

Attack Detection in Time Series for Recommender Systems

Sheng Zhang, Amit Chakrabarti, James Ford, Fillia Makedon
Department of Computer Science, Dartmouth College
{clap, ac, jford, makedon}@cs.dartmouth.edu

ABSTRACT

Recent research has identified significant vulnerabilities in recommender systems. Shilling attacks, in which attackers introduce biased ratings in order to influence future recommendations, have been shown to be effective against collaborative filtering algorithms. We postulate that the distribution of item ratings in time can reveal the presence of a wide range of shilling attacks given reasonable assumptions about their duration. To construct a time series of ratings for an item, we use a window size of k to group consecutive ratings for the item into disjoint windows and compute the sample average and sample entropy in each window. We derive a theoretically optimal window size to best detect an attack event if the number of attack profiles is known. For practical applications where this number is unknown, we propose a heuristic algorithm that adaptively changes the window size. Our experimental results demonstrate that monitoring rating distributions in time series is an effective approach for detecting shilling attacks.

Categories and Subject Descriptors: H.3.5 [Information Storage and Retrieval]: Online Information Services—*Commercial services*; k.4.4 [Computers and Society]: Electronic Commerce—*Security*

General Terms: Algorithms, Security

keywords: Recommender systems, shilling attacks, anomaly detection, time series

1. INTRODUCTION

Recommender systems have become popular in the past several years as an effective way to help people deal with information overload. However, since these systems are dependent on external sources of information, they are vulnerable to *shilling attacks*, in which attackers influence systems in a manner advantageous to themselves by introducing biased rating profiles. Shilling attacks can be classified as *push* and *nuke* attacks according to their intent—making a target item more or less likely to be recommended, respectively.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD'06, August 20–23, 2006, Philadelphia, Pennsylvania, USA.
Copyright 2006 ACM 1-59593-339-5/06/0008 ...\$5.00.

Because recommender systems are widely used in the realm of e-commerce, there is a natural motivation for producers of items to use these shilling attacks so that their items are recommended to users more often. Therefore, an important research challenge in recommender systems is to detect and defeat shilling attacks.

A considerable complication in detecting shilling attacks is that it is difficult to precisely and completely define the set of shilling attack patterns. New attacks will continue to arise over time, so an attack detection approach should avoid being restricted to any predefined set of attacks. Our goal in this paper is to seek methods that are able to detect a diverse and general set of recommendation attacks.

Our work begins with the following observation. If we assume that attack profiles are injected into the system within a relatively short period of time, most shilling attack models (discussed in detail in Sec. 2) share a common characteristic despite their diversity: over their attack period they induce changes in the rating distributions of target items (and possibly other items). For example, a push attack, regardless of its attack model, will cause the rating distribution of a target item to be concentrated on high ratings during its duration. Similarly, a target item's rating distribution will be concentrated on low ratings in a nuke attack. Our thesis is that examining the rating distribution for each item over a time series can yield a considerable diagnostic power in detecting a large set of attacks.

The idea of treating shilling attacks as events that disturb the rating distribution differs from previous methods that decide whether a user's rating profile is biased or normal by comparing it with others overall. Detecting attacks in time series has two key benefits. First, it enables detection of attacks that are difficult to isolate in previous methods where each attack profile is considered separately. Attack profiles generated by some attacks (such as sampling attacks) looks very similar to normal profiles, and thus are almost indistinguishable when only considering individual rating pattern. Second, unusual distributions in time series can reveal previously undefined or unknown attacks. This is a significant advance over heuristic rule-based categorizations or supervised classifications. We note that the time series approach may also find valuable non-malicious anomalies.

To construct time series that are appropriate for attack detection, we extract two useful properties of rating distributions: sample average and sample entropy (Sec. 3). Sample average captures the change in an item's likability, while sample entropy captures the distributional change (the degree of dispersal or concentration) in an item's ratings. We

construct time series for an item by taking every disjoint k consecutive ratings given to the item (according to their given time) as a window and computing sample average and sample entropy for each window. We show that observing the time series of these two properties exposes attack events.

We give a theoretical analysis to quantify the changes in sample average and sample entropy in time series when attack profiles are injected. Assuming that the number of attack profiles is known, an optimal window size is derived to maximally amplify changes caused by attacks, which helps to enhance the performance of attack detection (Sec. 4). For practical applications where this assumption does not hold, we propose a heuristic algorithm to estimate the number of attack profiles and adaptively adjust the window size (Sec. 5). We give experimental results in Sec. 6 and conclude with a discussion in Sec. 7.

2. RELATED WORK

In this section, we describe popular recommendation attack models and summarize the related work on shilling attacks. Five popular attack models are briefly introduced here in the context of a push attack. In *Random attacks* (see Fig. 1), a target item will be given the highest rating, but ratings to *filler items* (a proportion of the remaining items) in each rating profile are chosen randomly (usually from a normal distribution). *Average attacks* are a more sophisticated variation: the ratings for filler items in attack profiles are distributed around the mean for each item. *Segmented attacks* target users who are in favor of a particular item segment (e.g., readers expressing an interest in fantasy books) and bias a system’s recommendations to these users. Segmented attacks give the highest rating to the target and the item segment and the lowest rating to filler items. *Bandwagon attacks* can be viewed as an extension of random attacks. They take advantage of the Zipf’s law distribution on item popularity (the number of ratings received). Attackers in this model give the highest rating to selected frequently rated items and random ratings to filler items. Besides the above four models, there is a *Sampling attack* model, in which attack profiles are constructed from entire user profiles sampled from the actual rating database augmented by the highest rating for the pushed item.

Previous Research [5, 9, 11, 13] has evaluated the impact of shilling attacks on various collaborative filtering algorithms. Recently, several studies were aimed at detecting shilling attacks. Chirita *et al.* [6] and Mobasher *et al.* [10] proposed several empirical metrics for detecting random attacks and segmented attacks, respectively. Zhang *et al.* [13] developed a probabilistic approach to detect random attacks by computing the probability of each rating profile given a low-dimensional linear model extracted from ratings. Those profiles that have abnormally low probabilities are identified as attacks. While the existing approaches are effective in detecting random attacks (or segmented attacks), they are incapable of detecting average attacks and sampling attacks. To the best of our knowledge, there is no approach in the literature that provides a systematic methodology to detect a large variety of shilling attacks.

Time series have been exploited for detecting attacks in network traffic analysis [4, 8]. The general problem of finding time series discords (subsequences that are maximally different to all the rest of the time series subsequences) was studied in [7].

3. CONSTRUCTING A TIME SERIES

Our thesis is that the analysis of rating distributions in time is a powerful tool for the detection of recommendation attacks. The intuition behind this thesis is that all (known) attack models cause changes in the rating distributions of target items (and possibly other items). For example, Table 1 lists the effects of the five attack models surveyed in Sec. 2. Rating distributions of target items will always become concentrated on high ratings (in push attacks) or low ratings (in nuke attacks) whatever attack model is used. For filler items, distributions become concentrated on low ratings when segmented attacks are injected. When other attack models are used, the rating distributions of filler items may also be concentrated depending on the variance of the distribution that generates ratings.

To extract useful information from the rating distribution of an item, we use the following two properties: *the degree of dispersal or concentration* of the distribution and the *sample average*. The measure we use to capture the degree of dispersal or concentration of a rating distribution is the *sample entropy*. Assume that we have an empirical histogram $X = \{n_i, i = 1, \dots, r_{max}\}$, meaning that the i th possible rating occurs n_i times in the sample. Then the sample entropy is defined as $H(X) = -\sum_{i=1}^{r_{max}} (n_i/S) \log_2(n_i/S)$, where $S = \sum_{i=1}^{r_{max}} n_i$ is the total number of ratings in the histogram. The value of the sample entropy lies in the range $[0, \log_2 r_{max}]$. The value 0 is taken when all ratings are the same and the value $\log_2 r_{max}$ is taken when n_i is the same for all i . Using the above notation and assuming that the i th possible rating has the value i , the sample average is defined as $M(X) = (\sum_{i=1}^{r_{max}} n_i \times i)/S$.

To construct the time series of the above two measures for an item, we first sort all the ratings for the item by their time stamps, and then group every disjoint k consecutive ratings into a window. Here, k is referred to as the *window size*. For each window, we compute its sample average and sample entropy. Therefore, we obtain two time series for the selected item, each corresponding to one of the measures.

Denote as w_j the j th window for the selected item. If ratings to the item are i.i.d. from a distribution $P = \{p(x), x \in [1, r_{max}]\}$ with mean μ and variance σ^2 , we can use the following two propositions to show that $M(w_j)$ and $H(w_j)$ are asymptotically Normal as k increases. We note that nothing is assumed about the distribution P except the existence of a mean and variance.

PROPOSITION 1. *If ratings to an item are i.i.d. with mean μ and variance σ^2 , then $\frac{M(w_j) - \mu}{\sigma/\sqrt{k}} \rightarrow N(0, 1)$ as k increase. In other words, the sample average for the item can be approximated using a normal distribution with mean μ and standard deviation σ/\sqrt{k} . This proposition follows from the central limit theorem [12].*

PROPOSITION 2. *If ratings to an item are i.i.d. from a distribution P , then $\frac{H(w_j) - H}{\sqrt{\text{Var}(-\log_2 p(x))/\sqrt{k}}} \rightarrow N(0, 1)$ as k increases, where H is the true entropy. In other words, the sample entropy for the item can be approximated using a normal distribution with mean H and standard deviation $\sqrt{\text{Var}(-\log_2 p(x))/\sqrt{k}}$. This proposition follows from a result in [2] (see also [1]).*

As both sample average and sample entropy are asymptotically Normal, we can decide whether a window is an

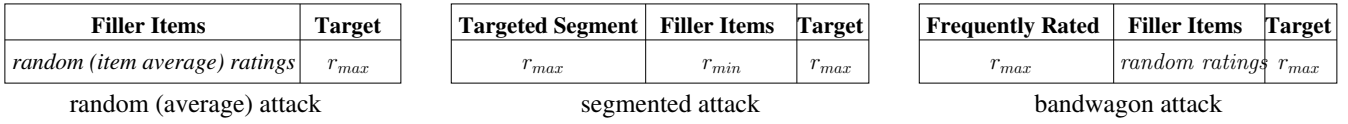


Figure 1: Popular shilling attack models (assuming that a target item is pushed).

Table 1: Effects on rating distributions by various attack models

Attack Model	Rating Distributions Affected
Random attack	Target items, filler items (possibly—depending on the variance of the used distribution)
Average attack	Target items, filler items (possibly)
Segmented attack	Target items, items in the targeted segment, filler items
Bandwagon attack	Target items, frequently rated items, filler items (possibly)
Sampling attack	Target items

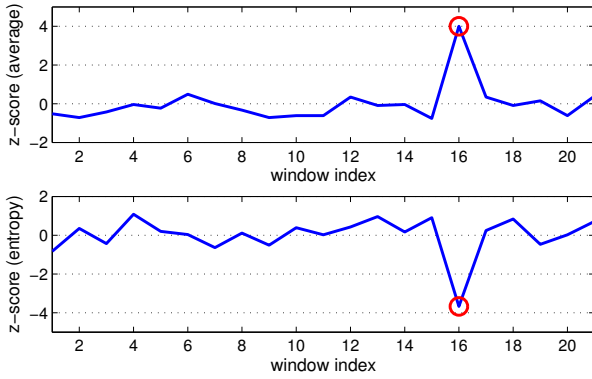


Figure 2: A push attack event stands out clearly when viewed through the lens of sample average and sample entropy. The window size is 50 and the window containing attacks is marked with a circle.

anomaly by testing whether the absolute value of its z-score (the difference from the mean divided by the standard deviation) for sample average (sample entropy) is larger than a threshold. The threshold is set to 2 in this paper, which corresponds to the 95.5% confidence level. We illustrate the effectiveness of this approach through the example in Fig. 2. In this example, 40 attack ratings are injected to give the highest rating to a target item. The window size is 50 and the window containing attacks is marked with a circle. Both plots show that the attack event stands out clearly, while z-scores for sample average and sample entropy of those windows containing normal ratings vary within a small range.

4. A THEORETICAL ANALYSIS

Having introduced how to construct the time series for an item, we now quantify the changes of sample average and sample entropy caused by an attack event. Our discussion below is focused on a target item, and the notation used previously for Proposition 1 and 2 will still apply here. We note that the following analysis also holds for other items affected by attacks, *e.g.*, filler items in segmented attacks.

When attack ratings for the target item are injected, more than one window may contain attack ratings. We focus our analysis on the window that has the largest number of attack ratings, and denote it as the *anomalous window*. In case

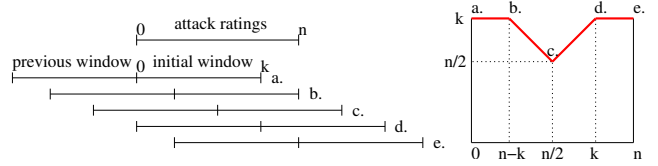


Figure 3: The number of attacks in the anomalous window when the start position of the initial window moves from the start of attacks to the end of attacks in case 2 ($\lceil n/2 \rceil < k < n$).

there are two or more candidates for the anomalous window, one is chosen randomly.

In the following, we will first compute the expected fraction of attack ratings in the anomalous window (Sec. 4.1); and then find the optimal window size k to maximize the absolute value of its z-scores for sample average (Sec. 4.2) and for sample entropy (Sec. 4.3). Maximizing the absolute value of these two z-scores helps to identify the anomalous window more accurately in our approach.

4.1 The expected fraction of attack ratings

Assuming that the number of attack profiles during an attack event is n , the number of attack ratings for the target item is also n . For ease of expression, we will initially assume that there are no normal (real) ratings given to the target item during the attack. In other words, the n attack ratings are consecutive. The more general situation in which real ratings and attack ratings are intermixed will be discussed in Sec. 4.4. Because the start position of attack ratings is random, we now compute the expected fraction (denoted as λ) of attack ratings in the anomalous window in each of the following three cases.

In case 1 where $n \geq 2k - 1$ ($k \leq \lceil n/2 \rceil$), there always exists a window that is filled with attack ratings. Therefore, we have $\lambda = 1$.

In case 2 where $\lceil n/2 \rceil < k < n$, we compute λ by moving a window (initial window in Fig. 3) from the start of attacks to the end of attacks. The right plot shows the number of attacks in the anomalous window when the initial window is in different positions. Overall, the expected number of attacks in the anomalous window is the area of the right plot divided by n , which gives us $2k - k^2/n - n/4$. Thus, λ in this case is $2 - k/n - n/(4k)$.

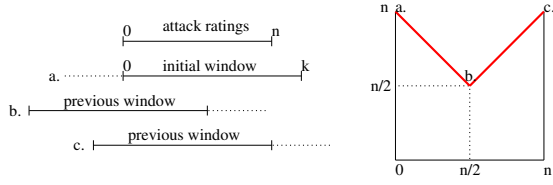


Figure 4: The number of attacks in the anomalous window when the start position of the initial window moves from the start of attacks to the end of attacks in case 3 ($k \geq n$).

In case 3 where $k \geq n$, λ can be computed in a similar way to the above. The expected number of attacks in the anomalous window is the area of the right plot in Fig. 4 divided by n , that is $3n/4$. Thus, $\lambda = (3n)/(4k)$.

Combining all three cases, we conclude that the expected fraction of attack ratings in the anomalous window is

$$\lambda = \begin{cases} 1 & k \leq \lceil n/2 \rceil \\ 2 - \frac{k}{n} - \frac{n}{4k} & \lceil n/2 \rceil < k < n \\ \frac{3n}{4k} & k \geq n \end{cases} \quad (1)$$

4.2 Sample average

Denote the anomalous window as \tilde{w} , and denote its z-score for sample average as $Z_M(\tilde{w})$. We now quantify the expectation of $Z_M(\tilde{w})$ and compute the optimal k that maximizes its absolute value. Assuming w.l.o.g. that the item is being pushed, by Proposition 1 it follows that

$$\begin{aligned} |E(Z_M(\tilde{w}))| &= \frac{E(M(\tilde{w})) - \mu}{\sigma/\sqrt{k}} = \frac{(1-\lambda)\mu + \lambda r_{max} - \mu}{\sigma/\sqrt{k}} \\ &= \frac{\sqrt{k}\lambda(r_{max} - \mu)}{\sigma}. \end{aligned}$$

Since μ and σ are fixed, maximizing $|E(Z_M(\tilde{w}))|$ is reduced to maximizing the term $\sqrt{k}\lambda$. Because λ has three different representations when k is changed (recall from Eq. 1), we do the optimization in each case.

In case 1 where $k \leq \lceil n/2 \rceil$, $\lambda = 1$. Thus, we have $\sqrt{k}\lambda = \sqrt{k}$. This is maximized when $k = \lceil n/2 \rceil$. The maximum value is $\sqrt{\lceil n/2 \rceil} \approx (\sqrt{2}/2)\sqrt{n}$.

In case 2 where $\lceil n/2 \rceil < k < n$, $\lambda = 2 - k/n - n/(4k)$. It follows that $\sqrt{k}\lambda = 2\sqrt{k} - k\sqrt{k}/n - n/(4\sqrt{k})$. With basic calculus, we can determine that the above term is maximized when $k = \frac{2+\sqrt{7}}{6}n \approx 0.7743n$. The corresponding maximum value is about $0.7944\sqrt{n}$.

In case 3 where $k \geq n$, $\lambda = (3n)/(4k)$. Thus, we have $\sqrt{k}\lambda = (3n)/(4\sqrt{k})$. This case is maximized when $k = n$, with a maximum value of $0.75\sqrt{n}$.

Combining all the cases, we get the following theorem.

THEOREM 1. *The absolute value of the expected z-score for sample average in the anomalous window, $|E(Z_M(\tilde{w}))|$, is maximized when the window size is equal to $\frac{2+\sqrt{7}}{6}$ times the number of attacks, i.e., $k = \frac{2+\sqrt{7}}{6}n$.*

4.3 Sample entropy

Denote the z-score of \tilde{w} for sample entropy as $Z_H(\tilde{w})$. We now quantify $|E(Z_H(\tilde{w}))|$ and compute the optimal k to maximize this value.

If the fraction of attack ratings in the anomalous window is λ' , the entropy of \tilde{w} can be bounded using the following idea. To generate a rating in \tilde{w} , we first toss a coin. With probability $1 - \lambda'$, a normal rating is generated from the original distribution P ; and with probability λ' , the highest rating r_{max} is generated. If the random variable that describes the outcome of the coin toss is denoted as C , we have,

$$\begin{aligned} H(\tilde{w}) &\leq H(\tilde{w}, C) = H(C) + H(\tilde{w}|C) \\ &= H_2(\lambda') + (1 - \lambda')H \\ &\leq 1 + (1 - \lambda')H, \end{aligned} \quad (2)$$

where $H_2(\lambda') = -\lambda' \log_2 \lambda' - (1 - \lambda') \log_2 (1 - \lambda')$. On the other hand,

$$H(\tilde{w}) \geq H(\tilde{w}|C) = (1 - \lambda')H. \quad (3)$$

As $\lambda = E(\lambda')$, by Proposition 2 we have

$$|E(Z_H(\tilde{w}))| = \frac{H - E(H(\tilde{w}))}{\sqrt{\text{Var}(-\log_2 p(x))}/\sqrt{k}},$$

By Inequality (2, 3), $|E(Z_H(\tilde{w}))|$ can be bounded as

$$\frac{\sqrt{k}(\lambda H - 1)}{\sqrt{\text{Var}(-\log_2 p(x))}} \leq |E(Z_H(\tilde{w}))| \leq \frac{\sqrt{k}\lambda H}{\sqrt{\text{Var}(-\log_2 p(x))}}. \quad (4)$$

Using the same reasoning in Sec. 4.2, the upper bound of Inequality (4) is maximized when $k = \frac{2+\sqrt{7}}{6}n$. Similarly, the lower bound is maximized when $k = \frac{2H-1+\sqrt{7H^2-4H+1}}{6H}n$. Because this term converges to $\frac{2+\sqrt{7}}{6}n$ when H is large, we obtain the following theorem.

THEOREM 2. *The absolute value of the expected z-score for sample entropy in the anomalous window, $|E(Z_H(\tilde{w}))|$, is maximized when $k \approx \frac{2+\sqrt{7}}{6}n$ (for large H).*

Taken in conjunction with Theorem 1, this shows that an optimal window size can be found to simultaneously maximize both the absolute value of the expected z-score for sample average and for sample entropy in the anomalous window.

4.4 An Extension

The above analysis can be easily extended to the case in which real ratings and attack ratings are intermixed in time. Define ω ($0 < \omega \leq 1$) as the ratio of attack ratings to the total number of ratings (including both attack and real ratings) given to the item during an attack event. Because the number of attack ratings is n , the total number of ratings is n/ω and is denoted as the *length of an attack event*. Assuming that the ratio of attack ratings to real ratings is fixed within any sub-series of consecutive ratings, the following Corollary can be obtained using a similar reasoning to those in the previous two Subsections.

COROLLARY 1. *If the ratio of attack ratings to the total number of ratings given to the item during an attack event is ω , both $|E(Z_M(\tilde{w}))|$ and $|E(Z_H(\tilde{w}))|$ are maximized when $k \approx \frac{2+\sqrt{7}}{6} \frac{n}{\omega}$.*

5. A HEURISTIC APPROACH

The previous section shows that when the length of an attack event is known, an optimal window size can be found to best detect rating distribution changes caused by attacks.

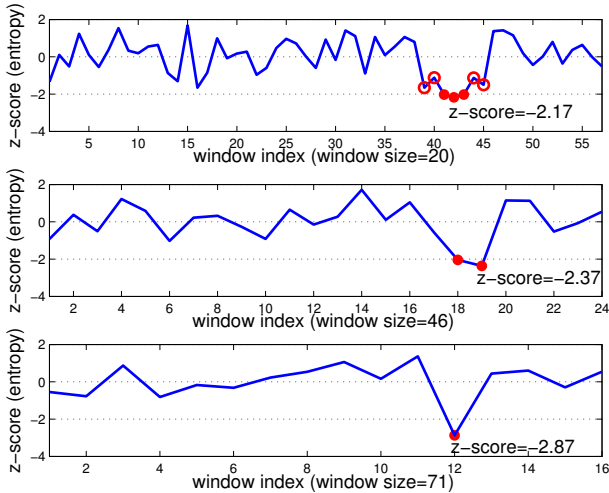


Figure 5: A push attack event stands out more clearly after a heuristic approach is used to repeatedly estimate the length of an attack event and adjust the window size appropriately.

However, this assumption does not hold for practical applications. In this section, we propose a two-step heuristic approach to first estimate the length of an attack event and then adaptively adjust the window size.

At first, a default window size is chosen and a time series for sample average (sample entropy) is constructed for the item. The default window size can be set as the largest number of attack profiles that is considered negligible. Since the default window size is relatively small, rating distributions of several windows will be affected by a typical attack event. Meanwhile, there might be some detected anomalies in normal windows. However, we argue that it is unlikely we will find a series of consecutive spikes (with the same direction) in normal windows. Therefore, the length of an attack event can be estimated as the largest number of consecutive anomalies (with the same direction) times the default window size. After estimating the length of an attack event, we reset the window size as $\frac{2+\sqrt{7}}{6}$ times that length (using the result of Corollary 1).

Note that we can always reestimate the length of the event using the current window size and then adjust the window size accordingly until there is no series of two or more consecutive anomalies.

We illustrate the effectiveness of this heuristic approach in Fig. 5. It plots the z-score (for sample entropy) of the same item used in Fig. 2. The number of attacks to push the item is 100, and the fraction of attack ratings (ω) is $2/3$. Those windows containing attacks are marked with a circle and windows identified as anomalies are marked with a filled circle. The first plot shows the time series with a default window size of 20. This gives us an estimate that the length of the attack event is $3 \times 20 = 60$. Therefore, we adjust the window size to 46 ($\approx 0.7743 \times 60$) and obtain the second time series. After reestimating the length of the attack event as 92 (46×2), we finally obtain the third time series with a window size 71 ($\approx 0.7743 \times 92$). Compared with the first time series, this one reveals a much clearer presence of the attack event. The z-score of the anomalous window

is -2.87 while the lowest z-score of the windows containing attacks in the first time series is -2.17 .

In each iteration of the above heuristic approach, the window size increases by a factor at least $2 \times \frac{2+\sqrt{7}}{6}$. Therefore, the total number of iterations needed is $O(\log \frac{n}{w})$.

6. EXPERIMENTS

We selected all the items (618 in total) with at least 500 ratings from a MovieLens data set consisting of about 1 million ratings. Ratings are discrete-valued between 1 and 5. We sort ratings for each item by their time stamp. To simulate an attack event, we insert a number of attack ratings for an item into its normal ratings. The start position of an attack event is random, and the ratio of attack ratings to the total number of ratings (given to the item) during the attack event is set to ω .

For each time series of sample average and sample entropy, we mark a window as an anomaly if the absolute value of its z-score is larger than 2. Two metrics, *detection rate* and *false alarm rate*, are used to evaluate the detection performance. The detection rate is defined as the number of detected attack events divided by the number of attack events. An attack event is considered to be detected if a window containing attacks is marked as an anomaly. The false alarm rate is defined as the number of normal windows that are predicted as anomalies divided by the number of normal windows.

We first measure how well our method detects attacks by using it to observe the time series of target items.¹ We execute a push (nuke) attack for each of the 618 items and attempt to detect the attack by examining its time series. The total number of detected attack events and the total number of false alarm cases are computed over all the items. The window size is set to 20 because this is about the minimum number of attacks that will have a considerable effect in this data set according to our experiments and results in [9]. The ratio of attack ratings (ω) is set to $2/3$. The number of attack ratings to push the item varies from 20 to 200. Five trials were performed in total, and detection results are plotted in Fig. 6. It shows that sample average yields better detection results when fewer attacks are injected. As the number of attacks increases, the detection performance using sample average drops quickly, while the performance using sample entropy is still stable.

Next, we show that attacks may also be detected by observing an item’s rating distribution change when the variance of the attack ratings (which need not be extreme values) to the item is sufficiently small. In average attacks, ratings given to a filler item are usually Normal with a mean equal to the item average. We generate 50 attack ratings for each item in this way (assuming that the item is a filler item) and try to detect attacks by analyzing its times series using sample entropy. Table 2 presents the results when the standard deviation of the Normal distribution varies.

Finally, we examine how our heuristic approach (estimating the length of an attack event and adjusting the window size) compares with using a fixed window size. Push attacks are used and the number of attack ratings in an attack event is randomly set between 50 and 200. The default window

¹This experiment also applies to the items that receives extremely high or low ratings in attacks, *e.g.*, filler items in segmented attacks.

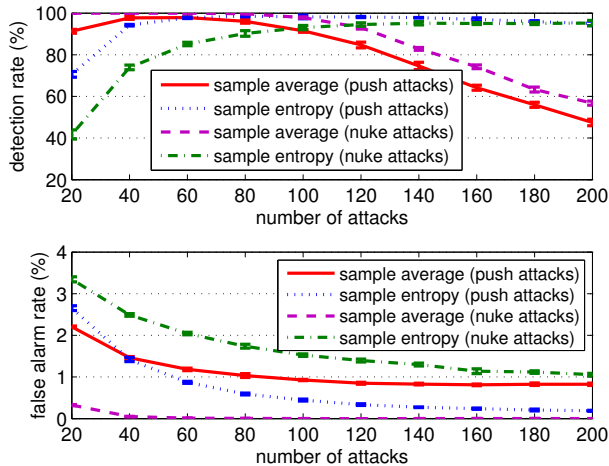


Figure 6: Detection results when the number of attack ratings in an attack event varies.

Table 2: Attack events can also be detected when the standard deviation of attack ratings (not necessarily extreme) given to an item is small.

Std	Detection rate	False alarm rate
0	98.54% \pm 0.40%	0.92% \pm 0.03%
0.2	91.10% \pm 1.29%	1.42% \pm 0.04%
0.4	76.76% \pm 0.60%	2.30% \pm 0.05%
0.6	46.28% \pm 2.11%	3.29% \pm 0.03%

size is set to 20. Five trials were performed on all the items, and results (for sample entropy) are listed in Table 3. The detection rates of these two approaches are almost the same, while the heuristic approach yields a lower false alarm rate.

We also compare these two approaches on a 24,983 users-by-100 items Jester data set. Ratings in Jester are originally continuous-valued between -10 and 10. We rounded ratings to integers and randomly permuted the ratings given to each item (because time stamps are not available). We use the same experimental setting as that of the previous experiment, and list results in Table 4, which shows that the false alarm rate decreases significantly when the heuristic approach is used. Results in Table 3 & 4 verify that our heuristic approach is effective in improving attack detection performance.

7. DISCUSSION

Our approach of attack detection is based on the assumption that the duration of an attack event is relatively short so that rating distributions of some items will be changed during that duration. We argue that this is a reasonable assumption because of the following two points. First, shilling attackers usually hope their attack ratings can take effect as soon as possible, since normal user rating patterns may also change during attacks. Second, if an attack event cannot induce considerable changes in rating distributions of target items within a period of time, then the effect of this attack event will probably be negligible overall.

Another assumption we make is that rating distributions of items are stationary. This assumption corresponds well

Table 3: Detection results (for sample entropy) of the heuristic approach (estimating the length of an attack event and adjusting the window size) versus results of using a fixed window size in MovieLens.

	Detection rate	False alarm rate
Fixed window size	97.60% \pm 0.24%	0.42% \pm 0.03%
Heuristic approach	97.48% \pm 0.29%	0.36% \pm 0.05%
Improvement	-0.12%	15.64%

Table 4: Detection results (for sample entropy) of the heuristic approach versus results of using a fixed window size in Jester.

	Detection rate	False alarm rate
Fixed window size	100% \pm 0%	1.18% \pm 0.04%
Heuristic approach	100% \pm 0%	0.25% \pm 0.03%
Improvement	0%	78.47%

to the items we have tested in MovieLens and Jester. However, in a large-scale recommender system, ratings to a given item may have a trend and/or a seasonality over time. More complex models (*e.g.*, the Auto-Regressive Integrated Moving Average [3]) are needed to incorporate such trends.

Acknowledgment: This material is based in part upon work supported by the National Science Foundation under award number IDM 0308229 and CCF 0448277, and startup funds from Dartmouth College.

8. REFERENCES

- [1] A. Antos and I. Kontoyiannis. Convergence properties of functional estimates for discrete distributions. *Random Structures and Algorithms*, 19(3-4):163-193, 2001.
- [2] G. Basarin. On a statistical estimate for the entropy of a sequence of independent random variables. *Theory of Probability and Its Applications*, 4(3):333-336, 1959.
- [3] P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting*. Springer, 2nd edition, 2002.
- [4] J. D. Brutlag. Aberrant behavior detection in time series for network monitoring. In *Proc. of the 14th USENIX Systems Administration Conference*, pages 139-146, 2000.
- [5] R. Burke, B. Mobasher, R. Bhaumik, and C. Williams. Segment-based injection attacks against collaborative filtering recommender systems. In *Proc. of the IEEE Int. Conf. on Data Mining*, pages 577-580, 2005.
- [6] P.-A. Chirita, W. Nejdl, and C. Zamfir. Preventing shilling attacks in online recommender systems. In *Proc. of ACM Int. Workshop on Web Information and Data Management*, pages 67-74, 2005.
- [7] E. Keogh, J. Lin, and A. Fu. HOT SAX: Finding the most unusual time series subsequence: Algorithms and applications. In *Proc. of the IEEE Int. Conf. on Data Mining*, pages 226-233, 2005.
- [8] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. In *Proc. of SIGCOMM*, pages 217-228, 2005.
- [9] S. K. Lam and J. Riedl. Shilling recommender systems for fun and profit. In *Proc. of the 13th WWW*, pages 393-402, 2004.
- [10] B. Mobasher, R. Burke, C. Williams, and R. Bhaumik. Analysis and detection of segment-focused attacks against collaborative recommendation. In *Proceedings of the 2005 WebKDD Workshop (Lecture Notes in Computer Science)*, 2006.
- [11] M. O'Mahony, N. Hurley, N. Kushmerick, and G. Silvestre. Collaborative recommendation: A robust analysis. *ACM Transactions on Internet Technology*, 4(4):344-377, 2004.
- [12] L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer, 2004.
- [13] S. Zhang, J. Ford, and F. Makedon. Analysis of a low-dimensional linear model under recommendation attacks. In *Proc. of the 29th ACM SIGIR*, 2006.