Exploration vs. Exploitation in Coupon Personalization

by

Aliaa Atwi

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Engineer in Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2018

Author .... Signature redacted ....
Department of Electrical Engineering and Computer Science
January 15, 2018

Certified by .... Signature redacted .....
Devavrat Shah
Professor of Electrical Engineering and Computer Science
Thesis Reader

Accepted by .... Signature redacted
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Exploration vs. Exploitation in Coupon Personalization

by

Aliaa Atwi

Submitted to the Department of Electrical Engineering and Computer Science
on January 15, 2018, in partial fulfillment of the
requirements for the degree of
Engineer in Computer Science

Abstract

Personalized offers aim to maximize profit by taking into account customer preferences inferred from past purchase behavior. For large retailers with extensive product offerings, learning customer preferences can be challenging due to relatively short purchase histories of most customers. To alleviate the dearth of data, we propose exploiting similarities among products and among customers to reduce problem dimensions. We also propose that retailers use personalized offers not only to maximize expected profit, but to actively learn their customers' preferences. An offer that does not maximize expected profit given current information may still provide valuable insights about customer preferences. This information enables more profitable coupon allocation and higher profits in the long run. In this thesis we 1) derive approximate inference algorithms to learn customer preferences from purchase data in real time, 2) formulate the retailers' offer allocation problem as a multiarmed bandit and explore solution strategies.

Thesis Reader: Devavrat Shah
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I am grateful to Prof. John Hauser for his guidance and support throughout this research journey, and to Prof. Devavrat Shah for helpful feedback on the manuscript. I am also thankful to Prof. Leslie Kolodziejski and to Janet Fischer for their mentorship. Finally, I am grateful to my friends and family for their unending love and support. I could not have done it without you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Direct promotion offers are often used to induce trial and encourage brand loyalty, ideally without rewarding purchases that would have otherwise occurred. Personalized offers aim to maximize profit by taking into account customer preferences inferred from past purchase behavior.

Large retailers face unique challenges in offer personalization due to their extensive product lines. For example, the average supermarket carries approximately 38,900 SKUs [1], but most customers purchase a very small subset of these products in a given time frame. This is especially true for new customers. The relative dearth of purchase data makes it difficult to accurately learn customer preferences for effective coupon personalization. The problem of interest is how a large retailer can allocate personalized offers in real time with relatively little data about customer preferences.

To alleviate the dearth of data, we propose that retailers exploit the similarities between products and between customers. For example, a customer who purchased gluten-free pasta is likely to also prefer gluten-free bread. Instead of learning a

[1] Food Marketing Institute, Supermarket facts 2016, www.fmi.org

customer's preference for each product separately, the retailer can learn preferences for attributes that are shared by many products. Similarities between customers can also be exploited. For example, a customer who purchased baby formula is likely to be a parent and have preferences that are similar to other parents.

In addition, we propose that retailers use personalized offers not only to maximize expected profit, but also to actively learn about customer preferences. An offer that does not maximize expected profit given current information may still provide valuable insights about customer preferences. This information enables more profitable coupon allocation and higher profits in the long run. The tradeoff between exploration and exploitation can be captured in the multi-armed bandit framework.

Our goals in this thesis are to provide approximate inference algorithms that allow retailers to learn customer preferences in real time, and to explore scalable solution strategies for the retailers' multiarmed bandit problem. We are also interested in the case of dynamic customer preferences. Preference dynamics occur, for example, when customers learn about product quality through consumption.

Chapter 2

Consumer Model and Real-Time learning of Preference Parameters

2.1 Consumer Model

In each purchase occasion, the consumer sets out to purchase products from an exogenously chosen set of categories. Products are characterized by their levels on n attributes which can be shared across different categories. An attribute is a characteristic of the product (e.g. color), made up of various levels or degrees of that characteristic (e.g. red, yellow, blue). Specifically, a product i is characterized by a vector $x_i \in \mathcal{R}^n$ that describes its levels on the n attributes. Attribute levels are assumed to be fixed in time, except for price, which can change if the retailer decides to offer a coupon on the product. A consumer is characterized by a vector w of her preference weights for attributes. The consumer's expected utility from consuming product i is $w'x_i$. To account for unobserved fluctuations, the consumer's utility u_{it} at each time t is modeled by adding a random shock ϵ_{it} to the expected utility. The random shock is assumed to be zero-mean Gaussian, with a variance β_i^2. Both w and ϵ_{it} are known to the customer, but not directly observed by the retailer. In a given category with $m > 2$ products, the consumer will purchase the product y_t with the maximum utility u_{it}, according to the linear-in-parameters multinomial Probit model:

$$\mathsf{u}_{it} = w'x_{it} + \epsilon_{it} \qquad \text{for i=1,...,m}$$
$$\mathsf{y}_t = \arg\max_i \mathsf{u}_{it},$$

The retailer would like to learn w in real time from sequential purchase observations. To avoid storing a growing purchase history for each consumer, the retailer summarizes its beliefs about w as an independent Gaussian random vector, and updates the mean and covariance after each purchase occasion. Consider the prior :

$$P(w) = \prod_{j=1}^{n} \mathcal{N}(w_j; \mu_j, \sigma_j^2) = \mathcal{N}(w; \mu, \Sigma).$$

The likelihood of observing the purchase $y_1 = i$ is:

$$P(y_1 = i|w; X) = P(u_{i1} > u_{21}, \dots, u_{m1}) = \int_{u_{i1}} \prod_{j \neq i} \Phi\left(\frac{u_{i1} - \vec{x_{j1}}\vec{\mu}}{\beta_j^2 + \sum_{k=1}^{n} x_{jk1}^2 \sigma_k^2}\right)$$

Using Bayes' rule, the posterior is given by:

$$p(w|y; x) \propto p(y|w; x) p(w).$$

The posterior is not Gaussian and cannot be represented compactly, making the inference task computationally inefficient.

A common way to tackle this problem is by using sampling techniques such as MCMC. However, the slow convergence of MCMC makes it unsuitable for real time learning and decisions.

2.2 Proposed approach - Expectation Propagation

We propose approximating the posterior at each iteration with a variational distribution $\tilde{p}(w) = \prod_{i=1}^{N} \mathcal{N}(w_i; \tilde{\mu_i}, \tilde{\sigma}_i^2)$ and passing this independent Gaussian as the prior for the next iteration. The approximate posterior $\tilde{p}(w)$ is chosen to minimize the KL divergence $D(\tilde{p}(w)||p(w))$.
Equivalently, consider the joint pdf of the all the random variables at time t:

$$p(y, \vec{u}, \vec{s}, \vec{w}; X) = p(y|\vec{u}) \prod_{i=1}^{m} p(u_i|s_i) p(s_i|\vec{w}; \vec{x_i}) p(\vec{w})$$

$$= \underbrace{\delta(y = \arg\max_i \vec{u})} \prod_{i=1}^{m} \underbrace{\delta(s_i = x_i \vec{w})} \underbrace{\mathcal{N}(u_i; s_i, \beta^2)} \prod_{i=1}^{n} \underbrace{\mathcal{N}(w_i; \mu_i, \sigma_i^2)} \quad = \prod_k t_k$$

This equation shows that the joint pdf can be expressed as the product of factors (functions) t_k. Since the posterior is proportional to the joint pdf, approximating the joint pdf by a function $q(y, \vec{u}, \vec{s}, \vec{w}; X) = \prod_k \tilde{t_k}$, where all the approximate factors $\tilde{t_k}$ are gaussian guarantees that approximate joint pdf is gaussian, and hence it guarantees that the approximate posterior is gaussian.

<u>Expectation Propagation Algorithm:</u> [Min01a][Min01b]

1. Initialize term approximations $\tilde{t_i}$ to any gaussian terms (for example: the constant 1).

2. Compute the approximate joint pdf from the product of $\tilde{t}_i$:

$$q(y, \vec{u}, \vec{s}, \vec{w}; X) = \prod_i \tilde{t}_i$$

3. Repeat until all $\tilde{t}_i$ converge:

   (a) choose a $\tilde{t}_i$ to refine.

   (b) Remove $\tilde{t}_i$ from the joint pdf to get an 'old' joint pdf

$$q^{\setminus i}(y, \vec{u}, \vec{s}, \vec{w}; X) = \frac{q(y, \vec{u}, \vec{s}, \vec{w}; X)}{\tilde{t}_i} = \prod_{k \neq i} \tilde{t}_k$$

   (c) Find a new q by projecting $t_i q^{\setminus i}(y, \vec{u}, \vec{s}, \vec{w}; X)$ on the class of independent gaussian distributions.

   (d) Update

$$\tilde{t}_i = \frac{q(y, \vec{u}, \vec{s}, \vec{w}; X)}{q^{\setminus i}(y, \vec{u}, \vec{s}, \vec{w}; X)}$$

2.3 Learning Preference Weights of Simple Consumer - Binary Probit Choice Model

In this section, we focus on learning the preference weights for a single consumer deciding whether to purchase a single product.

The product is characterized by a vector of attributes x, and the consumer by a length-N vector of hidden preference weights w.

The consumer derives a random utility

$$U = w^T x + \epsilon$$

from the product, where ϵ is a 0 mean gaussian utility shock. For now, assume that its variance is known and denote its value by β^2. The learning method described extends with small modifications to the case when only a conjugate prior on β is known.

The consumer makes a purchase if the random utility is positive. Let y be 1 if the consumer purchases the product, and -1 otherwise. The likelihood of observing y is given by:

$$P(y|w;x) = \Phi\left(\frac{y \cdot w^T x}{\beta}\right).$$

We would like to learn w from sequential purchase observations. For the purpose of Bayesian learning, assume a priori that w is a gaussian vector with independent components, i.e:

$$P(w) = \prod_{i=1}^{N} \mathcal{N}(w_i; \mu_i, \sigma_i^2).$$

By Bayes' rule, the posterior satisfies

$$p(w|y;x) \propto p(y|w;x)p(w).$$

The posterior is not Gaussian and cannot be represented compactly, making the inference task inefficient.

As proposed, we approximate the posterior after each purchase occasion with a Gaussian and pass this Gaussian as the prior for the next iteration. This approach, minimizes the KL divergence between the posterior and its gaussian approximation by moment matching, i.e. by using the mean and variance of the posterior marginals as parameters for the Gaussian approximation. Finding the mean and variance of the posterior marginals is non-trivial, and we resort to expectation propagation and iterative message passing on factor graphs to accomplish this task.

2.4 Sum Product Algorithm on Factor Graphs [KFL01]

The sum-product algorithm is an efficient way to compute marginals of joint distributions. It exploits the fact that a joint distribution can often be decomposed into the product of several local functions (factors) and uses the distributive law to simplify computations. It also allows reusing intermediate results (partial sums) in computations of different marginals.
For example, consider a function $g(x_1, x_2, \ldots, x_4)$. In the most general case, computing the marginal $g(x_1)$, involves taking summations over all possible tuples $(x_2, \ldots, x_n)$. However, if $g(x_1, \ldots, x_n)$ can be expressed as $f_1(x_1)f_2(x_2, x_3)f_3(x_3, x_4)$, then a more efficient way to marginalize is to invoke distributivity leading to

$$g(x_1) = f_1(x_1) \sum_{x_3} f_2(x_2, x_3) \sum_{x_4} f_3(x_3, x_4)$$

A factor graph is a graphical representation of the way that the joint can be decomposed into a product of factors. It is a bipartite graph consisting of factor nodes and variable nodes. Each local function is represented with a factor node, and each variable by a variable node. An edge connects a variable node x to factor node f if and only if x is an argument of f. When the factor graph is cycle free, it not only encodes the factorization of the joint pdf, but also the expressions needed to compute its marginals.

The algorithm to compute a single marginal (for instance $g(x_1)$) in a cycle-free factor graph proceeds as follows. Treat x_1 as the root of the tree. Messages are passed on the tree from the leaves to the root. Starting at the leaves, each variable message sends a trivial identity message to its parents, and each factor node sends a description of itself to its parent. An intermediate variable node waits for messages to arrive from all its children, takes their product and passes it on to its parent. An intermediate factor node waits for messages to arrive from all its children, takes their product, then sums the result over all variables except the parent variable before passing it on. The algorithm proceeds all the way to the top until variable node x_1 forms the product of all messages received from its children. The result is $g(x_1)$. [1]

This procedure can be repeated for each marginal, or alternatively, a more efficient approach that avoids computing intermediate messages multiple times can be used. Instead of designating one node as a the root and having parent-child edges, treat all nodes equally. Each neighbor of a node at some point assumes the role of parent or child. Starting at the leaves, each node remains idle until it has received messages from all its neighbors except one. This remaining neighbor assumes the role of a parent, and messages are passed to it as in the single marginal algorithm. The algorithm terminates once two messages have been passed on every edge, one in each direction.

For the purpose of outlining the update equations, consider the following notation: Let f be factor, x an argument (neighbor) of f, $n(x)$ the neighbors of x, $n(f)$ the neighbors of f, $m_{x\to f}$ the message from x to f, and $m_{f\to x}$ the message from f to x.

$$m_{x\to f} = \prod_{h\in n(x)/\{f\}} m_{h\to x} \tag{2.1}$$

$$m_{f\to x} = \sum_{y\in n(f)/\{x\}} f \prod_{y\in n(f)/\{x\}} m_{y\to f} \tag{2.2}$$

Finally, finding the marginal at variable x is equivalent to taking the product of

[1] Note that all the messages passed are functions of a single variable.

all incoming messages to x,

$$p(x) = \prod_{h \in n(x)} m_{h \to x}, \tag{2.3}$$

or equivalently, the product on one edge connected to x of an incoming and an outgoing message,

$$p(x) = m_{x \to f} m_{f \to x}. \tag{2.4}$$

If the factor graph has cycles, the procedure is not guaranteed to converge to the exact marginals, and may not converge at all. However, it has been used successfully in practice in many areas to yield very good approximations of the marginals.

Binary Probit Factor Graph [GCBH10]

The factor graph below describes the joint PDF

$$p(y, t, s, w; x) = \underbrace{p(y|t)}_{\text{q}} \underbrace{p(t|s)}_{\text{h}} \underbrace{p(s|w; x)}_{\text{g}} \underbrace{p(w)}_{\text{f}}$$

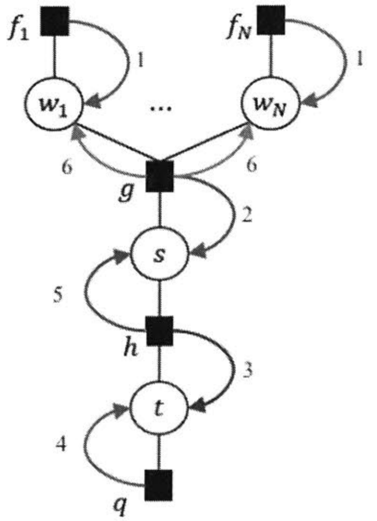
Factor f_i samples weights w from the Gaussian prior $p(w)$.
Factor g calculates the score s for x as $w^T x$ such that $p(s|w; x) = \delta(s = w^T x)$.
Factor h adds Gaussian noise to s to obtain t such that $p(t|s) = \mathcal{N}(t; s, \beta^2)$.
Factor q sets y to 1 if the utility is positive, such that $p(y|t) = \delta(y = sign(t))$.

2.4.1 Update Equations for Binary Probit

(From [GCBH10] with minor modification to allow for non-binary features)

Starting with a prior $P(w) = \prod_{i=1}^{N} \mathcal{N}(w_i; \mu_i, \sigma_i^2)$, the message passing algorithm results in a Gaussian approximate posterior $\tilde{P}(w) = \prod_{i=1}^{N} \mathcal{N}(w_i; \tilde{\mu}_i, \tilde{\sigma}_i^2)$, where the

Figure 2-1: Binary Probit Factory Graph



posterior parameters are given by:

$$\tilde{\mu_i} = \mu_i + yx_i \cdot \frac{\sigma_i^2}{\Sigma} \cdot v\left(\frac{y \cdot x^T\mu}{\Sigma}\right) \tag{2.5}$$

$$\tilde{\sigma_i^2} = \sigma_i^2 \cdot \left[1 - x_i\frac{\sigma_i^2}{\Sigma^2} \cdot w\left(\frac{y \cdot x^T\mu}{\Sigma}\right)\right], \tag{2.6}$$

where,

$$\Sigma^2 = \beta^2 + \sum_{i=1}^{N} x_i^2\sigma_i^2$$

$$v(t) = \frac{\mathcal{N}(t;0,1)}{\Phi(t;0,1)}$$

$$w(t) = v(t) \cdot [v(t) + t] \tag{2.7}$$

.

Derivation of update equations

The update equations above are obtained from calculating the messages labeled 1-6 in the binary probit factor graph.

Since all of the messages are Gaussian (and those that are non Gaussian are approximated by Gaussians), means and variances can be conveniently passed as messages instead of complete PDFs.[2]

Message 1:

The messages $f_i \to w_i$ are simply: $\begin{cases} \mu_i \\ \sigma_i^2 \end{cases}$

Message 2:

Using the sum-product algorithm update equation 3 gives:

$$\begin{aligned} m_{g \to s} &= \int_{\vec{w}} \delta(s = w^T x) \prod_{i=1}^{N} \mathcal{N}(w_i; \mu_i, \sigma_i^2) d\vec{w} \\ &= \mathcal{N}\left(s; \sum_i x_i \mu_i, \sum_i x_i^2 \sigma_i^2\right) \end{aligned}$$

where the second equality follows from recognizing that $x^T w$ is a linear combination of independent Gaussians and hence a Gaussian.

Message 3:

Using the sum-product algorithm update equation 3 gives:

$$\begin{aligned} m_{h \to t} &= \int_{s} \mathcal{N}(t; s, \beta^2) \cdot \mathcal{N}(s; \sum_i x_i \mu_i, \sum_i x_i^2 \sigma_i^2) ds \\ &\propto \mathcal{N}\left(t; \sum_i x_i \mu_i,\ \beta^2 + \sum_i x_i^2 \sigma_i^2\right) \end{aligned}$$

[2] In general, as long as all continuous distributions involved are in the exponential family, we can pass sufficient statistics instead of complete distributions

where the proportionality term follows from equation (4.1) in the Appendix.

Message 4: [HMG06]

According to the sum-product algorithm, the true message $m_{q \to t}$ is simply the description of factor q itself, $\delta\left(y = sign(t)\right)$, which is non-Gaussian. Following the expectation propagation algorithm, we approximate this message by a Gaussian as follows:

- Approximate the marginal $p(t)$ by a Gaussian $\hat{p}(t)$ through moment matching.

- Using equation (2.4), we know that $p(t) = m_{t \to q} \cdot m_{q \to t}$. So $\hat{p}(t) = m_{t \to q} \cdot \hat{m}_{q \to t}$. We can use this to find the approximate message $\hat{m}_{q \to t}$ as $\frac{\hat{p}(t)}{m_{t \to q}}$. Since both $\hat{p}(t)$ and $m_{t \to q}$ are Gaussian, $\hat{m}_{q \to t}$ is also Gaussian

First, the marginal $p(t)$ satisfies the following:

$$\begin{aligned} p(t) &= \delta(y = sign(t)) \cdot \mathcal{N}\left(t; \sum_i x_i \mu_i, \beta^2 + \sum_i x_i^2 \sigma_i^2\right) \\ &= \begin{cases} TN^+\left(t; \sum_i x_i \mu_i,\ \beta^2 + \sum_i x_i^2 \sigma_i^2\right) & \text{for } t > 0 \text{ and } y = +1 \\ TN^-\left(t; \sum_i x_i \mu_i,\ \beta^2 + \sum_i x_i^2 \sigma_i^2\right) & \text{for } t < 0 \text{ and } y = -1 \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

where TN^+ (TN^-) is the positive (negative) truncated normal PDF.

For compactness, we denote $U := \sum_i x_i \mu_i$ and as before $\Sigma^2 := \beta^2 + \sum_i x_i^2 \sigma_i^2$.

By moment matching, we approximate the truncated normal by a Gaussian with mean $\tilde{U}$ and $\tilde{\Sigma}^2$:

$$\begin{aligned}
\tilde{U} &= U + y\Sigma \cdot v\left(y \cdot \frac{\mu}{\sigma}\right) \\
\tilde{\Sigma}^2 &= \Sigma^2\left[1 - w\left(y \cdot \frac{\mu}{\Sigma}\right)\right]
\end{aligned}$$

These are simply the expressions for the mean and variance of a truncated normal, where $v(.)$ and $w(.)$ are defined in (2.7). As per expectation propagation, the approximate message is then,

$$\begin{aligned}
\hat{m}_{q\to t} &= \frac{\hat{p}(t)}{m_{t\to q}} \\
&= \frac{\mathcal{N}(t;\tilde{U},\tilde{\Sigma}^2)}{\mathcal{N}(t;U;\Sigma^2)} \\
&\propto \mathcal{N}\left(t;\frac{\Sigma^2\tilde{U} - \tilde{\Sigma}^2 U}{\Sigma^2 - \tilde{\Sigma}^2}, \frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2 - \tilde{\Sigma}^2}\right)
\end{aligned}$$

where the last proportionality term follows from equation (4.2) in the Appendix.

Message 5:

The message is given by:

$$\begin{aligned}
m_{h\to s} &= \int_t \mathcal{N}(t;s,\beta^2)\mathcal{N}\left(t;\frac{\Sigma^2\tilde{U} - \tilde{\Sigma}^2 U}{\Sigma^2 - \tilde{\Sigma}^2}, \frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2 - \tilde{\Sigma}^2}\right) dt \\
&= \int_t \mathcal{N}\left(s;\frac{\Sigma^2\tilde{U} - \tilde{\Sigma}^2 U}{\Sigma^2 - \tilde{\Sigma}^2},\ \beta^2 + \frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2 - \tilde{\Sigma}^2}\right)\mathcal{N}\left(t;U_3,\Sigma_3^2\right) dt \\
&= \mathcal{N}\left(s;\frac{\Sigma^2\tilde{U} - \tilde{\Sigma}^2 U}{\Sigma^2 - \tilde{\Sigma}^2},\ \beta^2 + \frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2 - \tilde{\Sigma}^2}\right)
\end{aligned}$$

where the second equality results from equation (4.1) in the Appendix, and U_3 and Σ_3^2 are defined as they appear in equation (4.1) (exact expressions omitted for space). The last equality holds since $\mathcal{N}\left(t; U_3, \Sigma_3^2\right)$ normalizes to 1, and integration is over t not s.

Message 6:

Finally, factor g sends a message to each of $w_1, \ldots, w_N$. To send a message to w_i it takes the product of its own description with all messages it received from all other $w_{j,j\neq i}$ as well as the message it received from s. W.L.O.G:

$$
\begin{aligned}
m_{q\rightarrow w_1} &= \int_{w_2\ldots w_N,s} \delta(s = w^T x) \prod_{i=2}^{N} \mathcal{N}\left(w_i; \mu_i, \sigma_i^2\right) \cdot \mathcal{N}\left(s; \frac{\Sigma^2\tilde{U} - \tilde{\Sigma}^2 U}{\Sigma^2 - \tilde{\Sigma}^2},\ \beta^2 \frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2 - \tilde{\Sigma}^2}\right) dw_2, \ldots, w_N, s \\
&= \int_{w_2,\ldots,w_N,s} \delta\left(w_1 = \frac{s - w_2x_2 - \ldots - w_Nx_N}{x_1}\right) \prod_{i=2}^{N} \mathcal{N}\left(w_i; \mu_i, \sigma_i^2\right) \\
&\quad \cdot \mathcal{N}\left(s; \frac{\Sigma^2\tilde{U} - \tilde{\Sigma}^2 U}{\Sigma^2 - \tilde{\Sigma}^2},\ \beta^2 \frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2 - \tilde{\Sigma}^2}\right) dw_2, \ldots, w_N, s \\
&= \mathcal{N}\left(w_1; \frac{1}{x_1}\left[\frac{\Sigma^2\tilde{U} - \tilde{\Sigma}^2 U}{\Sigma^2 - \tilde{\Sigma}^2} - \sum_{i=2}^{N} x_i\mu_i\right],\ \frac{1}{x_1^2}\left[\beta^2 + \frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2 - \tilde{\Sigma}^2} + \sum_{i=2}^{N} x_i^2\sigma_i^2\right]\right)
\end{aligned}
$$

where the second equality follows by reordering terms inside $\delta(s = w^T x)$, and the third equality follows by treating $\frac{s - w_2x_2 - \ldots - w_Nx_N}{x_1}$ as a linear combination combination of independent Gaussian terms, with the marginal Gaussian distributions appearing inside the integral.

Resulting Approximate Posterior Marginals:

The approximate marginal posterior at w_i is

$$\begin{aligned}
p(w_1) &= m_{q\to w_1} \cdot \mathcal{N}(w_1; \mu_1, \sigma_i^2) \\
&= \mathcal{N}\left(w_1; \frac{1}{x_1}\left[\frac{\Sigma^2\tilde{U} - \tilde{\Sigma}^2 U}{\Sigma^2 - \tilde{\Sigma}^2} - \sum_{i=2}^{N} x_i\mu_i\right], \frac{1}{x_1^2}\left[\beta^2 + \frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2 - \tilde{\Sigma}^2} + \sum_{i=2}^{N} x_i^2\sigma_i^2\right]\right) \\
&\quad \cdot \mathcal{N}(w_i; \mu_i, \sigma_i^2) \\
&= \mathcal{N}\left(w_1; \tilde{\mu_1}, \tilde{\sigma_1}^2\right)
\end{aligned}$$

Using equation (4.1) in the Appendix, the posterior variance is given by:

$$\begin{aligned}
\tilde{\sigma_i}^2 &= \frac{\sigma^2\left[\frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2-\tilde{\Sigma}^2} + \sum_{i=2}^{N} x_i^2\sigma_i^2 + \beta^2\right]}{\sigma_1^2 x_1^2 + \beta^2 + \sum_{i=2}^{N} x_i^2\sigma_i^2 + \frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2-\tilde{\Sigma}^2}} \\
&= \frac{\sigma_1^2\Sigma^2\tilde{\Sigma}^2 + \sigma_1^2\left(\Sigma^2 - x_1^2\sigma_1^2\right)\left(\Sigma^2 - \tilde{\Sigma}^2\right)}{\Sigma^2\tilde{\Sigma}^2 + \Sigma^2\left(\Sigma^2 - \tilde{\Sigma}^2\right)} \\
&= \frac{\sigma_1^2\left[\Sigma^2\tilde{\Sigma}^2 + \Sigma^2\left(\Sigma^2 - \tilde{\Sigma}^2\right) - x_1^2\sigma_1^2\left(\Sigma^2 - \tilde{\Sigma}^2\right)\right]}{\Sigma^2\tilde{\Sigma}^2 + \Sigma^2\left(\Sigma^2 - \tilde{\Sigma}^2\right)} \\
&= \sigma_1^2\left[1 - \frac{x_1^2\sigma_1^2\left(\Sigma^2 - \tilde{\Sigma}^2\right)}{\Sigma^4}\right] \\
&= \sigma_1^2\left[1 - x_1^2\frac{\sigma_1^2}{\Sigma^2}\left(1 - \frac{\tilde{\Sigma}^2}{\Sigma^2}\right)\right] \\
&= \sigma_1^2 \cdot \left[1 - x_1\frac{\sigma_1^2}{\Sigma^2} \cdot w\left(\frac{y \cdot x^T\mu}{\Sigma}\right)\right],
\end{aligned}$$

where the last equality was obtained by substituting the expression for $\tilde{\Sigma}^2$ and simplifying the expression.

Similarly, we can find the posterior mean as follows:

$$
\begin{aligned}
\mu_{new} &= \frac{\sigma_1^{-2}\mu_1 + \frac{x_1^2}{\beta^2+\frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2-\tilde{\Sigma}^2}+\sum\limits_{i=2}^{N} x_i^2\sigma_i^2} \cdot \frac{1}{x_1}\left[\frac{\Sigma^2\tilde{U}-\tilde{\Sigma}^2U}{\Sigma^2-\tilde{\Sigma}^2} - \sum\limits_{i=2}^{N} x_i\mu_i\right]}{\frac{x_1^2}{\beta^2+\frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2-\tilde{\Sigma}^2}+\sum\limits_{i=2}^{N} x_i^2\sigma_i^2} + \sigma_1^{-2}} \\
&= \frac{\mu_1\left(\beta^2 + \frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2-\tilde{\Sigma}^2} + \sum_{i=2}^{N} x_i^2\sigma_i^2\right) + \frac{\sigma_1^2 X_1^2}{x_1}\left[\frac{\Sigma^2\tilde{U}-\tilde{\Sigma}^2U}{\Sigma^2-\tilde{\Sigma}^2} - \sum\limits_{i=2}^{N} x_i\mu_i\right]}{\sigma_1^2x_1^2 + \sum_{i=2}^{N} x_1^2\sigma_i^2 + \beta^2 + \frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2-\tilde{\Sigma}^2}} \\
&= \frac{\mu_1\left[\Sigma^2 - \sigma_1^2x_1^2 + \frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2-\tilde{\Sigma}^2}\right] + \sigma_1^2x_1\left[\frac{\Sigma^2\tilde{U}-\tilde{\Sigma}^2U}{\Sigma^2-\tilde{\Sigma}^2} - \sum\limits_{i=2}^{N} x_i\mu_i\right]}{\Sigma^2 + \frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2-\tilde{\Sigma}^2}} \\
&= \mu_1 + \frac{-\sigma_1^2x_1^2\mu_1 + \sigma_1^2x_1\left[\frac{\Sigma^2\tilde{U}-\tilde{\Sigma}^2U}{\Sigma^2-\tilde{\Sigma}^2} - \sum\limits_{i=2}^{N} x_i\mu_i\right]}{\frac{\Sigma^2\tilde{\Sigma}^2}{\Sigma^2-\tilde{\Sigma}^2}} \\
&= \mu_1 + \frac{\sigma_1^2x_1\left[-\mu\left(\Sigma^2-\tilde{\Sigma}^2\right) + \Sigma^2\tilde{\mu} - \tilde{\Sigma}^2\mu\right]}{\Sigma^2\left(\Sigma^2-\tilde{\Sigma}^2\right) + \Sigma^2\tilde{\Sigma}^2} \\
&= \mu_1 + \frac{\sigma_1^2x_1\Sigma^2(\tilde{\mu}-\mu)}{\Sigma^2(\Sigma^2-\tilde{\Sigma}^2) + \Sigma^2\tilde{\Sigma}^2} \\
&= \mu_1 + \frac{\sigma_1^2x_1}{\Sigma^2}\left(\tilde{\mu}-\mu\right) \\
&= \mu_1 + \frac{x_1y\sigma_1^2}{\Sigma}v\left(\frac{y\mu}{\Sigma}\right)
\end{aligned}
$$

2.4.2 Gibbs Sampling with Data Augmentation

An alternative method for learning the preference parameters is to use Gibbs sampling. The procedure below is for one consumer and multiple products (indexed by j). Let X be an MxN matrix where each row x_j represents the attributes (indexed by i) of a single product. y_j is 1 if product j is purchased and -1 otherwise. U_j is the utility from consuming product j. The prior on w is $\mathcal{N}\left(\vec{\mu_0}, \sigma^2 \Lambda_0^{-1}\right)$.

Procedure

1. Choose an initial w^0

2. For $m = 1, 2, \ldots$

   (a) For each product j sample $U_j^{(m)} \sim \begin{cases} TN^+\left(U_j; w^{m-1}x_j; \beta^2\right) & \text{if } y = 1, \\ TN^-\left(U_j; w^{m-1}x_j; \beta^2\right) & \text{if } y = -1. \end{cases}$

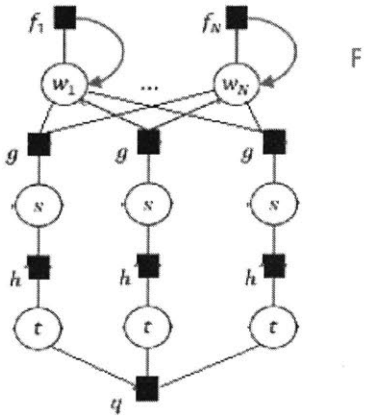
   (b) Sample $w^{(m)} \sim \mathcal{N}\left(w^m; \left(X^TX + \Lambda_0\right)^{-1}\left(X^TU^{(m)} + \Lambda_0\mu_0\right), \left(X^TX + \Lambda_0\right)\right)$

This is the procedure for the first purchase occasion. The posterior is non-Gaussian, but we have samples from it. One possibility is to again approximate the posterior by a Gaussian (by fitting the samples) and pass that as the prior for the next iteration.

2.4.3 Multinomial Probit Formulation

In this section, we extend the expectation propagation update equations to the case when the customer can choose among multiple products, i.e. to the multinomial probit choice model. The main difference with the binary probit case is the presence of cycles in the factor graph. There are no closed form expressions for the final values of the preference parameters. Instead, we need to run the message passing algorithm iteratively until convergence, which is not guaranteed. The messages passed in the

Figure 2-2: Linear in Parameters Multinomial Probit Factor Graph



multinomial probit factor graph are exactly the same as those in the binary probit, except for message 4, which we derive below.

Without a loss of generality, assume that the product purchased is m. The true message $m_{q\to t_j}$ according to the sum product algorithm is:

$$m_{q\to t_j} = \int \delta\left(t_m > t_1, \ldots, t_{m-1}\right) \prod_{i\neq j} m_{t_i\to q}\, d\, t_{i\neq j}$$

However, this is not gaussian. As in the binary case outlined above, expectation propagation approximates the message by a gaussian by moment matching. The approximate message is:

$$\hat{m_{q\to t_j}} = \frac{\hat{P_{t_j}}}{m_{t_j\to q}},$$

where P_{t_j} is the non-gaussian marginal density of t_j, and $\hat{P}_{t_j}$ is the gaussian approximation of the marginal density. For $j \neq m$,

$$\begin{aligned}
P_{t_j} &= m_{q \to t_j} m_{t_j \to q} \\
&= \int \delta\left(t_m > t_1, \ldots, t_{m-1}\right) \prod_{k=1}^{m} \mathcal{N}\left(t_k; \sum_i x_{ki}\mu_i, \beta_k^2 + \sum_i x_{ki}^2 \sigma_i^2\right) \; d\, ti \neq j \\
&= \int \delta\left(t_1 < t_m\right) \delta\left(t_2 < t_m\right) \ldots \delta\left(t_{m-1} < t_m\right) \\
&\cdot \prod_{k=1}^{m} \mathcal{N}\left(t_k; \sum_i x_{ki}\mu_i, \beta_k^2 + \sum_i x_{ki}^2 \sigma_i^2\right) \; d\, ti \neq j
\end{aligned}$$

For compactness, denote $\sum_i x_{ki}\mu_i$ by u_k and $\beta_k^2 + \sum_i x_{ki}^2 \sigma_i^2$ by s_k^2.

$$P_{t_j} = \int \delta\left(t_j < t_m\right) \mathcal{N}\left(t_j; u_j, s_j^2\right) \cdot \mathcal{N}\left(t_m; u_m, s_m^2\right) \prod_{k \neq m,j} \Phi\left(\frac{t_m - u_k}{s_k}\right) \; d\, t_m$$

We use moment matching to approximate P_{t_j} with a gaussian. The mean of t_j is

$$\begin{aligned}
E[t_j] &= \int t_j P_{t_j} \; d\, t_j \\
&= \int \mathcal{N}\left(t_m; u_m, s_m^2\right) \left[\int t_j \delta\left(t_j < t_m\right) \mathcal{N}\left(t_j; u_j, s_j^2\right) \; d\, t_j\right] \prod_{k \neq m,j} \Phi\left(\frac{t_m - u_k}{s_k}\right) \; d\, t_m \\
&= \int \mathcal{N}\left(t_m; u_m, s_m^2\right) \left[u_j - \frac{\phi\left(\frac{t_m - u_j}{s_j}\right)}{\Phi\left(\frac{t_m - u_j}{s_j}\right)}\right] \prod_{k \neq m,j} \Phi\left(\frac{t_m - u_k}{s_k}\right),
\end{aligned}$$

where the last equality follows from the formula of the expected value of a truncated normal, truncated from the right at t_m.

Similarly, the second moment is

$$
\begin{aligned}
E[t_j^2] &= \int t_j^2 P_{t_j} \; d\, t_j \\
&= \int \mathcal{N}\left(t_m; u_m, s_m^2\right) \left[\int t_j^2 \delta\left(t_j < t_m\right) \mathcal{N}\left(t_j; u_j, s_j^2\right) \; d\, t_j\right] \prod_{k \neq m,j} \Phi\left(\frac{t_m - u_k}{s_k}\right) \; d\, t_m \\
&= \int \mathcal{N}\left(t_m; u_m, s_m^2\right) \left[s_j^2 + u_j^2 - s_j\left(t_m - u_j\right) \frac{\phi\left(\frac{t_m - u_j}{s_j}\right)}{\Phi\left(\frac{t_m - u_j}{s_j}\right)}\right] \prod_{k \neq m,j} \Phi\left(\frac{t_m - u_k}{s_k}\right) \; d\, t_m,
\end{aligned}
$$

where the last equality follows from the formula of the second moment of a truncated normal, truncated from the right at t_m.

For $j = m$,

$$
\begin{aligned}
E[t_m] &= \int t_m \mathcal{N}\left(t_m; u_m, s_m^2\right) \prod_{k \neq m} \Phi\left(\frac{t_m - u_k}{s_k}\right) \; d\, t_m \\
E[t_m] &= \int t_m^2 \mathcal{N}\left(t_m; u_m, s_m^2\right) \prod_{k \neq m} \Phi\left(\frac{t_m - u_k}{s_k}\right) \; d\, t_m
\end{aligned}
$$

These integrals were computed numerically in MATLAB. Finally, the message from q to t_j is computed as

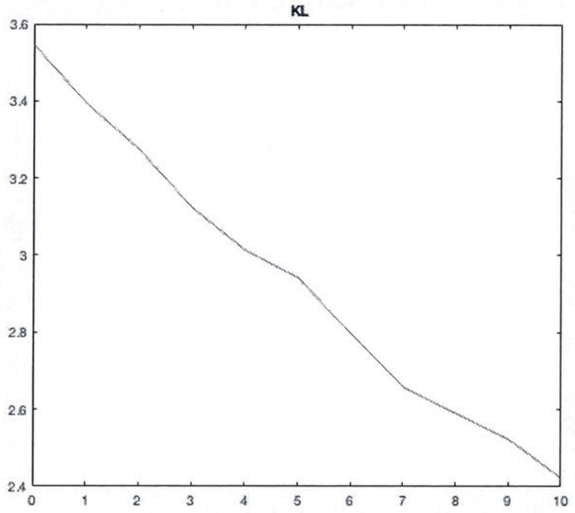
$$
\begin{aligned}
m_{q \hat{\to} t_j} &= \frac{\hat{P}_{t_j}}{m_{t_j \to q}} \\
&= \frac{\mathcal{N}(t_j, E[t_j], E[t_j^2] + E[t_j]^2)}{\mathcal{N}\left(t_j; \sum_i x_{ji}\mu_i, \beta_j^2 + \sum_i x_{ji}^2 \sigma_i^2\right)}
\end{aligned}
$$

2.4.4 Simulations

To confirm that the EP procedure outlined above converges in practice towards the true values of the mean and variance of the preference parameters, we ran simulation experiments. We measure the KL divergence between the true distribution of w and the approximate posterior over w obtained from EP after each purchase. The KL divergence decreases as we observe more customer choices, indicating that the approximate posterior approaches the true distribution.

The figure below shows the KL divergence after each of 10 purchase occasion for a sample run on the algorithm with 17 products, and 6 attributes including price. The parameters used are included in appendix B.

Figure 2-3: KL divergence between the true preference distribution and the approximate; posterior distribution obtained from Expectation Propagation after each purchase occasion



Chapter 3

Retailer's Optimization: Multi-Armed Bandit Problem

In each purchase occasion, the retailer provides a personalized coupon (price discount) for a at most one product in the category. The goal of the retailer is to maximize its longterm profits. A coupon is characterized by the pair (a_t, p_{a_t}), where a_t is the index of the product to be discounted, and p_{a_t} is the resulting reduced price on that product. Let r_t be the retailer's immediate profit in stage t

$$\mathsf{r}_t = p_{\mathsf{y}_t} - c_{\mathsf{y}_t},$$

where y_t is the consumers' choice as a function of a_t, and p_{a_t}, and c_{y_t} is the cost of product y_t to the retailer.

If the retailer had perfect knowledge of $\mathbf{w}$, the optimal policy would be to give the coupon that maximizes the expected immediate profit in each period:

$$(a_t, p_{a_t}) = \arg\max_{a_t, p_{a_t}} E_{\mathsf{y}_t(a_t, p_{a_t})} r_t.$$

In the absence of this knowledge, the retailer is faced with a tradeoff between optimizing immediate profit based on its partial knowledge of **w**, and trying to learn more about **w** to allow better coupon allocation in the longterm. This tradeoff between exploration and exploitation is captured by the multi-armed bandit framework.

3.1 Related Work

The multi-armed bandit problem was introduced by Thompson in 1933. It is concerned with a decision maker who is given a set of statistical processes (arms) with unknown reward profiles. The goal of the decision marker is to adaptively sample the arms in a sequence that maximizes the total longterm expected reward. At the heart of the problem is the need to balance exploring under-sampled arms to learn their reward profile, and exploiting arms that have demonstrated high rewards.

In their seminal work, Gittins and Jones have shown that the optimal solution for the discounted infinite horizon problem with finitely many independent arms takes the form of an index rule [GJ79] [Git79]. An index for each arm is calculated in each period, and the arm with the highest index is chosen. The index of an arm can be interpreted as the 'present value' of playing that arm infinitely many times into the future. Independence of the arms allows decoupling the problem into an optimal stopping problem over each arm. Whittle (1988) derived conditions for the indexability of multi-armed bandit problems in more general settings [Whi88].

In the setting we are considering, the rewards from different arms are correlated, so the Gittins index does not provide exact solutions for the bandit problem. Treating the rewards as independent may lead us to systematically underestimate the value of information for some arms. Moreover, the problem is not indexable.

The multi-armed bandit problem with correlated arms has been studied in the regret minimization literature, which focuses on the finite horizon bandit problem and tries to find policies that minimize regret asymptotically for long horizons. Regret is the difference between using a given policy and playing the optimal arm for the length of the horizon. In their seminal paper, Lai and Robbins (1985) proved a bound on the regret for bandit problems with independent arms. The bound is $\Omega(m \log T)$, where m is the number of arms and T is the horizon. They also show that the bound is tight by suggesting a policy that can achieve it. [LR85]

Intuitively speaking, however, allowing correlation between arms should result in more efficient exploration, creating more opportunities to exploit optimal arms and minimize regret. Mersereau et al. (2009) focused on the case where the expected rewards of all the arms are linear functions of the same (unobserved) random variable. In that case, the mean payoffs of different arms are perfectly correlated, making them equivalent in terms of exploration. A greedy policy, which myopically exploits the current estimated best arm without regard to exploration is thus optimal. The interesting result is that the regret of the greedy policy is $O(\sqrt{T})$ regardless of the number of arms [MRT09].

Finally, Rusmevichinetong and Tsitsiklis (2010), consider the case when the expected rewards of the arms are linear functions of multiple random variables [RT10]. The correlation between different arms is no longer perfect, and a greedy policy is no longer optimal. The authors show that if the arms are "strongly convex" a policy that goes through disjoint phases of exploration followed by phases of exploitation is optimal is the sense that it achieves the lower bound on regret $\Omega(n\sqrt{T})$ where n is the number of underlying random variables. In the general case, they prove that the 'uncertainty ellipsoid" (UE) algorithm achieves near optimal regret [1]. The uncertainty ellipsoid algorithm gives under-explored arms 'the benefit of the doubt'. More specifically, let $\hat{\vec{w}}$ be the current LMS estimate of $\vec{w}$. An uncertainty ellipsoid $\mathcal{E}_t$ around $\hat{\vec{w}}$ is defined by:

$$\mathcal{E}_t = \left\{ \vec{w} \in R^r : ||\vec{w} - \hat{\vec{w}}||_{M_t} \leq \rho(t) \right\},$$

where ρ is an appropriate slowly increasing function, and

$$M_t = \sum_{k=1}^{t-1} X_t X_t',$$

is the design matrix corresponding to the first t-1 time steps. It captures the likely error in the LMS estimate of $\vec{w}$. The uncertainty radius R_t^a associated with each arm a is

$$R_t^a = \max_{v \in \mathcal{E}_t} v' a$$

The UE policy chooses the arm a which maximizes $a' \hat{\vec{w}} + R_t^a$. The results from Rusmevichinetong and Tsitsiklis (2010) also do not carry over directly to our case, even though the perceived utility is a linear function of the product characteristics. The reason is that the retailer cannot observe these utilities directly. Instead, it observes the product with the highest utility.

3.2 Thompson Sampling

Thompson sampling is a heuristic proposed by William R. Thompson in 1933 to address the exploration-exploitation trade-off [Tho33]. It has received renewed at-

[1] a logarithmic factor away from the lower bound on regret

tention due to its applicability to general reward distributions and correlation structures, and because of how naturally it interfaces with Bayesian systems. Thompson sampling is a randomized policy which, at each step, samples an arm according to the posterior probability of it yielding the highest rewards.

Let $(y_1, \ldots, y_t)$ be the purchase history observed up to time t, and let a_t be the arm played at t. The arm played corresponds to the product that receives a coupon. As discussed in chapter 2, customer purchases are modeled by a linear in parameters multinomial probit. The parameters are the customer's preference weights w for different attributes of the products. Let $R_a(w)$ be the expected return from pulling arm a given that the customer's preference weights are w, i.e.:

$$R_a(w) = E[p_{y_t} - c_{yt} | w, a_t = a]$$

If the preference weights w were known, the optimal strategy would be to always pull the arm that maximizes expected return, i.e.$arg \max_a R_a(w)$. However, w is unknown. We start with a prior distribution on w, and perform bayesian updates on the distribution when purchases are observed as described in chapter 2. Let W_{at} be the probability that a is the winning arm at time t, i.e. the probability that giving a coupon for product a yields the maximum expected return (where expectation is taken over the shock ϵ_t). At $t = 0$, we can use the prior on w to compute:

$$W_{a0} = P\left(R_a = \max\{R_1, \ldots, R_m\}\right)$$

Upon receiving observations, the posterior on w can be used to compute:

$$W_{at} = P\left(R_a = \max\{R_1, \ldots, R_m\} | y_1, \ldots, y_t\right)$$

Thompson sampling pulls arm a at time t with probability W_at. The probability W_{at} can be computed by simulation. Let $I_a(w)$ be an indicator random variable that takes the value 1 if arm a yields the highest expected reward at time t, i.e.:

$$I_a(w) = \begin{cases} 1, & \text{if } R_a(w) = \max\{R_1, \dots, R_m\} \\ 0, & \text{otherwise.} \end{cases}$$

Then we can express W_{a0} and W_{at} in terms of $I_a(w)$:

$$\begin{aligned} W_{a0} &= E[I_a(w)] \\ W_{at} &= E[I_a(w)|y_1, \dots, y_t] \end{aligned}$$

Therefore, we can compute W_{at} by simulation as follows:

- Obtain L independent samples from the posterior distributions on w: $w^{(1)}, \dots, w^{(L)}$.
- Compute $W_{at}^{(L)} = \frac{1}{L} \sum_{l=1}^{L} I_a(w^{(l)})$.

By the law of large numbers, $W_{at} = \lim_{L \to \infty} W_{at}^{(L)}$.

In practice, it is faster to obtain one sample $w^{(o)}$ from the posterior $P(w|y_1, y_t)$, and play the arm $a = arg\max_a R_a(w^{(o)})$ [Sco10].

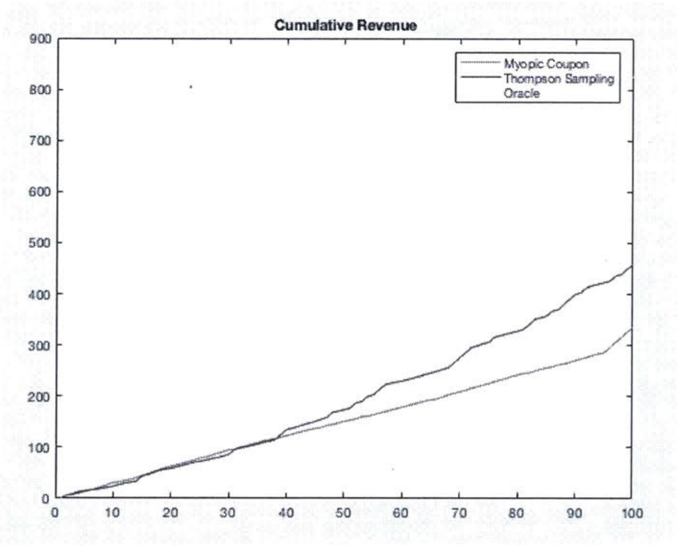
Other potential approaches are UCB_GLM, which extends the uncertainty ellipsoid method to generalized linear models [FCGS10], as well as the Bayes_UCB algorithm which is asymptotically optimal for simple problems like binary bandits, and has natural extensions to parametric multi-armed bandits [KCG12].

We conduct simulation studies that compare the performance of different solutions for realistic problem parameters. Natural benchmarks are the myopic retailer strategy which does not take into account the value of information, the strategy of a retailer with perfect knowledge of $\vec{w}$, and a one-step-look ahead approximate DP solution.

3.3 Simulations

We compared the performance of Thompson Sampling and the myopic approach (no exploration) on toy data. For a problem with 9 products, 6 attribute levels (including price), and 3 possible discount levels (no discount, 25% off, and 50% off) , Thompson Sampling achieved higher cumulative revenue over 100 purchase occasions than the myopic approach. The figure below shows the cumulative revenue from Thompson Sampling and the myopic approach compared to the cumulative revenue achieved by an oracle that knows the optimal arm to play. The parameters are included in appendix C.

Figure 3-1: Cumulative revenue in 100 purchase occasions using 1) Myopic Couponing Policy (no exploration) versus 2) Thompson Sampling versus 3) Oracle that always pulls optimal arm



Chapter 4

Conclusions and Future Work

In this thesis, we propose that in addition to encouraging purchase, coupons can be used to actively explore the preferences of customers, which can help with the cold start problem faced by large retailers. The tradeoff between exploration and exploitation is captured by the multiarmed bandit problem. We modeled customer choice by a linear in parameters multinomial probit model. Instead of learning the customers' preferences for different products independently, the linear in parameters Probit allows the retailer to learn the preferences for attributes shared by many products at once.

After each purchase occasion, the retailer has to update its beliefs about the customer preferences in real time. The true posterior distribution cannot be expressed in closed form making exact inference intractable. We proposed using expectation propagation to approximate the posterior with an independent gaussian using moment matching. Update equations for expectation propagation over a factor graph were derived for the binary probit and linear in parameters multinomial probit. Simulations show that the approximate posterior approaches the true distribution of the preference weights with each purchase occasion.

We use Thompson sampling to balance exploration and exploitation in deciding which product to coupon. Instead of myopically exploiting the current estimates 'best' (most profitable) arm, Thompson sampling explores all possible product in proportion to their likelihood of being optimal. Simulations show that Thompson sampling yields higher cumulative profits than the myopic approach for realistic toy examples.

Future work can explore dynamic customer preferences. If customers learn about their utility for products through experience, retailers can use coupons to induce trial and influence customer choices thereafter.

4.1 Appendix

4.1.1 Elementary Gaussian Formulas

To simplify expressions, $\mathcal{N}(\cdot;\mu,\sigma^2)$ will sometimes be represented in terms of its canonical parameters: the precision, $\pi := \sigma^{-2}$, and the precision adjusted mean, $\tau := \pi\mu$.

Product and Ratio of Two Gaussian PDFs

The product of two Gaussian PDFs is given by:

$$\mathcal{N}(x;\mu_1,\sigma_1^2)\mathcal{N}(x;\mu_2,\sigma_2^2) = \mathcal{N}\left(\mu_1;\mu_2,\sigma_1^2+\sigma_2^2\right)\mathcal{N}(x;\mu_3,\sigma_3^2) \tag{4.1}$$

Where,

$$\mu_3 = \frac{\sigma_1^{-2}\mu_1 + \sigma_2^{-2}\mu_2}{\sigma_1^{-2}+\sigma_2^{-2}}$$
$$\sigma_3^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}$$

Similarly, the ratio of two Gaussian PDFs is given by:

$$\frac{\mathcal{N}(x;\mu_1,\sigma_1^2)}{\mathcal{N}(x;\mu_2,\sigma_2^2)} \propto \mathcal{N}(x;\mu_3,\sigma_3^2) \tag{4.2}$$

Where,

$$\mu_3 = \frac{\sigma_2^2\mu_1 - \sigma_1^2\mu_2}{\sigma_2^2-\sigma_1^2}$$
$$\sigma_3^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_2^2-\sigma_1^2}$$

The canonical representation results in more compact expressions:

$$\tau_3 = \tau_1 - \tau_2$$

$$\pi_3 = \pi_1 - \pi_2.$$

Integrating out Gaussian Prior on the Mean of Gaussian R.V.

$$\int_{-\infty}^{\infty} \mathcal{N}(x; m_1, \sigma_1^2)\mathcal{N}(y; \alpha x, \sigma^2)dx \propto \mathcal{N}(y; \alpha m_1; \alpha^2\sigma_1^2 + \sigma_2^2) \tag{4.3}$$

4.2 Appendix B

Number of products = 17;

Number of attributes = 6;

Prior Mean = [1, 1, 1, 1, 1, 1]';

Prior Variance = [20, 20, 20, 20, 20, 20]';

Noise Covariance = $20 * I$, where I is a 17 dimensional identity matrix.

Real Mean = [-3 -2 -1 1 2 -2]';

Number of Purchase occasions = 10;

X= [1 0 1 0 0 2.5;

1 0 0 0 0 1.5;

1 0 0 0 1 1 ;

1 1 1 0 1 2.5;

1 0 1 1 0 3;

1 0 0 1 0 5;

1 1 1 0 0 4.5;

1 1 0 1 1 0.5;

1 0 0 1 0 0.5;

1 1 0 0 1 2.5;

1 0 1 0 1 2.5;

1 0 1 0 0 3;

1 1 0 1 0 3.5;

1 1 1 1 0 1.5;

1 1 0 0 0 1.5;

1 1 1 1 0 2.5;

```
1 1 0 0 0 1.8];
```

4.3 Appendix C

True mean = [10 3 6 2 -1 -1]';

Prior mean = [1 1 1 1 1 1]';

Prior variance = [20 20 20 20 20 20]';

Noise covariance = 4*I, where I is a 9x9 identity matrix.

DiscountLevels = [0 0.25 0.5];

X = [0 0 1 1 0 3;

0 1 0 0 0 3;

1 0 1 1 0 20;

0 0 1 0 1 3;

0 1 0 0 1 1;

0 1 1 0 1 4;

0 0 1 0 1 2;

0 1 0 1 1 4;

0 0 0 1 1 1];

Bibliography

[FCGS10] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári, *Parametric bandits: The generalized linear case*, Advances in Neural Information Processing Systems, 2010, pp. 586–594.

[GCBH10] Thore Graepel, Joaquin Q Candela, Thomas Borchert, and Ralf Herbrich, *Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine*, Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 13–20.

[Git79] John C Gittins, *Bandit processes and dynamic allocation indices*, Journal of the Royal Statistical Society. Series B (Methodological) (1979), 148–177.

[GJ79] J. C. GITTINS and D. M. JONES, *A dynamic allocation index for the discounted multiarmed bandit problem*, Biometrika **66** (1979), no. 3, 561–565.

[HMG06] Ralf Herbrich, Tom Minka, and Thore Graepel, *Trueskillł: A bayesian skill rating system*, Advances in Neural Information Processing Systems, 2006, pp. 569–576.

[KCG12] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier, *On bayesian upper confidence bounds for bandit problems*, International Conference on Artificial Intelligence and Statistics, 2012, pp. 592–600.

[KFL01] Frank R Kschischang, Brendan J Frey, and H-A Loeliger, *Factor graphs and the sum-product algorithm*, Information Theory, IEEE Transactions on **47** (2001), no. 2, 498–519.

[LR85] Tze Leung Lai and Herbert Robbins, *Asymptotically efficient adaptive allocation rules*, Advances in applied mathematics **6** (1985), no. 1, 4–22.

[Min01a] Thomas P. Minka, *Expectation propagation for approximate bayesian inference*, Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (San Francisco, CA, USA), UAI'01, Morgan Kaufmann Publishers Inc., 2001, pp. 362–369.

[Min01b] Thomas P Minka, *A family of algorithms for approximate bayesian inference*, Ph.D. thesis, Massachusetts Institute of Technology, 2001.

[MRT09] Adam J Mersereau, Paat Rusmevichientong, and John N Tsitsiklis, *A structured multiarmed bandit problem and the greedy policy*, IEEE Transactions on Automatic Control **54** (2009), no. 12, 2787–2802.

[RT10] Paat Rusmevichientong and John N Tsitsiklis, *Linearly parameterized bandits*, Mathematics of Operations Research **35** (2010), no. 2, 395–411.

[Sco10] Steven L Scott, *A modern bayesian look at the multi-armed bandit*, Applied Stochastic Models in Business and Industry **26** (2010), no. 6, 639–658.

[Tho33] William R Thompson, *On the likelihood that one unknown probability exceeds another in view of the evidence of two samples*, Biometrika (1933), 285–294.

[Whi88] Peter Whittle, *Restless bandits: Activity allocation in a changing world*, Journal of applied probability (1988), 287–298.