Today, we would like to talk about a recommender system having the functionality of price personalization.
Seventeen years have been passed after the birth of Grouplens, but we think that recommender systems still have many limitations. One of such limitations is that a recommender system is a system only to recommend and cannot behave like clerks in real store. As such an action, we chose Price Personalization, a pricing scheme that allows sellers to adjust the price for an item depending on the customer or transaction.
This is an outline of our talk. We begin with talking about price personalization and its merits. We then show a formalization and an implementation of a personalized pricing recommender system. Finally, we briefly summarize our experimental results, and conclude our talk.
Let’s move on to price personalization and its merits.
Price discrimination is a pricing scheme where different prices are charged for the same item. In cases of traditional price discrimination, prices are changed based on factors such as the sales location or customer demographics. Price personalization is a kind of price discrimination and is more personalized. Sellers can obtain the additional profit by offering a discount only to customers who will not buy at a standard price but who will buy at a discounted price.
Resale is an activity that customers buy items at low prices and then resell them at higher prices. Because resale is an obstacle to implement price discrimination, resale activity must be blocked. In a case of traditional price discrimination, physical factors have been mainly used for blocking. In a case of price personalization, many kinds of approaches have been adopted. In our approach, a system tries to predict whether customers will resell or not.
To show a merit of price personalization for customers, we discuss the commercial viability of managing a recommender system. For this commercial viability, the profit obtained by the increase of customer loyalty must be larger than the cost for managing recommender systems. However, because the effect of loyalty on the profit is indirect and uncertain, the additional profit might be inadequate to compensate for the cost. In such a case, a recommender system falls into its dark-side. A dark recommender system may recommend more expensive items instead of offering lower-cost items that will satisfy customers’ needs.
What can we do for such a dark recommender system? When I thinking about this problem, I here the voice from somewhere “use the personalization.” Additional profit brought by introducing price personalization enhances the commercial viability of a recommender system; thereby Decreasing the sellers’ incentive of making dark recommendation. We further insist that customers have the additional benefit of being offered price discounts.
We then formalize the task of a personalized pricing recommender system.
A personalized pricing recommender system is a recommender system having the functionality of price personalization. We implement the simplest PPRS because this is the first attempt to develop a PPRS. A PPRS is passively invoked for an item that a customer is currently viewing or accessing. There are only two levels of prices: a standard and a discounted. For each target item, a specific customer can be offered a discounted price only when the customer first views the item. This rule blocks the repetition of revisiting until discount prices are offered.
Objective of a Personalized Pricing Recommender System

**maximize the cumulative rewards by iterating the process below**

1) select an item
2) predict a customer type for the pair of the customer and the item
3) determine whether to offer a discounted or a standard price based on the predicted customer type
4) decide whether to buy the item
5) receive a reward based on the customer’s decision and the customer type

An objective of a PPRS task is to maximize the cumulative rewards by iterating this process. A customer selects an item. A PPRS predict customer type and determine whether to offer a discounted or a standard price. A customer decides whether to buy the item. Finally, A PPRS receives a reward based on the customer’s decision and the customer type. We then sequentially show a customer type and a reward.
Customer Type

There are three customer types.

**Standard:** Customers who will buy an item regardless of whether the price is standard or discounted

⭐ A standard price should be offered to obtain more profit

**Discount:** Price-sensitive customers who will buy an item only if a discounted price is offered.

⭐ A discounted price should be offered so that a customer to buy an item

**Indifferent:** Customers who will not buy an item whether or not it is discounted

⭐ A standard price should be offered to block the customers to resale, because these customers will not consume the item for oneself

There are three customer types. Standard customers will buy an item at a standard price. Discount customers will buy if a discount is offered. Indifferent customers will not intended to buy. Standard and discounted prices should be offered to a standard and a discount customers, respectively. For indifferent customers, a standard price should be offered to block the customers to resale, because the customers will not consume the item for oneself.
This is a table of rewards. If a standard customer and a discount customer buy an item, a system receives rewards, $\alpha$ and $\beta$, respectively. These correspond to profits gained by selling items. In a case of an indifferent customer, a system receives a reward $\gamma$, if the customer doesn’t buy the item. This $\gamma$ corresponds to a potential profit by blocking resale.
We then show our implementation of a PPRS.
This is the inputs and outputs of the prediction model for the customer type.

A customer–item pair to predict its customer type is specified. Three types of data sets are used for prediction: preference DB, customer data, purchasing history. Preference DB is used for building a recommendation model, such as pLSA or matrix decomposition. Then, this model parameters and customer data are combined into features. Purchasing histories are used as target values. From these features and target values, classification model can be learned by a standard classification algorithm, such as a logistic regression.
However, there are three technical problems in the prediction of customer types: ambiguity in observation, exploitation–exploration trade–off, and class imbalance problem. We sequentially show these problems.
The ambiguity in observation is the impossibility to detect true customer type by observing customers’ behavior. A true customer type is unknown to a PPRS and must be guessed from the customers’ responses. When a system offers a discount price, indifferent customers do not buy an item; thus, a system can perceive the customer is an indifferent type. However, both standard and discount customers buy an item; thus, a system cannot differentiate these two types.
To solve the problem of the ambiguity in classification, we take a multi-stage classification approach. There are two types of classifiers: a standard classifier and a discount classifier.

A standard classifier and a discount classifier are learned from the customers’ responses to offers at a standard price and at a discount price, respectively. A customer-item pair is firstly classified into a standard type or a non-standard type. If it is classified as a non-standard type, it is further classified into a discount type or an indifferent type.

In our experiment, a prescreening stage is added.
We show the second technical problem: exploitation–exploration trade-off. A PPRS collects purchasing histories while predicting customer type. Because current prediction of customer types might be incorrect, a system must sometimes take non-best actions to collect data. On the other hand, because too frequent non-best actions will reduce the total rewards, a system should fundamentally take the best action to earn rewards. A system must take into account the balance between these two actions.
A Multi-armed bandit problem treats the adjustment of exploitation-exploration trade-offs.

We adopted the most naive approach, $\varepsilon$-Greedy. A system selects exploitation actions with the probability $1-\varepsilon$ and selects exploration actions with the probability $\varepsilon$.

Consider the case that the prediction of a standard classifier is a standard customer.

If selecting exploitation, the customer is treated as a standard type as predicted and a PPRS offers a standard price.

If selecting exploration, the input pair is passed to a discount classifier.
A class imbalance problem is the decline in accuracy when the class distribution is highly skewed. In a case of a standard classifier, many standard customers wrongly classified as non-standard ones, because standard customers are much fewer than non-standard ones. We adopted a class weighting approach to alleviate this problem. This is simply to adjust a decision threshold accordingly.
Finally, we briefly summarize our experimental results.
Experimental Condition

Quasi-synthetic data from MovieLens’ 1M dataset

- Preference data and customers’ demographics are imported from Movielens dataset
- Customers’ purchasing histories are artificially generated so that satisfy the following conditions:
  1. Preference for the target items would become stronger in the order of a standard customer, a discount customer, and an indifferent customer
  2. The determination of purchasing activities was assumed to depend on the customers’ preference for the target items and their demographics
  3. Almost all customers are indifferent, and the number of discount customers is slightly larger than that of standard customers

Though this purchasing history is simple, it is not trivial for a system to be able to obtain additional reward because of three technical problems.

Please refer to our manuscripts about the details of conditions.

We tested our PPRS on quasi–synthetic data from MovieLens’ 1M dataset. Preference data and customers’ demographics are imported from Movielens dataset. However, customers’ purchasing histories are artificially generated so that satisfy the following conditions. We’d like insist that though this purchasing history is simple, it is not trivial for a system to be able to obtain additional reward because of three technical problems.
Main Experimental Results

- Our PPRS could successfully obtain additional rewards by adopting a personalized pricing scheme.
- We could adjust parameters by observing customers’ behaviors, even though true customer types could not be observed.
- Higher-weighting on standard customers than non-standards in a standard classifier, because it is important not to miss loyal customers.
- Lower-weighing on discount customers than indifferent ones in a discount classifier, because discounts should be offered for customers who are certainly discount types.
- Exploration probability $\varepsilon$ heavily affects the total rewards.

This is a brief summary of our experimental results. Our PPRS could successfully obtain additional rewards by adopting a personalized pricing scheme. We could adjust parameters by observing customers’ behaviors, even though true customer types could not be observed. Regarding parameters, we observed these results.
Conclusion

Contributions

- We added the function to take actions other than recommendation, i.e., price personalization, to a RS
- We discussed how it improves the commercial viability of managing a RS, and thereby improving the reliability to a RS
- We implemented a simple system and tested on a quasi-synthetic data set

Future Work

- If utilities other than prices can be considered as rewards, a framework of a PPRS could be made applicable to broader actions
- Recommender systems have started to provide not only simple recommendations but also more sophisticated actions; such evolved systems could be called “Attendant Systems.”

These are our contributions.
If utilities other than prices can be considered as rewards, a framework of a PPRS could be made applicable to broader actions. Recommender systems have started to provide not only simple recommendations but also more sophisticated actions; such evolved systems could be called “Attendant Systems.”
May the Personalization Be with You
May Not Be with Your Adversaries

Errata: reference [9] should be
R. Kleinberg and T. Leighton. The value of knowing a demand curve:
Bounds on regret for online posted-price auctions. In Proc. of the 44th

That’s all I have to say. Thank you for your attention.