H, \ldots, 1/H), \quad k = 1, \ldots, K \tag{17}$$

   - **Shrinkage prior** on latent dimension weights:

     $$\lambda_r^{(h)} = \prod_{m=1}^r \frac{1}{\vartheta_m^{(h)}}, \quad \begin{cases} \vartheta_1^{(h)} \sim \text{Ga}(a_1, 1) \\ \vartheta_{m>1}^{(h)} \sim \text{Ga}(a_2, 1) \end{cases} \tag{18}$$

     for $r = 1, \ldots, R$ and $h = 1, \ldots, H$.

Typical settings setting for hyperparameters include $a_1 = 2.5$, $a_2 = 3.5$ for the multiplicative inverse gamma prior, $\sigma_l^2 = 10$ for the similarity variance, $\alpha_c = 1$ for the CRP concentration and $\alpha_v = \sum_{i=1}^n n_{iv}/n$ for empirical Dirichlet priors. This combination of priors provides a flexible yet regularized framework that adapts model complexity to the data, maintains computational tractability ,controls overfitting through shrinkage and regularization and incorporates domain knowledge through empirical Bayes adjustments.

2.3 Posterior compuatation

Using the above hierarchical priors with typical hyperparameter settings, we have following posterior distribution:

$$p(\Theta|\mathscr{D}) \propto \left[\prod_{i=1}^n p(\mathbf{y}_i|\mathbf{p}_{C_i}) p(A_i|\pi^{(G_i)}) \right] p(\Theta) \tag{19}$$

where $\Theta = \{\mathbf{C}, \mathbf{p}, \nu, \mathbf{Z}, \mathbf{X}, \lambda\}$ contains all parameters and $\mathscr{D} = \{\mathbf{y}_i, A_i\}_{i=1}^n$ is the observed data. As the posterior is intractable, the imploy two part Gibbs sampler with data augmentation. This is done in two broad steps:

1. **Updating parameters given clusters**: Given $G_1, \ldots G_n$ and $\mathscr{L}(A_1), \ldots \mathscr{L}(A_n)$, the quantities $\mathbf{Z}$, $X^{(h)}$ and $\lambda^{(h)}$ are updated using Polya-Gamma data augmentation (Polson et al., 2013).

   $$\int \frac{(e^\psi)^a}{(1+e^\psi)^b} d\psi = 2^{-b} e^{\kappa\psi} \int e^{-\omega\psi^2/2} p(\omega) d\omega \tag{20}$$

   where $\omega \sim \text{PG}(b,0)$ and $\kappa = a - b/2$ This enables a conjugate prior.

2. **Cluster Assignment Updates**: The cluster assignments are updated following the algorithm given in Neal (2000) for CRP mixtures. That is,

$$p(\text{new cluster}) \propto \alpha_c \frac{\Gamma(\sum_v \alpha_v)}{\prod_v \Gamma(\alpha_v)} \frac{\prod_v \Gamma(\alpha_v + n_{iv})}{\Gamma(\sum_v (\alpha_v + n_{iv}))} \tag{21}$$

One thing to note here is that the MIG prior automatically prunes redundant dimensions via the equation:

$$\lambda_r^{(h)} = \prod_{m=1}^{r} \vartheta_m^{(h)-1} \to 0 \text{ as } r \text{ increases} \tag{22}$$

The entire detailed algorithm for Gibbs sampling is given below.

Algorithm 1 Iterative Posterior Computation with Gibbs Sampler

---

1: Initialize parameters and cluster assignments
2: **repeat**
3:     **Part I: Update parameters given clusters**
4:     **for** $k = 1$ to K **do**
5:         Sample $\mathbf{p}_k \sim \text{Dirichlet}(\alpha_1 + \sum_{i:C_i=k} n_{i1}, \dots)$
6:     **end for**
7:     **for** $i = 1$ to n **do**
8:         Sample G_i from Multinomial$(\tilde{\nu}_1, \dots, \tilde{\nu}_H)$ where:
9:         $\tilde{\nu}_h \propto \nu_{hC_i} \prod_l \pi_l^{(h)\mathscr{L}(A_i)_l}(1 - \pi_l^{(h)})^{1-\mathscr{L}(A_i)_l}$
10:    **end for**
11:    **for** $h = 1$ to H **do**
12:        Sample Polya-Gamma augmented variables:
13:        $\omega_l^{(h)} \sim \text{PG}(n_h, Z_l + \mathscr{L}(X^{(h)}\Lambda^{(h)}X^{(h)\top})_l)$
14:        Update $\mathbf{Z}$ via Gaussian conditional:
15:        $Z_l \sim \mathscr{N}(\mu_{Z_l}, \sigma_{Z_l}^2)$
16:        Update latent coordinates via:
17:        $\mathbf{X}_v^{(h)} \sim \mathscr{N}(\Sigma\eta_v^{(h)}, \Sigma)$
18:        where $\Sigma = (\mathbf{X}_{(-v)}^{(h)\top}\Omega^{(h)}\mathbf{X}_{(-v)}^{(h)} + \Lambda^{(h)-1})^{-1}$
19:    **end for**
20:    **Part II: Update cluster assignments**
21:    **for** $i = 1$ to n **do**
22:        Remove C_i and renumber clusters if empty
23:        Sample new C_i from:
24:        $p(C_i = k | \cdots) \propto \begin{cases} \frac{n_{k,-i}}{n-1+\alpha_c} \prod_v p_{kv}^{n_{iv}} \nu_{G_i k} & \text{existing } k \\ \frac{\alpha_c}{n-1+\alpha_c} p(\mathbf{y}_i|\text{new } k)p(G_i|\text{new } k) & \text{new cluster} \end{cases}$
25:    **end for**
26: **until** convergence

2.4 MCMC diagnostics

The paper suggests the following MCMC convergence diagnostics:

- 5,000 MCMC iterations with 1,000 burn-in

- Trace plots for key parameters

- Relabeling to address potential label switching

- Posterior mode estimation for cluster assignments

3 Simulations

The paper uses this model on a simulated data as well as a business intelligence data set which comprises of monoproduct choice data and co-subscription networks for $n = 130$ agencies, and $V = 15$ different products. Here, in this report, we implemented the model on simulated data.

3.1 Generating data

Following the simulation setup in the paper, we simulated $n = 200$ agencies divided in to $k = 4$ latent clusters. The total number of products was $V = 15$. The monoproduct subscription data $y_{is} = 1, 2, \ldots 200$, $s = 1, \ldots 500$ was generated from a categorical random variable $\mathscr{Y}_k^0$ with probability mass function $\mathbf{p}_k^0$, where $k = 1$ for agencies $i = 1, \ldots 50$, $k = 2$ for agencies $i = 51, \ldots 100$, $k = 3$ for agencies $k = 101, \ldots 150$ and $k = 4$ for $i = 151, \ldots 200$. The number of monoproduct customers for each agency i was taken to be 500. Co-subscription networks were simulated from a 3 component $(\pi^{o(i)}, i = 1, 2, 3)$ mixture given model in (5). Here, $\pi^{o(1)}$ was characterized by on dense community among 10 possible products, $\pi^{o(2)}$ represented four hub products, and $\pi^{o(3)}$ was very similar to $\pi^{o(2)}$ with the exception of product $\nu = 4$, which was held out. The eigen model construction in (6) was avoided in defining $\pi^{o(i)}$ to evaluate how well it characterizes the co-subscription probabilities. While simulating the data $\mathscr{L}(A_i)$, the cluster mixing probability vectors ν_1^0 and ν_2^0 were both taken to be $(0.9, 0.05, 0.05)$, $\nu_3^0 = (0.05, 0.9, 0.05)$ and $\nu_4^0 = (0.05, 0.05, 0.9)$. This was done so that the first scenario $\pi^{o(1)}$ was highly likely in agencies belonging to cluster 1 and 2 (one high probability), and scenarios $\pi^{o(2)}$ and $\pi^{o(3)}$ were more likely in clusters 3 and 4.

3.2 Fitting the model

The simulated data were analyzed using the proposed Bayesian model. The hyperparameters were set to be $a_1 = 2.5$, $a_2 = 3.5$, and $\sigma_l^2 = 10$. The priors were chosen to center the inference around empirical averages. The co-subscription network parameters were centered using the logit of average edge probabilities, i.e., quantities μ_l were defined as

$$\mu_l = \text{logit} \left\{ \sum_{i=1}^{n} \mathscr{L}(A_i)_l / n \right\}.$$

Dirichlet priors for product preferences were set using $\boldsymbol{\alpha} = (1, \ldots, 1)$ with averaged monoproduct frequencies across agencies. A CRP prior with concentration parameter $\alpha_c = 1$ was used for clustering, with results shown to be robust across various α_c values.

The MCMC algorithm was initialized by randomly assigning agencies to $K = 4$ clusters. The MCMC sampler ran for 10,000 iterations with a burn-in of 1,000, using $H = 3$ mixture components and $R = 2$ latent dimensions, with thinning every 5 iterations. The model automatically pruned redundant components during inference.

3.3 Results

Comparing the mean values of the sampled parameters with the ground truth, we get the following values of accuracy measure :

6

Table 1: Model Performance Evaluation Metrics

Metric	Value	Interpretation
Adjusted Rand Index (Clusters)	0.711	**Good** cluster recovery
Adjusted Rand Index (Scenarios)	0.000	**Poor** scenario separation
Product Probability RMSE	0.101	**Good** preference estimation
Edge Probability RMSE (Scenario 1)	0.358	**Moderate** network accuracy
Edge Probability RMSE (Scenario 2)	0.374	**Moderate** network accuracy
Edge Probability RMSE (Scenario 3)	0.373	**Moderate** network accuracy
Mixing Weight RMSE	0.475	**Fair** weight estimation

Key Conclusions

- **Cluster recovery is successful** ($\text{ARI}_C = 0.711$), indicating the model effectively identifies product preference groups.

- **Scenario identification fails completely** ($\text{ARI}_G = 0$), suggesting the co-subscription patterns are not being distinguished.

- **Product preference probabilities** are well-estimated ($\text{RMSE} = 0.101$).

- **Network edge probabilities** show moderate accuracy ($\text{RMSE} \approx 0.37$ across scenarios).

- **Mixing weight estimation** needs improvement ($\text{RMSE} = 0.475$).

Moreover the number of points per cluster were correctly identified as 50 per cluster by the last iteration.

Some other interesting plots are : The significant overlap shows us that the degree of connectivity
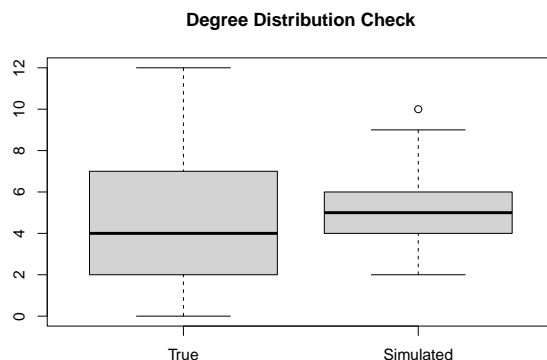

Figure 1: Comparison of Degree distribution

in the cross-purchasing network was correctly estimated by the chain.

There are a lot of variables involved in this estimation, but taking a look at the ACFs of just a few, we get the following :

We see that there is a good degree of convergence in this chain. This is further supported by trace plots of the variables. Taking one such variable being sampled for, we see that the trace plot converges well towards the true value.
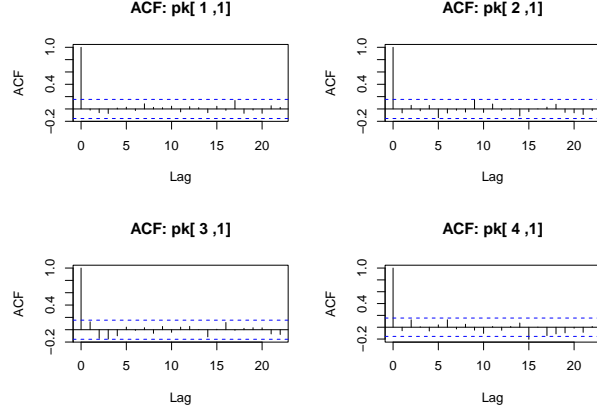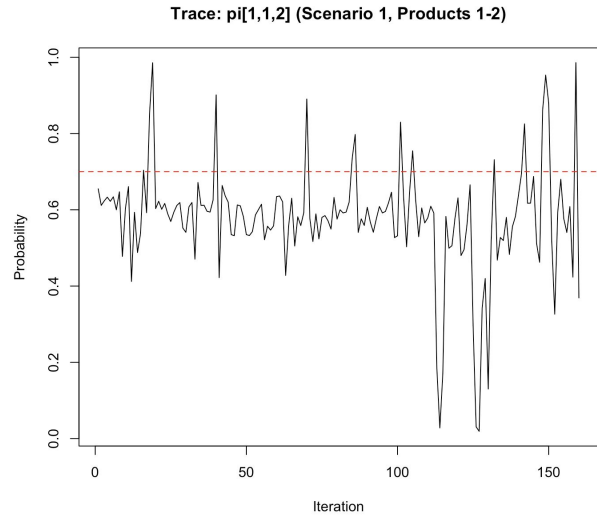
7

Figure 2: ACFs

Figure 3: Trace plot

4 Conclusion

The approach's novel integration of CRP clustering with latent eigenmodel mixtures provides a flexible framework for joint modeling of categorical choices and network data. While the current implementation assumes customer independence and uses fixed edge thresholds, the methodology offers natural extensions for incorporating demographics, temporal dynamics, and cost-benefit analysis. These results demonstrate both the statistical rigor and practical business value of the proposed technique for targeted marketing in insurance markets.

5 Contribution

- **Contribution to finding papers** : Dwija

- **Contribution to understanding the methodology** : Aditya, Dwija, Saee

- **Contribution to coding/understanding and explaining the publicly available codes** : Aditya

- **Generating the figures and tables** : Aditya. Saee

- **Writing the report** : Saee, Dwija

References

Kaishev, V., Nielsen, J., and Thuring, F. (2013). Optimal customer selection for cross-selling of financial services products. *Expert Systems with Applications*, 40:1748–1757.

Kamakura, W. (2008). Cross-selling. *Journal of Relationship Marketing*, 6:41–58.

Kamakura, W. A., Ramaswami, S. N., and Srivastava, R. K. (1991). Applying latent trait analysis in the evaluation of prospects for cross-selling of financial services. *International Journal of Research in Marketing*, 8(4):329–349.

Neal, R. (2000). Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9.

Polson, N. G., Scott, J. G., and Windle, J. (2013). Bayesian inference for logistic models using polya-gamma latent variables.

Thuring, F., Nielsen, J., Guillén, M., and Bolancé, C. (2012). Selecting prospects for cross-selling financial products using multivariate credibility. *Expert Systems with Applications*, 39(10):8809–8816.

Verhoef, P. and Donkers, B. (2001). Predicting customer potential value: An application in the insurance industry. *Decision Support Systems*, 32:189–199.