This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

Evading User-Specific Offensive Web Pages via Large-Scale Collaborations

Mingwei Xu[1], Qinghua Li[2], XueZhi Jiang[1], Yong Cui[1]
[1] Department of Computer Science and Technology, Tsinghua University
[2] Department of Computer Science and Engineering, Pennsylvania State University
[1] {xmw, cy}@csnet1.cs.tsinghua.edu.cn; [2] qxl118@psu.edu; [1] jxz07@netlab.edu.cn

***Abstract*—Web pages polluted by unhealthy contents (e.g. pornography or violence) have offended many users and become a social headache. This paper presents a collaborative rating system and a light-weight algorithm to detect polluted pages and thus improve user experience of web browsing. It mainly tackles two challenges. First, the system should cater to web users' different tastes and judging standards on which polluted pages they like or dislike. Second, the system should be resilient to dishonest ratings and collusions. The model and the algorithm are evaluated by simulations which show that they can work well.**

Keywords- Offensive Web Pages; collaborative; model

I. INTRODUCTION

With the sweeping popularity of World Wide Web, a large amount of harmful web pages have emerged. These pages can be grossly classified into two categories, phishing pages and polluted pages. Phishing pages usually allure users to expose their credit card numbers or deceive users into paying for nothing. As to polluted pages, the typical examples include pages contaminated by pornography and violence, which have offended many users, and are especially harmful to young kids. Polluted web page has become a social headache.

Some work has been done to detect phishing websites, such as the Microsoft Phishing Filter and the Netcraft Toolbar. However, few attentions have been paid to polluted pages. Though at the website level it is easy to block accesses to certain well known polluted sites, e.g. porn sites, most polluted pages reside in ordinary websites, especially in BBS-like forums and the rapidly developing personal Blogs. The specific pages should be detected rather than the whole website. Data mining might help to detect some polluted pages, especially to text-based contents. But it fails to work well when page content is shown in a picture.

In this paper, we mainly focus on detecting polluted pages. We observed that most polluted pages are visited by multiple users at different time and places. If the first few victims of a page could warn subsequent visitors, the latter could evade that page. Based on this idea, we build up a defense system which enables web users to collaboratively filter polluted pages. Several challenges make such a system difficult to build. First, it is improper and impossible to find one polluted page set for all users, because users have different tastes and judging standards. So the system should cater to diversified user tastes and judging standards, and detect user-specific polluted pages. Second, malicious users might make dishonest ratings. A user might slander a normal page to be polluted, or a powerful entity might manipulate a group of users to prejudicially attack one website or protect another. So the system should be resilient with dishonest users and even colluding groups.

This paper presents a collaborative rating system which can detect the polluted pages for users. Offensive Web Pages (OWPs) are used to denote polluted pages in later parts. The remainder is arranged as follows. Section II introduces related work. Section III states the OWP problem, and presents our collaboration model, prediction algorithm and attack analysis. Section IV evaluates the model and the algorithm by simulation. Section V concludes this paper.

II. RELATED WORK

Collaborative filtering scheme has not only been used in recommender systems to help users find content of interest from a potentially overwhelming set of choices [1], but also been used in P2P file-sharing systems to detect insecure objects, such as spam emails [2] and polluted files [3].

Existing methods of avoiding bias from unfair ratings can be grossly classified into two categories, endogenous and exogenous. The endogenous methods assume that unfair ratings can be recognized by their statistical properties, and they exclude or give low weight to presumed unfair ratings [4, 5]. The shortness of the endogenous methods is that the statistical property fails to function correctly when more than 50% ratings are unfair. In fact, reference [4] first uses collaborative filtering to limit this method within a trustable community. The exogenous category covers methods where external factors, such as the reputation of the rater, are used to determine the weight given to ratings [3]. Many peer reputation systems [3] use pair-wise similarity as the external factor to determine peers' reputations. However, this method is too heavy to enable large-scale collaboration.

III. MODEL, ALGORITHM AND ANALYSIS

Before stepping into the collaboration model, it is helpful to formalize the OWP problem.

Web Page Attribute The different types of page content pollutions are defined as different page attributes, which can be denoted as:

$$A = (a_1, ...a_i, ...a_n) \quad (1)$$

Page attributes have continuous values ranging from 0 to 1. Attribute value denotes the degree of pollution. Value 0 means no pollution, while value 1 means full pollution.

Web User Opinion User opinion of web page at any attribute is modeled as a binary value, *clean* or *offensive*:

$$O=(o_1,...o_i,...o_n) \quad o_i \in \{clean, offensive\} \quad (2)$$

o_i denotes the opinion of page attribute a_i. Binary option simplifies the user's decision.

OWP Problem For each user and at any page attribute, web pages are classified into two categories, offensive pages and clean pages. The classification is user-specific because of different users' tastes and judging standards. The goal of solving the OWP problem is to predict user-specific offensive pages before users visit them.

A. Collaboration Model

In our model, collaborating users leave a rating after they visit a web page. All ratings are correlated to predict potential offensive pages for each user.

To achieve high efficiency, collaboration should happen within a collaboration domain. A collaboration domain concerns with one page attribute of one website. Ratings to pages in one website should only be used to predict OWPs in the same website. One reason is that a website usually has a stable viewer community. It is more likely to find common OWPs within this community. The other reason is that a user might have subjective likes or dislikes between different websites, but this prejudice is often stable to one website. So limiting collaboration into one website can eliminate such prejudice asymmetry. In the following, we discuss our collaboration model and OWP prediction algorithm in one collaboration domain. The attribute concerned is denoted by a. The page set and the user set are denoted by P and U, which have n_p and n_u elements.

Our collaboration model is illustrated in Fig.1. There are two components, the collaborator and the predictor. From collaborator k's view, P^k is the union of the clean subset P_c^k and the offensive subset P_o^k. After visiting a page, a collaborator makes a rating based on the rule:

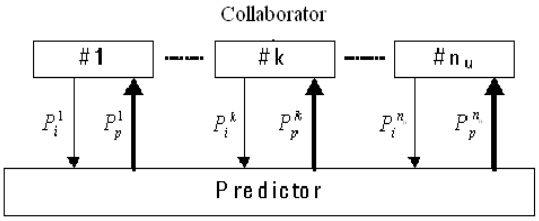

Fig.1. Collaboration Model

Honest Rating Rule Given a page p, collaborator k rates: 1) clean if $p \in P_c^k$; 2) offensive if $p \in P_o^k$.

The pages collaborator *k* has visited form a set P_i^k which is the union of the clean subset $P_{i,c}^k$ and the offensive subset $P_{i,o}^k$: $P_i^k = P_{i,c}^k \bigcup P_{i,o}^k$.

The predictor collects ratings, correlates them and predicts potential offensive pages for each collaborator. The prediction set for collaborator k is denoted by P_p^k, which is the union of the clean subset $P_{p,c}^k$ and the offensive subset $P_{p,o}^k$: $P_p^k = P_{p,c}^k \bigcup P_{p,o}^k$.

An accurate prediction obeys:

$$\begin{cases} P_p^k \subseteq (P - P_i^k) \\ P_{p,c}^k \subseteq (P_c^k - P_{i,c}^k) \\ P_{p,o}^k \subseteq (P_o^k - P_{i,o}^k) \end{cases} \quad (3)$$

Several factors might undermine the prediction accuracy. First, web page content might gradually change. Second, collaborator's taste and judging standard might change as well. Therefore heuristically only fresh ratings should be used for prediction. Another factor is dishonest ratings that violate the Honest Rating Rule. This is a common problem in collaborative rating systems, and we discuss it later.

B. OWP Prediction Algorithm

In this subsection, we give an algorithm that can be used by the predictor. For collaborator k, the attribute value of any page in P_c^k is lower than P_o^k. So the basic heuristic is that there is an offensive threshold T_k between attribute values of pages in P_c^k and. Any page with attribute value above T_k will be taken by collaborator k as offensive. To estimate T_k, we have to estimate pages' attribute value which is quantified by page's *Global Offensive Ratio* (*GOR*) approximately.

Definition 1 *GOR* is the ratio of offensive ratings in all ratings to a page.

Given enough honest ratings, *GOR* of a page will approximate the page's attribute value with high probability. It is helpful to set a rating threshold T_R. Only pages who receive more than T_R ratings are considered into the prediction algorithm.

For collaborator k, the pages in P_o^k can be ranked by *GOR*. The smallest value GOR_{MIN} can be used as k's indirect offensive threshold. However, random errors may exist. Two methods can be used limit random errors. First, a *slow-start threshold* T_{SS} could be set. A collaborator will receive predictions only after it has made more than T_{SS} ratings. We will discuss this in Section IV. Second, random error can be further dampened by substituting GOR_{MIN} with another parameter as the indirect offensive threshold, *Local Offensive Bottom* (*LOB*).

Definition 2 Collaborator k's *Local Offensive Bottom* (LOB_k) is the $\lceil 0.05n \rceil^{th}$ smallest *GOR* in $P_{i,o}^k$, $n = |P_{i,o}^k|$. When $P_{i,o}^k = \varnothing$, $LOB_k = 1$, $NOOF_k$=*TRUE*.

The heuristic is effective to eliminate false positives when there are less than 5% random errors, but it is weak when large-scale dishonest ratings exist. To avoid false positives resulted from this, a check should be made before prediction to see if the collaboration network is trustable. We define another parameter *Local Clean Top* (*LCT*):

Definition 3 Collaborator *k's Local Clean Top* (LCT_k) is the $\lceil 0.05n \rceil^{th}$ largest *GOR* in $P_{i,c}^k$, $n = |P_{i,c}^k|$. When $P_{i,c}^k = \varnothing$, $LCT_k = 0$, $NOCL_k$=*TRUE*.

A natural check is that for any collaborator *k* LOB_k should be larger than LCT_k. We call this *Trust Check*. The trust check enables collaborators to launch self-protection when needed. Though it can not result in ZERO false positive, it does greatly reduce that especially when large-scale dishonest collaborators exist. Its effect will be shown in the evaluation part. As a result, we have the following prediction rule ($P_i = \bigcup_{k=1}^{n_u} P_i^k$):

Prediction Rule Given collaborator *k* and page j ($j \in P_i \&\& j \notin P_i^k$): 1) if ($NOCL_k$==*TRUE*), j is offensive to k; 2) if($LOB_k > LCT_k$) and $GOR_j > LOB_k$, *j* is offensive to *k*.

C. Attack Analysis

Our collaboration model and prediction algorithm predict potential offensive pages for collaborators based on their past ratings. Malicious collaborators might try to exploit them to achieve selfish goals. In this paper, we consider two basic attack models.

Ballot Stuffing Attackers make exaggeratedly positive ratings to selected websites. No matter whether those pages are offensive to them or not, they just make clean ratings.

Bad Mouthing Attackers make exaggeratedly negative ratings to selected websites. Contrary to Ballot Stuffing, they just make offensive ratings.

For generality, we suppose both the two kinds of attackers exist. Accordingly, collaborators are divided into three groups, the honest group (*HG*), the ballot stuffing group (*BSG*), and the bad mouthing group (*BMG*). The upper attack models ensure that the two dishonest groups have no page-dependent behaviors. A group can not launch ballot stuffing to some pages but bad mouthing to others. Suppose the three groups' ratios are denoted by r_1, r_2, and r_3, which satisfy $r_1+r_2+r_3=1$, and page's *Partial Offensive Ratio* (*POR*) in the three groups are denoted by POR_H, POR_{BS}, and POR_{BM}. Then:

$$GOR = r_1 \times POR_H + r_2 \times POR_{BS} + r_3 \times POR_{BM} \quad (4)$$

According to the attack model, $POR_{BS}=0$, $POR_{BM}=1$, then:

$$GOR = r_1 \times POR_H + r_3 \quad (5)$$

According to (5), in any attack scenario, *GOR* statistically depends on honest rating. Since user's preference is calculated from *GOR*, the increase of *GOR* will cause the increase of the latter. Thus, the affection to prediction accuracy is small. However, when dishonest collaborators overtake most of the collaboration network, there might be too much random error which would undermine prediction accuracy. We will evaluate this later by simulation.

In conclusion, our model and algorithm have good resilience to *Ballot Stuffing*, and *Bad Mouthing* attacks, even in a colluding way.

IV. EVALUATION

In this section, we evaluate our collaboration model and prediction algorithm by a discrete-event simulator.

A. Setting, Metric and Method

Since the collaboration model and OWP prediction algorithm work in the same way in different collaboration domains, we limit our simulations within one domain. We simulate three typical types of websites. The first are BBS-like forums, in which some pages are contaminated. The second are news websites, in which pages are often uncontaminated. The last are porn websites, in which most pages are contaminated.

Each simulation lasts for 50 days. Each hour 20 new web pages are added into the website. Fresh pages are more likely to be visited. The probability that a page added in the latest i^{th} 24-hour (day) is selected is twice that of a page added in the latest $(i+1)^{th}$ 24-hour (i=1, 2, …). Since pages added 10 days ago have a very low probability to be visited, we *set page lifetime* as 10 days. Each page's attribute value stays the same in a simulation process.

In a day, each collaborator continuously visits 15 pages (according to Alexa [8]). It takes 1 minute to read a page. The start time of one day's browsing is randomly distributed. Each collaborator does not visit the same web page twice in a day. Fig.2 and Fig.3 respectively describe the honest and dishonest rating behavior. When an honest collaborator is warned that a page is offensive, he will not read the page. But the system will automatically make an offensive rating. Collaborators' *LOB* and *LCT* are updated every hour.

```
1:  Select a page;
2:  Query the page's GOR;
3:  Judge if it is offensive by the Prediction Rule;
4:  if(offensive)
5:      Make an offensive rating;
6:      continue;
7:  else
8:      Read the page;
9:      if(the page is offensive)
10:         Make an offensive rating;
11:     else
12:         Make a clean rating;
13:     end if;
14: end if
15: Goto 1;
```

Fig.2 Honest Collaborator Behavior

```
1:  Select a page;
2:  Read the page;
3:  if (this user belongs to BSG)
4:      Make a clean rating;
5:  else if (this user belongs to BMG)
6:      Make an offensive rating;
7:  end if
8:  Goto 1;
```

Fig.3 Dishonest Collaborator Behavior

We use two metrics, *Prediction Efficiency (PE)* and *False Positive (FP)*. Let no denotes the overall times that honest collaborators visit an offensive page, n_{cp} denotes the overall times that offensive pages are correctly predicted to honest collaborators, n_p denotes the overall times that predictions are made to honest collaborators, and n_{fp} denotes the overall times these predictions are false. Then: $PE=n_{cp}/n_o, FP=n_{fp}/n_p$.

Prediction Efficiency is used to evaluate how well our collaboration model and algorithm can protect web browsing, while False Positive is used to evaluate prediction accuracy. Note that a page will be predicted only after it has received more than T_R ratings. This threshold reduces random errors, but also undermines prediction efficiency. Let n_{o*} denotes the overall times that honest collaborators visit an offensive page before the page has received T_R ratings. We define a metric to model the rating threshold's influence: $PE*=n_{cp}/(n_o-n_{o*})$.

Similarly, the Trust Check might also undermine prediction efficiency. When this check fails, honest collaborators will reject predictions. Let n_{o**} denotes the overall times that honest collaborators visit an offensive page when the Trust Check fails. We define a metric to model the Trust Check's influence: $PE**=n_{cp}/(n_o-n_{o*}-n_{o**})$.

Our simulation has two goals. One is to study how the *rating threshold* T_R and the *slow-start threshold* T_{SS} will influence system performance. The other is to investigate how well our model and algorithm can adapt to various deploying scenarios.

B. Results

Fig.4 illustrates a dynamics simulation process. Each point denotes a metric (*PE, PE*, PE***, or *FP*) value in a day.

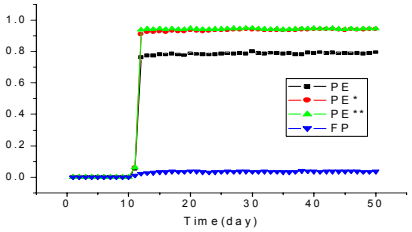

Fig.4 A Simulation Process

In the first 10 days of simulation, no predictions are made due to *slow start threshold*. From the 11th day, the collaboration system quickly converges to a stable state. This happens in all simulations. So in later figures, all metrics use the stable values (the average value of the last 40 days), except for special explanations. Default parameters in our simulations: Page attribute and T_k are randomly distributed within [0, 1]; n_u=10000, T_R=50, T_{SS}=150.

BBS-like Website

Collaborator Scale. We change the collaborator scale from 2500 to 40000. The results are shown in Fig.5. As the scale increases, *PE* is most significantly improved because n_{o*}/n_o decreases fast.

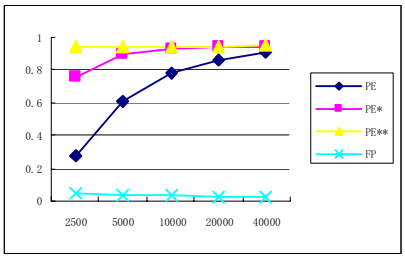

Fig.5 Performance under different collaborator scales.

When the collaboration scale is small, *(PE*-PE)* and *(PE**-PE*)* are big, showing that TR and the Trust Check induce much efficiency loss. The latter does so because in this case there are more random errors, which make the Trust Check fails. This shows that our prediction algorithm can work best for *hot* pages. *FP* also decreases slightly with scale increment, but it remains low (<5%) all through.

T_k Distribution. We further analyzed the result at point 10000 in Fig.5. We classify collaborators into 10 ranks by the range of T_k, as shown by the horizontal axis in Fig.6. As T_k becomes larger, *PE* almost remains unchanged, but *FP* increases especially after 0.6. This is because larger T_k means smaller peer group which shares the same offensive pages, and in turn induces larger random error. The number of pages false predicted is still low because no is small. But when fairness is concerned, our model seems to be biased towards collaborators with lower T_k. We run another group of simulations in which all T_k are randomly distributed within a narrower range. In this case, *FP* is greatly reduced for all T_k ranges, P_E almost keep unchanged as Fig.6. The comparison shows that our model can work better in a more homogeneous collaborator community.

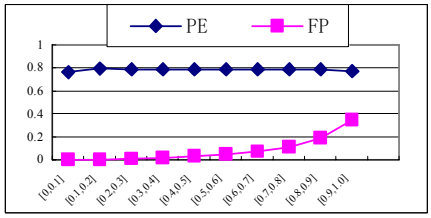

Fig.6 Performances for different collaborators.

Attack Resilience. We independently change the ratio of ballot stuffing and bad mouthing collaborators from 0% to 90%, and run a group of simulations for each attack. The results are given in Fig.7 and Fig.8.

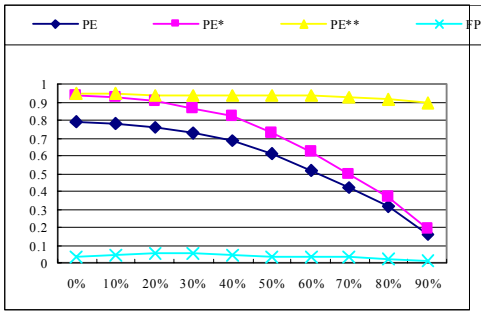

Fig. 7 Performances with Ballot Stuffing attack.

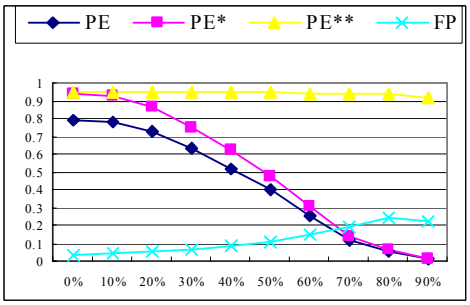

Fig.8 Performances with Bad Mouthing attack.

In both cases, *PE* gradually decreases, but faster in the latter. We think the decrease is due to smaller peer groups which share the same offensive pages. Even so, we can still obtain a 60%, 40% gain at the ratio 50% in the two cases. *Ballot stuffing* has little influence on *FP*. But *bad mouthing* induces larger *FP*. In both attacks, (*PE***-*PE**) is very large, showing that *the Trust Check* plays a significant role in reducing *FP*.

Slow Start. We change T_{SS} from 15 to 150 and run two groups of simulations. In both cases, T_{SS} has only very slight influences on *PE* and *FP*.

FP Damping. We change T_R from 5 to 150. When T_R is larger than 25, *FP* is under 5%.

News Website

Page attribute is set as randomly distributed within [0, 0.1]. We run the same groups of simulations as BBS-like Website and find similar results. We pay special attention to the performance under bad mouthing attack. When bad mouthing attackers increase to 90%, both *PE* and *FP* decrease close to 0. The attack goal is to achieve more false positives, however, it fails to do so.

Porn Website

Page attribute is set as randomly distributed within [0.9, 1]. We pay special attention to the performance under ballot stuffing attack. Fig.9 shows the results. PE only slightly decreases as the ratio increases. So the attack fails.

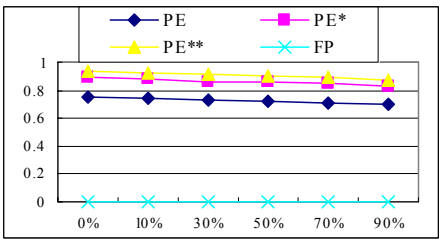

Fig.9 Performances with Ballot Stuffing attacks

V. CONCLUSION

In this paper, we proposed a collaboration system to detect user-specific OWPs. We evaluated our collaboration model and algorithm by analysis and simulation, and got the following conclusions. 1) The model and algorithm can accurately predict OWPs for users of different tastes and judging standards when most collaborators are honest. 2) The model and algorithm have good resilience to Ballot Stuffing, and Bad Mouthing attacks. Even when the attacker ratio achieves 50%, they can still obtain 60%, and 40% prediction efficiency, while at the same time keep false positive low. We also find that the larger the collaboration scale, the harder the colluding attack is.

ACKNOWLEDGMENT

This research is supported by Natural Science Foundation of China under No. 90604024, the Key Project of Chinese Ministry of Education under No. 106012 and NCET, and the 863 project under No. 2006AA01Z209.

REFERENCES

[1] Herlocker, J., Konstan, J., Terveen, L., and Riedl, J. Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems 22 (2004), ACM Press, 5-53.

[2] Zhou F., Zhuang L., Zhao B.Y., Huang L., Joseph A.D., and Kubiatowicz J.. Approximate object location and spam filtering on P2P systems. In: Proc. USENIX Middleware Conf., Rio de Janeiro, Brazil, 2003, 1–20.

[3] Kevin Walsh and Emin Gun Sirer. Experience with an Object Reputation System for Peer-to-Peer Filesharing. NSDI 2006.

[4] C. Dellarocas. Immunizing Online Reputation Reporting Systems Against Unfair Ratings and Discriminatory Behavior. In ACM Conference on Electronic Commerce, pages 150.157, 2000.

[5] M. Chen and J. Singh. Computing and Using Reputations for Internet Ratings. In Proceedings of the Third ACM Conference on Electronic Commerce (EC'01). ACM, October 2001.

[6] F. Cornelli et al. Choosing Reputable Servents in a P2P Network. In Proceedings of the eleventh international conference on World Wide Web (WWW'02). ACM, May 2002.

[7] F. Cornelli, E. Damiani, and S. D. Capitani. Choosing reputable servents in a p2p network. In Proc. of the Eleventh International World Wide Web Conference, 2002.

[8] http://www.alexa.com/