Detection of Fraud Ranking in Mobile Applications Using REVRANK Algorithm

S. Sabitha\textsuperscript{1}, M. Rajeshwari\textsuperscript{2} & M. Sandhiya\textsuperscript{3}

\textsuperscript{1,2,3}A.V.C College of Engineering

\textbf{Abstract:} Now a day’s application usage has become more popular. Users download their application based on rating and ranking on popularity list of particular application. The app developer who wants their application in the popularity list may request their friends to download and give fake rank/rate. This may lead to improper ranking/rating in the internet, which affects the genuine applications. Therefore preventing the fraud ranking plays a vital role. In this paper, we proposed an application to detect fraud ranking for mobile applications using REVRANK algorithm and counting sort algorithm. REVRANK algorithm identifies dominant reviews and converts this to ranking according to number of views and reviews for a particular app. Counting sort algorithms counts the number of legitimate downloads. By blending these two algorithms, genuine reviews/ratings are given to applications.

\textbf{INTRODUCTION}

Online web spam refers to all forms of malicious manipulation of user generated data so as to impede usage patterns of the data. The plenty of mobile Apps has grown at a breath taking rate over the past few years. To stimulate the development of mobile Apps, many App stores launched everyday App leader boards, which demonstrate the chart rankings of most trendy Apps. Indeed, the App leader board’s is one of the most vital methods for promoting mobile Apps. A higher rank on the leader board usually leads to large number of downloads and million dollars in the revenue. Therefore, App developers tend to investigate various ways such as advertising campaigns to promote their Apps in order to have their Apps ranked as high as achievable in such App leader boards. Hence, we have characterized a few fraud evidences from Apps’ historical ranking records, and develop three functions to mine such ranking based fraud evidences. However, the ranking based data can be affected by App developer reputation and some legitimate promotion campaigns, such as Limited Time Discount. Result is not enough to use ranking - based data. Therefore, we further propose two types of fraud data based on Apps’ rating and review history, which replicate some anomaly patterns from Apps’ historical rating and review records.

To overcome these problems, we have implemented REVRANK algorithm and counting sort algorithm. Implementing this algorithm in the websites, will predict - when the app is created and its uploaded date and time. If a user creates an application, they will distribute it to the public website. The most excellent way to do so is to upload the application to the app stores like Google Play. Similar to this, if a user uploads an application, we initially propose an algorithm to classify the leading sessions of each App based on its historical ranking records and then, it analyzing the Apps’ ranking behaviors, we found best mobile application in fraud ranking software.

\textbf{ALGORITHM}

\textbf{A.REVRANK ALGORITHM}

It is based on a collaborative principle. Given multiple reviews of a product, REVRANK identifies the most important concepts. The challenge lies in finding those concepts that are important but infrequent. The main idea employed by REVRANK is to use the given collection of reviews along with an external balanced corpus in order to define a reference virtual core (VC) review. It is not the best possible review on this product, but is, in some sense, the best review that can be extracted or generated from the given collection (hence our usage of the term ‘virtual’: the collection may not contain a single review that equals to the core review and the virtual core review may change with the addition of a individual new review to the collection). We do not generate the VC review explicitly; all reviews, including the VC one, are described as feature vectors. The feature set is the lexicon of dominant terms contained in the reviews, so that vector coordinates equals to the overall set of dominant terms. Reviews are then ranked conforming to affinity metric between their vectors and the Virtual Core vector. Our approach is inspired by classic information retrieval, where a document is described
as a bag of words, and each word in each document is assigned a score (typically tf-idf (Salton and McGill 1983)) that shines the word’s importance in this document. The document is then represented by a vector whose coordinates equals to the words it contains, each coordinate having the word’s score as its value. Similarity of vectors denotes parallel of documents. The key novelty in our approach is in showing how to define, compute and use a virtual core review to locate the review ranking problem. Our features (dominant terms) constitute a compact lexicon containing the key concepts related to the reviews of a specific product. The lexicon typically contains concepts of numerous semantic types: absolute references to the book and the plot, references to similar books or to other books by the equivalent author, and other important contextual elements. We identify this lexicon in an unsupervised and efficient manner, using a measure encouraged by tf-idf with the use of an external balanced reference corpus. There are three major differences from the classic tf-idf: First, at the status of key concepts extraction, we view the collection of reviews for a precise book as a single long document. Second, rather than computing inverted frequencies on the reviews corpus, we use an external balanced reference corpus. Finally, instead of counting the documents in the inverted corpus, we calculate the term-frequency in the external balanced corpus. The theory behind that we first want to determine the set of key concepts in reviews of a specific book, rather than directly retrieving a single document/review. Using standard tf-idf by viewing each review as a document, we would perhaps be able to discover important terms that develop in very few reviews, but we may not be able to discover central terms that are common to several reviews (since the idf tends to punish terms that emerge in more than one review). Reviews that make use of such terms are expected to be more helpful than those that does not, because in a sense the existence of such terms guarantees that the review also covers the main form of the subject at hand. The REVRANK algorithm has three simple stages, described in detail below: (i) identify dominant terms; (ii) design each review to a feature vector defined by the dominant terms; and (iii) use some affinity metric to rank the reviews.

Figure 1: Analyse of revrank algorithm

B.COUNTING SORT ALGORITHM

Counting sort is an algorithm for sorting a set of objects according to keys that are little integers; that is, it is an integer sorting algorithm. It proceed by counting the number of objects that have been each distinct key value, and using arithmetic on those counts to decide the positions of each key value in the output procession. Its running time is linear in the number of elements and the difference between the maximum and minimum key values, so it is only suited for direct use in situations where the variation in keys is not decidedly greater than the number of items. However, it is often used as a subroutine in other sorting algorithm, radix sort that can handle larger keys more effectively. Because counting sort helps key values as indexes into an array, it is no more a comparison sort, and the \( \Omega(n \log n) \) lower bound for comparison sorting does not apply to it.\(^{12}\) Bucket sort may used for many of the same tasks as counting sort, with a similar time analysis; however, correlated to counting sort, bucket sort requires linked lists, arrays or a large amount of reallocated memory to hold the sets of elements within each bucket, whereas counting sort rather stores a single number (the count of items) per bucket.

PROPOSED SYSTEM

Now- a -days mobile application should have phony App ratings. So we first propose a simple yet effective algorithm to identify the leading sessions of each App based on its classical ranking records. Then, with the analysis of Apps ranking behaviors, we find that the fraud Apps often have different ranking patterns in each dominant session compared with normal Apps. This system contains two major categories. Application upload and application download. If user may be a developer he can upload an application for the purpose of sales and must be contact with the admin under some criteria’s. Otherwise the user may download any application from any appstores. The mining process starts in the downloading session which maintains the number of applications. In this process we should maintain the two main algorithms namely Revrank algorithm, count sort algorithm. These algorithms should be used to analyze the application, how many times the application is viewed and how many times it is downloaded respectively.
ADVANTAGES OF PROPOSED SYSTEM

- The proposed framework is scalable and can be extended with other domain generated evidences for ranking fraud detection.
- Experimental results show the effectiveness of the proposed system, the scalability of the detection algorithm as well as some regularity of ranking fraud activities.
- There is no existing benchmark to decide which leading sessions or Apps really contain ranking fraud.
- Avoid ranking duplication.

CONCLUSION

This paper introduces a system which is built up and it is actually a positioning extortion discovery framework for mobile Apps. In particular, initially it is demonstrated that positioning misrepresentation is happening in driving sessions and gave a system to dig driving sessions for each App from its chronicled positioning records. At that point, it is recognized that positioning based confirmations, rating based proofs and survey based confirmations are used for identifying positioning extortion. In addition, a unique model is proposed which is an improvement based total system to incorporate every one of the proofs for assessing the validity of driving sessions from portable Apps. A novel point of view of this methodology is that every one of the proofs can be displayed by measurable theory test, in this way it is anything but difficult to be reached out with different confirmations from space information to distinguish positioning misrepresentation. At last, the proposed framework is accepted with broad examinations on certifiable App information gathered from the Apple's App store. Exploratory results demonstrated the adequacy of the proposed methodology. Later on, to concentrate more viable misrepresentation confirms and dissect the idle relationship among rating, survey and rankings is planned. In addition, amplification of positioning misrepresentation location approach is performed with other portable App related administrations, for example, mobile Apps suggestion, for improving client experience.

REFERENCES