Can Machine Learning Techniques Provide Better Learning Support for Elderly People?

Kohei Hatano(✉)

Kyushu University, Fukuoka, Japan
hatano@artsci.kyushu-u.ac.jp

Abstract. Computer-based support for learning of elderly people is now considered as an important issue in the super-aged society. Extra cares are needed for elderly people's learning compared to younger people, since they might have difficulty in using computers, reduced cognitive ability and other physical problems which make them less motivated. Key components of a better learning support system are sensing the contexts surrounding elderly people and providing appropriate feedbacks to them. In this paper, we review some existing techniques of the contextual bandit framework in the machine learning literature, which could be potentially useful for online decision making scenarios given contexts. We also discuss issues and challenges to apply the framework.

Keywords: Elderly people · Machine learning · Learning support
Contextual bandit

1 Introduction

Many countries now face the super-aged society due to the developments of health cares, medicines, and so on. Elderly people have more opportunities to work or to play important roles in the communities. Infrastructures for life-long learning of people will be essential components to support and promote such activities in the super-aged society. In particular, for elderly people, extra cares are needed. For example, they might have difficulty using computers, reduced cognitive ability, and reduced memory capacity, which prevent them to keep their motivation for learning. Therefore, the system needs to be more aware of the context surrounding them and should take actions adaptively to support and motivate their learning.

Although there already exist various systems for support learning, the systems are designed mainly for younger or middle-aged people, say, in the classrooms or in the offices. On the other hand, elderly people could learn in more diverse contexts, e.g., in different physical and mental conditions. So, the systems need more adaptivity to the contexts and more options and choices to support elderly people. Adapting elderly people's situations is a non-trivial task and it

N. Streitz and S. Konomi (Eds.): DAPI 2018, LNCS 10922, pp. 178–187, 2018.
https://doi.org/10.1007/978-3-319-91131-1_14

would be hard to specify the behaviors of the system manually for each context. For this non-trivial task, machine learning techniques are useful in general. They can learn to optimize the system from the data and feedbacks, under an appropriate design of learning framework.

In this paper, we survey some of recent machine learning techniques to seek candidates to support learning by elderly people in the super-aged society. To aware and adapt contexts, the system need to learn better behaviors in an online fashion rather than offline, where the term online means that the system modifies behaviors during the course of interactions with users. Among such online machine learning techniques, those of the contextual bandit techniques could fit the demand of the adaptivity. We will explain the mathematical formulation of the contextual bandit problems and some standard algorithms as well as evaluation methods for online bandit techniques and applications such as online news recommendation tasks. Then, we will discuss several issues to apply the contextual bandit framework to support elderly people's learning. In particular, we highlight differences between typical applications of the contextual bandit framework and learning supports for elderly people. Finally, we will discuss future research directions.

2 Related Work

To the best of our knowledge, we are not aware of existing researches on computer-based support for learning of elderly people. However, in the ubiquitous or mobile learning research areas, there are a lot of related results concerning contexts surrounding users. Among them, Ogata and Yano proposed context-aware systems for learning Japanese polite expressions [16], where the contexts include gender, work, age, places, social status of the user, the conversational partner, and relationship between them. Syvänen et al. proposed a mobile learning system which adaptively changes the user interface based on the context of users such as devices like PCs, PDAs and so on [22]. Researchers pay attention how contexts affect computer-based learning. Economides [10] defines various features or contexts that characterize learning in pervasive and ubiquitous learning environments. Hood et al. examined how contexts affect self-regulated learning behaviors in a MOOC environment [12]. As for elderly people's learning, Pachman and Ke investigated designs of multimedia training which help elderly people to learn better with additional representational supports [17]. In the literature of recommendation systems, Adomavicius surveyed various approaches in context-aware recommendations systems [1].

3 Contextual Bandit: Formulations and Results

In this section, we will review contextual bandit techniques in the machine learning and related literature. For a detailed survey, see, e.g., the work of Bubeck and Cesa-Bianchi [6].

3.1 Formulations

The typical contextual bandit framework is described as a game between the player and the environment. The player has a fixed set of actions $\mathcal{A} = \{1, \ldots, K\}$[1]. The protocol of the game is given as follows for each $t = 1, \ldots, T$.

1. The environment gives the player a *context* $\boldsymbol{x}_t \in X \subset \mathbb{R}^n$, which is a feature vector containing the contextual information of the trial.
2. The player chooses an action $\boldsymbol{a}_t \in \mathcal{A}$.
3. The environment assigns a reward $r_a \in [0, 1]$ for each action a in $\mathcal{A}$[2]. Equivalently, the environment assigns a vector $\boldsymbol{r} \in [0, 1]^K$. Note that the reward vector is not revealed to the player.
4. The player gets a reward r_{a_t}. Here, the player only knows the reward of the chosen action and does not know the rewards of other actions.

Assumption on the environment. In the typical contextual bandit setting, the environment is assumed to be *stochastic*. More precisely, we assume that a fixed and possibly unknown distribution D over $X \times [0, 1]^K$ and each context $(\boldsymbol{x}_t, \boldsymbol{r}_t)$ is drawn independently and randomly according to the distribution D (i.i.d. assumption).

Goal: Let $\mathit{\Pi} \subseteq \{\pi : X \to \mathcal{A}\}$ is a fixed set of functions corresponding to candidates of the player's strategies. The goal of the player is to minimize the regret:

$$\mathit{regret} = \max_{\pi^* \in \mathit{\Pi}} E\left[\sum_{t=1}^{T} r_{t,\pi^*(\boldsymbol{x}_t)}\right] - E\left[\sum_{t=1} r_{t,a_t}\right],$$

where the expectation is taken w.r.t. the distribution D and the randomness of the player.

3.2 Context-Free Bandits

In the special cases where no context is given, the problem is called the multi-armed bandit problem and studied extensively. In this simpler setting, at each trial t, the reward $r_{t,a}$ of each action $a \in \mathcal{A} = \{1, \ldots, K\}$ is drawn randomly. More precisely, $\boldsymbol{r}$ is drawn according to the distribution D over $\mathcal{A}$. Let $\mu_a = E_{\boldsymbol{r} \sim D}[r_a]$ for each $a \in \mathcal{A}$. Then the regret is given as $\max_{a^* \in \mathcal{A}} \mu_{a^*} T - \sum_{t=1}^{T} \mu_{a_t}$. Let a^* be the best action, i.e., $\mu_{a^*} = \max_{a \in \mathcal{A}} \mu_a$. Then the goal is now to pick up actions as whose expected reward is close to μ_{a^*} as many as possible.

The main issue in the (context-free) bandit problem is how to manage the trade-off between exploration and exploitation in limited trials. Here, if we explore various actions many time, we will get more accurate information on rewards of actions, but we might lose the opportunity to exploit actions which turned out to be good. On the other hand, if we exploit actions which seems to

[1] In general, the set of actions might vary in time (see, e.g., [14]).
[2] For simplicity, the reward is normalized in $[0, 1]$.

be good in a small amount of trials, we might miss chances to find much better actions.

Let us begin with a simple strategy by exploring first and then exploiting. A basic tool in the probability theory is the Hoeffding bound (see, e.g., [9]). Suppose that we pick up an action a for m times and $\hat{\mu}_a$ is the empirical mean of the m rewards. Then, by the Hoeffding bound, we have $\Pr\{|\hat{\mu}_a - \mu_a| \leq \varepsilon\} \leq 2e^{-2\varepsilon^2 m}$. Let $\Delta = \min_{a \in \mathcal{A}}(\mu_{a^*} - \mu_a)$. So, if we pick up each action for $4/\Delta^2$ times, it can be easily shown that (together with the union bound), with probability at least $1 - 0.04K$, the action with the highest empirical mean reward is indeed the best action. More generally, we could consider the following strategy: (i) pick up each action for $\varepsilon T/K$ times, and (ii) choose the action with the best empirical mean reward for the rest of $(1 - \varepsilon)T$ trials. This strategy is called *the ε-greedy* strategy [21]. It is known that, by setting $\varepsilon = (2K \ln T)/\Delta T$, the regret of the ε-greedy strategy is $O((K/\Delta) \ln T)$, which is close to optimal. A disadvantage of this strategy is that it needs to know the parameter Δ, which is not known in advance. Yet, the ε-greedy strategy is simple and thus easy to implement and it often shows competitive performances in practice.

Algorithm 1. UCB (bandit algorithm without context information)[4]

1. Choose each action $a \in \mathcal{A}$ once and let $m_a = 1$ for each $a \in \mathcal{A}$.
2. For each trial $t = 1, \ldots, T$:
   (a) Choose the action
   $$a_t = \arg\max_{a \in \mathcal{A}} \hat{\mu}_a + \sqrt{\frac{\ln t}{2m_a}},$$
   where $\hat{\mu}_a$ is the mean reward of action a.
   (b) Update $m_a = m_a + 1$ and the empirical mean reward $\hat{\mu}_{a_t}$ of a_t.

The UCB (upper confidence bound) [4] is another widely known strategy. The UCB strategy maintains a confidence interval for the reward of each action. More specifically, given m_a observations of rewards of the action $a \in \mathcal{A}$, by using the Hoeffding inequality, it holds that with probability $1 - O(1/t)$, the expectation μ_a of the reward of a is at most

$$\hat{\mu}_a + \sqrt{\frac{\ln t}{2m_a}},$$

which is called the upper confidence bound of μ_a. The idea of the UCB strategy is to pick up the action a with the maximum upper confidence bound at each trial. Note that, if the sample of the action $a \in \mathcal{A}$ is small, the upper bound tends to be large and thus a is likely to be chosen. If sufficiently many instances are chosen for a, then the upper confidence bound shrinks and the confidence for the bound gets higher. Auer showed that the UCB strategy achieves the regret bound $O((1/\Delta) \ln T)$, which is almost optimal. There are variants of the UCB

strategy such as KL-UCB [7] which achieve optimal regret bounds for various probability models on rewards.

The third approach to the context-free bandit problem is based on the Bayesian statistics. That is, by assuming a prior probability distribution over rewards of actions in $\mathcal{A}$, choices of arms are based on the posterior probability of rewards. A representative algorithm is the Thompson sampling [2,13].

3.3 Contextual Bandit

Now we return to the contextual bandit problem. We review a variant of the UCB for the contextual cases, called the LinUCB [14].

We will assume the following model concerning the context and the reward. Given a context $\boldsymbol{x}_{t,a}$ for each action $a \in \mathcal{A}$, the expectation of the reward r_a is given as

$$E[r_a \mid \boldsymbol{x}_{t,a}] = \boldsymbol{\theta}_a^{*\top} \boldsymbol{x}_{t,a},$$

where $\boldsymbol{\theta}_a^*$ is an unknown but fixed vector and the expectation is taken w.r.t. the conditional distribution induced from D given $\boldsymbol{x}_{t,a}$.

The LinUCB strategy is based on the similar idea of the UCB. Given contexts $\boldsymbol{x}_{1,a}, \ldots, \boldsymbol{x}_{t,a} \in \mathbb{R}^n$ and rewards $r_{1,a}, \ldots, r_{t,a} \in \mathbb{R}$, the Ridge regression estimates $\boldsymbol{\theta}_a^*$ as

$$\hat{\boldsymbol{\theta}}_a = \arg\min \sum_{i=1}^{t} \|r_{i,a} - \boldsymbol{x}_{t,a}^\top \boldsymbol{\theta}_a\|_2^2 + \|\boldsymbol{\theta}\|_2^2 = \boldsymbol{A}_{t,a}^{-1} \boldsymbol{b}_{t,a},$$

where $\boldsymbol{A}_{t,a} = \sum_{\tau=1}^{t} \boldsymbol{x}_{t,a}^\top \boldsymbol{x}_{t,a} + \boldsymbol{I}_n$, $\boldsymbol{b}_{t,a} = \sum_{\tau=1}^{t} r_{t,a} \boldsymbol{x}_{t,a}$ and $\boldsymbol{I}_n$ is the n-dimensional identity matrix. It can be easily verified that $E[\hat{\boldsymbol{\theta}}_a] = \boldsymbol{\theta}_a^*$ and if $r_{t,a}$s are independent given $\boldsymbol{x}_{t,a}$s for $a \in \mathcal{A}$, the variance of $\hat{\boldsymbol{\theta}}_a \top \boldsymbol{x}_a$ is $\boldsymbol{x}_a^\top A_{t,a}^{-1} \boldsymbol{x}_a$. Therefore, for an appropriate choice of $\alpha \in \mathbb{R}_+$,

$$\hat{\boldsymbol{\theta}}_a^\top \boldsymbol{x}_{t,a} + \alpha \sqrt{\boldsymbol{x}_{t,a}^\top A_{t,a}^{-1} \boldsymbol{x}_{t,a}}$$

is an upper bound of $\boldsymbol{\theta}^{*\top} \boldsymbol{x}_{t,a}$ with high probability. Like the UCB strategy, the LinUCB strategy chooses an action with the maximum upper confidence bound. The details of the algorithm is shown in Algorithm 2. The time complexity of the LinUCB strategy per trial is $O(Kn^2)$. Apparently, computing the inversion of the matrix $A_{t,a}$ takes $O(n^3)$ time. This can be reduced by updating the inverse of the matrix using the Woodbury formula (see, e.g., [18]). The regret bound of the LinUCB strategy itself is not known. However, a modified version of the LinUCB strategy has regret bound $O(\sqrt{nT(\ln TK)^3})$ [3,8].

Li et al. also considered a more involved model where the expected rewards are not only linear in contexts and latent vectors associated with particular actions, but also depend linearly in additional contexts and shared a latent vector [14].

Algorithm 2. LinUCB (Li et al. [14])

Input: $\alpha \in \mathbb{R}_+$

1. For $t = 1, \ldots, T$:
   (a) Receive the feature $\boldsymbol{x}_{t,a}$ of each arm $a \in \mathcal{A}_t$.
   (b) For each $a_t \in \mathcal{A}_t$:
      i. If a_t is new, let $\boldsymbol{A}_a = \boldsymbol{I}_n$, where $\boldsymbol{I}_n$ is the identity matrix in $\mathbb{R}^{n \times n}$ and $\boldsymbol{b}_a = \boldsymbol{0} \in \mathbb{R}^n$.
      ii. Let $\hat{\boldsymbol{\theta}}_a = \boldsymbol{A}_a^{-1} \boldsymbol{b}_a$.
   (c) Choose the arm

$$a_t = \arg\max_{a \in \mathcal{A}_t} \hat{\boldsymbol{\theta}}^\top \boldsymbol{x}_{t,a} + \alpha \sqrt{\boldsymbol{x}_{t,a}^\top \boldsymbol{A}_a^{-1} \boldsymbol{x}_{t,a}}.$$

   (d) Receive reward $r_t \in \mathbb{R}_+$.
   (e) Update

$$\boldsymbol{A}_{a_t} = \boldsymbol{A}_{a_t} + \boldsymbol{x}_{t,a} \boldsymbol{x}_{t,a}^\top \text{ and } \boldsymbol{b}_{a_t} = \boldsymbol{b}_{a_t} + r_t \boldsymbol{x}_{t,a_t}.$$

3.4 Evaluation Methods for Contextual Bandit Algorithms

It is a non-trivial task to evaluate online algorithms using real data sets. When we apply an online algorithm to a real environment (say, a recommendation system), we will obtain a sequence of real feedbacks only for the applied algorithm and thus the obtained data is not suitable for testing other online algorithms. In particular, in the contextual bandit setting, algorithms cannot observe feedbacks for actions not taken by them.

A typical approach to evaluating online algorithms is to construct a simulator reflecting real environments. However, this approach is non-trivial as well since it needs knowledge about the real environments, which are unknown in practice.

Li et al. [14,15] developed a simple but effective method for obtaining test data sets for any online contextual bandit methods. The method is to obtain a sequence of events by running the online algorithm which always chooses random actions. The details are shown in Algorithm 3.

Algorithm 3. Uniform logging strategy [14,15]

1. Let S be the empty sequence of events.
2. For each round $t = 1, 2, \ldots, L$, repeat:
   (a) Observe a context $\boldsymbol{x}_a$ for each action $a \in \mathcal{A}$.
   (b) Choose an action $a_t \in \mathcal{A}$ uniformly randomly.
   (c) Observe reward r_{t,a_t}.
   (d) Add the event $(\boldsymbol{x}_t, a_t, r_{t,a_t})$ into the sequence S.
3. Output S.

Then we explain how to use the sequence of events S obtained by the random online strategy to evaluate an online strategy. Given an online strategy π, the evaluation strategy just runs π over S under the following manner: For any $\ell = 1, \ldots, L = |S|$, and for the ℓ-th event $(\boldsymbol{x}_\ell, a_\ell, r_{\ell,a_\ell})$ in S, (i) if the strategy π takes the action a_ℓ given $\boldsymbol{x}_\ell$, then evaluation strategy picks up $(\boldsymbol{x}_\ell, a_\ell, r_{\ell,a_\ell})$ and adds it into the history of chosen events. A precise description of the evaluation strategy is given in Algorithm 4.

Algorithm 4. Evaluation strategy [14, 15]

Input: Sequence S of events obtained by the uniform logging strategy (Algorithm 3) and an online strategy π.

1. Let h_0 be the empty sequence of the events and $T = 0$.
2. For each round $t = 1, 2, \ldots, L(= |S)$ repeat:
   (a) Get the t-th event $(\boldsymbol{x}_t, a_t, r_t)$ from the sequence S.
   (b) If $\pi(h_{t-1}, \boldsymbol{x}_t) = a_t$
      i. Update $h_t = (h_{t-1}, (x_t, a_t, r_t))$.
      ii. Let $T = T + 1$.
   (c) Else, let $h_t = h_{t-1}$.
3. Output h_T.

A notable advantage of the evaluation strategy (Algorithm 4) is that it provides an *offline* evaluation method of online strategies. More precisely, under the probabilistic setting of the environment, the evaluation is *unbiased*, in the sense that the evaluation is obtained as if we run the online strategy in the online environment. The formal statement is as follows:

Theorem 1 (Li et al. [14,15]). *For any distribution D over the set of $\mathcal{X}$ contexts and the set $[0,1]^K$ of rewards of all actions, any randomized online strategy π, and any $T \geq 1$, the following statement holds.*

1. *For any history of events $h_T = ((\boldsymbol{x}_1, a_1, r_{1,a_1}), \ldots, (\boldsymbol{x}_T, a_T, r_{T,a_T}))$, the probability that h_t is obtained by the evaluation strategy is exactly equal to probability that h_T is generated by the distribution D and the strategy π, i.e.,*

$$Pr_{D,evaluation(S,\pi)}[h_T] = \Pr_{D,\pi}[h_T].$$

2. *The expected length of the sequence of events S obtained by the uniform logging strategy is KT.*

Note that, in fact, the first statement of Theorem 1 holds for any logging strategy which chooses actions randomly and independently according to some distribution over actions, not restricted to the uniform distribution. The uniform assumption affects the second statement of the theorem. For non-uniform extensions of logging strategies, see, e.g., [20].

3.5 Experimental Results

Li et al. applied the LinUCB and the evaluation strategy to the recommendation of news [14]. The data set is obtained from about 40 million events from the Yahoo! Front Page. The action set consists of 20 news articles. The contexts of users are initially represented as more than 1,000 features including gender, age, location, and user history of Yahoo! The features of each news (action) consists of about 100 components including categories of the news. Then, by some dimensionality reduction techniques, each context for each news is concisely represented as a 6-dimensional binary vector. The reward is defined as the click through rate (CTR), which is the ratio of clicks against the total recommended news. The events are obtained by the uniform logging strategy. LinUCB strategy obtained higher CTR than the random strategy, ε-greedy strategy, the UCB.

Other applications of the contextual bandit framework include whole page presentations of search results (including not only search results, but size, layout, and etc.) [23], personalized health feedback [19], task assignments in crowdsourcing [11] and the mobile context-aware recommender systems [5].

4 Discussion

There are several issues to formulate the problem of supporting elderly people's learning as an instance of the contextual bandit.

1. **What is exactly the task? (How do we formulate the problem in a mathematical sense?):** An informal goal would be to motivate and support learning of people by giving actions according to the contexts. This goal can be mathematically reduced to maximizing the sum of rewards under the assumption that reward function is appropriately defined. The design of the reward function directly affects how the system should behave. The CTR is not necessarily a good objective for learning. The reward should promote or encourage learning of elderly people and reflect the amount of accomplishments. For example, an indicator of accomplishment of some learning task would be reasonable.
2. **How to define/sense the context:** These issues are also quite important. A good definition of contexts might need knowledge of experts in the user-machine interface, psychology, education, health cares. Also, sensing the contexts is another critical issue. For example, to measure the internal state of a user, eye-tracking devices or some other sophisticated devices might be necessary.
3. **How to make actions given a context?** This issue depends the learning task and contexts. For example, actions could be recommendations of some educational materials, reminders, etc.
4. **What kind of feedbacks of users expected?** This is a crucial factor as well. Like computational advertisements, clicks from users to some contents could be positive/negative feedbacks. Again, an indicator of success/failure of some learning task is a choice. In some cases, combinations of users' actions might be viewed as feedbacks.

5 Conclusions

In this paper, we briefly review the mathematical framework of the contextual bandit problem and its applications. We also discussed how to apply the contextual bandit framework to support elderly people's learning. In fact, it is a non-trivial challenge to define contexts and the objective to optimize (rewards) for the problem. An elderly people friendly learning support system might be useful for people with some handicaps with appropriate modifications.

Acknowledgment. This work is supported in part by JSPS KAKENHI Grant Number JP16K00305.

References

1. Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A.: Context-aware recommender systems. AI Mag. **32**(3), 67–80 (2011)
2. Agrawal, S., Goyal, N.: Analysis of thompson sampling for the multi-armed bandit problem. In: Proceedings of the 25th Annual Conference on Learning Theory (COLT 2012), PMLR, vol. 23, pp. 39.1–39.26 (2012)
3. Auer, P.: Using confidence bounds for exploitation-exploration trade-offs. J. Mach. Learn. Res. **3**, 397–422 (2002)
4. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Mach. Learn. **47**, 235–256 (2002)
5. Bouneffouf, D., Bouzeghoub, A., Gançarski, A.L.: A contextual-bandit algorithm for mobile context-aware recommender system. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012. LNCS, vol. 7665, pp. 324–331. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34487-9_40
6. Bubeck, S., Cesa-Bianchi, N.: Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Found. Trends Mach. Learn. **5**(1), 1–122 (2012)
7. Cappé, O., Garivier, A., Maillard, O.-A., Munos, R., Stoltz, G.: Kullback-Leibler upper confidence bounds for optimal sequential allocation. Ann. Stat. **41**(3), 1516–1541 (2013)
8. Chu, W., Li, L., Reyzin, L., Schapire, R.: Contextual bandits with linear payoff functions. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011). Proceedings of Machine Learning Research, PMLR, vol. 15, pp. 208–214 (2011)
9. Dubhashi, D.P., Panconesi, A.: Concentration of Measure for the Analysis of Randomized Algorithms. Cambridge University Press, Cambridge (2009)
10. Economides, A.A.: Adaptive context-aware pervasive and ubiquitous learning. Int. J. Technol. Enhanced Learn. **1**(3), 169–192 (2009)
11. Hassan, U.U., Curry, E.: A multi-armed bandit approach to online spatial task assignment. In: Proceedings of the 11th IEEE International Conference on Ubiquitous Intelligence and Computing (UIC 2014), pp. 212–219 (2014)
12. Hood, N., Littlejohn, A., Milligan, C.: Context counts: How learners' contexts influence learning in a MOOC. Comput. Educ. **91**, 83–91 (2015)
13. Kaufmann, E., Korda, N., Munos, R.: Thompson sampling: an asymptotically optimal finite-time analysis. In: Bshouty, N.H., Stoltz, G., Vayatis, N., Zeugmann, T. (eds.) ALT 2012. LNCS (LNAI), vol. 7568, pp. 199–213. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34106-9_18

14. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th International Conference on World wide web (WWW 2010), pp. 661–670. ACM Press (2010)
15. Li, L., Chu, W., Langford, J., Wang, X.: Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM 2011), pp. 297–306 (2011)
16. Ogata, H., Yano, Y.: Context-aware support for computer-supported ubiquitous learning. In: Proceedings of the 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE 2004), pp. 23–25 (2004)
17. Pachman, M., Ke, F.: Environmental support hypothesis in designing multimedia training for older adults: Is less always more? Comput. Educ. **58**(1), 100–110 (2012)
18. Petersen, K.B., Pedersen, M.S.: The Matrix Cookbook (2012)
19. Rabbi, M., Aung, M.H., Zhang, M., Choudhury, T.: MyBehavior: automatic personalized health feedback from user behaviors and preferences using smartphones. In: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UBICOMP 2015), UbiComp 2015, pp. 707–718. ACM (2015)
20. Strehl, A., Langford, J., Li, L., Kakade, S.M.: Learning from logged implicit exploration data. In: Advances in Neural Information Processing Systems (NIPS 2010), vol. 23, pp. 2217–2225 (2010)
21. Sutton, R.S., Barto, A.G.: Introduction to Reinforcement Learning, 1st edn. MIT Press, Cambridge (1998)
22. Syvänen, A., Beale, R., Sharples, M., Ahonen, M., Lonsdale, P.: Supporting pervasive learning environments: adaptability and context awareness in mobile learning. In: Proceedings of the 3rd IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE 2005), pp. 251–253. IEEE (2005)
23. Wang, Y., Yin, D., Jie, L., Wang, P., Yamada, M., Chang, Y., Mei, Q.: Beyond ranking: optimizing whole-page presentation. In: Proceedings of the 9th ACM International Conference on Web Search and Data Mining (WSDM 2016), pp. 103–112. ACM (2016)