Using Multi-armed Bandit to Solve Cold-start Problems in Recommender Systems at Telco

Hai Thanh Nguyen[1] and Anders Kofod-Petersen[1,2]

[1] Telenor Research
7052 Trondheim, Norway
{HaiThanh.Nguyen|Anders.Kofod-Petersen}@telenor.com
[2] Department of Computer and Information Science (IDI),
Norwegian University of Science and Technology (NTNU),
7491 Trondheim, Norway
anderpe@idi.ntnu.no

Abstract. Recommending best-fit rate-plans for new users is a challenge for the Telco industry. Rate-plans differ from most traditional products in the way that a user normally only have one product at any given time. This, combined with no background knowledge on new users hinders traditional recommender systems. Many Telcos today use either trivial approaches, such as picking a random plan or the most common plan in use. The work presented here shows that these methods perform poorly. We propose a new approach based on the multi-armed bandit algorithms to automatically recommend rate-plans for new users. An experiment is conducted on two different real-world datasets from two brands of a major international Telco operator showing promising results.

Key words: multi-armed bandit, cold-start, recommender systems, telecom, and rate-plan.

1 Introduction

The Telco industry do not at first glance appear to be of particular interest from a recommender system perspective. Telcos do not commonly supply a lot of services; most general they supply subscriptions, or rate-plans; either pre-paid or post-paid. However, recommending the optimal rate-plans for users in general, and new users in particular can be challenging.

Suggesting a rate-plan for a new user is a typical *cold-start user problem* (following the separation suggested by Park et al., [1]). This problem has also been identified under slightly different names, such as: the *new user problem* [2], the *cold start problem* [3] or *new-user ramp-up problem* [4]. However, the fact that a customer traditionally only has one rate-plan at any given time increases the difficulty of this problem. Comparing this to a more traditional recommender problem where a user-item matrix might be sparse; in this example the matrix will be completely sparse.

To solve this cold-start problem, given the fact that no prior information on the new user exists, one might think of a random recommendation of rate-plans.

However, the chance that the recommended plan be accepted by the new user is small. In fact, given n available rate-plans the probability that a random pick-up plan is accepted is only $1/n$. We say this approach has too much randomness in its recommendations.

Another possibility for solving this problem is to use the distribution of selected plans from existing users. Concretely, it is sensible to recommend the most popular rate-plan to the new user. By doing this, we assume that there is a fixed distribution behind the choice of rate-plans by the new users. However, in reality and also in the experiment below, we can observe that this is not the case. We say this method exploits too much the most popular rate-plan.

The idea now is to have a better solution to control the randomness in the exploration of different rate-plans while keeping the exploitation of the most popular rate-plan at a time. This is the usual dilemma between *Exploitation* (of already available knowledge) versus *Exploration* (of uncertainty), encountered in sequential decision making under uncertainty problems. This has been studied for decades in the multi-armed bandit framework. The work presented here, attempts to tackle the cold-start user problem by recommending a plan that will appeal to the user in question, rather than *the best* plan. We approach this by applying the multi-armed bandit algorithms.

The multi-armed bandit (MAB) is a classical problem in decision theory [5,6,7]. It models a machine with K arms, each of which has a different and unknown distribution of rewards. The goal of the player is to repeatedly pull the arms to maximise the expected total reward. However, since the player does not know the distribution of rewards, he needs to explore different arms and at the same time exploit the current optimal arm (i.e. the arm with the current highest cumulative reward).

To evaluate our MAB approach in solving the cold-start user problem at Telco, we conduct experiments on real-world datasets and compare it with trivial approaches, which include the random and most popular method. Experimental results show that our proposed approach improves upon the trivial ones.

The paper is organised as follows: Section 2 gives an overview of related works; Section 3 provides a formal definition of the cold-start problem in the rate-plan recommender system at Telco. We describe our proposed approaches in Section 4. Section 5 presents some experimental results and discussions. The paper ends with a summary of our findings and a discussion on future work.

2 Related Work

Unfortunately, there are very few examples of research regarding rate-plan recommender systems for Telco, in particular with respect to the cold-start problem. Examples include, Thomas et al., who describe how to recommend best-fit recharges for pre-paid users [8]. Soonsiripanichkul et al., employes a naïve Bayes classifier to infer which rate-plan to suggest to existing users [9]. Both use existing data on customers' usage patterns and do not address the cold-start problem.

In general, one common strategy for mitigating the cold-start user problem is to gather demographic data. It is assumed that users who share a common background also share a common taste in products. Examples include Lekakos and Giaglis [10], where lifestyle information is employed. This includes age, marital status and education, as well as preferences on eight television genres. The authors report that this approach is the most effective way of dealing with the cold-start user problem in sparse environments.

A similar thought underlies the work by Lam et al., [11] where an *aspect model* (see e.g. [12]) including age, gender and job is used. This information is used to calculate a probability model that classifies users into user groups and the probability how well liked an item is by this user group.

Other examples of applying demographic information for mitigating the cold-start user problem exists, e.g. [13,14,15]. All the solutions above use similar demographic information; most commonly age, occupation and gender. Most of the solutions ask for less than five pieces of information. Even though five is a comparatively small number, the user must still answer these questions. Users do generally not like to answer a lot of questions, yet expect reasonable performance from the first interaction with the system [16].

Zigoris and Zhang [16], suggests to use a two part Bayesian model, where the prior probability is based on the existing user population and *data likelihood*, which is based on the data supplied by the user. Thus, when a new user enters the system, little is know about that user and the prior distribution is the main contributor. As the user interacts with the system the data data likelihood becomes more and more important. This approach performs well for cold-start users. Other similar approaches can by found in [17], suggesting a Markov mixture model, and [18] who suggests a statical user language model that integrates an individual model, a group model and a global model.

Our study differs from previous research on the cold-start problem, as no demographic information is taken into account. Only the information on selected plans of previous users is available to the recommender engine. This assumption makes the cold-start problems even harder to solve. However, we leave the issue of collecting more information from users and how to use it for cold-start recommender systems for future works.

3 Problem Definition

Recommending a rate-plan for a new mobile telephony user differs from traditional recommender systems. Traditionally, recommender systems are in a context where users can purchase and own several products, such as books; Rate-plans are different in the sense that one user can have any number of rate-plans, but typically only one plan at any given time. Further, the user will typically have the same product for an extended time period. Finally, no explicit rating for the rate-plans exist. We call this problem the *Cold Start Alternative Recommendation* (CSAR) problem and below is its formal definition.

Let $U = \{u_1, \ldots, u_T\}$ be the set of T new users. Assume that we have a set P of n rate-plans to recommend to a new user: $P = \{p_1, \ldots, p_n\}$ where each plan $p_i (i = 1...n)$ is described by m features $\{f_1, \ldots, f_m\}$, such as price, number of included SMS, number of voice minutes included and so on. Among k $(k \geq 1)$ suggested rate-plans, the new user can only select one plan at any given time.

Assume that at a given time t a new user u_t comes and the system recommends a rate-plan p_t without any knowledge on the new user. Depending on the user's needs, she will accept the offer or select another rate-plan. We want to design an algorithm that can find a best-fit rate-plan for the new user. Let $need_t$ be a vector described the user's demand: $need_t = (need_{t1}, \ldots, need_{tm})$, where each feature $need_{tj}$ corresponds to each feature f_j of the rate-plans. If we denote the similarity value between the recommended plan p_t and the actual demand of the new user u_t by a $similarity(need_t, p_t)$, then the objective when solving the CSAR problem is to select the rate-plans p_t that maximizes the following so called "cumulative reward" ($Reward$) over all T new users:

$$Reward_T = \sum_{t=1}^{T} (similarity(need_t, p_t))$$

The CSAR problem would be easy to solve if we knew about the user's needs $need_t$. The task then becomes straightforward by selecting the rate-plan that provides the maximal value of the $similarity(need_t, p_t)$ over all available plans.

As mentioned, it is not possible to calculate $similarity(need_t, p_t)$ since $need_t$ is not available. We suggest to study an approximated problem to the CSAR problem where we consider the similarity value between the recommended plan p_t and the actual selection of the new user p_t^*. By doing this, we wish to achieve a recommendations as close as possible to the actual choice made by the user. The actual choice is also considered as her temporary best-fit plan. Formally, we want to maximize the following so called "reward":

$$Reward_T = \sum_{t=1}^{T} (similarity(p_t, p_t^*))$$

There are many ways to define the similarity value between two vectors p_t and p_t^*. Below, we suggest to take into account the two most popular measurements which are i) *the indicator function* and ii) *the correlation value.*

Indicator function If we use the indicator function as the similarity measurement, then the problem becomes to design an algorithm that predict the rate-plan p_t^* chosen by the new user. The cummulative reward now is the following: $Reward_T^{(1)} = \sum_{t=1}^{T} (\mathbb{I}(p_t^* \neq p_t))$, where $\mathbb{I}(p_t^* \neq p_t)$ is an indicator function which is equal to 0 if $p_t^* \neq p_t$ and to 1, otherwise. To evaluate any algorithm solving this problem, we can use the classical precision measurement: $Precision_T = \frac{1}{T} Reward_T^{(1)}$

Correlation value In the second case, we study how similar is the recommended rate-plan p_t to the actual selection p_t^* of the new user in terms of the features. Generally, when a new user purchases a rate-plan, she looks at the features describing the different rate-plans including the recommended rate-plan p_t. Finally she picks up a plan p_t^* that we can assume is perceived as the temporary best-fit for her. Therefore, it is sensible to choose the correlation coefficient as a similarity measurement between plans and the task is to maximizes the following so called "cumulative reward": $Reward_T^{(2)} = \sum_{t=1}^{T}(Corr(p_t^*, p_t))$, where $Corr(p_t^*, p_t)$ is the correlation value between two vectors p_t^* and p_t.

Possible correlation values can be Pearson correlation or Kendall correlation. Since the actual demand of new users is not available at the time when they enter, it is fair to treat all the features equally in the correlation calculation. While solving this problem, we try to recommend a rate-plan that is sufficiently good in terms the features and that the user will accept. Thus, classic precision measurements are not applicable. We, therefore, define the Average-Feature Prediction (AFP) as a new evaluation measurement of how much of the features of the rate-plan chosen by T new users are predictable on average:

$$AFP_T = \frac{1}{T} Reward_T^{(2)}$$

4 Bandit Algorithms for the CSAR Problem

Based on the idea of the multi-armed bandit [5,6,7], in the following we translate the new CSAR problem into a bandit problem.

Let us consider a set P of n available rate-plans to recommend to T completely new users. Each plan is associated with an unknown distribution of being selected by users. The game of the recommender system is to repeatedly pick up one of the rate-plans and suggest to a new user whenever she enters the system. The ultimate goal is to maximize the cumulative reward. As defined in previous section, the reward for our recommender system is the similarity value $similarity(p_t, p_t^*)$. Note that the setting in present context is slightly different from traditional MABs. In a traditional MAB only the reward of the selected arm is revealed. In our case all the non-selected arms also get rewards after the recommendation is made. In fact, in the case of using the *indicator function*, then the non-selected rate-plans by users will get a zero reward. In the case of using the *correlation value*, the rewards of the non-selected rate-plans will be the correlation value between the two vectors p and p^*. However, since the distributions of the rate-plans being selected are still unknown, the idea of using the MAB algorithms for the CSAR problem is still valid. The following three MAB algorithms are being used:

ϵ-greedy [7] aims at picking up the rate-plan that is currently considered the best (i.e. the rate-plan that has the maximal average reward) with probability ϵ (exploit current knowledge), and pick it up uniformly at random with probability $1 - \epsilon$ (explore to improve knowledge). Typically, ϵ is varied along time so that the plans get greedier and greedier as knowledge is gathered.

UCB [7] consists of selecting the rate-plan that maximises the following function: $UCB_{tj} = \hat{\mu}_j + \sqrt{\frac{2\ln t}{t_j}}$ where t is the current time-step, μ_j is the average reward obtained when selecting plan j, t_j is the number of times the plan j has been selected so far. In this equation, $\hat{\mu}_j$ favours a greedy selection (exploitation) while the second term $\sqrt{\frac{2\ln t}{t_j}}$ favours exploration driven by uncertainty; it is a confidence interval on the true value of the expectation of reward for plan j.

EXP3 [19] selects a rate-plan according to a distribution, which is a mixture of the uniform distribution and a distribution that assigns each plan a probability mass exponential in the estimated cumulative rewards for that plan.

5 Experiments and Results

This section details the datasets used in the experiments; the experimental settings, in which the detail implementation of the proposed methods and of the competing algorithms are provided; and contains an analysis and discussion of the experimental results.

5.1 Dataset

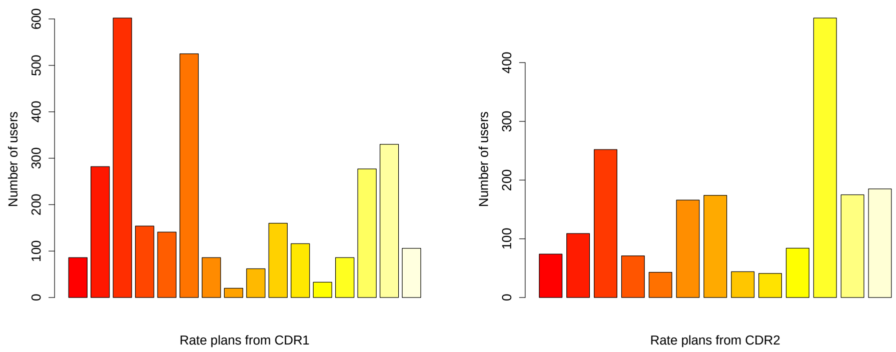

Fig. 1. Distributions of number of users with different rate-plans.

We use two different real-world client datasets from two brands of a major international Telco operator. These two datasets were collected during the first quarter of 2013. The first brand's dataset contains the descriptive features of 16 rate-plans, as well as information about the plans used by 3066 users. The second dataset contains the descriptive features of 13 rate-plans, as well as information about the plans used by 1894 users. In this work we have assumed that users have not picked their rate-plan at random – that is, they have each chosen a rate-plan that fits their need. Figure 1 shows the distributions of rate-plans.

Table 1. Features describing the rate-plans

Rate-plan Features	**Description**
Price per month	The fixed price of rate-plan per month (including 0)
Voice (minutes)	Number of voice minutes included per month (including 0)
Text (SMS)	Number of (SMS) messages included per month (including 0)
MMS	Number of MMS messages included per month (including 0)
MB	Number of MB included per month (including 0)
Voice (post cap, start)	Starting price per voice call after included
Voice (post cap, min)	Price of voice per minutes after included
SMS (post cap)	Price per SMS after included
MMS (post cap)	Price per MMS after incuded
Data (post cap, MB)	Price per MB after included
Speed (Mbit/s)	Speed of Internet allowed

Each of the used rate-plans are described by the 11 most important features, which are shown in Table 1. It is worth noticing that rate-plans broadly fall into three categories: *i*) pre-paid, where the customer pays a certain amount and receives a certain number of services that must be consumed within a certain time frame (e.g. 100 minutes, 100 SMS, 100 MB valid for 30 days); *ii*) traditional post-paid, where the customer pays a certain amount per month and pays for consumption; and *iii*) post-paid flat-rate, where all voice, SMS and MMS is included and the customer pays a certain amount depending on how much data is available (e.g. 100 voice minutes, 100 SMS/MMS and 1 GB of data per month).

5.2 Experimental Settings

Trivial approaches The first and the most naïve approach for the cold-start recommendation systems at Telcos is to choose randomly a rate-plan to recommend to a new user. This algorithm is very efficient, especially, when we do not have any description on users and the algorithm seems to be reasonable.

The second trivial approach is to recommend the most popular rate-plan (Most common) to the new user. This is a sensible approach in terms of the efficiency and many operators apply this.

The third trivial approach is to pick up the best-average-reward rate-plan (Best average) at a time (i.e. the current rate-plan that has the maximal average reward value) to recommend to a new user. In this case, we choose the Pearson correlation as a similarity measurement for the reward $similarity(p_t, p_t^*)$ value.

Multi-armed bandit algorithms ϵ-greedy estimates the average reward of each rate-plan. It then selects a random plan with probability ϵ_t, and choose plan of the highest average value of rewards with probability $1 - \epsilon_t$. The parameter ϵ_t is decreased over time t. In fact, the ϵ_t is calculated as follows: $\epsilon_t = min(1, (cn)/(d^2(t - n - 1)))$, where n is the number of rate-plans; c and d are chosen constants. In our experiment, we selected $c = 0.001$ and $d = 0.01$, which provided the best results.

The UCB algorithm estimates the value UCB_{tj} for each plan. It then choose the plan with the highest UCB_{tj} value to recommend to the new user.

Finally, EXP3 selects a plan according to a give distribution, as described in [19]. We select $\gamma = 0.01$ before drawing the probability to select the best plan to recommend to the new user.

Each of the six algorithms are run five times with different choices of parameters. The best results recorded are shown in Table 2.

5.3 Results and Analysis

Table 2 shows the performances of the six different approaches for the cold-start problem on the two different real-world client datasets DS1 and DS2. We present the precision result $Precision$ that indicates the accuracy of our recommendation and the prediction value AFP, which is the closeness of our recommended plan to the actual selected one in terms of the features.

Table 2. Precision and AFP for the two datasets

Method	**Precision**$_{DS1}$	**AFP**$_{DS1}$	**Precision**$_{DS2}$	**AFP**$_{DS2}$
Random	6.80	44.50	7.98	47.60
Most common	20.29	52.70	25.68	70.20
Best average	20.29	54.00	25.68	70.70
ϵ-greedy	20.28	54.00	25.60	70.07
EXP3	10.48	46.80	12.28	61.50
UCB	**43.04**	**69.20**	**45.08**	**75.30**

It can be seen from the table that the random approach provided very poor results in both datasets. In fact, it has only 6.80 percent precision and 44.50 percent prediction in the case of the DS1 dataset. This can be explained by the fact that the probability when a randomly recommended rate-plan being accepts is only $\frac{1}{16} = 6.25\%$. The is also true for the second dataset, where a randomly recommended rate-plan only has a $\frac{1}{13} = 7.67\%$ probability of being correct.

The most common (Most common) and the best average (Best average) performed better than the random one. Yet, the results are still not good. To explain this, we look at Figure 1. Clearly, both the DS1 and DS2 datasets have the most common rate-plans which has the maximal number of being selected by users. Beside, users also chose a variety of other different rate-plans. So following the most common, or the best average rate-plan would not be a good strategy.

The ϵ-greedy approach provides almost the same results as the most common and the best average approaches. The reason is that it was too greedy when setting up the ϵ_t to a too small value. This forces the ϵ-greedy algorithm to follow the best rate-plan (i.e. the rate-plan has the maximal average reward value) all the time.

The case of EXP3 shows even worse performance than the ϵ-greedy. This is probably because of a wrong assumption on the distribution of the selected rate-plans, which is a mixture of the uniform distribution and a distribution that assigns to each plan a probability mass exponential. This exemplifies the fact of being careful when selecting appropriate strategies for the MAB.

The UCB gave us a surprisingly good precision and prediction results. In fact, it increased the precision of the random approach to 39 percent and could predict more than 75 percent of the features of actual selected rate-plan by new users. The reason is that the UCB approach has a good strategy in balancing the exploitation of the best rate-plan at a time and the exploration of other different rate-plans which are also interest for the new users. To have a better explanation, by looking at the UCB algorithm as described in previous section, we see that the recommendation of a rate-plan is a result of solving the trade-off between the average reward and the number of times the plan has been selected so far by users. Therefore, beside the current best rate-plan, other good ones have a chance to be recommended, as well as the other rate-plans that already have been selected a few times. This UCB strategy resulted the distribution of recommended rate-plans closer to the real distribution, as shown in Figure 1, compared to most common and random approaches.

6 Conclusions and Future Research

This work approaches recommending rate-plans to completely new users at Telco, without any prior information on them. An experiment was conducted on two different real-world client datasets from two brands of a major international Telco operator. From the experimental results, we observed that the UCB algorithm clearly outperforms traditional naïve approaches, as well as other classical multi-arm bandit algorithms. This is still work in progress, and as such many issues still needs to be tackled. Improving the precision and AFP would still be preferable. Demographical information is likely to be required to improve this.

Acknowledegements

The authors would like to extend our gratitude to Professor Helge Langseth at the Department of Computer and Information Science, at the Norwegian University of Science and Technology (NTNU), and Dr. Humberto N. Castejón Martínez and Dr. Kenth Engø-Monsen at Telenor Research; without whom this work would not have been possible.

References

1. Park, S.T., Pennock, D., Madani, O., Good, N., DeCoste, D.: Naïve filterbots for robust cold-start recommendations. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining,2006. 699–705
2. Adomavicius, G., Tuzhilin, A.: Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering **17**(6) (2005) 734–749
3. Massa, P., Bhattacharjee, B.: Using trust in recommender systems: An experimental analysis. In Jensen, C.D., Poslad, S., Dimitrakos, T., eds.: Trust Management, Second International Conference, iTrust 2004. (2004) 221–235
4. Burke, R.: Hybrid recommender systems: Survey and experiments. user modeling and user-adapted interaction. User Modeling and User-Adapted Interaction **12**(4) (November 2002) 331–370
5. Lai, T.L., Robbins, H.: Asymptotically efficient adaptive allocation rules. Advances in Applied Mathematics **6**(1) (1985) 4–22
6. Katehakis, M., Veinott, J.A.: The multi-armed bandit problem: decomposition and computation. Mathematics of Operations Research **12**(2) (February 1987) 262–268
7. Auer, P., Cesa-Bainchi, M., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine Learning **47**(2-3) (May-June 2002) 235–256
8. Thomas, S., Wilson, J., Chaudhury, S.: Best-fit mobile recharge pack recommendation. In: National Conference on Communications (NCC). (Feb 2013) 1–5
9. Boonyarit Soonsiripanichkul, Nattapong Tongtep, T.T.: Mobile package recommendation using classification with feature discretization and threshold-based ensemble technique. Proceedings of the International Conference on Information and Communication Technology for Embedded Systems (ICICTES2014) (2014)
10. Lekakos, G., Giaglis, G.M.: A hybrid approach for improving predictive accuracy of collaborative filtering algorithms. User Modeling and User-Adapted Interaction **17**(1-2) (2007) 5–40
11. Lam, X.N., Vu, T., Le, T.D., Duong, A.D.: Addressing cold-start problem in recommendation systems. In: Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, ACM (2008) 208–211
12. Marlin, B.: Collaborative filtering: A machine learning perspective. Technical report, University of Toronto (2004)
13. Gao, F., Xing, C., Du, X., Wang, S.: Personalized service system based on hybrid filtering for digital library. Tsinghua Science & Technology **12**(1) (2007) 1–8
14. Agarwal, D., Chen, B.C.: Regression-based latent factor models. In: KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM SIGKDD, ACM (June 28 - July 1 2009) 19–28
15. Park, S.T., Chu, W.: Pairwise preference regression for cold-start recommendation. In: RecSys'09: Proceedings of ACM conference on Recommender systems. 21–28
16. Zigoris, P., Zhang, Y.: Bayesian adaptive user profiling with explicit & implicit feedback. In: CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management, ACM (2006) 397–404
17. Manavoglu, E., Pavlov, D., Giles, C.L.: Probabilistic user behavior models. In: ICDM '03: Proceedings of Third IEEE International Conference on Data Mining
18. Xue, G.R., Han, J., Yu, Y., Yang, Q.: User language model for collaborative personalized search. ACM Transactions on Information Systems **27** (2009) 1–28
19. Auer, P., Cesa-Bianchi, N., Freund, Y., Schapire, R.E.: The nonstochastic multi-armed bandit problem. SIAM Journal on Computing **32**(1) (2002) 48–77