Technique for Building a Priority Crawler

Amardeep Singh, Anshul Jain, Binny Tewani, Shobhit Goel

CSE Department, The LNM Institute of Information Technology
Rupa Ki Nangal, Post-Sumel,Via-jamdoli, Jaipur-302031, (Rajasthan) INDIA

singh.amardeep84@gmail.com
anshul.lnmiit@gmail.com
binnytewani@gmail.com
shobhitgoel01@gmail.com

Abstract— ***In*** **this paper, we put forward a technique for building a priority crawler. The World Wide Web today is growing at a phenomenal rate. Limited size data need to be chosen from the entire data set based on its quality and information provided so that later it can be used by the search engines to serve the end user queries. This technique selectively choose URL's to be kept in the limited size queue based on the data quality and information provided.**

Keywords**— priority crawler, URL dataset, Importance Matrix, limited size queue**

I. Introduction

Given Dataset contains data of 120 days with 2.4 million examples and 3.2 million features. Each Day has data of around 20000 URLs and each URL is represented as a set of up to 3.2 million features with different attributes and a class label. In this dataset, features attribute value contribute to the importance of URL. The no. of newly introduced attributes linearly increases with no. of URLs, which means that nearly every new URL introduces some new features which are not present in any previous URLs (refer Fig 2). We also noticed that in this dataset, there are some features which are not present in any URL while on the other hand, there are some feature which are present in every URL of a day's dataset. Here URLs are divided into two classes which are malicious and benign. URL with -1 class label represent that it is malicious while URL with +1 represent that it is benign. For the given dataset, we also noticed that the number of malicious URLs is much more than the number of benign URLs.

II. PROBLEM

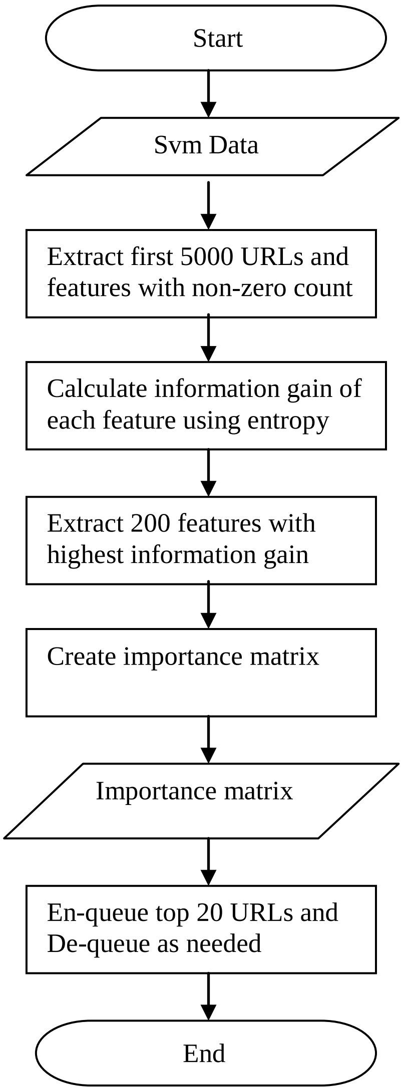
We are providing a technique for building a crawler which can access the huge URL data for each day and selectively chooses URL's to be kept in the limited size queue based on their Importance. **Crawler** will browse the data provided and select URL's to be kept in limited size queue.

.

III. MOTIVATION

Considering the increasing amount of data on the web, we need to have only quality and informative data to make it easier for the search engine to perform the queries. It was challenging to figure out which features to choose and what does features represent. We also tried it by implementing PCA but since it was already there in matlab, we tried something else. This task was interesting as we were motivated by the performance of Google to create a crawler comparable in performance to it which is a challenge.

IV. OVERVIEW


Start
Svm Data
Extract first 5000 URLs and features with non-zero count
Calculate information gain of each feature using entropy
Extract 200 features with highest information gain
Create importance matrix
Importance matrix
En-queue top 20 URLs and De-queue as needed
End

V. DESCRIPTION OF METHODS

Before applying algorithms on the dataset to determine the important URLs, it is important to pre-process the data to reduce the amount of processing time.

A. *Pre-processing*

In Pre Processing, we tried to optimize the representation and quality of data. We first extracted first 5000 URLs from the day0.svm to create our dataset. Then we calculated count of each feature in our dataset. Features whose count were zero had been removed. Now, for every feature whose count was non zero but were missing from the URL feature set, we made an entry in the dataset of that feature with attribute value zero. Now, our dataset was organized and reduced to 28266 features such that each column refers to a particular feature making it faster and easier for us to process it further.

Now, we further processed our data by removing all the features value from the dataset leaving only attributes as we no longer needed features value to determine which attribute belongs to which feature as all the attributes of a particular column for all the URLs referred to the same feature with first column referring to the class of the URL

Now to further reduce the number of features from the dataset, we decided to choose only those features which convey maximum information. For that, we decided to choose information gain as our basis to decide which feature conveys how much information. Now, to determine information gain of each feature, we calculated entropy of the entire dataset by using the Eq 1. Then, entropy of each feature is calculated. using the Eq 2. Now, difference of the value of entropy of the dataset and the entropy of the feature is information gain of that feature. Similarly we calculated information gain of all the features. Now, we sorted the values of information gain in the decreasing order and extracted the top 200 features. Then, in the dataset, for every URL, all the features except these 200 features are discarded. So now, we have a new dataset containing only 200 features attribute with highest information gain value.

B. *Heuristics and algorithms*

Heuristics we used in this technique was that in the dataset, all those features which were not present in any of the URLS were discarded from the dataset as they were not contributing any information in determining the importance of the URLs.

To determine the importance of the URLs of the dataset, we created an Importance matrix. Now, to create an information matrix, we needed to determine the importance of each URL for which we used Eq 4 in which we calculated the sum of the product of features attribute of the URL to the information gain of that feature. Now we have the importance matrix, which contain the URLs according to their importance, from which we En-queue the top 20 URL's.

C. *Mathematical Background Used*

To determine the entropy of the dataset, we used the following equation:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i \tag{1}$$

Where, *E(S)* is the entropy of the dataset, *Pi* is the probability of the class i in the dataset and c is the number of classes.

To determine the entropy of the features of the dataset, we have used the following equation:

$$E(T,X) = \sum_{c \in X} P(c) E(c) \tag{2}$$

Where, *E(T,X)* is the entropy of the feature *X* in the dataset *T*, *P(c)* is the probability of the feature c in the dataset and *E(c)* is the entropy of the dataset which contain feature *c*.

To determine Information gain of the features, we have used the following equation:

$$Gain(T,X) = E(S) - E(T,X) \tag{3}$$

Where, *Gain(T,X)* is the information gain of the feature *X* in the dataset *T*, *E(S)* is the entropy of the dataset *S* and *E(T,X)* is the entropy of the feature *X* in dataset *T*.

To determine the importance of the URLs,we have used the following equation:

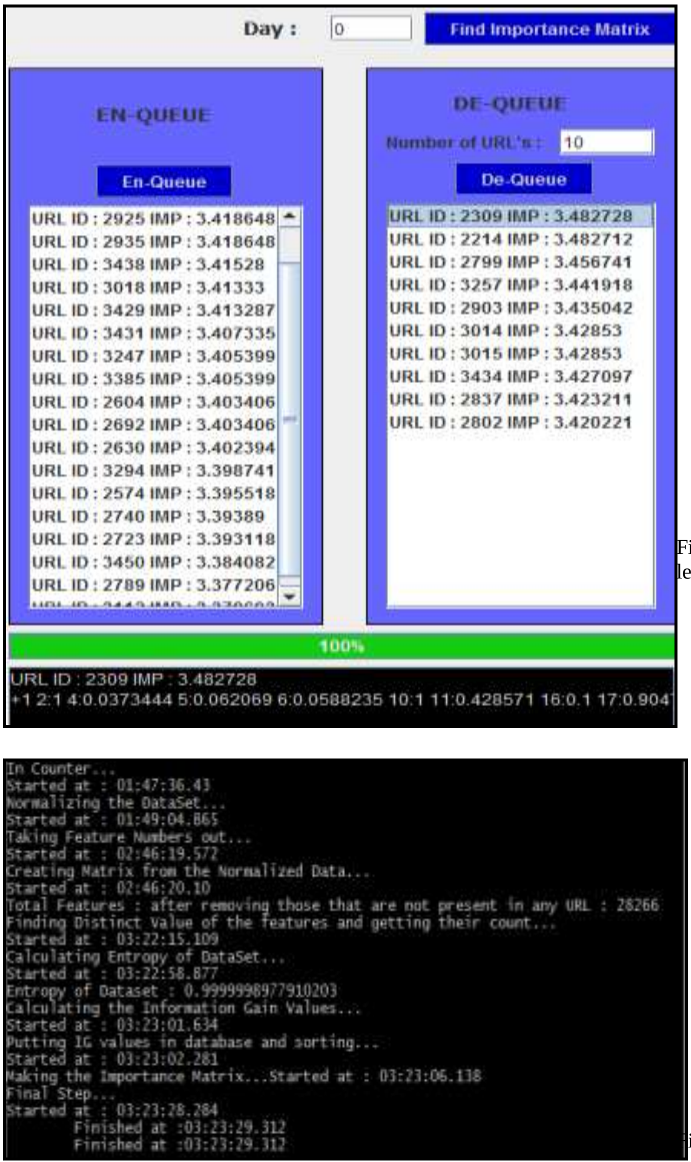
$$I(S) = \sum_{c \in X} A(c) Gain(T, c) \tag{4}$$

Where, *I(S)* is the importance of URL *S*, *A(c)* is the attribute value of feature *c* and *Gain(T,c)* is the information gain of feature *c* in dataset *T*.

D. *Additional Techniques used*

In addition to the methods stated above, we have also used some other techniques. We used MYSQL for sorting the data whenever needed and we used java-swing and AWT for designing the GUI for displaying the queued URL's as well the de-queued URL's according to the need. We used MATLAB for plotting graphs for the results obtained.

VI. RESULTS



Day : 0

Number of URL's : 10

```
URL ID : 2925 IMP : 3.418648
URL ID : 2935 IMP : 3.418648
URL ID : 3438 IMP : 3.41528
URL ID : 3018 IMP : 3.41333
URL ID : 3429 IMP : 3.413287
URL ID : 3431 IMP : 3.407335
URL ID : 3247 IMP : 3.405399
URL ID : 3385 IMP : 3.405399
URL ID : 2604 IMP : 3.403406
URL ID : 2692 IMP : 3.403406
URL ID : 2630 IMP : 3.402394
URL ID : 3294 IMP : 3.398741
URL ID : 2574 IMP : 3.395518
URL ID : 2740 IMP : 3.39389
URL ID : 2723 IMP : 3.393118
URL ID : 3450 IMP : 3.384082
URL ID : 2789 IMP : 3.377206
```

```
URL ID : 2309 IMP : 3.482728
URL ID : 2214 IMP : 3.482712
URL ID : 2799 IMP : 3.456741
URL ID : 3257 IMP : 3.441918
URL ID : 2903 IMP : 3.435042
URL ID : 3014 IMP : 3.42853
URL ID : 3015 IMP : 3.42853
URL ID : 3434 IMP : 3.427097
URL ID : 2837 IMP : 3.423211
URL ID : 2802 IMP : 3.420221
```

```
URL ID : 2309 IMP : 3.482728
+1 2:1 4:0.0373444 5:0.062069 6:0.0588235 10:1 11:0.428571 16:0.1 17:0.904
```

```
In Counter...
Started at : 01:47:36.43
Normalizing the DataSet...
Started at : 01:49:04.865
Taking Feature Numbers out...
Started at : 02:46:19.572
Creating Matrix from the Normalized Data...
Started at : 02:46:20.10
Total Features : after removing those that are not present in any URL : 28266
Finding Distinct Value of the features and getting their count...
Started at : 03:22:15.109
Calculating Entropy of DataSet...
Started at : 03:22:58.877
Entropy of Dataset : 0.9999998977910203
Calculating the Information Gain Values...
Started at : 03:23:01.634
Putting IG values in database and sorting...
Started at : 03:23:02.281
Making the Importance Matrix...Started at : 03:23:06.138
Final Step...
Started at : 03:23:28.284
        Finished at :03:23:29.312
        Finished at :03:23:29.312
```

Fig. 1 A screenshot of the GUI content showing the queue with top 20 URLs of the first 5000 URLs, with 10 URLs de-queued and processing statistics during the same process

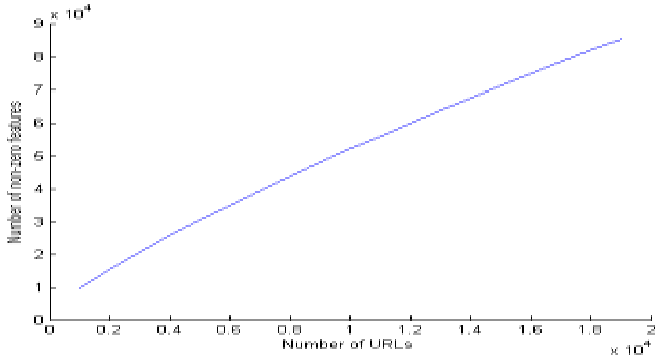


Fig. 2 A graph between Number of URLs (X-axis) and Number of non zero attributes (Y-axis) for data of Day2.svm.

No. of URLs	50	100	500	1000	2000
Features Left	1061	1765	5398	9384	14909
Uid1	6	6	6	784	28
Imp1	7.827	4.349	4.609	3.432	3.242
Uid2	38	38	291	689	1020
Imp2	7.732	4.317	4.578	3.432	3.217
Uid3	33	12	38	123	1114
Imp3	7.618	4.172	4.385	3.230	3.217
Uid4	12	33	33	98	335
Imp4	7.362	4.062	4.342	3.221	3.122
Uid5	5	85	288	382	1031
Imp5	4.932	3.968	4.342	3.15	3.094
Uid6	19	83	134	296	609
Imp6	4.544	3.841	4.342	3.107	3.080
Uid7	35	74	213	421	1233
Imp7	4.143	3.747	4.274	3.107	3.003

Fig. 3 A Table showing the changing pattern of the top 7 URLs and features left (after 1st Reduction) with the increase in no. of URLs

S.NO	1	2	3	4	5
Features taken	100	200	300	500	1000
No. of URLs	100	100	100	100	100
Uid1	28	28	28	73	73
Uid2	23	5	5	83	83
Uid3	5	83	83	50	50
Uid4	27	23	23	49	49
Uid5	83	27	27	84	84
Uid6	26	26	26	47	47
Uid7	21	21	21	85	85
Uid8	57	57	57	43	43
Uid9	90	13	13	42	42
Uid10	13	90	90	41	41
Time mm:ss	02:01	02:02	01:59	01:58	02:10

ig. 4 A table showing changing pattern of the top 10 URLs with the increase in the no. of features taken after finding IG values of features.

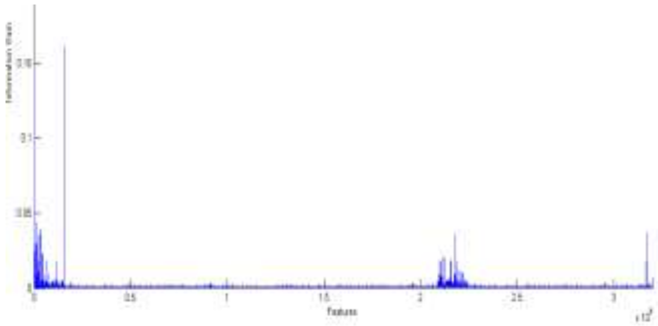


Fig. 5 A graph between features value (x-axis) and Information gain (Y-axis).

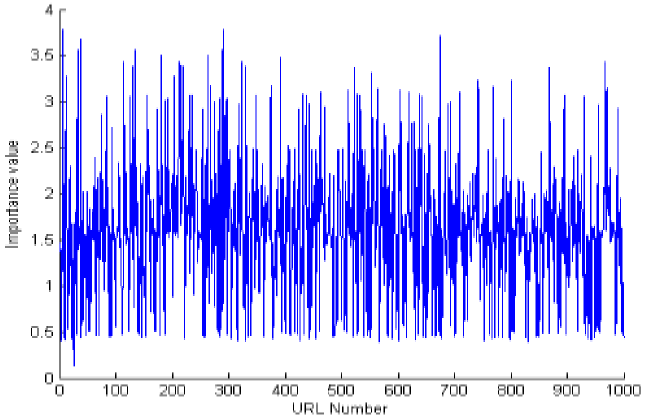
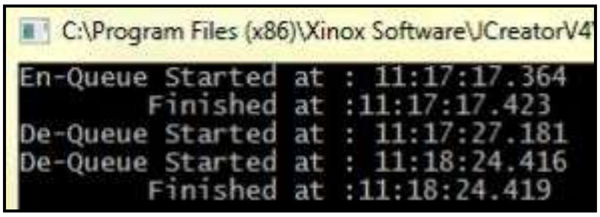


Fig. 6 A graph between URL Numbers (x-axis) and Inportance value (Y-axis).



```
En-Queue Started at : 11:17:17.364
         Finished at :11:17:17.423
De-Queue Started at : 11:17:27.181
De-Queue Started at : 11:18:24.416
         Finished at :11:18:24.419
```

Fig. 7 A screenshot displaying the en-queue and de-queue time for 5000 URL's.

INFERENCE

In fig 1, we have shown our GUI based crawler. The left list is showing the URL's which are presently in a queue and the right one is showing the list generated after de-queue. We observed that top 50 URL's belonged to class benign (i.e. +1).

As obvious from fig 2, no. of non-zero attributes linearly increases with no. of URLs which means that nearly every new URL has some new features with non-zero attribute which were zero for the earlier URLs.

In fig 4, it was observed that for the first 100 URLs, when no. of features taken after finding IG values was 500 or greater than 500, the importance of top 10 URLs was same, and for 200 and 300 features, the result was same for top 10 URLs but there were some discrepancies when 100 and 200 feature were taken, so we set our threshold to be 200, i.e. after finding IG values we are taking out 200 features.

. As we can see in fig 5, some specific range of features in the dataset convey maximum information, as information gain from them is maximum while most of the other range of features have very low information gain value. Features no. (0 to 2x10^5) and (21x10^5 to 22.5x10^5) convey max information among the dataset.

From analysing the Fig 6, we can say that Importance of URLs is uniformly distributed over the dataset as URLs with high importance are not constraint to a specific region and all regions have URLs of high importance as well as low importance.

VII. SURVEYS

1. Group Name : SAVA

We surveyed this group and according to the information they gave us, during the pre-processing phase, they reduced the number of features by considering only those features whose frequency was greater than 100.Then, they applied PCA function on the pre-processed data from which they generated the importance matrix. Then, they sorted the URLs according to their score in importance matrix and extracted the top 10 URLs.

They used the frequency of the features as the basis of determining the goodness of the features while we used information gain as our basis as there may exist features with frequency less than 100 which conveys good information so in doing so, they must have missed out many important URLs. They later applied PCA function on the pre-processed data to determine the importance of the URLs but we didn't use this method because we already tried applying PCA but later we realized that it is already there in matlab so we left it there only and calculated the importance of URLs by summing the product of information gain and features attribute. Hence, because of the difference in our pre-processed dataset, our results differ.

2. Group name : WEBSPIDERS

According to the information given to us by their group, they pre-processed the data by adding an entry for every feature with attribute 0 which were missing in the URL. For their data set, they considered the first 50 URLs with first 7000 attributes. First, they calculated entropy of the dataset and the features, and then the information gain of every feature. Then they selected the top 100 features with highest information gain and considering only these 100 feature, they generated the importance matrix from which they extracted the top 10 URLs.

For the test dataset, they chose first 50 URLs and the first 7000 features while we chose 5000 URLs and removed only those features for which frequency was zero. Later, after finding the information gain, they chose 100 features with highest information gain while we chose 200 features as we noticed in our analysis for 100 URLs, that there were some discrepancies in the result when 100 features were taken to when 200 features were taken. Since, they considered only 50 URLs while we considered 5000 URLs, there is so much difference in the range of our results.

3. Group name : STML

On surveying this team, we found out that they considered all the 16000 URLs in their dataset. They first pre-processed the data and then they calculated eigen value of every feature. Then they took out 100 features with highest eigen value. Then, they calculated the importance of each URL by summing the product of features attribute to its corresponding eigen value using which they created the importance matrix from which they later extracted the top 10 URLs.

For their dataset, they considered all the 16000 URLs while we took the first 5000 URLs. They used the eigen value based approach while we used the entropy based approach. Our results differs so much as range of our dataset is different .

4. Group name : BLITZ

This group on being surveyed said that they pre-processed the data by choosing first 100 URLs and by removing those features that are present in every URL and also those present in none of the URL.Then they applied gini index to select the most informative features and removed the non contributing features. They later applied important matrix ranksum to determine the 10 most important URLs

They chose the first 100 URLs as their dataset while we worked on first 5000 URLs. They applied gini index to determine the importance of feature while we used information gain as our basis to select features. Hence because of all these reasons stated above there is difference in our results.

5. Group name :A3M

On being surveyed, they told us that they selected their dataset by first applying random sampling on all the 16000 URLs of day0.svm dataset by which they reduced the size of dataset to 5040 URLs. Then they removed all the features from the dataset whose frequency was 0 and added entry for missing feature with attribute 0 whose frequency was non-zero. Then they calculated the information gain of the features using the entropies of the dataset and the features. Then they used information gain to select 100 features with highest information gain using which they created importance matrix from which they extracted the most important URLs.

Their dataset contained 5040 URLs as the sampled data of the 16000 URLs while our dataset contained first 5000 URLs. They used first 100 features with highest information gain to calculate the importance of the URL while we used 200 features with highest information gain as we noticed in our analysis of first 100 URLs that there were some discrepancies in the result when 100 features were taken to when 200 features were taken. Hence our results differ because of the difference in dataset as well as because of the difference in the number of features considered for each URL.

References

1. http://en.wikipedia.org/wiki/Entropy_(information_theory).
2. Efficient Crawling Through URL Ordering by
Junghoo Cho, Hector Garcia-Molina and Lawrence Page.
3. Pattern Recognition class notes.
4. http://en.wikipedia.org/wiki/Decision_tree_learning
5. Pattern Classification by Richard O. Duda,, Peter E. Hart, David G. Stork
6. Data Mining: Concepts and Techniques, 2nd ed. By Jaiwei Han, Micheline Kamber.