Artif Intell Rev (2007) 28:69–86
DOI 10.1007/s10462-008-9075-4

Collaborative web search: a robustness analysis

Michael P. O'Mahony · Barry Smyth

Published online: 13 September 2008

Abstract Collaborative web search utilises past search histories in a community of like-minded users to improve the quality of search results. Search results that have been selected by community members for past queries are promoted in response to similar queries that occur in the future. The I-SPY system is one example of such a collaborative approach to search. As is the case with all open systems, however, it is difficult to establish the integrity of those who access a system and thus the potential for malicious attack exists. In this paper we investigate the robustness of the I-SPY system to attack. In particular, we consider attack scenarios whereby malicious agents seek to promote particular result pages within a community. In addition, we analyse robustness in the context of community homogeneity, and we show that this key characteristic of communities has implications for system robustness.

Keywords Collaborative web search · Personalisation · Malicious attack · Robustness

1 Introduction

Traditional search engines operate in a "one-size-fits-all" manner and return the same search results for the same query, irrespective of the interests and needs of those individuals using the service. The rationale for collaborative web search (CWS) lies in the query repetition and selection regularity that exists in the searches of communities of like-minded users. The I-SPY system described by Smyth et al. (2004) is one example of such a collaborative approach to search. I-SPY is a meta-search framework where submitted queries are first passed to base-level search engines (e.g. Google, Yahoo! etc.). The result lists obtained are

M. P. O'Mahony (✉) · B. Smyth
School of Computer Science and Informatics,
University College Dublin,
Dublin, Ireland
e-mail: michael.p.omahony@ucd.ie

B. Smyth
e-mail: barry.smyth@ucd.ie

then adapted and re-ranked by I-SPY according to the heuristic that results which have been selected for similar queries in the past are also likely to be relevant for other community members. In Smyth et al. (2004, 2005), the benefits of this collaborative approach to search has been established.

The relatively uniform interests and needs that exist in community domains can also, however, be exploited by malicious agents seeking to manipulate search output. For example, users may seek to promote their own work to other community members in order to enhance their reputation or to achieve financial gain. The lack of robustness exhibited by collaborative recommender systems against attack has been highlighted by Lam and Riedl (2004), Mobasher et al. (2005) and O'Mahony et al. (2005). It has been shown that attacks on these systems are capable of significantly biasing the recommendations that are made for target items. All collaborative systems are vulnerable given the manner in which they operate, i.e. they depend on user supplied input in order to filter information or to make suggestions. Since it is practically impossible to assess the integrity of those who use a system, there is no guarantee that the data inserted into a system's database is an accurate reflection of true user preferences and therefore the possibility of biased data being deliberately inserted into system databases exists.

In this paper we examine the robustness of the I-SPY system against attack. We introduce attack models that are designed to associate target result pages with the query-space of particular communities, with the objective of promoting such pages in the result lists of future search sessions initiated by community members. In addition, we analyse the robustness of collaborative search in terms of community homogeneity. This is a key characteristic of communities and depends on such factors as the maturity of communities, user participation levels and community focus. Some communities will naturally have a tight focus (e.g. communities catering for such niche-interests as military history, model aviation etc.) while others will have a broader focus. By definition, all communities strive for a certain degree of speciality in order to satisfy members' interests and needs and it is clearly important to gain an understanding of the relationship between community focus and robustness against attack.

The paper is organised as follows. Section 2 presents an overview of the I-SPY system architecture. Attack models are described in Sect. 3 and an empirical analysis of these models is provided in Sect. 4. Related work dealing with the robustness of other collaborative systems is discussed in Sect. 5. Finally, conclusions and scope for future work are presented in Sect. 6.

2 Collaborative web search

In this section, we provide a brief overview of the I-SPY system architecture and illustrate system functionality by showing example output for a particular community of users. For a more comprehensive description of I-SPY, which is outside the scope of this work, please refer to Coyle and Smyth (2007) and Smyth et al. (2003, 2004).

2.1 Collaborative web search architecture

The collaborative search technique implemented in the I-SPY system is conceived of as a form of meta-search. Each new user query, q_T, is submitted to a set of underlying search engines and their results are combined to form a meta-search result-list, R_M. I-SPY then processes R_M to produce a new result-list, R_T, which reflects the learned preferences of a community of like-minded users. This is achieved by recording the page selections made for past search queries, thereby enabling such pages to be promoted when similar queries are submitted in future.

Briefly, the I-SPY algorithm proceeds as follows. The key data structure used in the system is the community *hit-matrix*, H, where H_{ij} represents the number of times that page p_j has been selected as a result for query q_i. Thus, the rows of H correspond to the (unique) queries submitted by community members and maintain an account of all page selections made for each query. Note that no record is maintained of which user selected which page; in effect, the hit-matrix serves as an anonymous account of community preferences.

When a new query, q_T, is submitted to the system, I-SPY computes the similarity between q_T and all queries, q_i, in the hit-matrix. Query similarity is computed using the Jaccard index (Eq. 1), which measures the term overlap between queries. I-SPY then selects a subset of the most similar queries to q_T to form a set of *candidate queries*. In this paper, we consider two query selection strategies. In the first strategy, we select those queries whose similarity to q_T exceeds a particular threshold; we refer to this approach as *threshold selection*. The second strategy is a nearest-neighbour approach and involves selecting the k queries from the hit-matrix with the highest (non-zero) similarity to q_T; we refer to this strategy as *best-k selection*.

$$\mathrm{Sim}(q_T, q_i) = \frac{|q_T \cap q_i|}{|q_T \cup q_i|} \tag{1}$$

The pages associated with selected candidate queries are called *promotion candidates*. These pages are assumed to be relevant to the new query because they have been relevant for similar queries, and for the same community of searchers, in the past. The next step in the I-SPY algorithm is to compute the relevance of each of the promotion candidates to the new query, q_T.

The relevance of page p_j to query q_i is calculated as the relative number of times that p_j has been selected for q_i (Eq. 2). Further, the relevance of p_j to q_T is a combination of $\mathrm{Relevance}(p_j, q_i)$ for all candidate q_i's $(q_1, \ldots, q_n)$ deemed similar to q_T, as shown in Eq. 3, where $\mathrm{Exists}(p_j, q_i) = 1$ if $H_{ij} > 0$, and 0 otherwise. Each $\mathrm{Relevance}(p_j, q_i)$ is weighted by $\mathrm{Sim}(q_T, q_i)$ to discount the relevance of results from less similar queries.

$$\mathrm{Relevance}(p_j, q_i) = \frac{H_{ij}}{\sum_{\forall j} H_{ij}} \tag{2}$$

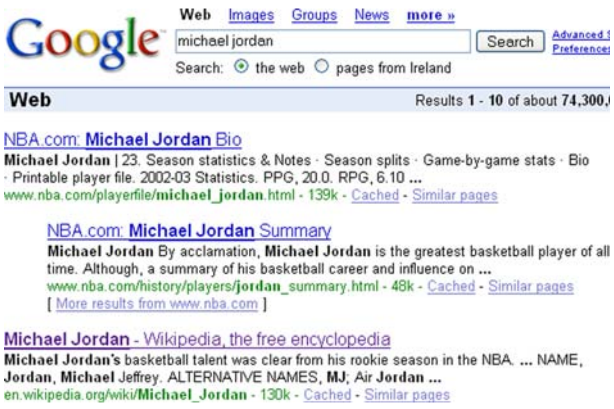
$$W\mathrm{Rel}(p_j, q_T, q_1, \ldots, q_n) = \frac{\sum_{i=1\ldots n} \mathrm{Relevance}(p_j, q_i) \times \mathrm{Sim}(q_T, q_i)}{\sum_{i=1\ldots n} \mathrm{Exists}(p_j, q_i) \times \mathrm{Sim}(q_T, q_i)} \tag{3}$$

This weighted relevance metric is used to rank-order the promotion candidates. These ranked pages are then listed ahead of the remaining meta-search results, which are themselves ranked according to a standard meta-search scoring metric, to give R_T.

2.2 An example session

Figure 1 shows the results of a typical search for the query *"michael jordan"* using the Google search engine. As can be seen, the first three results all relate to Michael Jordan, the renowned American basketball player, which is not surprising given the high profile of this particular individual.

Searchers from academic backgrounds, however, may be more interested in another Michael Jordan—the professor at the University of California, Berkeley. This is the fundamental issue with with the "one-size-fits-all" manner of operation, in that the same results are


Web Images Groups News more »

michael jordan Search Advanced S Preferences

Search: ⦿ the web ○ pages from Ireland

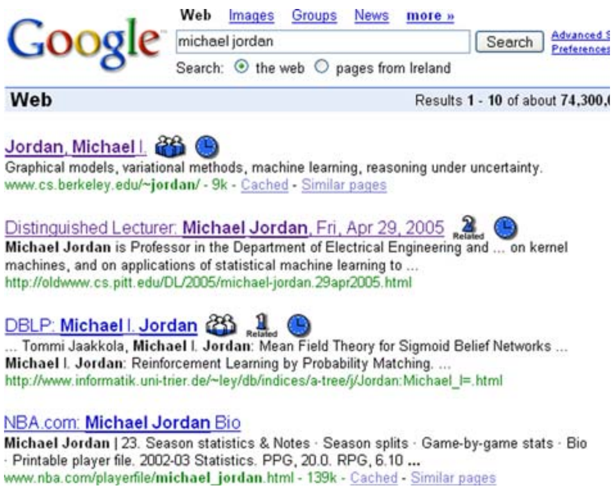
Web Results 1 - 10 of about 74,300,

NBA.com: Michael Jordan Bio
Michael Jordan | 23. Season statistics & Notes · Season splits · Game-by-game stats · Bio · Printable player file. 2002-03 Statistics. PPG, 20.0. RPG, 6.10 ...
www.nba.com/playerfile/michael_jordan.html - 139k - Cached - Similar pages

NBA.com: Michael Jordan Summary
Michael Jordan By acclamation, Michael Jordan is the greatest basketball player of all time. Although, a summary of his basketball career and influence on ...
www.nba.com/history/players/jordan_summary.html - 48k - Cached - Similar pages
[More results from www.nba.com]

Michael Jordan - Wikipedia, the free encyclopedia
Michael Jordan's basketball talent was clear from his rookie season in the NBA. ... NAME, Jordan, Michael Jeffrey. ALTERNATIVE NAMES, MJ; Air Jordan ...
en.wikipedia.org/wiki/Michael_Jordan - 130k - Cached - Similar pages

Fig. 1 The Google result page for the *"michael jordan"* query


Web Images Groups News more »

michael jordan Search Advanced S Preferences

Search: ⦿ the web ○ pages from Ireland

Web Results 1 - 10 of about 74,300,

Jordan, Michael I.
Graphical models, variational methods, machine learning, reasoning under uncertainty.
www.cs.berkeley.edu/~jordan/ - 9k - Cached - Similar pages

Distinguished Lecturer: Michael Jordan, Fri, Apr 29, 2005 2 Related
Michael Jordan is Professor in the Department of Electrical Engineering and ... on kernel machines, and on applications of statistical machine learning to ...
http://oldwww.cs.pitt.edu/DL/2005/michael-jordan.29apr2005.html

DBLP: Michael I. Jordan 1 Related
... Tommi Jaakkola, Michael I. Jordan: Mean Field Theory for Sigmoid Belief Networks ... Michael I. Jordan: Reinforcement Learning by Probability Matching. ...
http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/j/Jordan:Michael_I=.html

NBA.com: Michael Jordan Bio
Michael Jordan | 23. Season statistics & Notes · Season splits · Game-by-game stats · Bio · Printable player file. 2002-03 Statistics. PPG, 20.0. RPG, 6.10 ...
www.nba.com/playerfile/michael_jordan.html - 139k - Cached - Similar pages

Fig. 2 The I-SPY result page for the *"michael jordan"* query

presented for queries irrespective of the interests and tastes of those conducting the search. The benefit of collaborative Web search is that it allows results to be re-ranked based on community context. The output of the I-SPY system for a computer-science based community (a test community established at the School of Computer Science and Informatics, University College Dublin, Ireland) for the same query is shown in Fig. 2. As can be seen, the Web page of the professor is now ranked first (it is currently ranked 8th on Google). This re-ranking was made possible by analysing the past query-result patterns of community members, and thus the top-3 listed results refer to the professor, while the 4th result relates to the former basketball star. This example illustrates the power of collaborative Web search, where useful re-orderings of search results can be achieved in order to suit the particular needs and tastes of community users.

3 Attack models

We assume that the objective of attacks is to promote a single target page within a particular I-SPY community. For example, consider the author of a Web page related to the Java programming language, who seeks to promote his page above others for all relevant search sessions. In order to ensure successful promotions in the context of the I-SPY system, the key requirement for attackers is to associate the target page in the system's hit-matrix with queries that are representative of community search activity. We refer to such queries as *attack queries*.

To illustrate the attack process, consider an attack that seeks to promote a particular target page, p_T. Since I-SPY updates the community hit-matrix following all search activity (i.e. query–result page selections), attackers can insert any number of attack queries into the system by simply selecting p_T from the result-lists returned from multiple search sessions. Each search session can involve different combinations of query terms,[1] thereby permitting the target page to be associated with a diverse range of queries within the hit-matrix.

By definition, given the focused nature of typical communities, the selection of suitable terms with which to form attack queries should not pose a significant challenge. For example, queries formed using terms such as *php*, *mysql*, *C++* etc. would be good candidates in the case of a technology-related community. By using such popular or frequently occurring terms, high similarities (Eq. 1) between new search queries submitted by community members and attack queries can be achieved, thereby increasing the weighted relevance and ranking (Eq. 3) of the associated target page in the search results.

Note that the effort involved in implementing attacks is unlikely to be high, since the insertion of queries into the I-SPY system can be readily automated, thereby enabling a large number of such insertions using many combinations of query terms to be made. Further, the fact that I-SPY maintains an anonymous database of community preferences is beneficial for attackers since checks on individual user activity are consequently not possible. Thus, in this paper, we consider attack *cost* solely as a function of the number of queries that are inserted in the course of an attack, and the number of terms that are present in each.

We propose two attack models which are described in the following sections. Initially, we assume that full knowledge concerning the distribution of query term usage in a community is known to attackers. In Sect. 4, we also analyse the effectiveness of attacks when full query term distribution knowledge is not available to attackers, by simulating varying degrees of such knowledge on the part of attackers.

3.1 Promote-always attack

This attack model seeks to promote a particular target page for all searches submitted by community members, irrespective of the actual search terms that are used. We define attack query size, or the number of terms included in attack queries, as l. In order to ensure maximal coverage of the community query-space, the most frequently used search terms are selected for attack queries. If multiple attack queries are inserted, each consists of a different combination of query terms. The first attack query consists of those l terms with the highest combined frequency of occurrence, the second attack query consists of those l terms with the second-highest combined frequency of occurrence, etc.

[1] We assume that the target page, p_T, contains all attack query terms so that p_T is indexed by such terms by the underlying search engines.

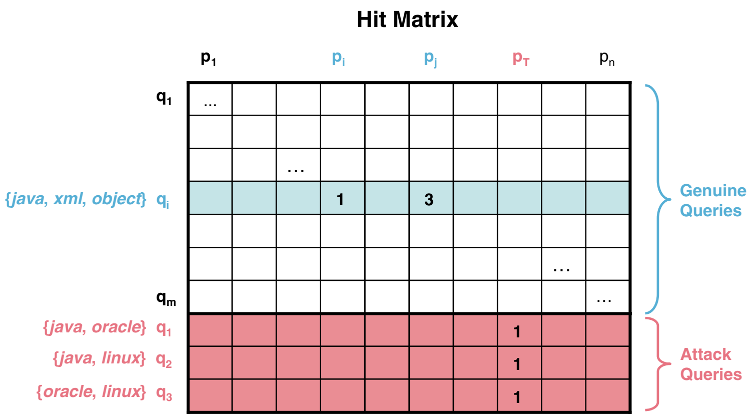


Fig. 3 A sample community hit-matrix showing genuine and attack queries. where the target page (p_T) has been selected for each of the attack queries

3.2 Term-specific attack

The objective of this attack is to promote a particular target page when a specific term is used in search queries. Attack queries are comprised of the specific term in question, along with combinations of $l - 1$ other terms which are selected so as to optimally cover the query-space as described above.

3.3 Example attacks

To illustrate the approach adopted in these attacks, consider the hit-matrix shown for a particular community in Fig. 3. Queries q_1 to q_m represent genuine user queries and the result selections for one particular query, $q_i = \{java, xml, object\}$, are shown. For this query, we can see that page p_i has been selected on one occasion and page p_j has been selected on three occasions.

Consider a promote-always attack which involves the creation of $n = 3$ attack queries of size $l = 2$. In the approach adopted in this paper, we first compute the number of different terms that are required to create the desired attack. In this case, we need 3 terms since ${}^nC_l = {}^3C_2 = 3$. Suppose that *java*, *oracle* and *linux* were the three most frequently occurring query terms (in descending order) in this particular community. Thus, the attack queries would consist of the following combinations of these terms: {*java*, *oracle*}, {*java*, *linux*} and {*oracle*, *linux*}. These attack queries are also shown in the figure, where for each, only one page—the target page (p_T)—has been selected.

In the case of a term-specific attack, where, for example, the attacker wished to promote the target page for all queries containing the term *mysql*, the following approach is employed. Assuming, as before, that *java*, *oracle* and *linux* were the three most frequently occurring query terms in the community and that 3 attack queries of size $l = 2$ were to be created, the attack queries this time would be {*mysql, java*}, {*mysql, oracle*} and {*mysql, linux*}.

When a new query, q_T, is submitted to the system, I-SPY first computes the similarity between q_T and all queries contained in the hit-matrix. Since attack queries are comprised of the most frequently occurring community terms, the likelihood of term overlap (i.e. similarity) between these queries and q_T is relatively high. Recall from Sect. 2.1 that the most similar queries to q_T are selected as candidate queries; if at least one attack query is included in the candidate set, then the target page will be promoted by the system in response to q_T (the exact ranking of each promoted page is governed by Eq. 3).

4 Evaluation

We begin by providing details on the configuration parameters used in the I-SPY system, the evaluation dataset and the experimental methodology and metrics that were used to evaluate system robustness.

4.1 I-SPY configuration

I-SPY was configured to display a maximum of 10 search results on each results screen, of which at most 3 were promoted results. For example, if the system was able to make 5 promotions for a particular search query, the top-3 promotions would be displayed at the top of the first screen and the remaining 2 would be displayed at the top of the second screen.

The following parameters were used for I-SPY's query selection strategies (Sect. 2.1). In the case of the threshold selection strategy, a query-similarity threshold of $th = 0.5$ was used. The motivation for this relatively high threshold value is to avoid spurious matches between unrelated queries; for example the queries $\{$*jaguar*, *automobiles*$\}$ and $\{$*jaguar*, *wildlife*$\}$ have a similarity of 0.33 but are clearly unrelated. This particular threshold value of 0.5 was found to provide good performance by Smyth et al. (2005).

For best-k selection, we experiment with retrieving various numbers of the most related queries. Following the example of Smyth and Balfe (2006), we consider the following values of k: 1, 5, 10, 15 and 20.

4.2 Dataset

The dataset used in our evaluations was obtained from a trial of the I-SPY system which took place over an extended period among the 50 staff members of a local software company. I-SPY was configured to draw on Google and HotBot as a source of search results. A new community hit-matrix was created for participants, which was trained on search log data prior to the start of the trial.

The dataset consists of 4,744 queries, of which 2,256 were unique, and associated result pages (URL's), of which 3,587 were unique. The total number of unique query terms in the dataset was 2,673.

4.3 Metrics and methodology

For both attack models, the objective was to promote a new target page which was not already present in the community hit-matrix. Consequently, no promotions were possible for the target page pre-attack. We evaluate the effectiveness of attacks using the *hit ratio* metric proposed by Mobasher et al. (2005), which measures the percentage of times that a target page appears in top-N promotions. Since I-SPY includes at most 3 promotions on each results page (Sect. 4.1), we present hit ratio results for $N = 3$, 6, 9 and 12, which measures the percentage of times that the target page was promoted to the 1st, 2nd, 3rd or 4th page of search results, respectively. (Of course, from an attacker's perspective, the promotion of the target page to the first screen of search results (top-3) represents the greatest attack success).

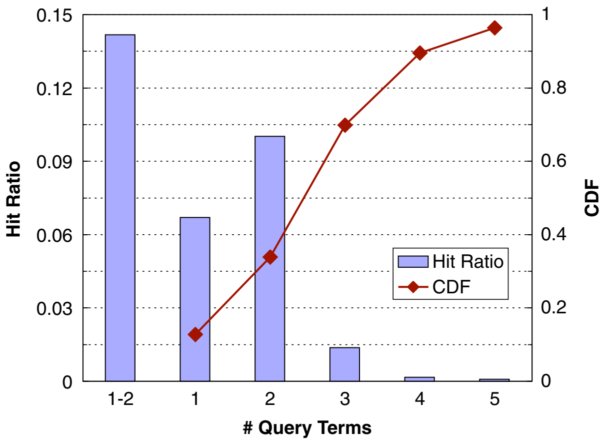


Fig. 4 Hit ratios achieved for the term-specific attack model versus attack query size. The cumulative distribution function (CDF) of genuine query sizes is also shown

The following experimental methodology was employed. For the promote-always attack, test and training sets consisting of randomly selected percentage of queries and associated result selections were drawn from the dataset. Attack queries were added to the training set and the average hit ratio for the target page was calculated over all test set queries. Test set size was 10% and a 10-fold cross validation was performed.

In the case of the term-specific attack involving a particular specific term, t, the test set consisted of all dataset queries that contained term t. These test set queries were submitted in turn to the full dataset (with attack queries inserted), and the average hit ratio for the target page was calculated. This procedure was then repeated for all the unique query terms contained in the dataset, and the average hit ratio over all such terms was calculated. Note that, with this approach, test set sizes are not constant across terms since the frequency of occurrence of each term in the dataset will vary. In Sect. 4.5, we consider the effect of the popularity of term t on the hit ratio achieved by the attack.

4.4 Attack query size versus robustness

We first examine the effect of attack query size on robustness. Top-3 hit ratio results are shown in Fig. 4 for the term-specific attack using threshold selection, for a fixed quantity (12) of attack queries inserted. The highest hit ratios were achieved for query sizes of $l = 1$ or $l = 2$ terms, with the greatest effect observed when a 50–150% combination of these query sizes was used (labeled "1–2" in the figure). Attack queries formed using 3 or more terms had little effect. The cumulative distribution function of genuine query sizes is also shown in Fig. 4, which shows that 70% of queries consisted of three terms or less. (Genuine queries consisted of three terms on average, with a standard deviation of 1.3. The mode was also three.) Given the relatively stringent overlap criteria imposed by the high query-similarity threshold value of $th = 0.5$, attack queries consisting of smaller combinations of frequently occurring terms had a greater probability of exceeding the similarity threshold with test (genuine) queries. Similar results were found for best-k query selection and also for the promote-always attack model. In the remainder of this section, attacks are implemented using the combination of query sizes discussed above.

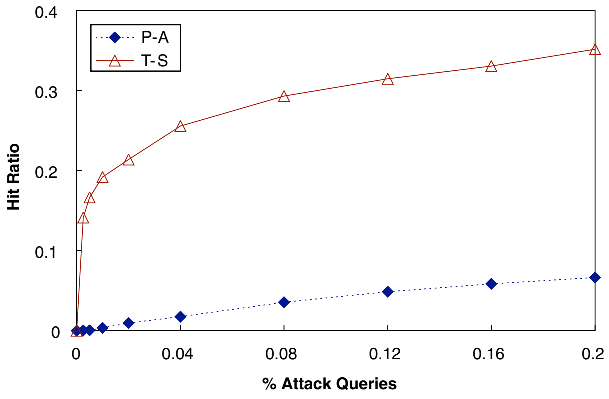


Fig. 5 Top-3 hit ratio results using threshold selection achieved for the promote-always (P-A) and term-specific (T-S) attack models versus the quantity of attack queries inserted

4.5 Attack models versus robustness

We now compare the performance of the two attack models at various attack sizes (expressed as a percentage of the number of attack queries inserted to the total number of genuine queries present in the dataset). Top-3 hit ratio results using threshold selection are shown in Fig. 5. In the case of the promote-always, a hit ratio of only 7% was achieved when the largest quantity (20%) of attack queries was inserted. Given the frequency of occurrence of query terms in the dataset follows a Zipfian distribution (i.e. only a small number of terms appear frequently in search queries; see Fig. 10), attack queries consisting of even the most-frequently occurring combinations of query terms failed to achieve similarities in excess of the threshold value (0.5) with test queries and thus, the attack was not successful in promoting the target page.

In contrast, the term-specific attack performed significantly better, achieving hit ratios of 14% at the smallest attack size (0.25%, or 12 attack queries inserted) and 35% at the largest attack size (20%). The improved performance of this attack model was to be expected, given its more focused nature, where promotions were attempted only in cases where a particular target term was used in search sessions. Thus, the task of achieving attack-genuine query similarities in excess of the threshold was made easier, since all test and attack queries had at least one term (the target term) in common.

The attack was equally successful in promoting pages for target terms that were used either frequently or infrequently by genuine users in search queries. For example, the most frequently used term in the dataset was *java*, which was present in 411 (genuine) queries. The hit ratio achieved when *java* was used as the target term was 35.5%. By way of contrast, there were 31 dataset terms which occurred in 10 different queries. The average hit ratio achieved across these 31 terms was 32.3%. Thus, irrespective of the popularity of the target term selected by the attacker, the target page was promoted into the top-3 results in >30% of searches which included that term.

The term-specific attack was also very straightforward to implement since 50% of the attack queries created consisted of only 1 term—the target term, with the remaining 50% consisting of combinations of the target term plus only a single additional term. Thus, we can conclude that I-SPY was indeed vulnerable to this attack, given that small numbers of easily created attack queries were capable of significantly biasing the output of the system. In addition, these findings have significance from the perspective of attack detection, where the correct classification of relatively small quantities of attack data is likely to prove problematic.

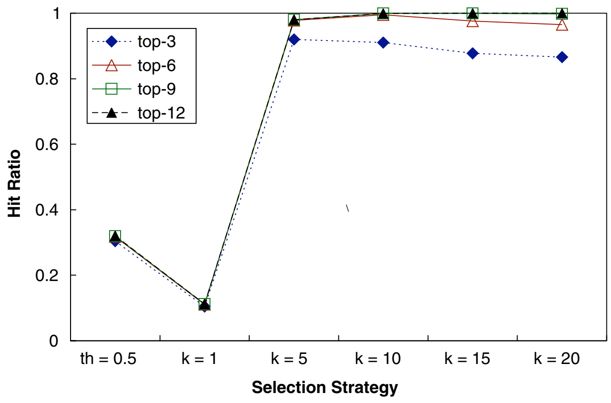


Fig. 6 Hit ratio results achieved for the term-specific (T-S) attack model using threshold and best-k query selection strategies

4.6 Varying query selection strategies

In this section, we analyse the robustness against attack in the context of the two query selections strategies that have been typically employed in I-SPY implementations—namely, threshold and best-k selection. Figure 6 shows results for the term-specific attack model for a fixed quantity (attack size = 10%) of attack queries inserted into the system. Not surprisingly, best-1 selection, where only the single, most similar candidate query was selected, provided the greatest degree of robustness against attack (hit ratio = 0.11), followed by threshold selection (hit ratio = 0.31). Thereafter, best-k selection for $k > 1$, where successively greater numbers of candidate queries were selected (and thereby increasing the likelihood of at least one attack query being selected), offered very poor robustness, with the target page being promoted on almost all occasions.

Interestingly, there was little difference between top-3 and top-6 to 12 hit ratio results, which means that the target page was typically promoted to the first screen of search results (representing optimal success from the attacker's perspective). This finding can be explained by examining the total number of pages which the system was able to promote for each of the selection strategies. From Fig. 8, it can be seen that the total number of promotable pages did not change significantly across the different query selection strategies; for example, 2.7 pages on average were promoted for $th = 0.5$, 4.7 pages for $k = 5$ and rising to only 7.3 pages for $k = 20$.[2] These relatively low numbers of promotable pages gave rise to the small differences observed between the top-N hit ratio results.

Results for the promote-always attack model are shown in Fig. 7. Overall, for the reasons set out in Sect. 4.5, this attack did not perform as well as the term-specific attack. In addition, there was a significant difference observed between the different top-N hit ratio results realised by the promote-always attack for the best-k (for $k > 1$) query selection strategy. For example, a hit ratio of 0.63 was achieved for $N = 12$, compared to 0.14 for $N = 12$ ($k = 20$). Referring to Fig. 8, in contrast to the term-specific attack, the number of pages that the system was able to promote grew significantly as k was increased (16.9 pages for $k = 20$ compared to 7.1 pages for $k = 5$). Consequently, it became more difficult to promote the target page into the top-3 results for high values of k and robustness was seen to improve for values of $k > 5$. As N was increased, however, this effect began to diminish and for top-12 results, hit ratios were more-or-less consistent for $k > 5$.

[2] Note that selecting k candidate queries does not imply that k pages will be promoted by I-SPY, given that the same page can be associated with multiple queries in the system hit-matrix.

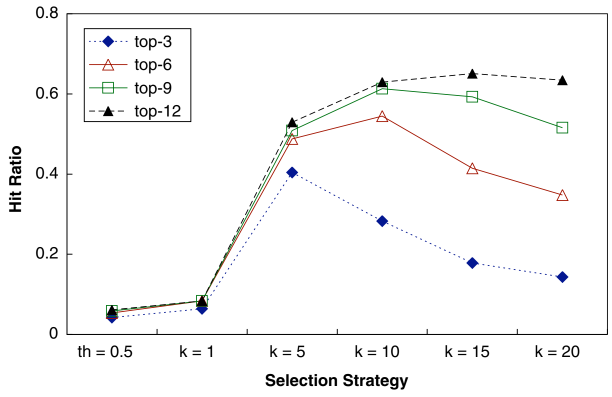


Fig. 7 Hit ratio results achieved for the promote-always (P-A) attack model using threshold and best-*k* query selection strategies

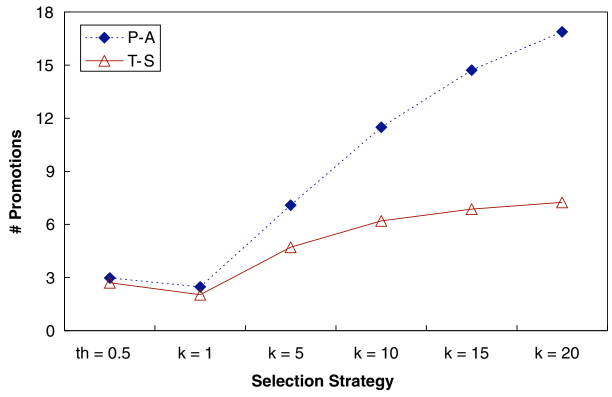


Fig. 8 The number of page promotions made for the promote-always (P-A) and term-specific (T-S) attack models using threshold and best-*k* query selection strategies

Query selection is a key component of the I-SPY algorithm in the sense that it enables the selection of only the most relevant queries from the hit-matrix for any new query that is submitted. It is critical that, if users are to maintain confidence in the system, high precision is achieved by I-SPY when pages are selected for promotion. In previous studies conducted by Smyth et al. (2005) and Smyth and Balfe (2006), threshold selection with $th = 0.5$ and best-k selection for $k > 1$ were able to deliver good performance in this regard. While best-1 selection provided the greatest degree of robustness against the term-specific attack, the utility of promotions made by I-SPY, however, were the least effective for particular value of k. Thus, we can conclude that none of the query selection strategies evaluated above delivered satisfactory robustness, while maintaining good performance in terms of result promotion.

4.7 Varying query term distribution knowledge

In the above analysis, it was assumed that full knowledge relating to query term distribution (i.e. the frequency of occurrence of terms in the dataset) was available to attackers. While this assumption is favourable from an attacker's perspective, it should be noted that search engines often provide such information (to some degree) to users. For example, additional and related search terms are "suggested" by Google Suggest and Yahoo! Search Suggestions to assist users when formulating search queries. Indeed, some I-SPY implementations provide such information directly in the form of lists of recent and popular queries that were submitted by community members.

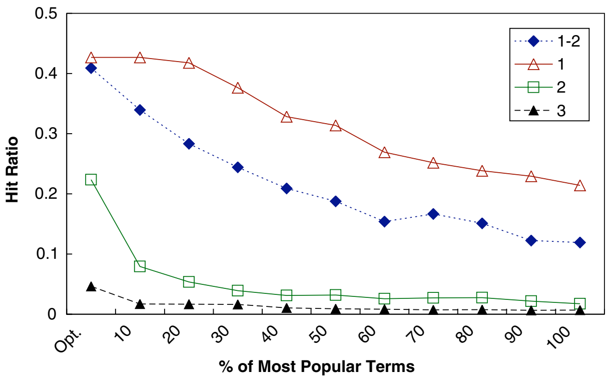


Fig. 9 The effect of query term distribution knowledge on hit ratios achieved by the promote-always attack. Attack queries consisted of terms which are drawn from the top-$X\%$ (shown on the x-axis) of the most frequently occurring terms

In this section, we assume that query term distribution knowledge is only partially available. In order to simulate such partial knowledge, a number of attacks were implemented where it was assumed that only a certain percentage of the most frequently occurring terms contained in the dataset was known. In each of the attacks, attack queries were constructed using terms drawn uniformly at random from the known set of the most frequently occurring terms.

For all simulated attacks, a fixed number of attack queries (attack size $=$ 10%) were inserted. Figure 9 shows the effect of query term distribution knowledge on robustness for the promote-always attack using best-k ($k = 5$) query selection. Similar trends were observed for other query selection strategies. The percentage of the most frequently occurring terms from which attack query terms were drawn is shown on the x-axis. Results for optimal attacks (i.e. when full query term distribution knowledge was known) are also shown. Note that drawing from the top-100% of the most frequently occurring terms simulates a complete lack of query term distribution knowledge.

Hit ratio results are shown for attack queries consisting of $l = 1$, 2 and 3 terms. Results for the 50–50% combination of query sizes of $l = 1$ and 2 terms are also shown (labeled "1–2" in the figure). As expected, all attacks became less successful as knowledge decreased, since the number of common terms (and therefore similarity) between attack and test queries grew less as more infrequently occurring terms were used in attack queries. It can also be seen that hit ratios for attacks consisting of greater numbers of query terms diminished at a greater rate as knowledge decreased. For example, the most successful attack consisted of queries formed using only a single term, where a hit ratio of 0.21 was achieved when a complete lack of knowledge was assumed ($x = 100\%$), and a hit ratio of 0.43 was achieved when full knowledge was assumed. Comparable hit ratios of 0.12 (no knowledge) and 0.41 (full knowledge) were realised for the next most successful attack, that based on combinations of query sizes of $l = 1$ and 2 terms. Attacks which consisted of queries with $l \geq 2$ terms met with little success as lack of knowledge increased.

In conclusion, it can be stated that attacks based on single term attack queries offer the best strategy for attackers in situations where the distribution of query terms is limited or unknown (at least this is the case for the particular dataset used in our evaluations). From an attacker's perspective, this finding is beneficial since (i) fewer terms need to be identified to create a given number of small attack queries than larger ones (assuming that each attack query created consists of a different set of terms) and (ii) such small and easy to create attack queries achieve the greatest success when full distribution knowledge is unknown, as would typically be the case in real-world scenarios.

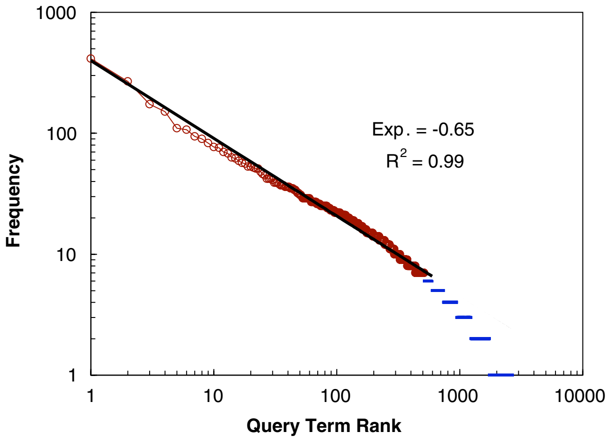


Fig. 10 The frequency of occurrence of query terms in the dataset. The distribution of the lower-ranked query terms is Zipfian, with an exponent of ≈ -0.65

4.8 Varying community homogeneity

Different communities have varying degrees of homogeneity, i.e. the amount of regularity and repetition that exists in the searches of like-minded community members. All communities, by definition, strive for a certain degree of focus in order to meet the needs and tastes of members. Broadly, community homogeneity depends on such factors as user participation levels, topic, community maturity (i.e. for how long a community has been in existence), the number of competing community-based applications offering services etc.

All of the above factors will be reflected in the data gathered from community activity. Communities with high degrees of homogeneity can be expected to exhibit significant query repetition, with many queries having at least some number of terms in common with others. In addition, tightly focussed communities will also exhibit higher URL selection regularity, with the same URL being selected for many related queries. In this section, we test the hypothesis that the more homogeneous communities are at greater vulnerability to attack, given that the more structured nature of the data in such communities ought to facilitate the creation of attack queries that effectively span the community query-space.

While we do not have data from other I-SPY communities, it is, however, possible to simulate homogeneity by varying the distribution of query term frequency of occurrence. This particular distribution, we believe, is a key measure of homogeneity in the sense that it underlies such factors as the distribution of (genuine) query–query similarity, the number of unique queries and terms used in searches etc. In essence, the query term distribution determines the connectivity of the community query-space.

As mentioned above, the distribution of the lowest 500-ranked query terms (the most significant part of the distribution since lower-rank query terms have the highest frequency of occurrence) in the dataset is Zipfian, with an exponent of ≈ -0.65 (Fig. 10). In the following experiments, we randomly assigned terms to queries according to Zipfian distributions, while maintaining the same number of queries, query sizes and result page distribution etc. as the original evaluation dataset. We conducted experiments with exponents ranging in values from -0.2 to -1.4. Exponents closer to zero resulted in more uniform distributions of query terms (i.e. more heterogeneous communities).

Top-3 hit ratio results are shown in Fig. 11 for promote-always attacks (attack size $=$ 10%). It is clear that robustness deteriorated substantially as community homogeneity increased. For example, a hit ratio of 0.25 was achieved for an exponent of -0.2, compared to a hit ratio

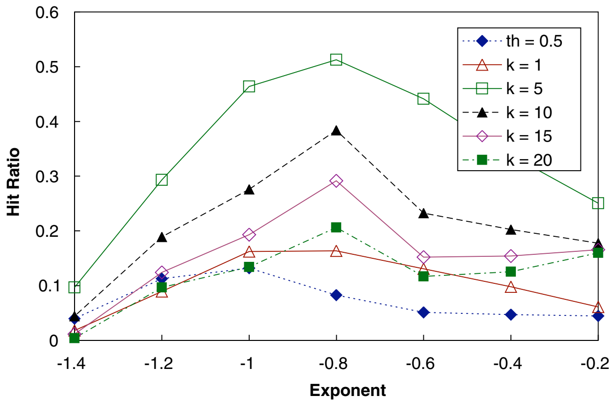


Fig. 11 Top-3 hit ratios achieved for different query selection strategies when subjected to a promote-always attack versus query term distribution

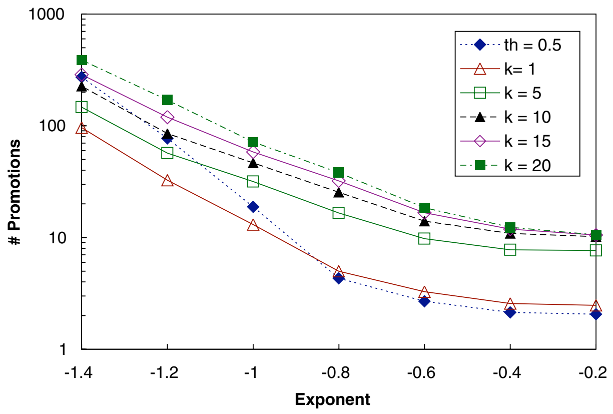


Fig. 12 The number of page promotions made for different query selection strategies when subjected to a promote-always attack versus query term distribution

of 0.51 for an exponent of −0.8 (best-5 query selection). The deterioration in robustness was also evident for all query selection strategies evaluated. Robustness began to improve for exponents >−0.8, which is explained by the large increase in the total number of promotions that I-SPY was able to make for test queries in such cases, as shown in Fig. 12. For example, 57 and 147 promotions were made for exponents of −1.2 and −1.4, respectively, compared to 17 promotions made for an exponent of −0.8 (best-5 query selection). These increases in the number of page promotions was due to the increased connectivity of the query-spaces of more homogeneous communities. Table 1 shows the (genuine) query–query similarity statistics for the simulated datasets, where, for example, average similarities of 0.001 were achieved for an exponent of −0.2, compared to 0.312 for an exponent of −1.4. In addition, the table shows the percentage of non-zero pairwise query–query similarities for each simulated dataset. Only 5.2% of similarities were >0 for an exponent of −0.8; in contrast, all queries were connected (i.e. each query had at least one term in common with at least one other query) for an exponent of −1.4. Consequently, it became much more difficult for target pages to be promoted into one of the top-3 results at the highest exponent values thus, smaller hit ratios were observed in such cases.

Table 1 Query–query similarity statistics for simulated datasets

Exp.	Sim. mean	Sim. std.	Non-zero similarities (%)		
			>0	>0.3	>0.5
−0.2	0.001	0.028	0.3	0.1	0.1
−0.4	0.001	0.030	0.4	0.1	0.1
−0.6	0.004	0.040	1.3	0.3	0.1
−0.8	0.013	0.065	5.2	1.1	0.1
−1.0	0.053	0.123	21.0	4.9	0.6
−1.2	0.147	0.177	55.0	15.8	2.3
−1.4	0.312	0.187	100.0	40.8	8.1

5 Related work

The collaborative Web search technique discussed in this paper shares a lot in common with other forms of recommendation systems (Burke 2002; Claypool et al. 1999; Resnick and Varian 1997) and, in particular, so-called collaborative filtering recommender systems (Herlocker et al. 1999; Resnick et al. 1994; Sarwar et al. 2001). Collaborative recommenders have been successfully employed in many on-line settings (e.g. Amazon, CDNow and Netflix) and work by gathering preferences and opinions from users and using these data to generate new suggestions; essentially, according to Shardanand and Maes (1995), they automate the "word-of-mouth" process. Research in the area of collaborative recommenders is now well-focussed on the robustness issue and, in this section, we review a snapshot of recent research activity in this regard.

A key difference between accepting recommendations from friends and those from on-line systems relates to trust. Over time, people are able to assess the reliability of friends and associates and can therefore place more value on recommendations from trusted and like-minded individuals, while disregarding those from others. In on-line systems, however, it is more difficult to make such assessments, given the anonymous or pseudo-anonymous nature of these environments.

Further, the cost of creating multiple identities within a single system is often cheap, and in such cases, the potential exists for *shillling* attacks (also referred to as *profile injection* attacks) to occur (Lam and Riedl 2004; O'Mahony et al. 2004). These attacks involve the creation of multiple attack profiles, which are typically designed to reflect the true preferences of genuine users for certain items, while the target item is assigned a biased (high or low) rating. It has been demonstrated by O'Mahony et al. (2005) that the creation of even small amounts of attack profiles can significantly manipulate system output and leads to the successful promotion (or demotion) of target items into top-N recommended lists.

In Bhaumik et al. (2007), the robustness of several navigation-orientated Web personalisation algorithms against attack was investigated. This research indicates that some commonly used algorithms were vulnerable to attacks created by generating false clickstreams designed to promote particular target pages. Recently, Koutrika et al. (2007) presented an analysis of tagging systems which found that these systems are also prone to manipulation by malicious agents.

All of the above systems (including I-SPY) rely on data that is gathered explicitly or implicitly from users in order to make recommendations or to facilitate search and navigation. Given this open manner of operation, these systems are vulnerable to manipulation through the insertion of biased data into system databases. Users are unlikely to remain loyal if it is perceived that recommendations of suspect quality are being made and thus, the integrity of these systems is of prime importance. In this regard, some recent research which proposes techniques to improve the robustness of collaborative recommender systems against attack, is of interest.

In O'Mahony et al. (2006), signal processing theory was applied to the problem of attack detection. Attack data, created according to a set of specific attack models, was deliberately inserted into a system in order to model the distributions of genuine and attack data, thereby enabling the selection of thresholds that optimised hit and false alarm rates for such attacks. In Chirita et al. (2005) and Mobasher et al. (2007), a number of generic and model-specific statistical measures to detect attack data were proposed. As with the previous approach, these measures were derived from the particular characteristics of the specific attack models under consideration, and consequently systems employing these measures would be vulnerable to new or unknown attack models that may be developed.

An unsupervised classification algorithm was introduced by Mehta et al. (2007) which considered the combined effect of attack profiles, with the expectation that the correlation between such profiles would be high relative to that between genuine profiles. The proposed algorithm, based on principal component analysis, achieved favourable performance when compared to Chirita et al. (2005) and O'Mahony et al. (2006). In Sandvig et al. (2007), an association rules based approach to recommendation was found to provide a significant degree of robustness against a range of widely studied attack models, particularly in the case of relatively small sized attacks.

Trust-based models of recommendation were proposed by O'Donovan and Smyth (2006), in which trust values were mined from existing preference data. Users were only selected as predictors for a target user in cases where they were able to accurately predict known ratings for the target user. Trust was also incorporated into the algorithm described by Massa and Avesani (2007), where trust statements were explicitly made by users in relation to others. This data was then used in conjunction with preference data to produce recommendations. Note that these approaches assume that profiles of individual user activity are maintained which is not the case in the anonymous mode of operation employed by I-SPY (Sect. 2.1). Nevertheless, these approaches offer interesting scope for further research.

The techniques described above provide a wide and useful basis for work on attack detection in other collaborative systems. Note that the individual approaches can be combined to enhance the strengths and overcome the limitations of each and, in future work, we will investigate the application of these techniques to the collaborative Web search domain in order to defeat the type of attacks that were introduced in this paper.

6 Conclusions

Collaborative Web search leverages the search histories of communities of like-minded users to adapt and re-rank search results to satisfy the needs of community members. Results that have been selected for queries in the past are promoted when similar queries are submitted in future search sessions.

The repetitive nature of community search activity can also, however, be usefully exploited by malicious agents in order to bias search output. In this paper, we have introduced attack models which seek to promote target pages within communities by linking such pages with frequently used search terms. Our findings, based on real-world community data collected from a local software company, indicated that I-SPY (using threshold selection) exhibited a significant degree of robustness against the promote-always attack model. I-SPY was, however, vulnerable to term-specific attacks consisting of small numbers of attack queries, even in cases where limited knowledge concerning query term distribution was available to attackers.

In addition, we analysed the robustness of collaborative Web search in terms of community homogeneity. Different communities will naturally vary in focus, and we have shown through simulation that the degree of specialisation exhibited by communities has a significant effect on robustness. We note, however, that a number of limitations are associated with the model of community homogeneity presented in this work. The model was governed by a single parameter, the exponent of the query term distribution function. While we believe that this particular distribution is a key factor in relation to community homogeneity, in future work we will take into account other factors such as result page and query size distributions.

It would also be useful, if data were available, to establish the range of homogeneity and robustness exhibited by diverse, real-world, communities. One approach we are planning to investigate is to perform a cluster analysis of our existing dataset to discover groups of related queries, and thereby obtain sub-communities with different degrees of focus.

Acknowledgements This work was carried out with the support of Science Foundation Ireland under Grant No. 03/IN.3/I361, which is gratefully acknowledged.

References

Bhaumik R, Burke R, Mobasher B (2007) Effectiveness of crawling attacks against web-based recommender systems. In: Proceedings of the 5th workshop on intelligent techniques for web personalization (ITWP-07)

Burke R (2002) Hybrid recommender systems: Survey and experiments. User Model User–Adapt Interact 12(4):331–370

Chirita PA, Nejdl W, Zamfir C (2005) Preventing shilling attacks in online recommender systems. In Proceedings of the ACM workshop on web information and data management (WIDM'2005) pp 67–74, Germany

Claypool M, Gokhale A, Miranda T, Murnikov P, Netes D, Sartin M (1999) Combining content and collaborative filters in an on-line newspaper. In: Proceedings of the ACM SIGIR workshop on recommender systems: algorithms and evaluation, 22nd international conference on research and development in information retrieval pp 15–22

Coyle M, Smyth B (2007) Supporting intelligent web search. ACM Trans Internet Technol 7(4) (Article No. 20)

Herlocker J, Konstan J, Borchers A, Riedl J (1999) An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd international ACM SIGIR conference on research and development in information retrieval pp 230–237, Berkeley

Koutrika G, Effendi FA, Gyongyi Z, Haymann P, Garcia-Molina H (2007) Combating spam in tagging systems. In: Proceedings of 3rd international workshop on adversarial information retrieval on the web (AIRWeb'07)

Lam SK, Riedl J (2004) Shilling recommender systems for fun and profit. In: Proceedings of the 13th international world wide web conference pp 393–402, New York

Massa P, Avesani P (2007) Trust-aware recommender systems. In: Proceedings of the ACM conference on recommender systems (RecSys'07) pp 17–24, Cyprus

Mehta B, Hofmann T, Frankhauser P (2007) Lies and propaganda: Detecting spam users in collaborative filtering. In: Proceedings of the 12th international conference on intelligent user interfaces (IUI-07) pp 14–21

Mobasher B, Burke R, Bhaumik R, Williams C (2005) Effective attack models for shilling item-based collaborative filtering system. In: Proceedings of the 2005 WebKDD workshop (KDD'2005), Chicago

Mobasher B, Burke R, Bhaumik R, Williams C (2007) Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. ACM Trans Internet Technol 7(4) (Article 23)

O'Donovan J, Smyth B (2006) Is trust robust? An analysis of trust-based recommendation. In: Proceedings of the 11th international conference on intelligent user interfaces (IUI'06) pp 101–108, Sydney

O'Mahony MP, Hurley NJ, Kushmerick N, Silvestre GCM (2004) Collaborative recommendation: a robustness analysis. ACM Trans Internet Technol (TOIT), Special Issue on Machine Learning for the Internet 4(4):344–377

O'Mahony MP, Hurley NJ, Silvestre GCM (2005) Recommender systems: attack types and strategies. In: Proceedings of the 20th national conference on artificial intelligence (AAAI-05) pp 334–339

O'Mahony MP, Hurley NJ, Silvestre GCM (2006) Detecting noise in recommender system databases. In: Proceedings of the international conference on intelligent user interfaces (IUI'06) pp 109–115

Resnick P, Varian HR (1997) Recommender systems—introduction to the special section. Commun ACM 40(3):56–58

Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) Grouplens: An open architecture for collaborative filtering of netnews. In: Proceedings of the ACM conference on computer supported cooperative work (CSCW'94) pp 175–186

Sandvig J, Mobasher B, Burke R (2007) Robustness of collaborative recommendation based on association rule mining. In: Proceedings of the ACM conference on recommender systems (RecSys'07) pp 105–111, Minneapolis

Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the tenth international world wide web conference pp 285–295

Shardanand U, Maes P (1995) Social information filtering: Algorithms for automating word of mouth. In: Proceedings of ACM conference on human factors in computing systems (CHI'95) pp 210–217, Denver

Smyth B, Balfe E (2006) Anonymous personalisation in collaborative web search. J Inform Retr 9(2):165–190

Smyth B, Balfe E, Briggs P, Coyle M, Freyne J (2003) Collaborative web search. In: Proceedings of the 18th international joint conference on artificial intelligence (IJCAI-03) pp 1417–1419, Mexico

Smyth B, Balfe E, Freyne J, Briggs P, Coyle M, Boydell O (2004) Exploiting query repetition & regularity in an adaptive community-based web search engine. User Model User-Adapt Interact: J Per Res 14:383–423

Smyth B, Balfe E, Boydell O, Bradley K, Briggs P, Coyle M, Freyne J (2005) A live-user evaluation of collaborative web search. In: Proceedings of the 19th international joint conference on artificial intelligence (IJCAI-05) pp 1419–1424