Personalized Web-page Rendering System

P. Swapna Raj B. Ravindran

Department of Computer Science and Engineering, Indian Institute of Technology Madras
pswapna@cse.iitm.ac.in, ravi@cse.iitm.ac.in

Abstract

Personalized rendering of web pages gives the users greater control to view only what they prefer. The goal of this work is to provide a tool that will let users customize the content on the pages. Our proposed model architecture learns user preferences through interaction and eventually learns to block content that is not of interest, or is offensive, to the user. This learning from interaction is achieved through a combination of reinforcement learning and data mining techniques. In this paper we look at customizing the rendering of advertisements. We provide the user a tool that customizes itself to their preferences, and blocks irrelevant advertisements and allow only those that are of interest to the user. We also demonstrate empirically that our tool customizes itself to hypothetical hand crafted users.

1 Introduction

There is tremendous amount of information available on the internet due to exponential growth of the World Wide Web. Users are now provided with more information than ever and it has become difficult for them to find the relevant or interesting information in the web page because of this information overload. The focus of our paper is on personalized rendering of web pages which provides solution to the user's problem by learning and adapting to the users preferences, as a result blocking only irrelevant contents and rendering informative content to the user.

The two most common aspects to personalize are structure and content personalization. Structure personalization involves the altering the location of available links as well as location of the content such as providing the most relevant link to the user in a prominent place. Content personalization, on the other hand changes the content of web site depending upon the user interest.

Building personalized web applications, which should respond to the need of specific user is a challenging task. Traditional personalized approaches allow individual user's to view content of their preference based on the browsing, interaction history and demographic information. Personalization requires implicitly or explicitly collecting user information and leveraging that knowledge. Users intentions, needs, preference should be captured, analyzed and processed to satisfy his needs.

There are various sources through which user information are gathered such as the Web portal, where the user information is collected during the registration process and corresponding user profile is created. Other sources include Social networking and user-generated sites which directly provide user information without having to interact with the user. However, these approaches to pool information requires lot of time and effort, and can discourage many users. Also we cannot rely on these sources alone since users tend to give incorrect data or they deliberately fail to fill the information because of privacy issues due to which an effective personalized advertisement system cannot be achieved.

We build a tool that *learns user preferences through interaction* with the user. This learning from interaction is achieved through a combination of supervised learning and reinforcement learning (RL) techniques. Our system thus involves supervised learning directed by an active learning component through which the system interacts with its user in such a way as to learn the users preferences through direct user feedback.

In this paper, we are addressing the customization of rendering online advertisements, since one of the major issue that any user encounters while accessing web pages are the showers of advertisements either in the form of pop-up advertisements or colorful banner advertisements that occur all over the web-page creating a clutter. Instead of having to see all ads that the advertisers push to the user; if the user gets control of viewing ads of his preference, it would prove beneficial to the user.

We are approaching the problem from an user's perspective. We are not modeling the task from advertisers perspective. There are no privacy issues in our model, since the tool resides on the user's machine and only gathers and stores data locally. Depending

International Conference on Management of Data
COMAD 2008, Mumbai, India, December 17–19, 2008

on how the agent is trained the same approach can be used to block content offensive to the user.

Our system's functionality is between In-stream supervision and Experience Sampling [3, 4]. In-stream supervision does implicit labeling by watching activities whereas Experience Sampling probes in real time for labels. The idea is to learn when to probe. Decision Theory (DT) is used to model this (DT selective supervision - Active learning).

The paper is organized as follows. In Section 2 we give a brief overview of the related work. We present our architecture of personalized advertisement system in Section 3 and some empirical results on a Internet Advertisement database in Section 4. We conclude with some discussions in Section 5.

2 Related Work

Personalization generally refers to making a website more responsive to the unique and individual needs of each user. In other words, Web personalization is the ability of users to modify the web page layout or specify what content should be displayed in the web page. The ultimate aim of personalization is the user satisfaction.

It is motivated by the recognition that a user has needs, and meeting them successfully is likely to save user's effort and time. There is great diversity in how personalization can be achieved. Information about the user can be obtained from a history of previous sessions, or through interaction in real time.

Personalization saves time by eliminating repetitive tasks by recognizing habits and shortening the path to engage in such habits. Also personalization provides better information: filter out information not relevant to a person; provide more specific information that is relevant to personal interests.

Web personalization can be either link or content personalization. Link personalization involves selecting the links that are more relevant to the user. E-commerce applications use link personalization to recommend items based on the clients buying history or some categorization of clients based on ratings and opinions. In Content personalization, content is personalized when pages present different information to different users. When substantial information in a page is personalized, other than link anchors we call it as Content personalization.

Personalization involves different technologies such as rule-based filtering, collaborative filtering and content-based filtering. In the rule based approach to personalized recommendation, marketing experts typically generate marketing rules which are used to perform inferencing based on the customer data [14]. However, it is difficult to obtain profitable, valid and rich marketing rule set from experts all the time thus Kim et al [6] have adopted the decision-tree induction technique to generate the marketing rules thus replacing marketing experts and provide personalized advertisements in Internet storefront. BroadVisions One-to-One System is a commercial product that uses a rule-based matching technique to provide proper advertisements to customers who stay connected by Internet radio, Internet television, Internet banking.

Content-based filtering systems are solely based on individual user's preferences. The system tracks each user's behaviour and recommends items to them that are similar to items the user liked in the past. The Recommender System developed by Robin and Marten [10] known as PRES (Personalized Recommender System) used content-based filtering techniques to search for article on the net about Home Improvement.

PRES System collects articles related to Home Improvement on the Internet and then it will create dynamic hyperlink to make it convenient and easier for a users. It then makes recommendation to the user by comparing user profile that had been set with selected content or preferences with the content of each document in the collection. In order to improve the accuracy, users can provide feedback based on the content they received. The document is then ranked based on similarity, novelty, proximity and relevancy. This makes a significant improvement in the interaction between system and the users.

The advantage of the technique is that it is user dependent. Since the preferences are explicitly set by users, user's have much more power over what and which kind of content they wish to view. The disadvantage of using content-based filtering is that system has to explicitly ask preferences, ratings, and recommendation from the user itself. Some terms in contents might have more than one meaning and this make the user profile less accurate.

Collaborative filtering systems [15, 2] invite users to rate objects or divulge their preferences and interests and then return information that is predicted to be of interest to them. This is based on the assumption that users with similar behavior (e.g., users that rate similar objects) have analogous interests. There are several commercial products based on collaborative filtering, such as GroupLens, Jester, LikeMinds Personalization Server etc. In this technique, a serious limitation is that a new customer has to provide preferences for a large number of items in order to view personalized advertisements. Also, for any new advertisement or product, some preference data from people in the data set is required before collaborative techniques can be applied. Content-based, rule-based, and collaborative filtering may also be used in combination, for deducing more accurate conclusions.

Personalization has become hype in areas such as electronic commerce, Interactive TV, Mobile, Travel support [8, 1, 16, 7] etc. Personalized advertisement techniques have been utilized by Digital TV environments, where the focus is on consumer clustering and targeted advertising. Typically the motive of interactive TV is to play/telecast ads that only suit the customer interest in their TV channels. In order to achieve this motive, demographic profile of customer with his preference and interactions with the TV are tracked. Setup box is used to store consumer data locally to maintain privacy issues and this data is transferred to the Server where data mining techniques like clustering are performed. The extracted behavioral rules are then sent to consumer's setup box for classification of each consumer in the family and the ads are telecast to match the consumer interests accordingly [13].

Personalization of web content for wireless mobile devices has also been gaining interest. The functionalities that we perform with our Personal Computers can now be done with the mobile device through access to web; like booking travel tickets, performing transaction etc. However, hand held devices can be taken to higher level of personalization wherein they adopt user preference according to his needs, priority and environment. Services such as banking, route planning, travelling, location tracking and city guide could be achieved through extended service description, adequate personalization, advanced profiling, proactive service discovery and execution [24].

A system that allows a user to view personalized web content on any hand held device has been proposed by Xinyi et al. They provide a user interface to specify and personalize web content which they store in interest knowledge. The extraction and tracking system will then track different websites and retrieve the latest content and store in the retrieved content. Optimization subsystem optimizes the retrieved content for target device like desktop, palm, mobile phone or pocket PC [25].

Personalization travel support system using Reinforcement learning was proposed by Anongnart et al. This system applies RL to analyze, learn customer behaviors and recommend products to meet customers need. They have employed two learning approaches in their study which are personalization learner by group properties (learning from all users in one group to find group interest of travel information) and personalization learner by user behavior (learn from user profile, user behavior to find unique interest of each user) [18]

Typically users were monitored according to the click-through history [9, 12] in order to provide them with better services. Later both Web usage and content mining were both integrated to provide more effective personalization to the user. Some of the approaches to personalization using this hybrid approach are mentioned [20, 5, 11]. Taghipour et al employed Q learning approach to Hybrid approach to web personalization [20, 21]. They make use of conceptual relationships among web pages with semantic knowledge about the user behavior. Existing methods were used to obtain conceptual structure of the website and framed state, action and reward function to capture user behavior.

AD ROSA system for automatic web banner personalization was introduced by Kazienko et al. AD ROSA integrates both web usage and content mining techniques [5]. In this approach they combine conceptual and usage information to improve the quality of web recommendation. Historical user session's are clustered and the centroid represents one usage pattern of publishers web site. Likewise historical visited ads corresponding to the user session are also aggregated thus mapping one web usage pattern to one ad visiting pattern.

Similarly conceptual space is generated by aggregating the term vectors from publishers web pages and the associated terms in the Advertisers web site are also extracted (content analysis in ad site) to gather general subject matter. Web page requested by the user is assigned to both the closest usage pattern and conceptual space. Ads related to closest conceptual space will divert users to website where the content is suitable to the user. While the closest usage pattern and its corresponding visiting pattern delivers advertisements that are most likely to be clicked by the user.

3 Proposed Framework

The focus of our work is on personalized rendering of web-pages. The content of a website can be tailored for a user by gathering user-information during interaction with the user, which is then used to deliver appropriate content to the user. In this paper we look at one of the most annoying features of web pages, the advertisements. Increasingly, Internet consumers are bombarded with Internet advertisements such as banner advertisements, pop-up ads. Our focus is to customize the rendering of the ads or images seen by a user visiting a web-page. The images on the web page can be broadly looked at as ads and non-ads (integral part of the web page which reflects the context of the page).

The goal of the system is to incrementally learn to identify the ads that the user prefers, as well as, the images that constitute non-ads. Only then the preferred or good-ads along with the non-ads will be rendered on the web-page, thereby reducing distracting clutter.

In our approach to personalized rendering of web pages, we use concepts from both Data mining and Reinforcement learning. The proposed architecture consists of two stages: classification stage followed by a decision making stage. Many data mining algorithms have their roots in statistical machine learning (ML), and as such haven't shed much of their heritage.

Traditionally ML algorithms have used certain inherent statistical properties of the data to evaluate the goodness of the patterns discovered by the algorithms. These are known as intrinsic measures, examples of which include prediction accuracy in the case of classification or regression tasks, purity in the case of clustering algorithms, support and confidence in the case of mining associations etc.

The problem with an intrinsic measure is that we are not sure how good the patterns are in decision making. The data mining stage tries to optimize intrinsic measure such as predictive accuracy, while the pattern evaluation stage evaluates pattern for its support in decision making.

Supervised learning otherwise termed as passive learner can predict the right class labels provided the side information (class labels) is known beforehand. However, obtaining the class labels beforehand is a costly procedure. Supervised learning does not support interactive learning alone. Our system thus involves supervised learning directed by an active learning component in order to learn user preferences through direct user feedback. Active learning reduces annotation costs for supervised learning by concentrating labeling efforts on the most informative data. Typically RL is used to pick samples to obtain side information for.

The first stage is classification, where initially the ad and the non-ad are separated. This stage should identify whether the random image shown is an ad or non-ad and accordingly take corresponding action in the following stage.

We adopt k-Nearest Neighbor (k-NN) classifier in the classification stage. k-NN classifier is an instance-based learning algorithm that is based on a distance function for pairs of observations, such as the Euclidean distance. k-NN classification decision is based on a small neighborhood of similar objects. The most significant advantage of k-NN classifier is that no retraining is required since it is an easy classifier to change incrementally. k-NN does not expect the samples to come from the data distribution as Artificial Neural Network (ANN) does since classification boundary depends on the nearest samples and not all data points in k-NN.

The second stage is a decision stage that decides what to do with the images (render or not-render to the user). Once the images chosen are rendered, the user picks the bad ones. These are used to now refine the classification problem. The key issue here is when to render a bad-ad? Rendering it might give an immediate penalty if the user rejects it. But it could be advantageous in two ways: (a) The user might accept it, in which case we have a new sample of a good-ad; (b) the user still rejects it, but this allows the agent to delineate the good and bad classes better.

The decision stage is modeled using Reinforcement Learning concept, where the agent learns in course of time which action should be taken when it comes across a particular image. There is no real characterization of the user via a labeled data set hence we choose to use RL to decide the data points to be used and their labels for effective decision making. Learning is driven by a meta-learning RL kind of framework.

Meta-learning allows us to adjust the bias of the mining algorithm. This in turn affects the performance of the algorithm, since the patterns of the algorithm is heavily influenced by the bias of the algorithm. There is a wealth of work on different meta-learning models in the literature for different settings [22, 23, 17].

The learning algorithm adopted is the Q-learning algorithm [19]. Q-learning is a popular Reinforcement Learning algorithm that does not need a model of its environment. Q-learning algorithms works by estimating the values of state-action pairs. The value Q(s, a) is defined to be the expected discounted sum of future payoffs obtained by taking action a from state s and following an optimal policy thereafter. Once these values have been learned, the optimal action from any state is the one with the highest Q-value.

Q-learning works by successively improving its evaluations of the quality of particular actions at particular states. The agent learns the Q-values as the training occurs and uses the learnt Q-value for making better decision when a new image is picked. Q-values can thus provide estimation of how successful that action might be. We use Q-learning since it follows off-policy control wherein sample from trajectory can be handled unlike in Sarsa learning algorithm where complete trajectory is considered. This feature of Q-learning is vital since there are no well defined trajectories in our problem.

The internet advertisements data set on which we conduct our experiments is taken from the UCI repository. The data set represents a set of possible advertisements on Internet pages. The data is described by image features, and some weak contextual features and does not represent content information of ads. The features encode the geometry of the image (if available) as well as phrases occurring in the URL, the image's URL and alt text, the anchor text, and words occurring near the anchor text.

Our work focuses on directly interacting with the user to learn his/her preferences. The user contributes by helping the agent learn the users behavior to certain ads. The agent cannot discriminate between the images at the start (unaware of the user preferences in the start). However, we train the agent to do so by obtaining some feedback from the user in the form of reward or penalty which in some sense behaves as the misclassification cost indirectly.

This problem has been set up as an RL problem as discussed earlier. Key steps to formulating this as an RL problem: (a) states; (b) actions; and (c) rewards.

State representation is 3-tuple of the form (n, g, b) where n, g and b constitute the count of non-ads, good-ads and bad-ads in the k nearest neighbors to that image. State representation reflects the confidence of

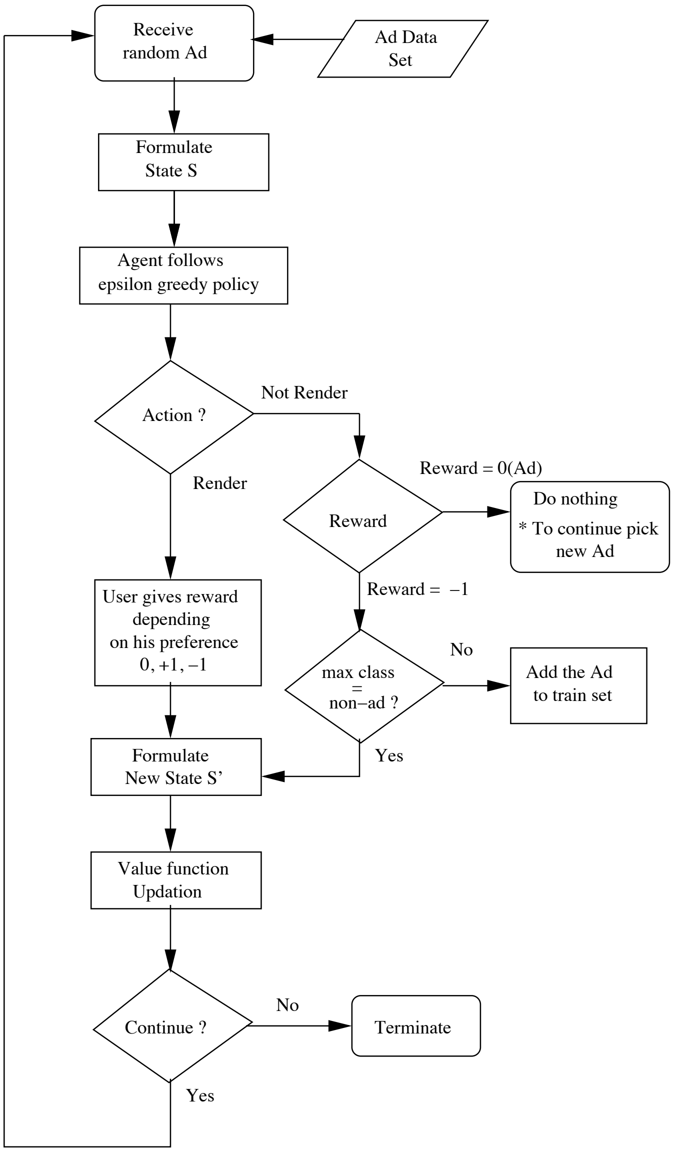

Figure 1: Flow chart of our work

the class labels generated. For instance, if $k = 30$, and for an image and if the state is (15, 10, 5), it represents that there are 15 non-ads, 10 good-ads and 5 bad-ads among 30 nearest neighbors for that image. We chose the above state representation since k-NN helps encode the certainty/uncertainty about the class label by looking at the k nearest neighbors and following a voting scheme.

There are two kinds of actions here: one is the render/not render actions, and other is what must be done to change the state of the meta-learning process. Rewards are relatively straight forward in our formulation. If the user rejects an image, then the reward is -1. If the user accepts an image explicitly, then the reward is $+1$. If the user ignores the image, then the reward is 0 (this is the case typically for all non-ad images). We consider a formulation where only the purported ads get rewarded, i.e. the images the agent thinks are ads and still renders them.

The flowchart in figure 1 indicates the process flow in our work and the corresponding algorithm is given in Algorithm 1. We refer to the step numbers in the algorithm 1 as we describe our framework. Initially there are only two kinds of images to start with: good-ads and non-ads. The input ad-data set has missing values which are taken care of by averaging technique. We perform stratified sampling of the data set, such that 2/3rd of the data is used as train set and rest used for evaluate/operation purposes.

We initialize polarity of all train images to some positive constant (good-ads = 5, non-ads = 10). We consider the non-ads to be significant than the ads since we know that they can be rendered for sure, hence the polarity to start with is high for non-ad. The Q function for each possible state, action combination is initialized to zero initially (steps 2 and 3 of algorithm 1).

During training, a random image is received by the agent (we simulate this by randomly picking an ad, step 5), the agent then picks the k nearest neighbors from the stored data points. The state information for this image is obtained by looking at how many of the k neighbors are non-ads (n), good-ads (g) and bad-ads (b) . Initially as mentioned all ads are good-ads, there are no bad-ads. We formulate a state with this information (step 6). Now with the state (n, g, b), the agent picks an action based on epsilon greedy action selection mechanism (step 7). The actions are render the ad or not render. There are two cases. The first case is when action picked is render (step 9). The agent then shows the image to the user and receives his feedback (step 10). The user can do one of three things

1. Ignore it : $reward = 0$
2. Like it : $reward = 1$
3. Dislike it : $reward = -1$

The typical action for a non-ad is ignore.

The second case is when action picked is not render (step 14). The image is not shown to the user. The user can do one of two things

1. Be unaware of it : $reward = 0$
2. Miss it : $reward = -1$

If the true class of the image is non-ad, then user will miss it. Otherwise, the user is unaware of it.

After we obtain feedback from the user in terms of reward values, the agent then updates the value function. There are two cases. The first one is when agent renders the image and depending on the rewards received, the value function updation is performed and

Algorithm 1: Algorithm for Web Page rendering

```
1  Initialize, for all s ∈ S, a ∈ A;
2  Initialize weights for each representative
   sample of training data. /* Ad = + 5;
   Non-Ad = + 10 */;
3  Q(s, a) ← 0;
4  Repeat(for each episode);
5  image = Pick a random image from
   training set;
6  s = Generate_State(image);
7  Choose a from s using policy derived from
   Q (epsilon greedy);
8  Take action a;
9  if a == render then
10     Observe user generated r;
11     Update_Weights(image, r);
12     s' = Generate_State(image);
13     Q(s, a) ← Q(s, a) + α [r + γ max_a'
       Q(s', a') - Q(s, a)];
14 else
15     Observe r;
16     if r is negative then
17         if k-NN classification of the image is
           not a non-ad then
18             Add example to representative
               set;
19         else
20             Update_Weights(image, r);
21             s' = Generate_State(image);
22             Q(s, a) ← Q(s, a) + α [r + γ
               max_a' Q(s', a') - Q(s, a)];
23 /* State computation is specific to the Ad
   problem */;
24 Function Generate − State(image);
25 Find the k nearest neighbors of the image;
26 Retrieve corresponding class-labels;
27 Count the number of non-ads (n), good-ads
   (g) and bad-ads (b) and form a vector (n, g,
   b);
28 /* Update Weight function */;
29 Function Update_Weights(image, r);
30 Find the k nearest neighbors of the image;
31 Update the weights of all the neighboring
   images slightly positive or negative (with
   respect to r) by some amount proportional
   to their distance from this image;
32 Flip the labels according to the weight
   changes in the training set;
```

the other case is when agent does not render an image. If the agent renders the image and obtain positive or negative reward from user, then agent retrieves k nearest neighbors that contributed to this state computation and modifies their polarity slightly positive or negative by some amount proportional to their distance from this image (step 11).

According to the change in polarity the labels are flipped (say if polarity because of weight updation changed from positive to negative we flip the label to bad-ad) (step 32). Then the state representation is recomputed (n', g', b') (step 12). The agent finds the max Q-value in the new state and updates the value of the render action in the old state (n, g, b) (step 13).

If on the other hand, agent obtained zero reward, then agent retrieves k nearest neighbors that contributed to this state computation. Move their polarity slightly positive by some amount proportional to their distance from this image. This is done since the agent assumes that the nearest neighbors are non-ads as well. Agent recomputes the state representation (n', g', b'). The agent finds the max Q-value in the new state and updates the value of the render action in the old state (n, g, b) as mentioned above.

If the agent does not render an image, and obtains a negative reward, and if the majority classification of this image is not non-ad ($g + b > n$) then this image is added to the training set and polarity is initialized to the same constant used at the start for this image (steps 16, 17 and 18). We add this to the training set since we have obtained a good sample that could help us delineate between good and bad-ad. However, if the majority classification is non-ad (i.e., agent is confident that it is a non-ad), then agent updates the polarity slightly upwards and then updates the Q-value of the do not render action (steps 20, 21 and 22).

If the agent does not render an image, and obtains a zero reward, the agent does nothing (not rendering good/bad-ad). The second pass to find k nearest neighbors and flipping their labels according to weight change is to provide a target for updating the value function.

Our system can filter content at the users end given the diversity of ways in which ads are embedded (popup ads, banners) provided the system is able to recognize them as ads.

4 Experimental Setup

The experiments were conducted on the internet advertisements data set from the UCI repository. The data set represents a set of possible advertisements on Internet pages. The features encode the geometry of the image (if available) as well as phrases occurring in the URL, the image's URL and alt text, the anchor text, and words occurring near the anchor text. The data set consists of 3279 (2821 non-ads, 458 ads) number of instances, described by about 1558 features (3 continuous; others binary). One or more of the three continuous features are missing in 28% of the instances. Each of the instance is either labeled as ad or non-ad.

Our customized advertisement tool follows a two stage process with a classification stage, identifying the category of image as ad or non-ad; followed by a decision stage where the actions could be render or not render the ad. The data mining approach that we have used is the k-NN classifier, since no retraining is required (incremental learner) and also since sampling need not follow actual distribution. We use the Q-learning algorithm from the Reinforcement learning to design the agent. We use epsilon greedy action selection mechanism (epsilon = 0.1 : Probability with which the agent can explore). k value is set to 50 and 75 in our experiments.

We design a hypothetical hand-crafted user with certain preference using if then rules. For example in our experiments we have two users. User 1 prefers to read peace related articles and news articles and does not have interest in Hollywood while User 2 is interested in web crawlers, meta-crawlers and share-wares. The hypothetical user is designed such that when an image or ad is shown; the user responds with reward (0 or $+1$ or -1) where reward 0 is indicative that non-ad was shown while reward $+1$ indicates that good-ad was shown to the user and reward of -1 indicates a bad-ad was shown to the user.

For each user the agent starts with a classifier that can identify ads vs. non-ads; and the initial information is not biased for a particular user. Stratified sampling is used to divide the dataset into train data and operation or evaluation data. In each run, two third of data is used as train set from which neighboring images for a given image are retrieved. The rest of the data are used as operation data to which the agent is exposed, which aids the learning process.

We randomly sample 100 images from the operation data and provide them to the agent to learn. Next we randomly sample 90 images (30 each of non-ad, good-ad and bad-ad) for evaluating the information learnt by the agent. We run the experiment for 500 steps ($50,000$ images, where 1 step = 100 images) which constitutes a single run. We average over 8 runs.

We plot graphs which depict the percentage of image categories shown to the users. We compute the count of non-ad, good-ad and bad-ad (n, g, b) shown among the 90 evaluative images after the agent has learnt from each of the 100 images. Average of (n, g, b) after each step across multiple runs is computed and the graphs are plotted with respect to each user. The graph which is indicative of the percentage of image categories shown to the users include three colored curves (blue representing non-ad, red representing good-ad while black is the bad-ad).

The graphs show that the percentage of non-ads, good-ads shown have increased with more training while bad-ads percentage has fallen. The agent learns to differentiate between the good and bad-ad only from 100th step onwards and later with more training it has well delineated between the good and bad-ad. Non-ads are learnt quicker. Hence with more training our RL agent learns to show the images that the user would prefer to see.

We also plot the graph depicting average reward obtained versus maximum possible reward. To obtain the maximum average reward we assume complete knowledge of the correct labels, and then pick the most rewarding action. The optimal average reward one could obtain is 30. Below shown are graphs for $k = 50$ and 75. Two sets of graphs for user 1 and user 2 with respect to $k = 50$ and 75 are plotted in figures 2 to 9.

Adblock is a content-filtering extension for the Mozilla Firefox and Mozilla Application Suite web browsers. Adblock allows users to prevent page elements, such as advertisements, from being downloaded and displayed. We should right-click on a banner and choose "Adblock" from the context menu, the banner won't be downloaded again. This is how Adblock operates. However we do not compare our approach with Adblock since the input dataset format we employ (each ad is an instance/tuple described by thousands of image features) is not compatible to Adblock.

5 Conclusion

In this paper we have presented a generic approach to rendering web-pages according to users preference by interaction with the user. The tool that we have proposed functions locally with respect to a single user. We used a two stage process: classification stage followed by decision making stage with feedback (meta-learning) to retrain the classifier to adapt to the user preference. The reason to have used two stage architecture is that the classification stage operates by optimizing the intrinsic measures alone it does not quantify how useful or actionable the pattern would be. Hence we adopt a decision stage as well.

We use k-NN classifier since it has its own advantages of being an incremental learner and also model free classifier hence no retraining is required. k-NN does not expect the samples to come from the data distribution as Artificial Neural Network (ANN) does since classification boundary depends on the nearest samples and not all data points in k-NN. We have proposed a general RL kind of meta-learning framework, however the state representation, action and reward could vary according to the problem domain.

In our case we used the k nearest neighbors to represent the state. The state representation could differ with different classifiers in our case the confidence of class labels is used as state. Typically RL is used to decide the data points to be used and their labels for effective decision making (active learning). We have illustrated the validity of our approach empirically on the Internet advertisement task. We choose to work on the most annoying aspect of the web-pages "Ads" and to provide a customized tool for the user which could allow him to view only ads of his preference and block other irrelevant ads.

References

[1] K. Chorianopoulos. Personalized and mobile digital tv applications. *Multimedia Tools Application*, pages 1–10, 2008.

[2] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22:5–53, 2004.

[3] E. Horvitz. Machine learning, reasoning, and intelligence in daily life:directions and challenges.

[4] A. Kapoor and E. Horvitz. Principles of lifelong learning for predictive user modeling.

[5] P. Kazienko and M. Adamski. *Personalized Web Advertising Method*. Springer, 2004.

[6] J. W. Kim, B. H. Lee, M. J. Shaw, H. L. Chang, and M. Nelson. Application of decision-tree induction techniques to personalized advertisements on internet storefronts. *International Journal of Electronic Commerce*, 5:45–62, 2001.

[7] K. Kim, J. M. Carroll, and M. B. Rosson. An empirical study of web personalization assistants: Supporting end-users in web information systems. *Proceedings IEEE 2002 Symposia on Human Centric Computing Languages and Environments*, pages 60–62, 2002.

[8] M. Lankhorst, H. V. Kranenburg, A. Salden, and A. Peddemors. Enabling technology for personalizing mobile services. *HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, 3:1464–1471, 2002.

[9] U. Lee, Z. Liu, and J. Cho. Automatic identification of user goals in web search. pages 391–400. ACM Press, 2005.

[10] R. V. Meteren and M. V. Someren. Using content-based filtering for recommendation.

[11] B. Mobasher, H. Dai, T. Luo, Y. Sun, and J. Zhun. Integrating web usage and content mining for more effective personalization. *In Proceedings of the International Conference on E-Commerce and Web Technologies*, pages 165–176, 2000.

[12] B. Piwowarski and H. Zaragoza. Predictive user click models based on click-through history. pages 175–182. ACM, 2007.

[13] K. C. Pramataris, D. A. Papakyriakopoulos, G. Lekakos, and N. A. Mylonopoulos. Personalized interactive tv advertising: The imedia business model. *Electronic Markets*, pages 17–25, 2001.

[14] Rainsberger and J. B. Rule-based personalization framework for integrating recommendation systems. 2005.

[15] P. Resnick and N. Iacovou. Grouplens: An open architecture for collaborative filtering of netnews. *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, 1994.

[16] B. B. Schafer, J. Konstan, and J. Riedi. Recommender systems in e-commerce. *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166, 1999.

[17] N. Schweighofer and K. Doya. Meta learning in reinforcement learning. *Neural Networks*, 16:5–9, 2003.

[18] A. Srivihok and P. Sukonmanee. Intelligent agent for e-tourism: Personalization travel support agent using reinforcement learning. *Proceeding of The 14th International World Wide Web Conference (WWW2005) Workshop*, pages 287–292, 2005.

[19] R. S. Sutton and A. G. Barto. *Reinforcement Learning : An Introduction*. MIT Press, 1998.

[20] N. Taghipour and A. Kardan. A hybrid web recommender system based on q-learning. *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 1164–1168, 2008.

[21] N. Taghipour, A. Kardan, and S. S. Ghidary. Usage-based web recommendations: a reinforcement learning approach. *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, pages 113–120, 2007.

[22] S. Thrun and L. Pratt. *Learning to learn*. Academic Publishers, 2001.

[23] R. Vilalta, C. Giraud-Carrier, P. Brazdil, and C. Soares. Using meta-learning to support data mining. *Proceedings of International Journal of Computer Science and Applications*, 2004.

[24] M. Wagner, W.-T. Balke, R. Hirschfeld, and W. Kellerer. A roadmap to advanced personalization of mobile services.

[25] X. Yin, W. S. Lee, and Z. Tan. Personalization of web content for wireless mobile device. *Wireless Communications and Networking Conference*, pages 2569–2574, 2004.

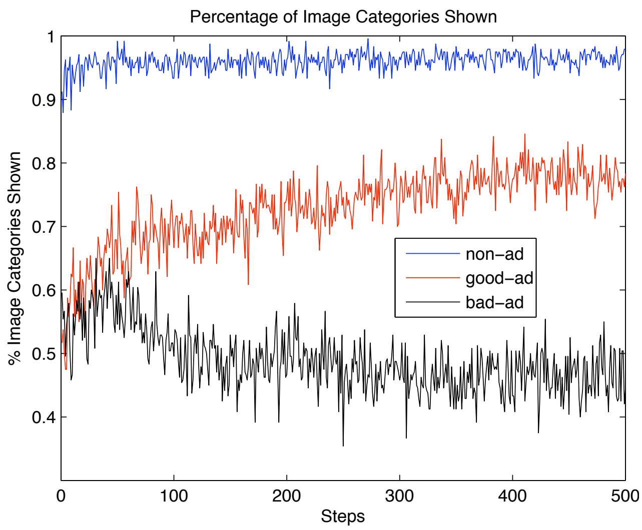

Figure 2: Percentage of image categories shown (User 1, k = 50)

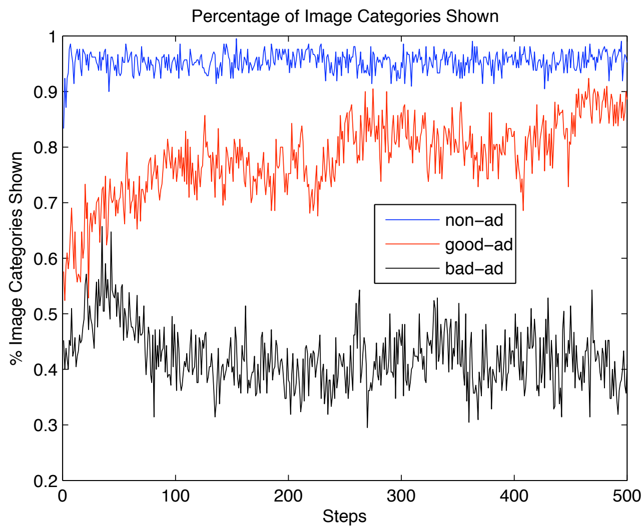

Figure 4: Percentage of image categories shown (User 2, k = 50)

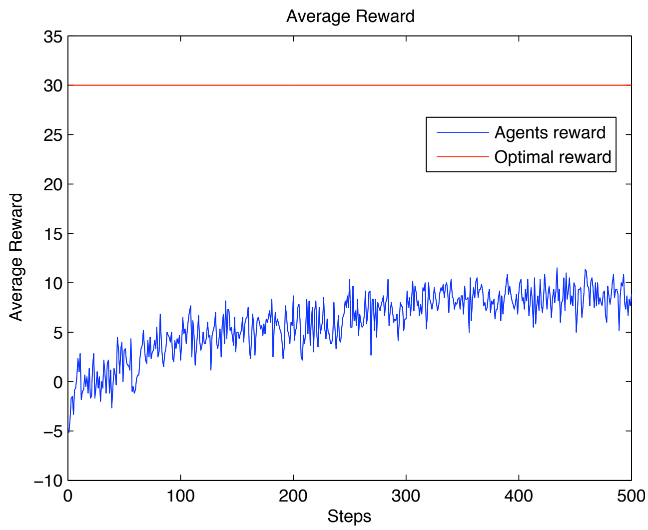

Figure 3: Average reward (User 1, k = 50)

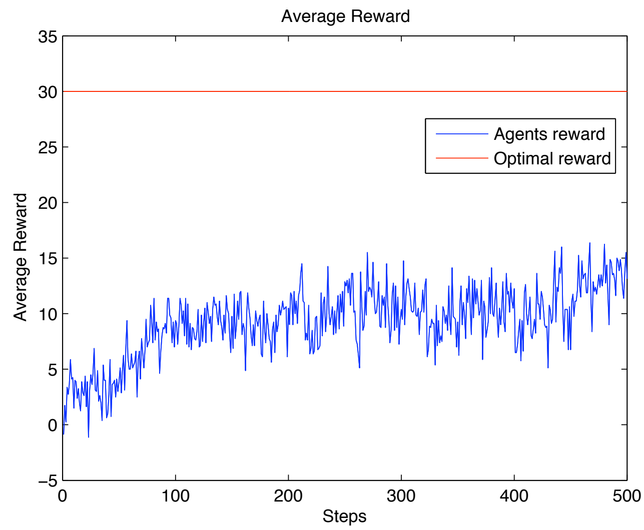

Figure 5: Average reward (User 2, k = 50)

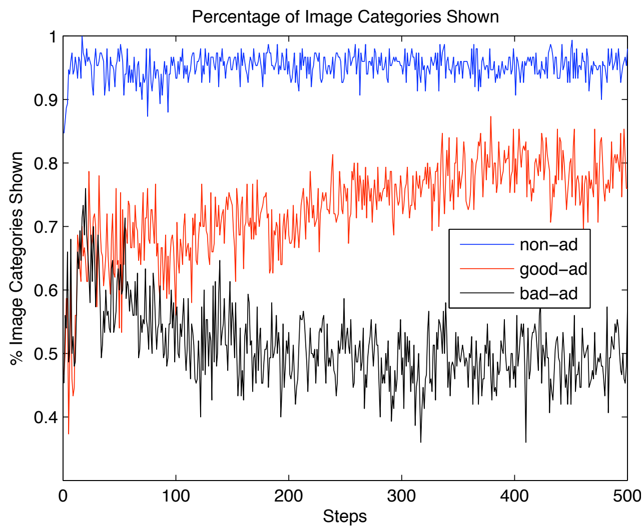

Figure 6: Percentage of image categories shown (User 1, k = 75)

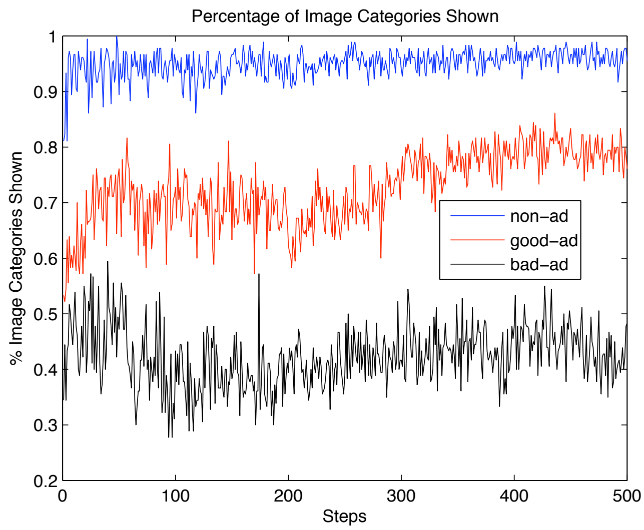

Figure 8: Percentage of image categories shown (User 2, k = 75)

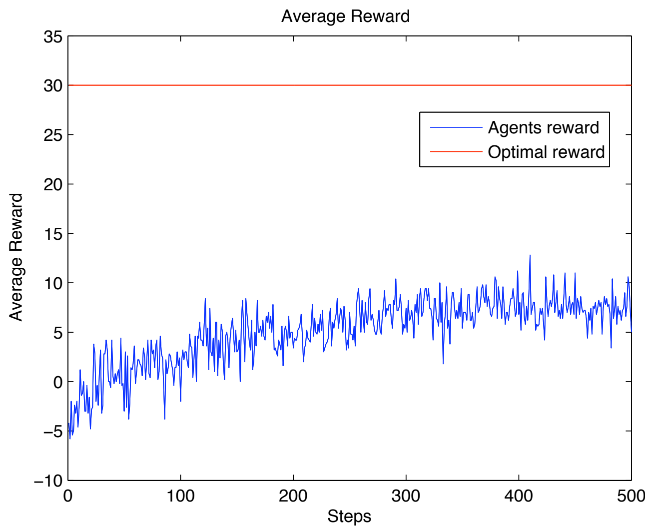

Figure 7: Average reward (User 1, k = 75)

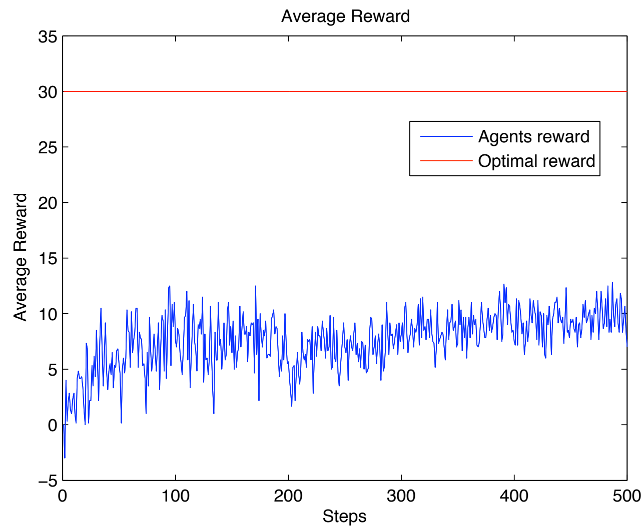

Figure 9: Average reward (User 2, k = 75)