Discovering Authorities as a function of time in Community Question Answering

Rajeev Ranjan

A Thesis Submitted to
Indian Institute of Technology Hyderabad
In Partial Fulfillment of the Requirements for
The Degree of Master of Technology

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Department of Computer Science and Engineering

June 2018

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

Rajeev Ranjan

(Signature)

(Rajeev Ranjan)

CS16MTECH11012

(Roll No.)

Approval Sheet

This Thesis entitled **Discovering Authorities as a function of time in Community Question Answering** by Rajeev Ranjan is approved for the degree of Master of Technology from IIT Hyderabad.

Maunendra.

(Maunendra Sankar Desarkar) Examiner
Dept. of Computer Science and Engg.
IITH

U.R

(U.RAMAKRISHNA) Examiner
Dept. of Computer Science and Engg.
IITH

(Dr. Subrahmanyam Kalyanasundaram) Adviser
Dept. of Computer Science and Engg.
IITH

~~( ) Co-Adviser~~
~~Dept. of Computer Science and Engg.~~
~~IITH~~

M.V.Pandurg R.

(M.V.PANDURANGA RAO) Chairman
Dept. of Computer Science and Engg.
IITH

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Subrahmanyam Kalyanasundaram for helping me to decide the problem to work on. This was a bigger problem than solving the problem itself. His suggestions, feedbacks, and constructive criticisms have been of utmost importance and value to me in order to sail through this journey of pursuing Master's degree.

I would also like to thank Dr. Manish Singh who has always been extremely helpful by sharing his views and thoughts whenever required through the course of my journey.

Finally, I would like to acknowledge contributions by Dr. Ramakrishna Upadrasta in the form of valuable suggestions during my past work demonstrations.

Abstract

Community Question Answering **(CQA)** websites such as *Stack Overflow* provide a great platform to ask questions and get answers. Such platforms serve the purpose of helping the community of people who look for answers. The platforms thrive as a result of a small group of people known as ***experts*** who provide quality answers to question posters.

There has been a lot of work to determine experts in CQA websites, make predictions about potential experts and predicting how their answering behaviour varies over time. There may be a situation where some experts were aggressive contributors to the community at some point in time. But gradually their contributions started to diminish and they may also finally stop contributing in future.

We have presented an approach to rank the experts in the community as a function of time. Expertise score of an expert shows how good the expert is. Higher the expertise score, better expert the person is. Our metrics to calculate the expertise score are modifications of the existing expertise ranking metrics such that it incorporates durations of inactivity of the expert in the community while ranking them.

Contents

Chapter 1

Introduction

CQAs are one of the major sources of relevant information which can be fruitful and accessible to people associated in some way with the respective field in the community.

Popular CQAs like Quora or Stack Overflow are such platforms with huge repositories of such fields and gigantic amount of information in each of the fields. The information on such platforms is particularly in the form of someone posting a question and few people answering it.

There exists a small community of users who are experts in their fields of interest. Such users are the ones giving quality answers to the questions posted in their fields of interest or expertise.

Link analysis algorithms [1, 2] such as HITS have played an important role in the ranking of web pages. HITS algorithm runs on the web graph and generates two values for each of the vertices in the graph. The values for each of the vertices in the graph are referred to as ***hub(H) scores*** and ***authority(A) scores***. Good authorities are those vertices in the graph which are good sources of information. Good hubs are vertices in the graph which link to many good authorities.

Application of such web page ranking algorithms in the graph representing the vertices as users of the community and edges as the relationships among users can give us two values for each user; hub and authority values. Higher the authority value of the vertex implies higher expertise level of the particular user. The experts hence are also referred to as authorities. Experts and authorities have been used interchangeably in this document and they mean the same.

The answer contribution pattern of experts varies over a period of time. There are a lot of instances where experts answer contribution frequency starts to decline over time and it may also happen that the user becomes completely inactive after a certain period of time. In Parallel, there are many new users or ***not so expert*** users who remain active by contributing to the community. Their rate of contribution may increase significantly with time.

Assume a situation where some experts were really aggressive contributors at some point in time. But gradually their contributions started to diminish and they finally stopped contributing (say after two years). But there remain some users who are not experts right now but they are active in the community for quite long. Even after a year or two has passed, the inactive experts will still show up as top position holders among experts when ranked using the existing web page ranking algorithms like HITS. At the same time, users who were not experts but are constantly contributing to the community gain little in terms of expertise score and recognition as they are suppressed by inactive experts whose expertise scores were very high in the past.

We have tried to come up with an idea of calculating the expertise score of the contributors in the community as a function of time so that people who are consistent contributors get proper recognition when ranked.

The idea is to penalize expertise scores of the experts who are inactive over a significant amount of time. This will make sure that expertise scores of people who are active are pushed up in the ranking. While referring to the people in the community who post answers and ask questions, the term ***user*** has also been used in this document. The terms ***user***, ***vertex***, ***node*** mean the same in this document and have been used interchangeably.

Chapter 2

User Relationship Graph

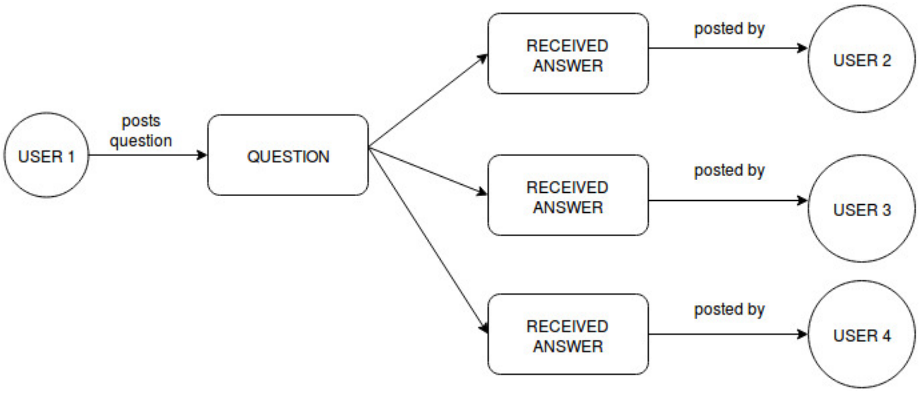


Figure 2.1: User relationship graph.

The relationship graph in Fig. 2.1 can be transformed into a simpler unweighted directed graph as shown in Fig. 2.2. Its worth noting that the sources of information or the answer posters have incoming edges towards them.

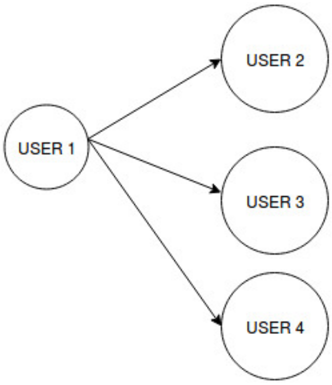


Figure 2.2: A simple directed graph.

Chapter 3

Existing Metrics for Expert Evaluation

3.1 HITS

3.1.1 How HITS works?

Consider Fig. 3.1 that may represent user relationship graph in a CQA. The value inside the square boxes in the figure represents the vertex number of the vertex in the graph. The set of vertices in the graph can be represented using V and set of edges can be represented using E. The HITS algorithm defines two values associated with every vertex in the input graph. They are the hub (H) and authority (A) values defined below. H(i) represents the hub score for vertex i. Similarly, A(i) represents the authority score for vertex i.

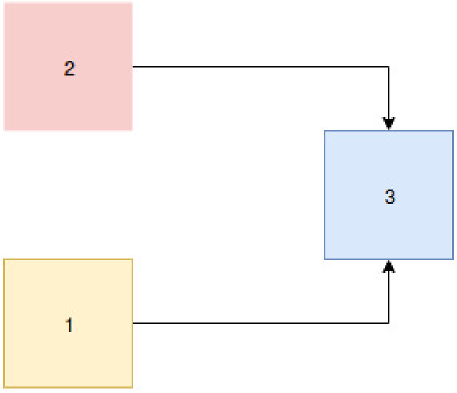


Figure 3.1: A directed graph (G say).

$$H(i) = \sum_{(i,j) \in E} A(j) \tag{3.1}$$

$$A(i) = \sum_{(j,i) \in E} H(j) \tag{3.2}$$

The above defined scores are calculated for every vertex in the graph recursively. In order to prevent the values of hubs and authorities to be large values, they are normalised after every iteration so that both the values are upper bounded by 1.

3.1.2 Initialisation

Figure 3.2 shows the initialisation of HITS algorithm. The values for vertices are represented as (H(i), A(i)) pair. The initial hub values for all the vertices are taken as 1 and all the authority values are taken as 0.

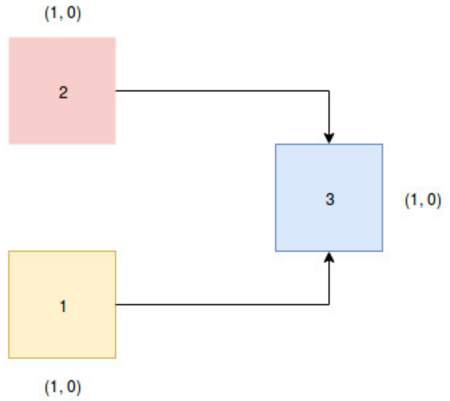


Figure 3.2: Initialisation of HITS.

3.1.3 Iterations

Iteration 1

Update authority value for each of the nodes. Fig. 3.3 shows the required update.

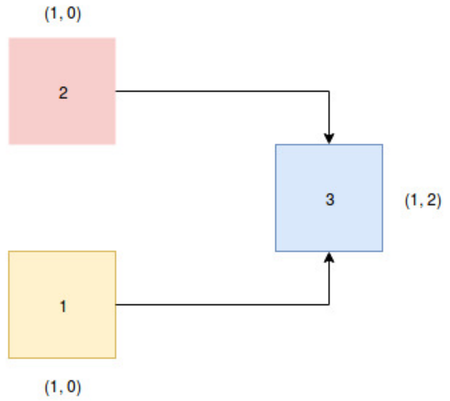


Figure 3.3: Authority score update.

Update hub value for each of the nodes. Fig. 3.4 shows the required update.

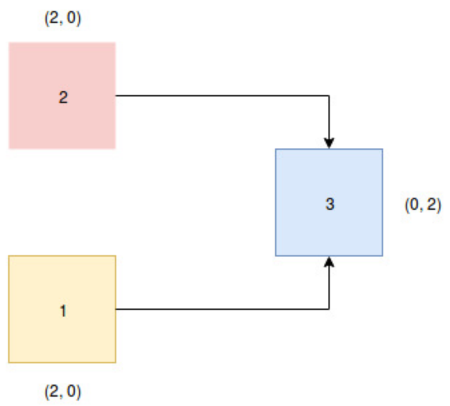


Figure 3.4: Hub score update.

The hub vector referred to as $\boldsymbol{u}$ and the authority vector referred to as $\boldsymbol{v}$ after the first iteration are shown in Tables 3.1 and 3.2 respectively.

Table 3.1: Hub scores after first iteration.

Vertex	H(i)
1	2
2	2
3	0

Table 3.2: Authority scores after first iteration.

Vertex	A(i)
1	0
2	0
3	2

The procedure described above is repeated several times after normalizing the scores at the end of each iteration. The final scores are obtained once the values are stabilized. Refer [9] for more details.

Alternatively, the above explained algorithm can be transformed into a series of matrix multiplications. The alternate approach is described below.

HITS Algorithm

- Create an adjacency matrix for the input graph (A say).
- Generate transpose of the matrix A, A^T.
- Create the hub vector , u which is a column vector that contains 1 for each row.

- Create the authority vector , v = $A^T * u$.
- while vectors u, v have not converged :
    - Update the hub vector, u = $A * v$.
    - Update the authority vector, v = $A^T * u$.

3.2 Z - Score

The metric Z - Score for a particular user or a vertex in the relationship graph is defined as shown in equation (3.3). n_a is the number of answers posted by the particular user and n_q is the number of questions asked by the corresponding user.

$$Z - Score = \frac{n_a - n_q}{\sqrt{(n_a + n_q)}}. \tag{3.3}$$

Z - Scores calculated for every vertex in the graph shown in Fig. 3.5 is illustrated below.

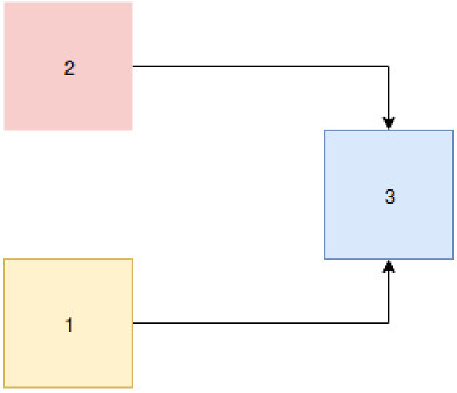


Figure 3.5: A directed graph (G say)

Z-Score for vertex 1 = $\frac{0-1}{\sqrt{(0+1)}}$ = -1.
Z-Score for vertex 2 = $\frac{0-1}{\sqrt{(0+1)}}$ = -1.
Z-Score for vertex 3 = $\frac{2-0}{\sqrt{(2+0)}}$ = $\sqrt{2}$.

3.3 Degree

This metric counts the number of incoming edges for a particular vertex in the graph. Degree score for all the vertices in Fig. 3.5 is calculated below.

- degree(v_1) = 0.
- degree(v_2) = 0.
- degree(v_3) = 2.

Chapter 4

User Feedback Metrics

Below mentioned metrics are the ones that are based on the feedbacks that an answer poster receives on a particular answer.

4.1 %Best

%Best = percentage of answers selected as ***'Best Answer'*** by the questions poster. 'Best Answer' is a tag present in the dataset for a particular answer in case it is chosen as such by the question poster.

4.2 Votes

Equation (4.1) shows the way to compute vote score for every user. $\boldsymbol{n_\mathbf{u}}$ denotes the number of upvotes and $\boldsymbol{n_\mathbf{d}}$ denotes the number of downvotes for the particular user.

$$votes = \frac{(n_u - n_d) * (\text{total \% of upvotes})}{\text{all answers attempted}} \tag{4.1}$$

Chapter 5

Comparison

One set of metrics consists of the expertise evaluation metrics based on some algorithms described in Chapter 3, while another set of metrics consists of user feedback metrics described in Chapter 4. Metrics in one set are compared against the ones in another set to obtain some conclusion. The comparison can be done using the Pearson Correlation coefficient, ***r*** between each possible pair of metrics by choosing the two from different sets.

Pearson Correlation coefficient is a measure of the linear correlation between two variables X and Y. It has a value between +1 and 1, where +1 is a total positive linear correlation, 0 is no linear correlation, and 1 is a total negative linear correlation.

The Pearson Correlation, r is defined below. Refer [8] for more details.

$$r = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}} \tag{5.1}$$

- x_i, y_i are individual sample points indexed with i.
- n is the sample size.

Chapter 6

Extension of the Existing Work

6.1 Forgetting Curve

Performance of a person after a certain time t can be approximated mathematically using an exponential function of time. Equation (6.1) captures the performance or the retention capability of the user at present who at some point in time was associated with the field of interest and 't' units of time have passed since then. More details can be found in [7].

$$P = Ae^{-Bt}; where \tag{6.1}$$

- A and B are constants.
- P is the performance of the person or the retention capability after t units of time have passed since the person lost contact with the area of interest.

One time unit has been taken as $\boldsymbol{w}$ days. A person is an expert till and until he/ she is constantly in touch with the gained knowledge. If not, the gained knowledge starts to fade away slowly with time. In the models discussed earlier, this point has not been taken into account to determine the score of an expert. According to the models discussed earlier, a person who was an expert ten years ago may still show up as an expert if ranked now even if the person may have lost his / her expertise completely.

6.2 Application of the Forgetting curve

The forgetting curve's equation (6.1) can be used to generate the expertise scores of users depending upon their activity or inactivity in the community.

The activities (answer and question posting in the community in particular) in the dataset have been sorted in descending order based on the date of creation of such activity. It is worth noting that the navigation over the sorted dataset begins with the latest activity in the dataset and ends with the first activity. This holds true for the basic and the distributed models described in this work further. The terms bucket and time bucket will be used interchangeably and they mean the same in this context.

The models described further illustrate how the forgetting curve equation (6.1) has been applied to them over the sorted and bucketed timeline of activity.

6.3 Proposed Models

6.3.1 Basic Retention Model

In the basic retention model, the first step is to compute the value of P (Ref. equation (6.1)). The value of P can be calculated as explained below.

Time is noted when the user last posted an answer. Let d be the difference of days between the time of the last activity in the community and the time of last answer posted by the particular user. The value of t in equation (6.1) can be calculated as shown below.

$$t = \frac{d}{w}$$

The values for A and B both in equation (6.1) have been fixed to 1 for this model. The retention value (P) which is a fraction and a function of time itself is used in the already available metrics in order to incorporate time in the expertise evaluation.

The calculated value of P is directly multiplied by the existing metrics in order to generate the expertise score. The prefix ***Basic*** or ***b*** refers to the model for which the metrics have been proposed. The modified metrics for this model are defined below.

- Basic HITS (b-HITS) score = P * HITS score.
- Basic Degree (b-degree) score = P * degree values.
- Basic Z - Score (b-Z-Score) score = P * Z-Score values.

6.3.2 Distributed Retention Model

In the distributed model, The first step is to compute P. The highest value for the parameter 'A' in equation (6.1) is 1 for this model. The value of A is distributed over the entire set of buckets as explained below while keeping B fixed at 1, thus calculating the value of $\mathrm{P_i}$ for each bucket i separately. The value of t for this model is just the count of the number of time buckets passed as we process the current time bucket when compared to the latest time bucket of the sorted timeline of the community.

$$\text{The final value of P for a user} = \sum P_{\mathrm{i}}$$

The final expertise score is the multiplication of values obtained by existing metrics and P.
If the person was active in the latest bucket, A will be 1 for that bucket for the particular user. If we calculate the score for somebody who was active between second and third buckets (the latest bucket is the first bucket), A will be 0.95 in such a case and so on. 'A' will decrease by 0.5 for every bucket as we move away from the latest bucket. The reason behind distributing the value of 'A' in such a way is that as we move away from the latest bucket towards past buckets, the points scored by the user for a past bucket should contribute lesser as compared to the points scored in the present bucket to the final expertise score of the user.

Metrics for this model has been named in the similar way as modified metrics have been named in the sub section 6.3.1. The prefix ***Distributed*** or ***d*** refers to the model for which the metrics have been proposed. The metrics for this model are mentioned below.

- Distributed HITS (d-HITS) score = P * HITS score.
- Distributed Degree (d-degree) score = P * degree values.
- Distributed Z - Score (d-Z-Score) score = P * Z-Score values.

6.3.3 New Model (M)

In this model, we have created the relationship graph as a weighted directed graph. The weight of the edges in the graph is defined in such a way that it represents timed score values. The detailed description about the graph below is more informative.

Proposed Relationship Graph

The proposed relationship graph is a weighted directed graph which is constructed out of the user relationship graph of the CQA. The weight $\in [0, 1]$ of an edge in the graph represents how active or inactive the user is in the community. Weight closer to 1 signifies the user being highly active, while weight closer to 0 signifies the user being highly inactive. The directed edge (u, v) represents user u has answered a question posted by user v.

Data preparation

- Sort the activities (answers and questions taken together) based on time of creation in increasing order to generate the sorted dataset.
- Divide the sorted dataset into time buckets of w days. Let us call this period of w days as one time unit.
- Let the number of time units be $\boldsymbol{n}$.

Symbols and notations used in the algorithm below

- G(V, E) represents a graph where $V = \{\text{vertices in the graph}\}$ and $E = \{\text{Edges in the graph}\}$.
- *e'* represents an arbitrary edge in the graph.

- wt(u, v) represents weight of the edge (u, v) in the graph.
- P is used as a multiplication factor. Refer equation (6.1).

Calculation of P

The symbols and notations in this section only refer to equation (6.1). The values for A and B for this model have been taken as 1. The value of t will be either 0 or 1 in this case. This is due to the fact that the weight of edges is updated for each w. Based on this information, the value of P can be calculated. The graph creation algorithm below will illustrate this fact more clearly.

Graph Creation

Algorithm

- Create an empty graph G.
- Count = 0.
- Begin with the first w in the sorted dataset.
- S = {}
- While count $\leq n$:
    - Add new *e'* in the form of (u, v) and wt(u, v) = 1 to G, where user u has answered a question during w, posted by a user v.
    - S = {x : $\nexists$ an edge (x, y) such that x $\in V$, $\forall y \in$ {sorted dataset} during the current w}.
    - $\forall x \in$ S, $\forall z \in V$, wt(x, z) = P * wt(x, z).
    - count = count - 1.
    - S = {}
    - Move to next w.

6.3.4 Analysis of the obtained graph

For any such user who is active during every time bucket, the outgoing edges from that particular user are never multiplied by the multiplier, P. Hence, for every outgoing edge out of that vertex, the weight of the edge will be 1.

The user will be penalised for every w it is inactive for. The weight of the edges associated with the user will be decreased every time it is multiplied by the multiplier. The weight of the edges hence corresponds to how active or inactive the users are in the community.

Chapter 7

Proposed Metrics for the model M

7.1 modified-HITS (m-HITS)

In the above created graph, the outgoing edge weight quantifies the amount of help a user has offered. So, we have drawn a one to one correspondence between the amount of help offered by a user to a user being informative.

A better informed person is expected to offer more help to the community. Based on the above correspondence, We have proposed a modified version of HITS algorithm.

7.1.1 Modified HITS algorithm (m-HITS)

- Create adjacency matrix for the input graph (A say).
- Find transpose of the matrix A^T.
- Create the authority vector, v that contains sum of weights of all the outgoing edges for every vertex in the graph. This is a column vector.
- Create the hub vector , u = A * v.
- while vectors u, v do not converge :
  - Update the authority vector, v = A^T * u.
  - Update the hub vector, u = A * v.

7.2 modified-Z-Score (m-Z-Score)

For each vertex in the graph, the proposed modified Z-score metric for a vertex *i*, m-Z-Score(i) is defined in the equation (7.1).

$$m - Z - Score(i) = \frac{\sum_{(i,j)\in E} weight(i,j) - \sum_{(j,i)\in E}(1 - weight(j,i) * \beta)}{\sum_{(i,j)\in E} weight(i,j) + \sum_{(j,i)\in E}(1 - weight(j,i) * \beta)} \tag{7.1}$$

As compared to the original Z-Score metric, here the number of answers for a particular user is considered to be equivalent to $\sum_{(i,j)\in E} weight(i,j)$, and the number of questions is considered to be equivalent to $\sum_{(j,i)\in E}(1 - weight(j,i) * \beta$. β is a parameter. It has been set to 1 for the current work.

7.3 modified-Degree (m-Degree)

For every node i in the graph, m-degree(i) is defined as shown in the equation (7.2).

$$m - Degree(i) = \sum_{(i,j)\in E} weight(i,j) \tag{7.2}$$

Chapter 8

Applications

The models proposed can be applied to any of the CQAs and experts with their expertise scores can be computed.

Apart from applying it to the CQA dataset as a whole, various communities and sub-communities in the dataset can be extracted and then the application can be done to them. For example, a dataset for Computer Science and Engineering can have communities for data structures, Java, machine learning or other computer science topics.

In this work, We have used Louvain Method for community detection in order to detect communities in the dataset. It is available as a python package. The proposed models have been applied to the communities thus extracted and more refined results have been obtained.

Application of the models in such communities can help us determine experts within a particular community (For ex. people who are experts in java or c++) in a dataset. Such applications can give us more meaningful results.

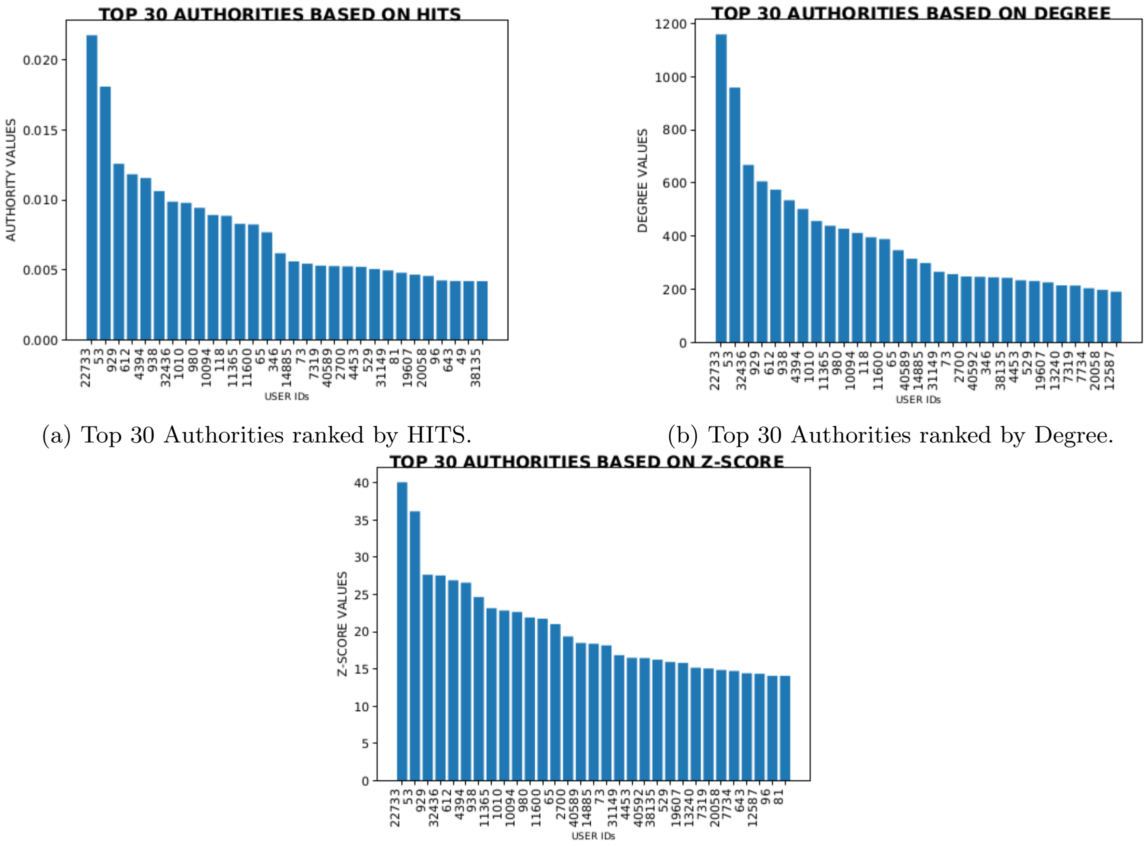
Chapter 9

Experimental Results

Dataset Source : https://data.stackexchange.com/

Dataset : askubuntu.com

9.1 Results obtained after implementing the existing work



(a) Top 30 Authorities ranked by HITS.

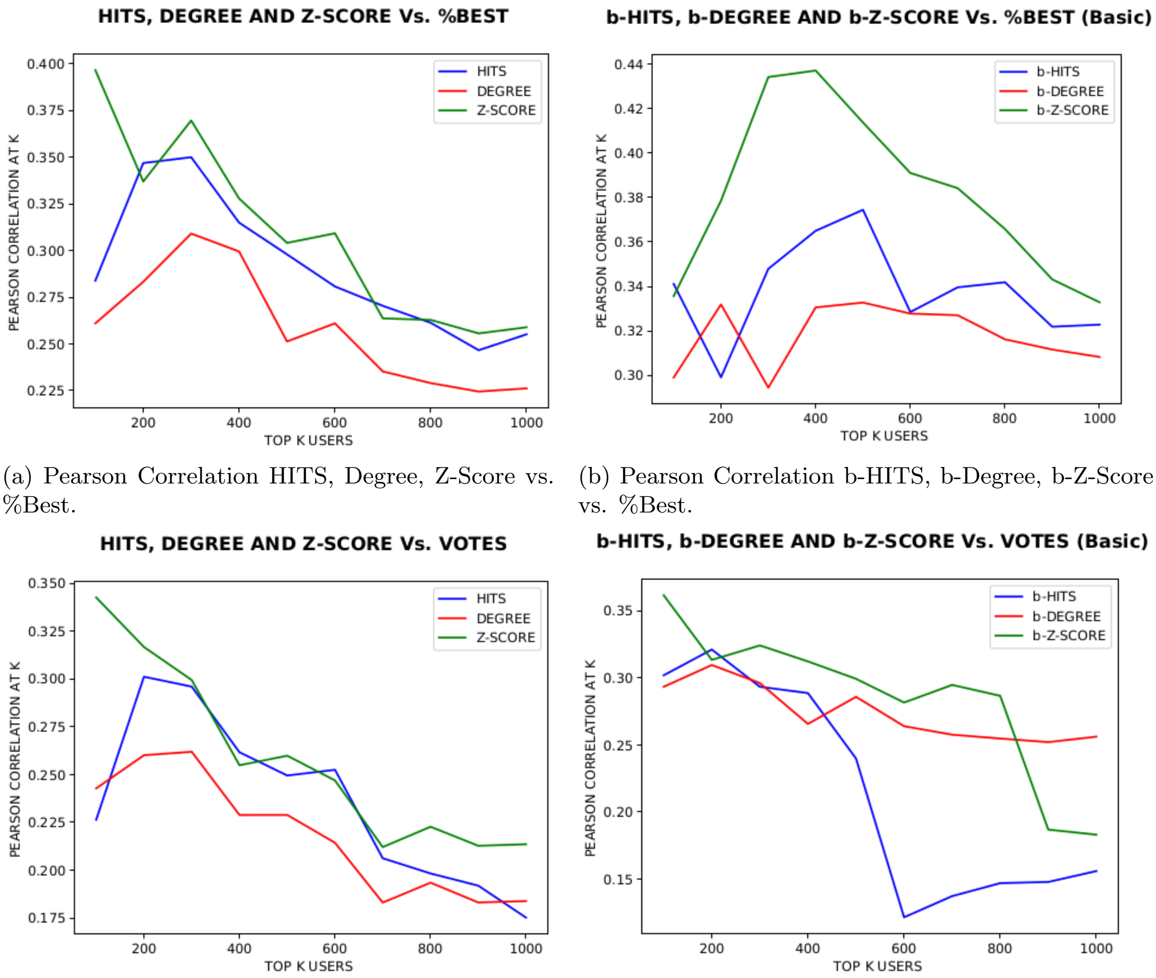
(b) Top 30 Authorities ranked by Degree.

(c) Top 30 Authorities ranked by Z - Score.

Figure 9.1: Expert Ranking based on HITS, Degree and Z-Score.

The results shown in Fig. 9.1 are the ranks of various experts who have been ranked based on the existing metrics.

9.2 Results obtained for Basic Model



(a) Pearson Correlation HITS, Degree, Z-Score vs. %Best.

(b) Pearson Correlation b-HITS, b-Degree, b-Z-Score vs. %Best.

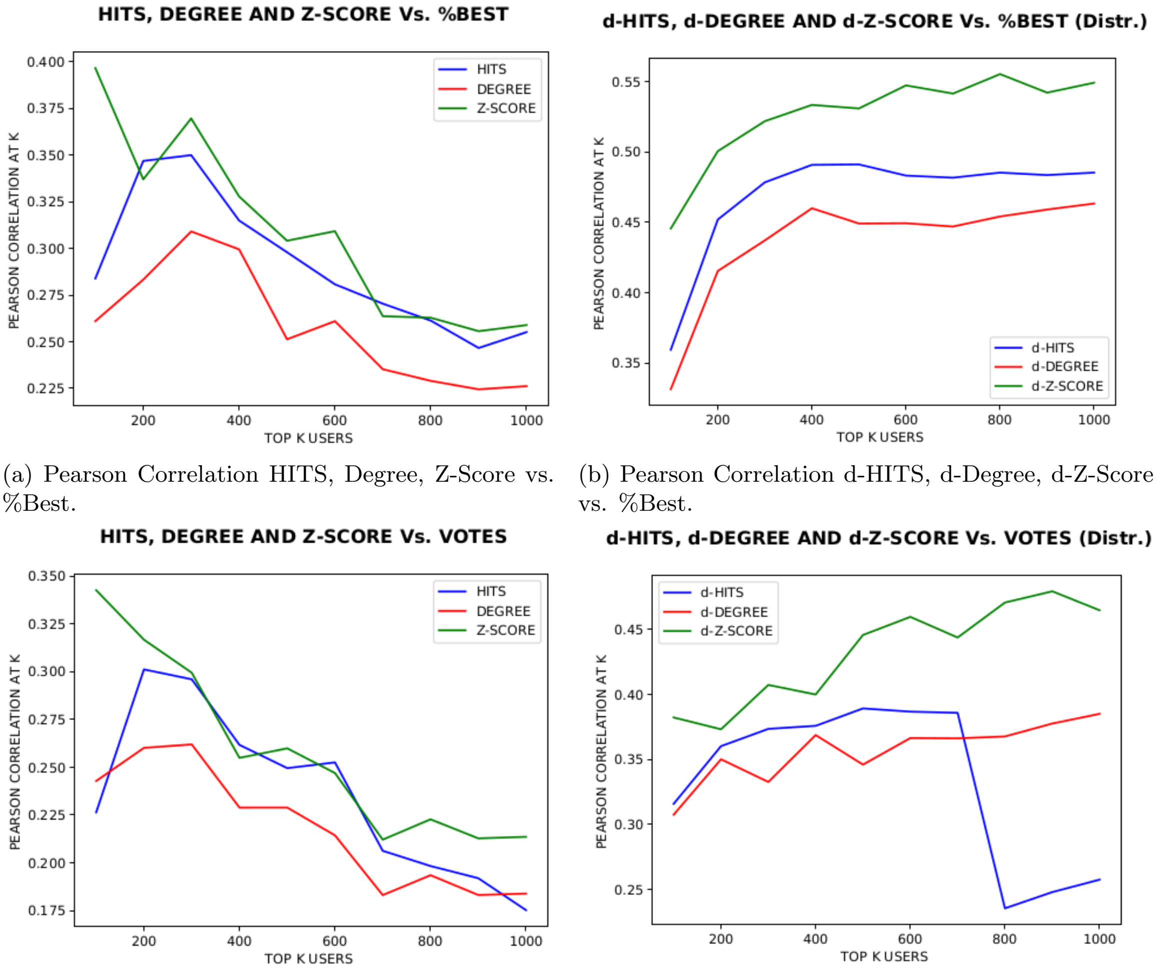
(c) Pearson Correlation HITS, Degree, Z-Score vs. Votes.

(d) Pearson Correlation b-HITS, b-Degree, b-Z-Score vs. Votes.

Figure 9.2: Pearson Correlation of existing and modified metrics against %Best and votes for the basic model.

Results shown in Figs. 9.2(a) and 9.2(c) are obtained after implementing the existing work. The Figs. 9.2(b) and 9.2(d) show results obtained after implementing the proposed basic model.

9.3 Results obtained for Distributed Model



(a) Pearson Correlation HITS, Degree, Z-Score vs. %Best.

(b) Pearson Correlation d-HITS, d-Degree, d-Z-Score vs. %Best.

(c) Pearson Correlation HITS, Degree, Z-Score vs. Votes.

(d) Pearson Correlation d-HITS, d-Degree, d-Z-Score vs. Votes.

Figure 9.3: Pearson Correlation of existing and modified metrics against %Best and votes for the distributed model.

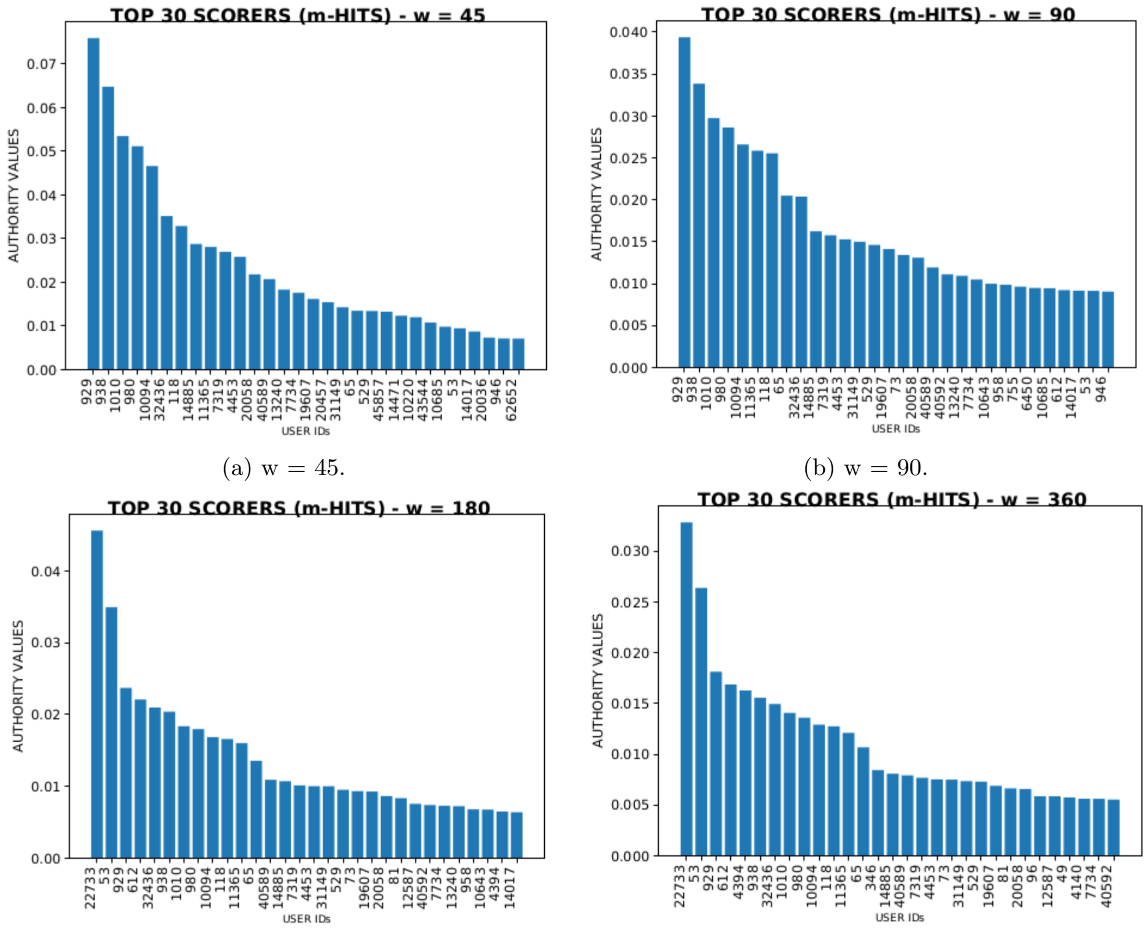
Results shown in Figs. 9.3(a) and 9.3(c) are obtained after implementing the existing work. The Figs. 9.3(b) and 9.3(d) show results obtained after implementing the proposed distributed model.

9.4 Results obtained for New Model(M)

w in the figures below refer to the number of days taken as one time unit.

9.4.1 Results obtained for the entire dataset taken as one community

Ranking of experts based on m-HITS by varying w



(a) w = 45.

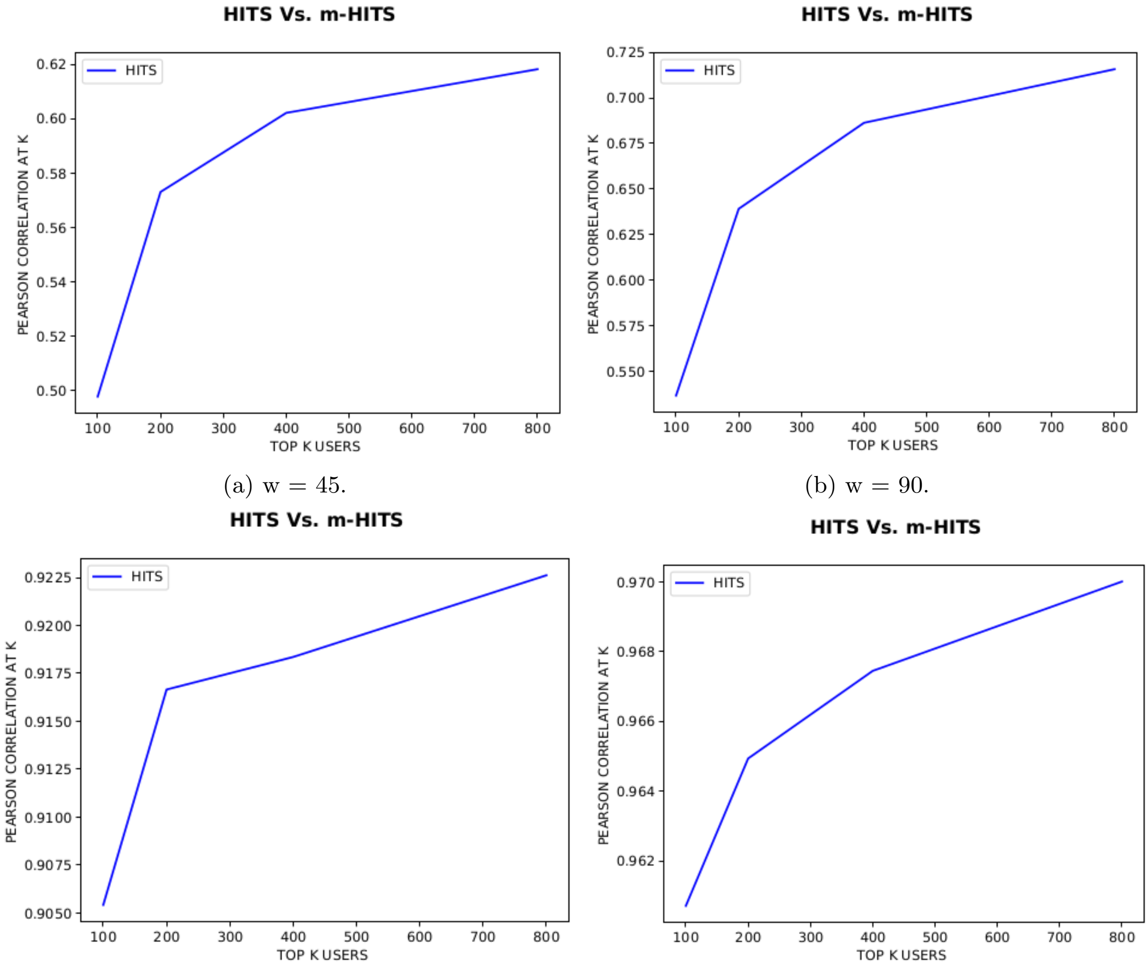
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.4: Ranking of experts on the basis of m-HITS for different w.

Pearson Correlation between HITS and m-HITS for different w



(a) w = 45.

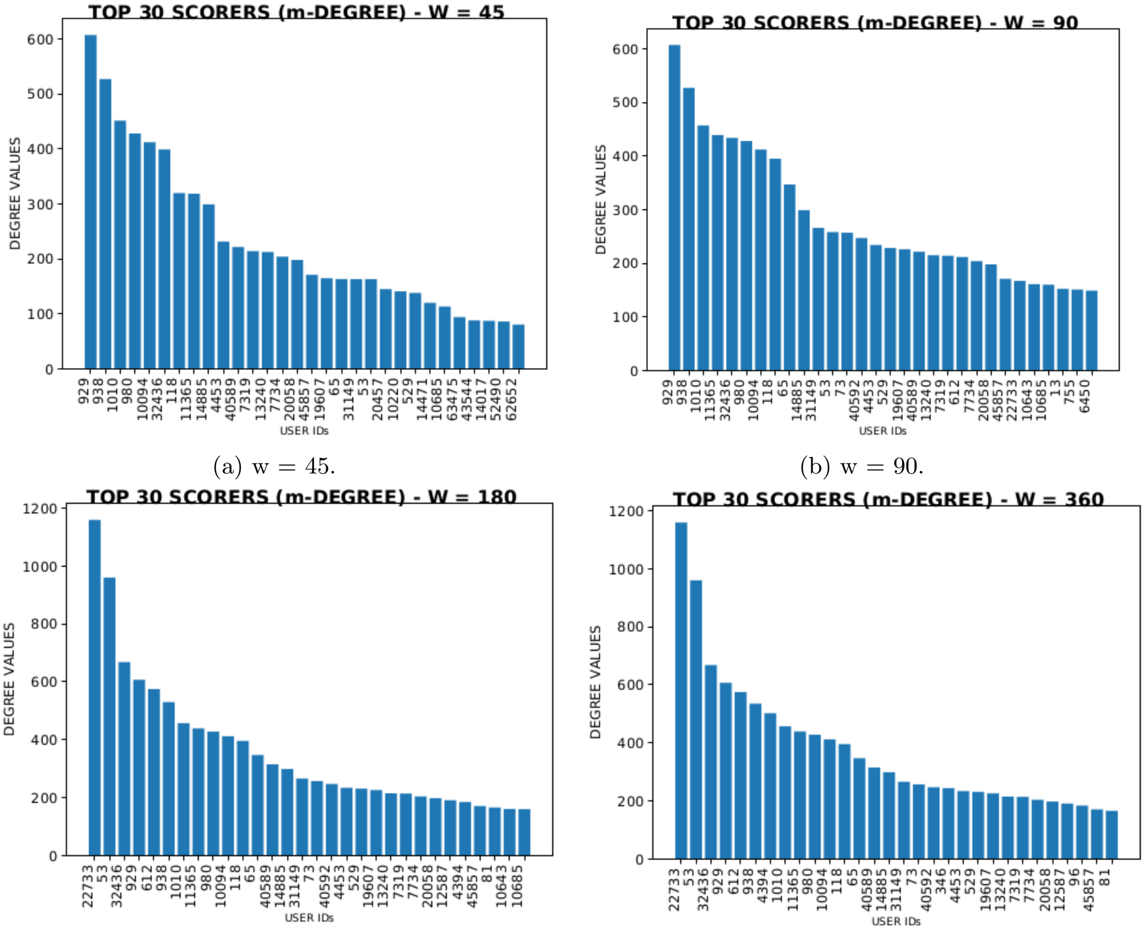
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.5: Pearson Correlation, HITS vs. m-HITS for different w.

Ranking of experts based on m-Degree by varying w



(a) w = 45.

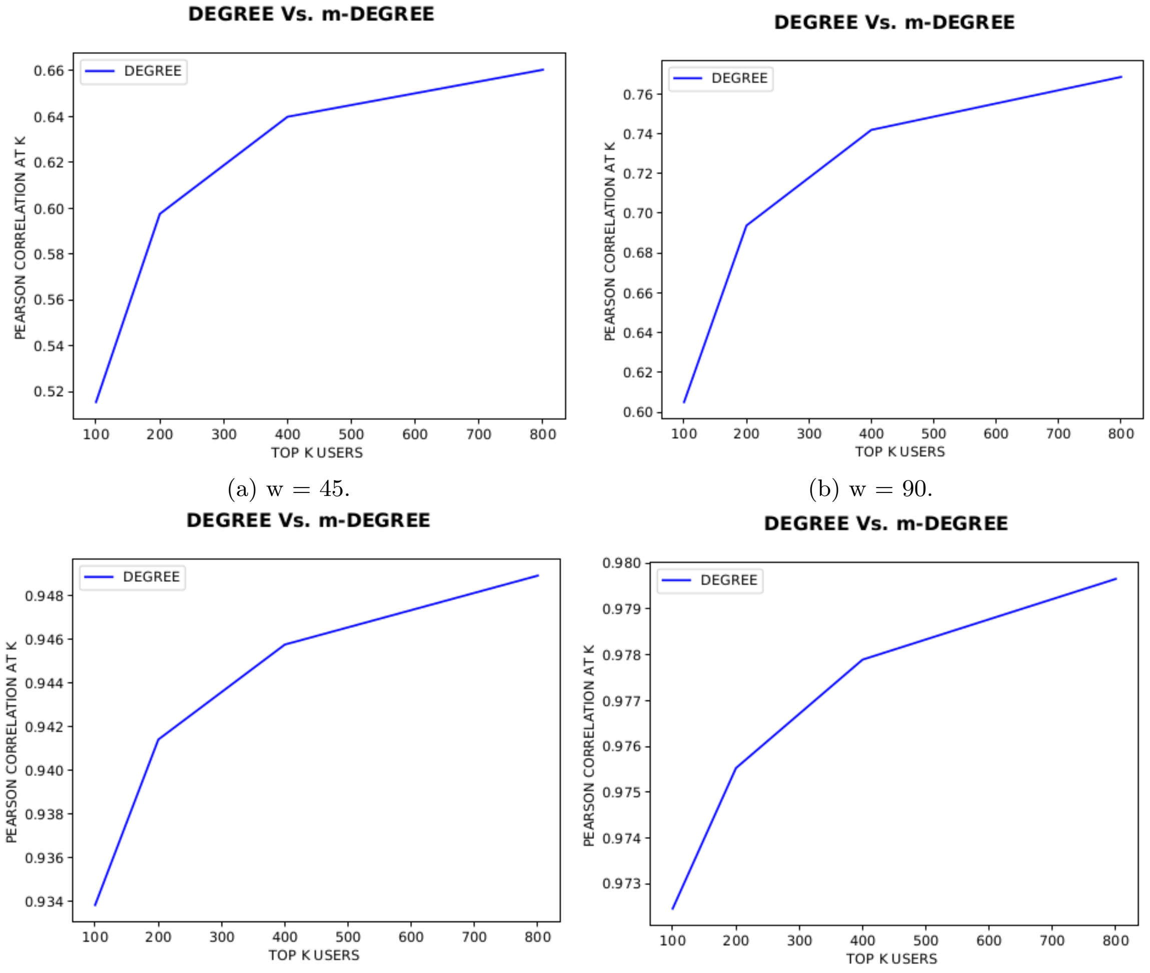
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.6: Ranking of experts on the basis of m-Degree for different w.

Pearson Correlation between Degree and m-Degree for different w



(a) w = 45.

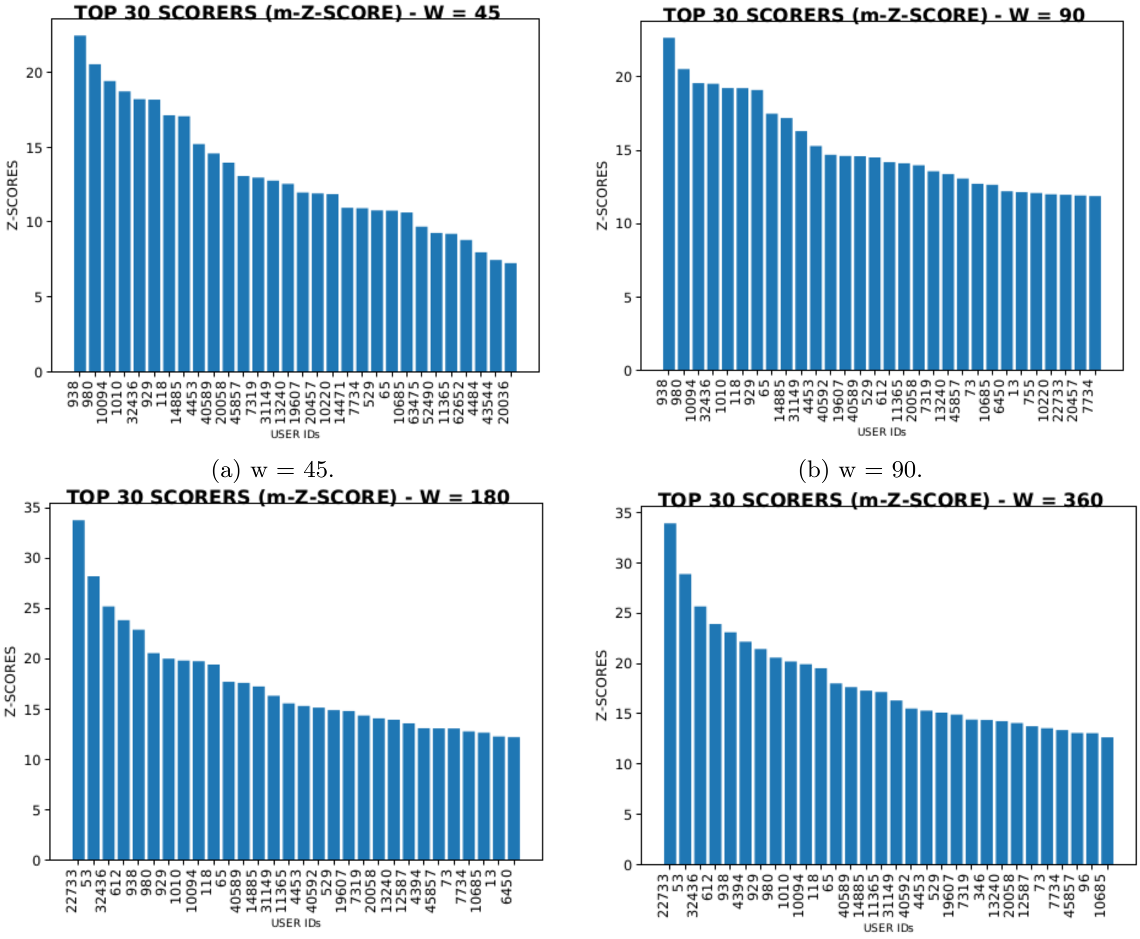
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.7: Pearson Correlation, Degree vs. m-Degree for different w.

Ranking of experts based on m-Z-Score by varying w



(a) w = 45.

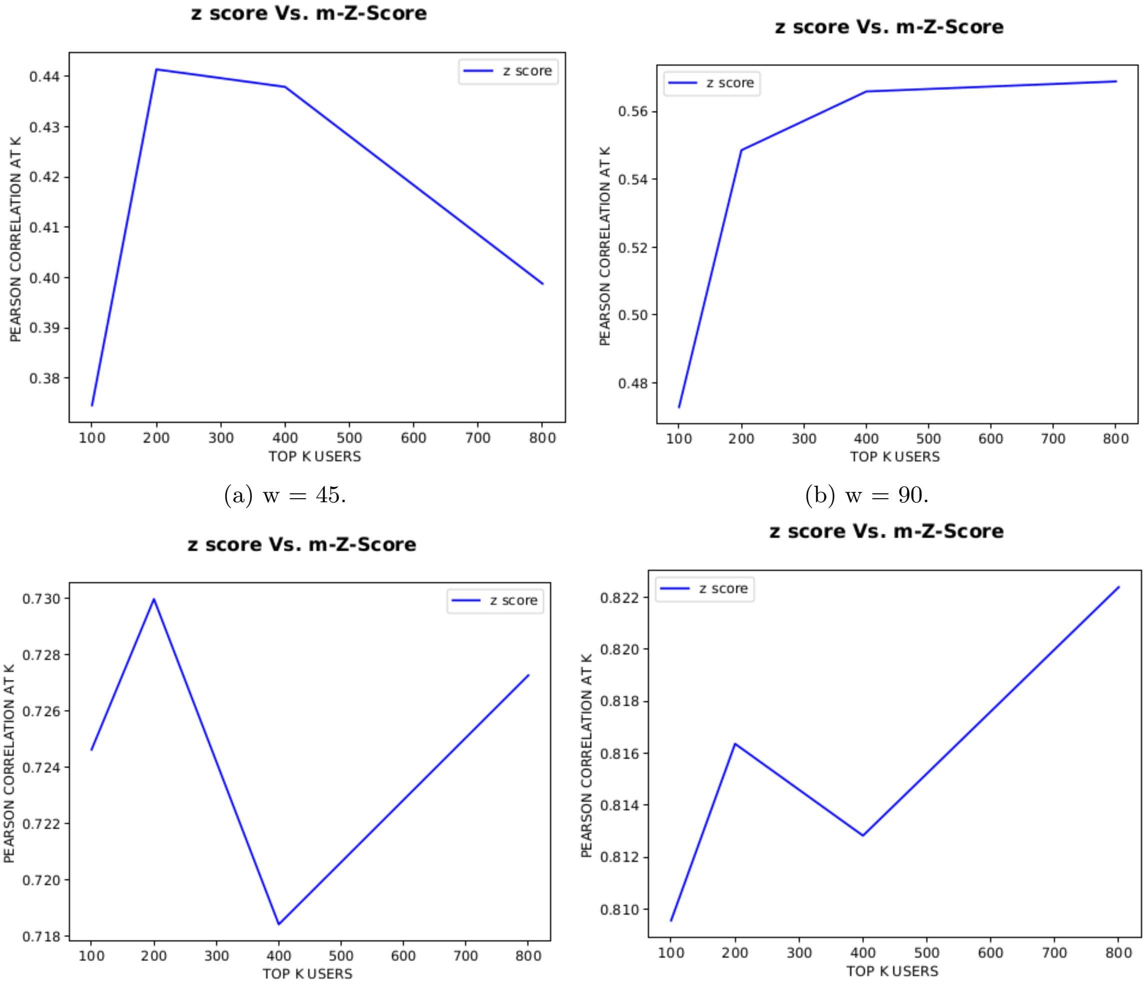
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.8: Ranking of experts on the basis of m-Z-Score for different w.

Pearson Correlation between Z-Score and m-Z-Score for different w



(a) w = 45.

(b) w = 90.

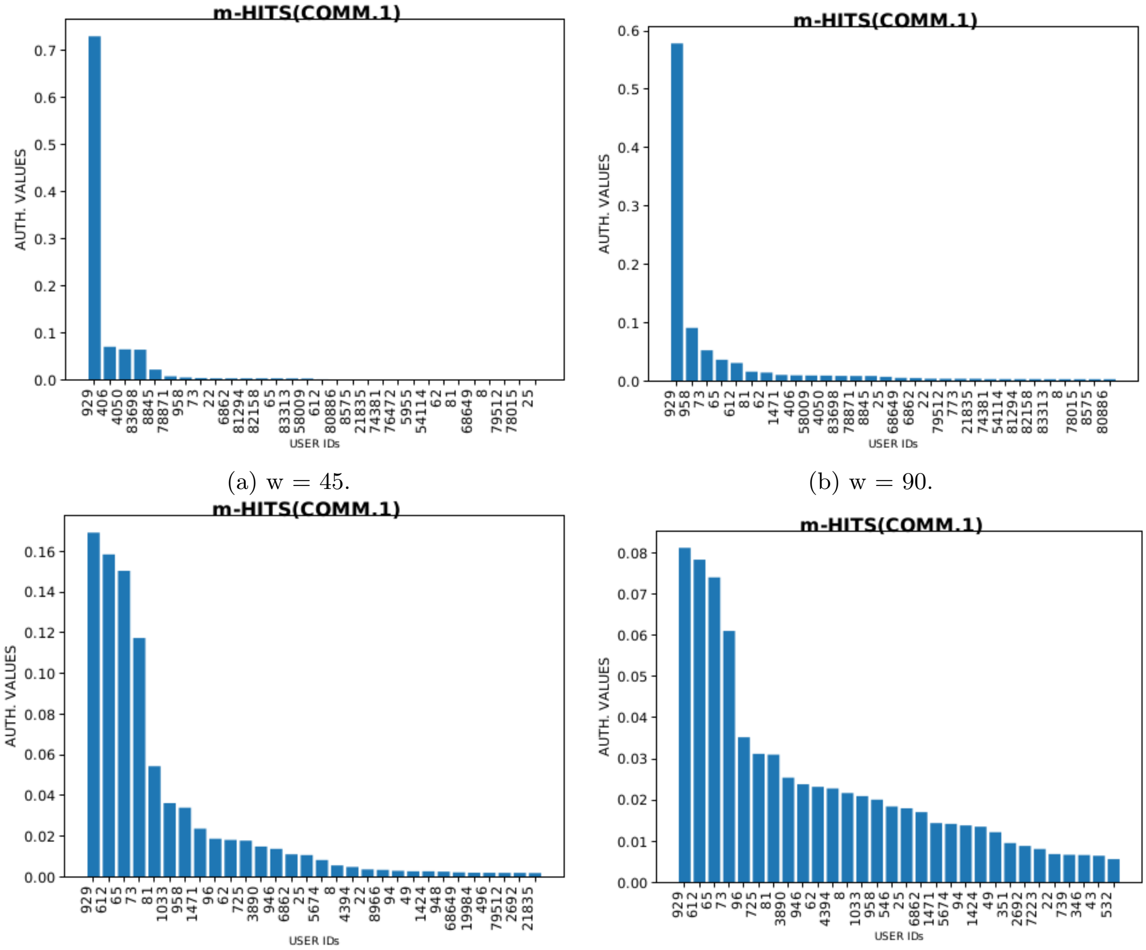
(c) w = 180.

(d) w = 360.

Figure 9.9: Pearson Correlation, Z-Score vs. m-Z-Score for different w.

9.4.2 Results obtained for different Communities in the dataset

Ranking of experts based on m-HITS by varying w (community 1)



(a) w = 45.

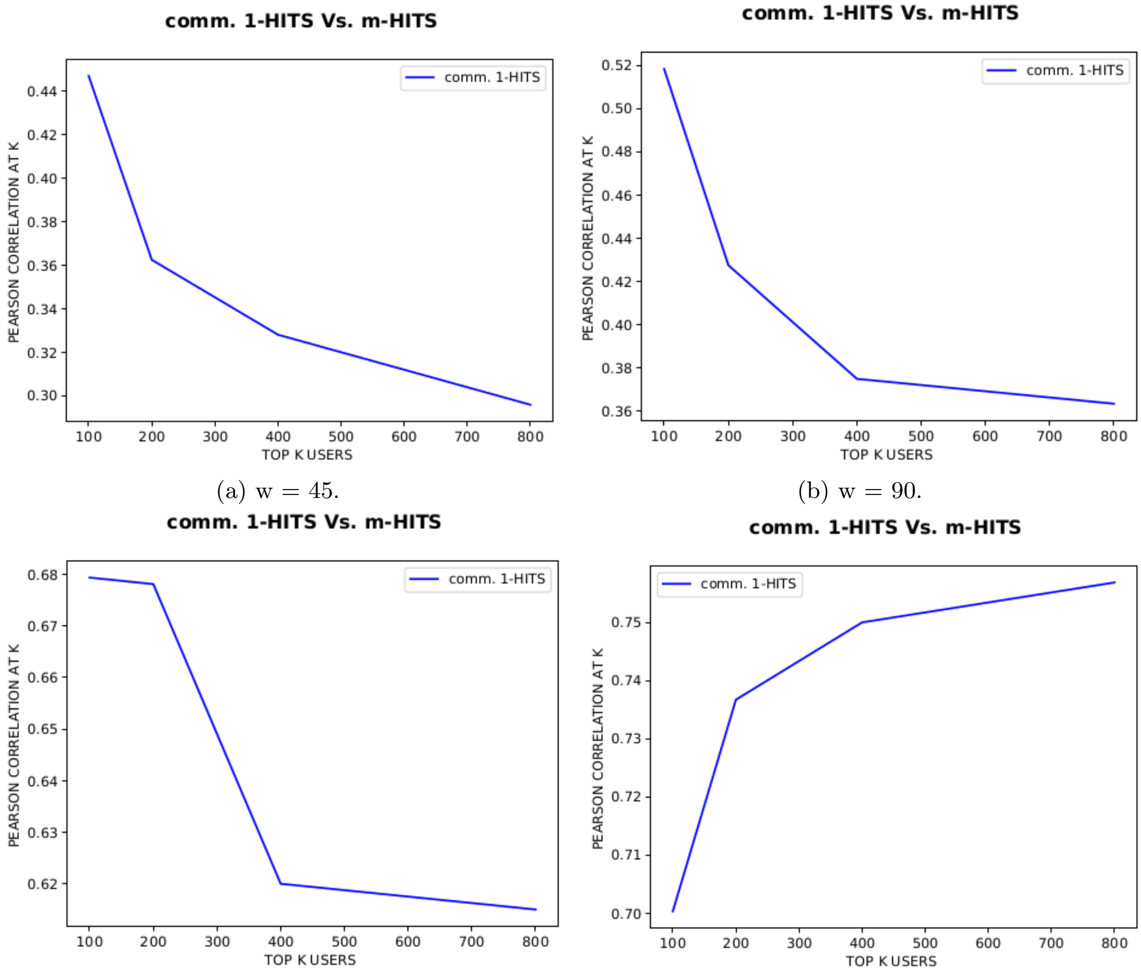
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.10: Ranking of experts on the basis of m-HITS for different w when applied to community 1.

Pearson Correlation between HITS and m-HITS for different w (Community 1)



(a) w = 45.

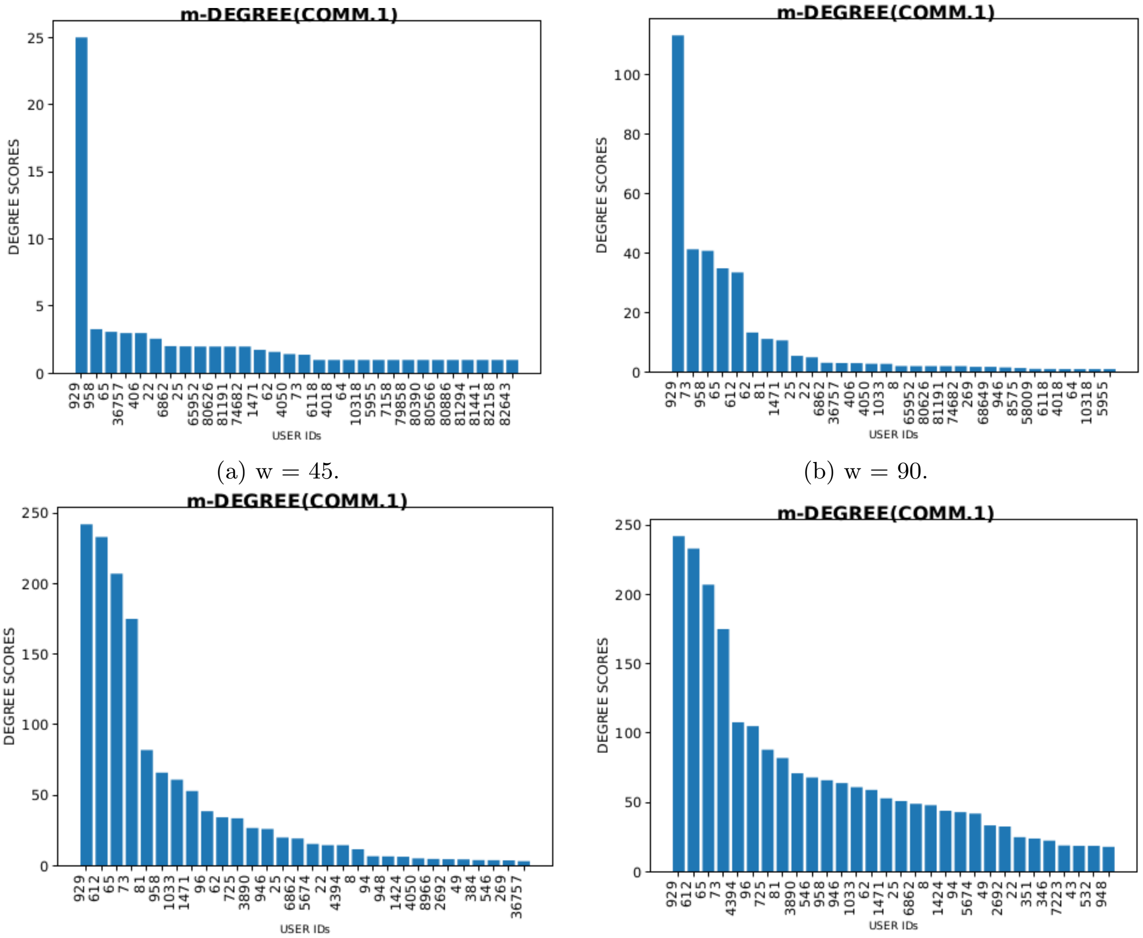
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.11: Pearson Correlation, HITS vs. m-HITS for different w when applied to Community 1.

Ranking of experts based on m-Degree by varying w (community 1)



(a) w = 45.

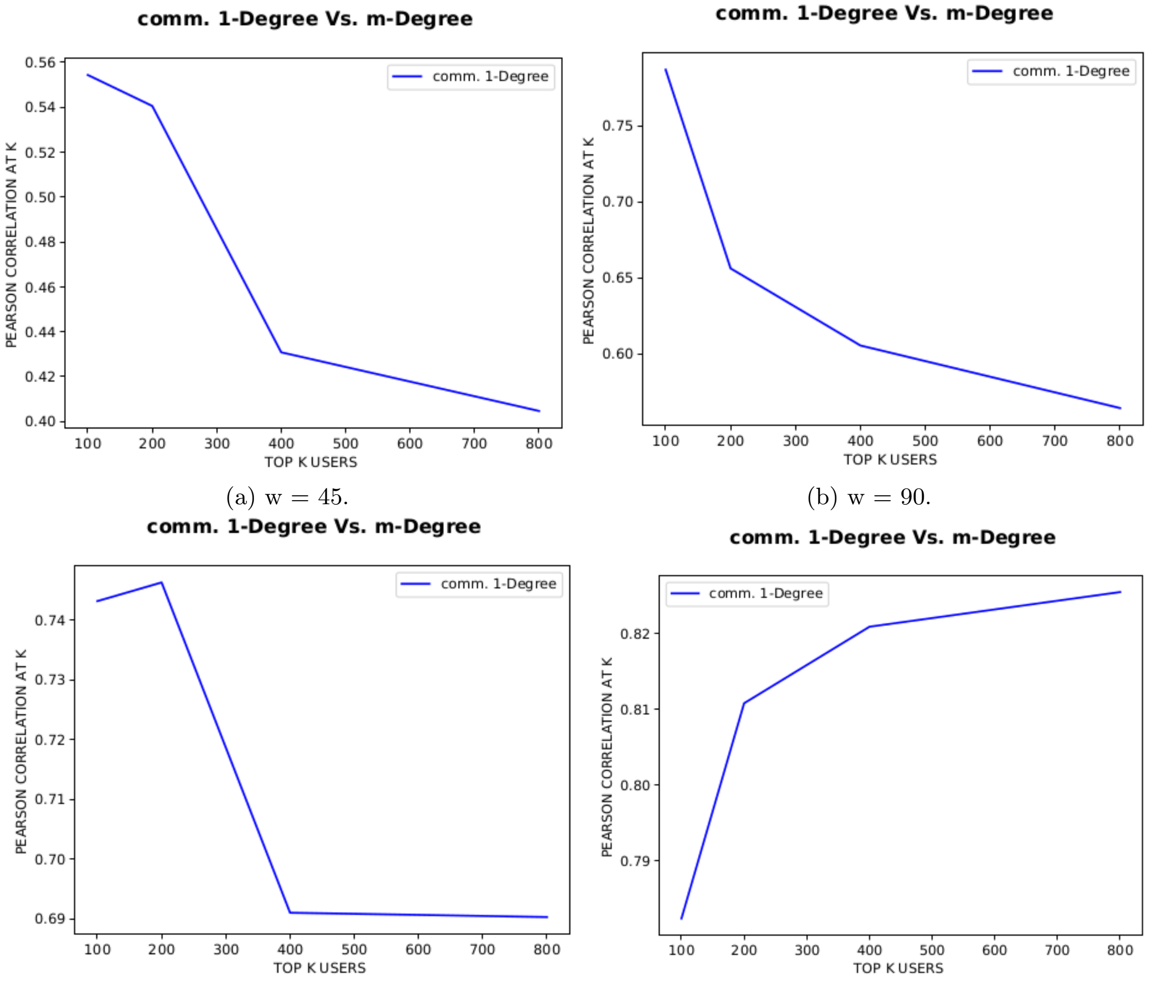
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.12: Ranking of experts on the basis of m-Degree for different w when applied to community 1.

Pearson Correlation between Degree and m-Degree for different w (Community 1)



(a) w = 45.

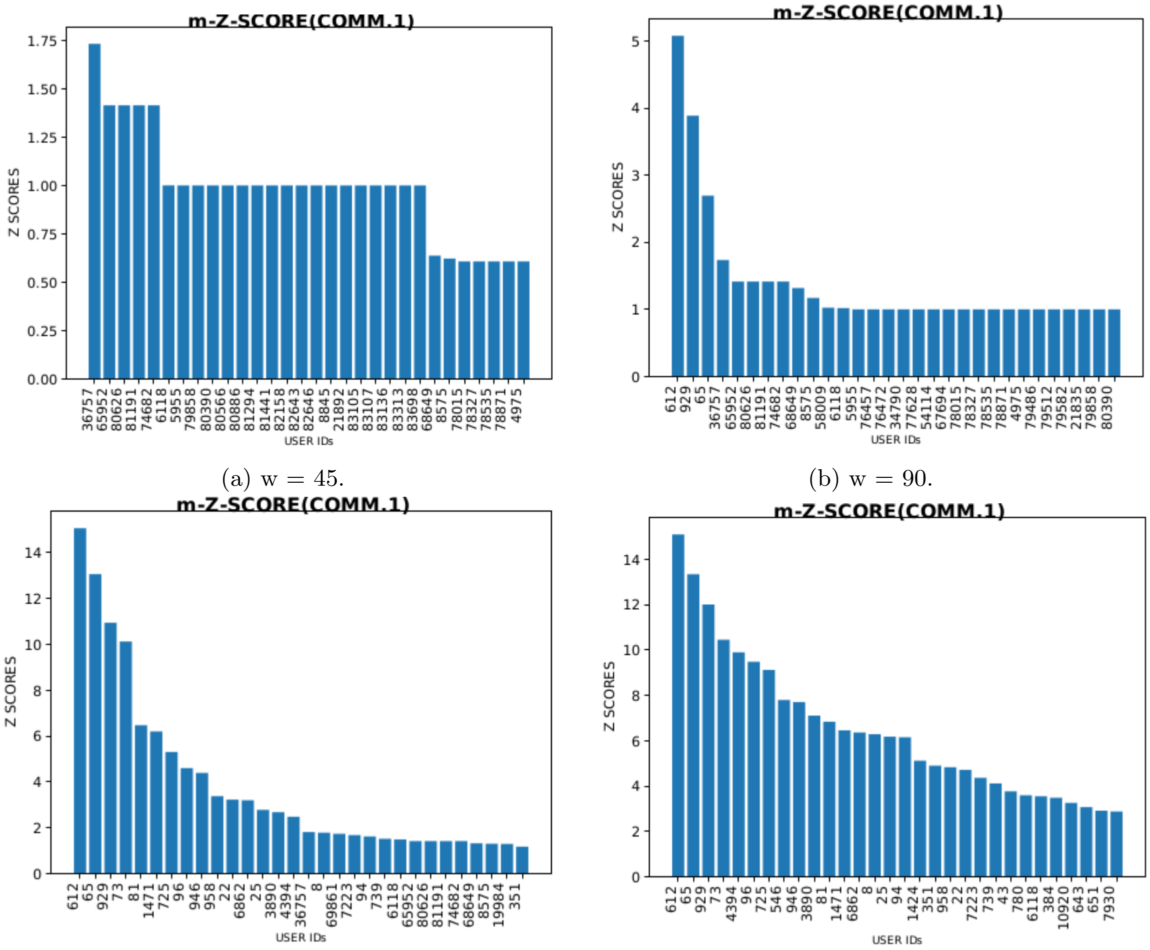
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.13: Pearson Correlation, Degree vs. m-Degree for different w when applied to Community 1.

Ranking of experts based on m-Z-Score by varying w (community 1)



(a) w = 45.

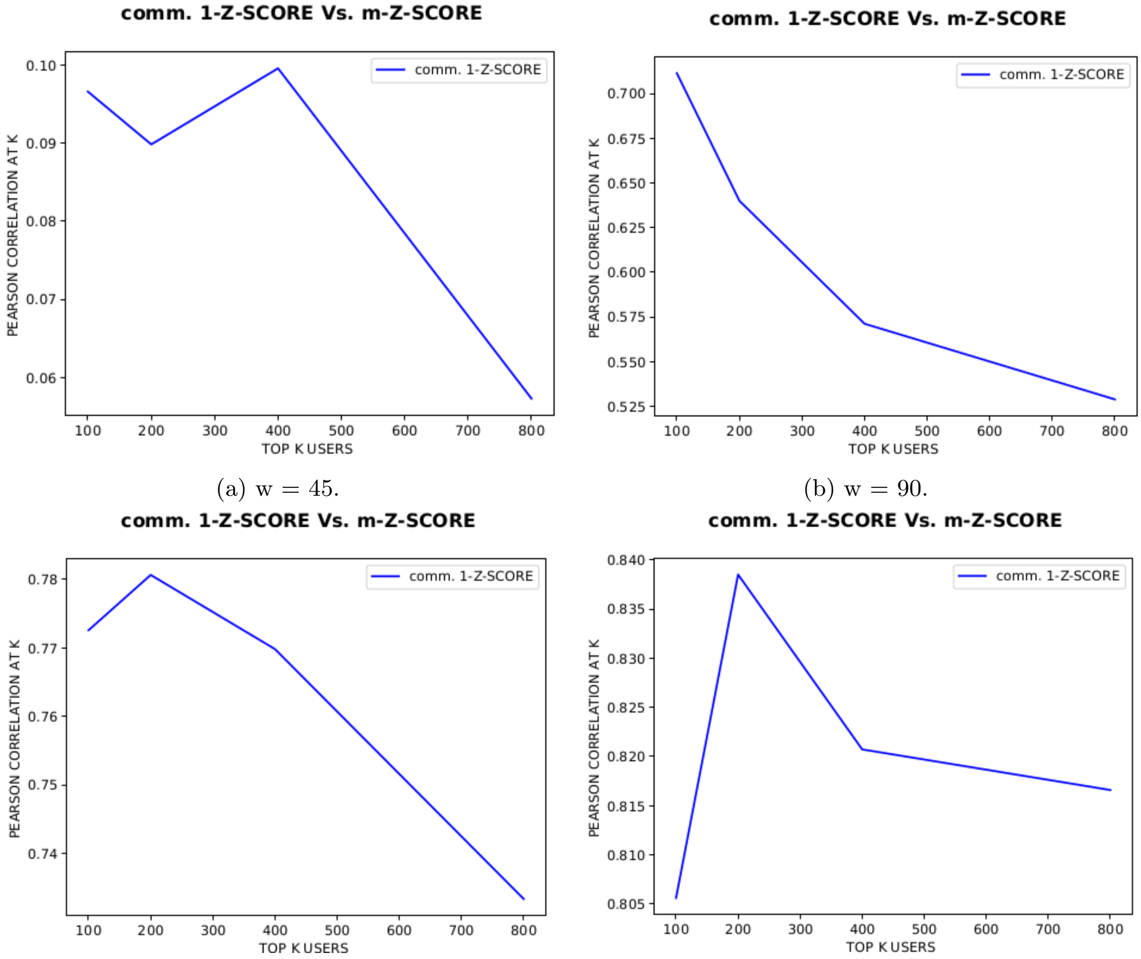
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.14: Ranking of experts on the basis of m-Z-Score for different w when applied to community 1.

Pearson Correlation between Z-Score and m-Z-Score for different w (Community 1)



(a) w = 45.

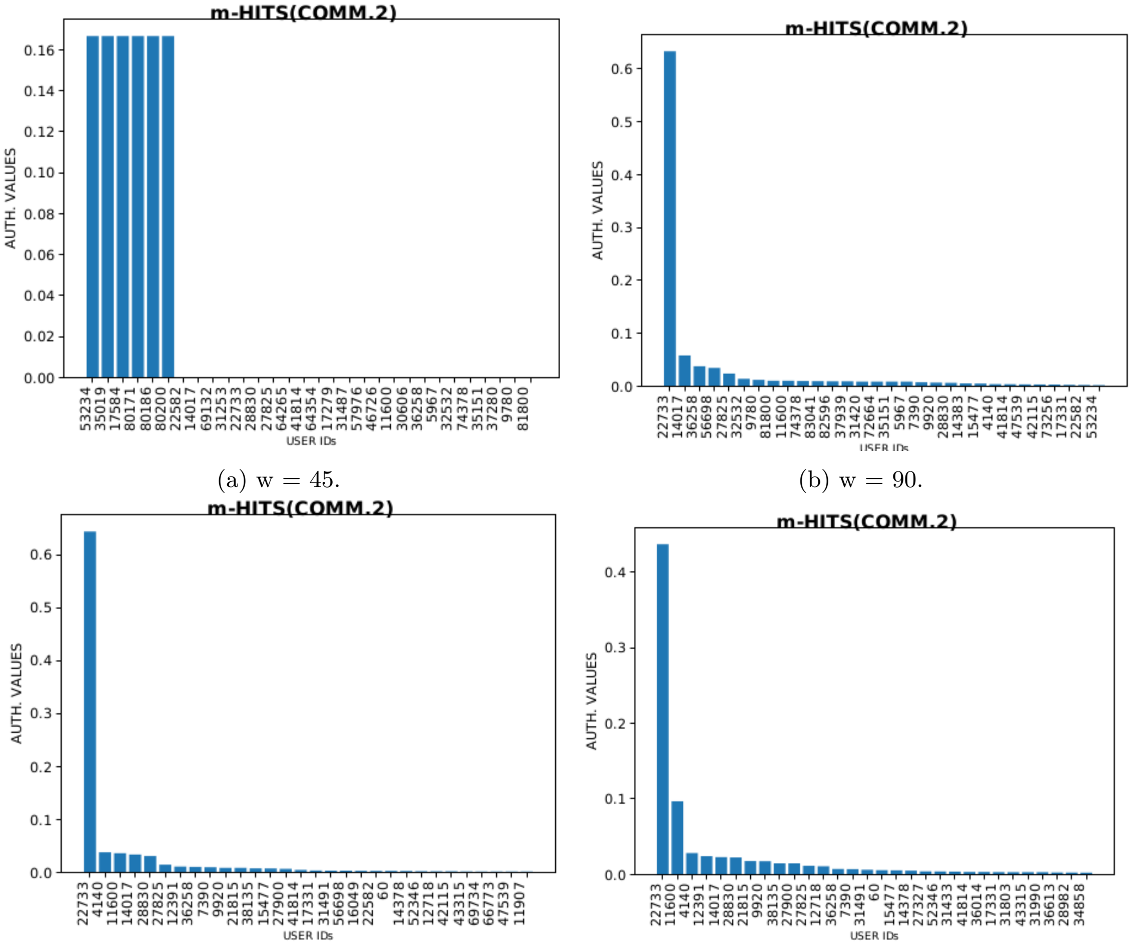
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.15: Pearson Correlation, Z-Score vs. m-Z-Score for different w when applied to Community 1.

Ranking of experts based on m-HITS by varying w (community 2)



(a) w = 45.

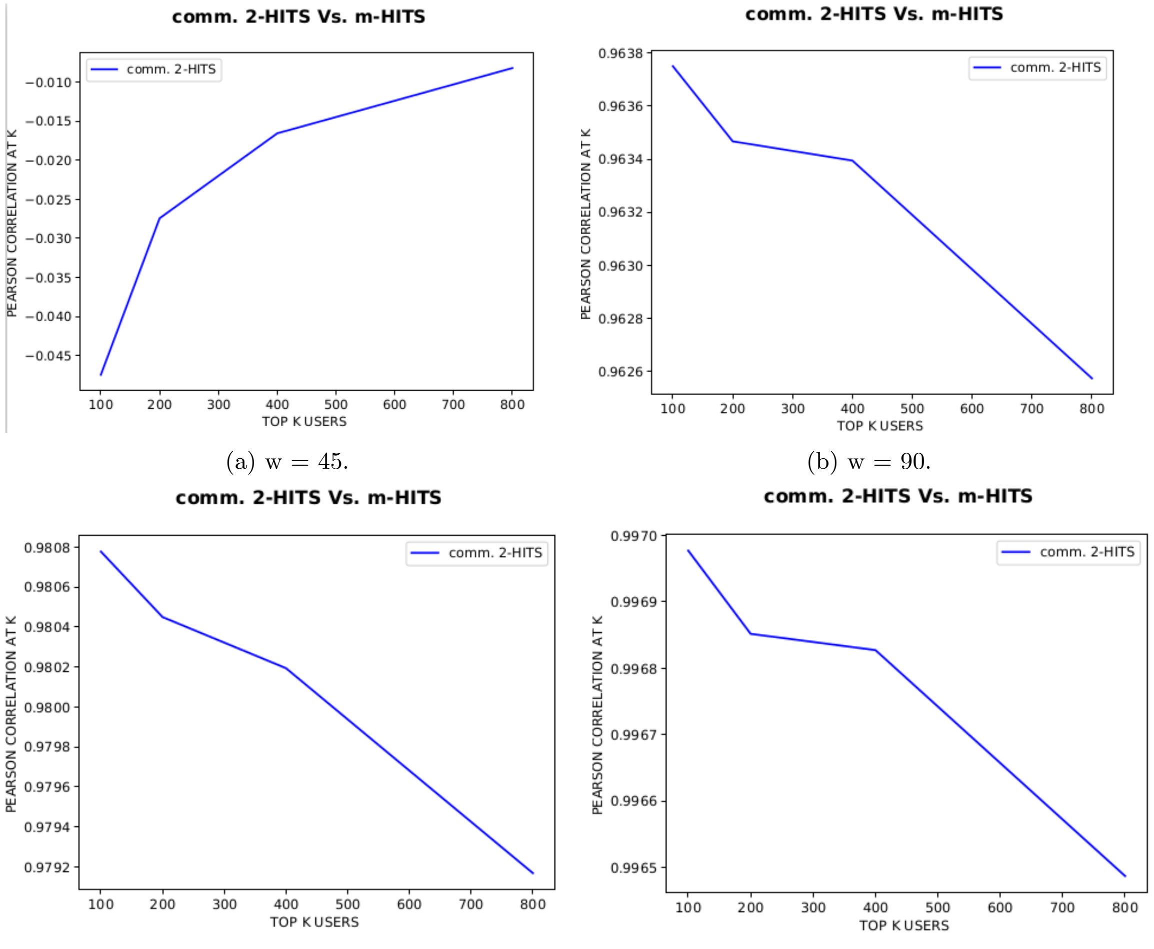
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.16: Ranking of experts on the basis of m-HITS for different w when applied to community 2.

Pearson Correlation between HITS and m-HITS for different w (Community 2)



(a) w = 45.

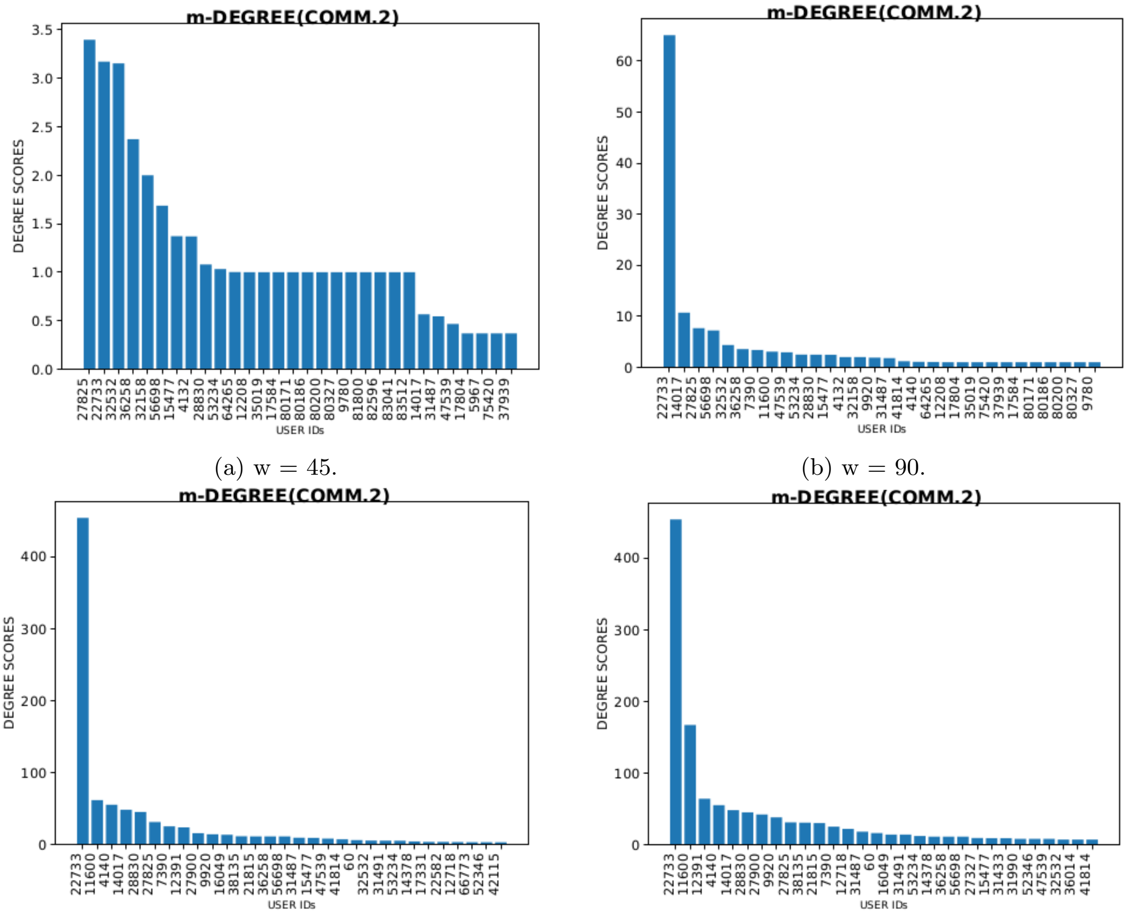
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.17: Pearson Correlation, HITS vs. m-HITS for different w when applied to Community 2.

Ranking of experts based on m-Degree by varying w (community 2)



(a) w = 45.

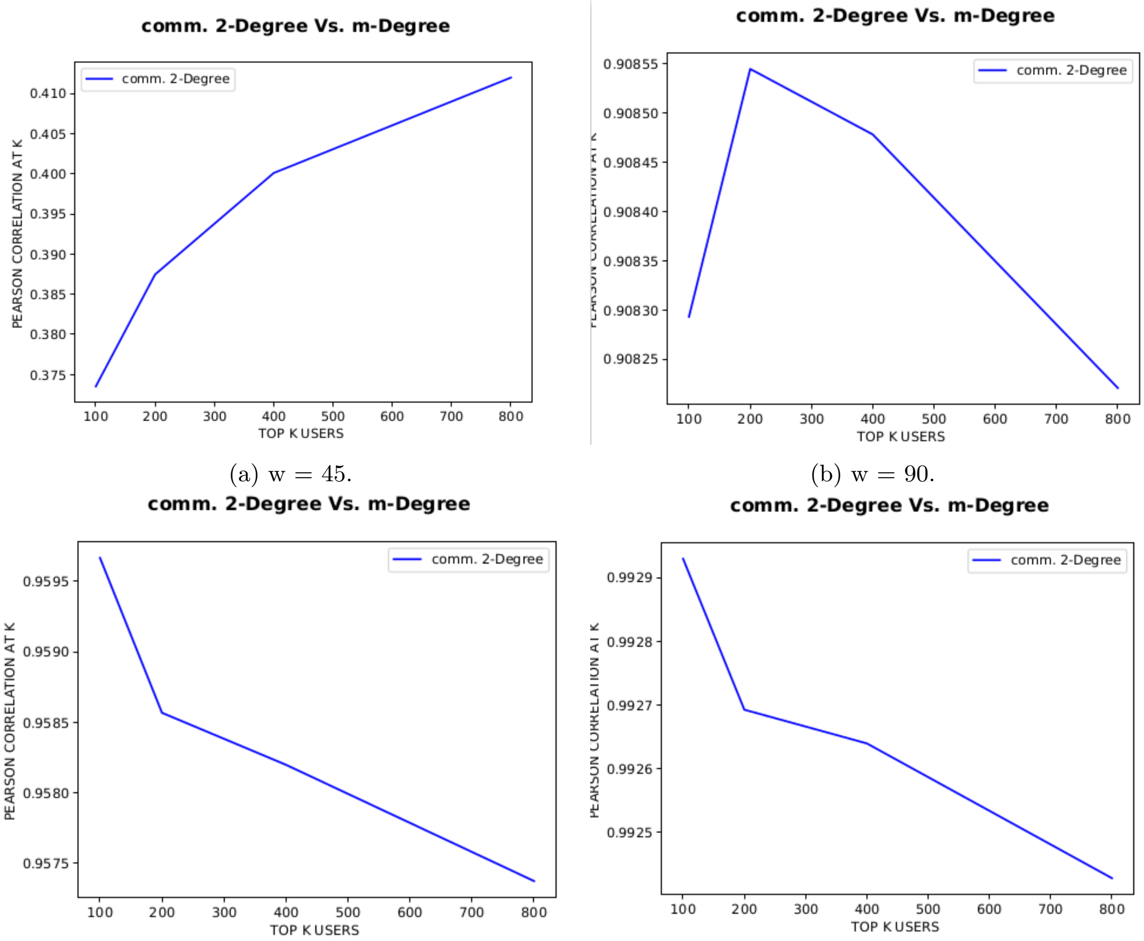
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.18: Ranking of experts on the basis of m-Degree for different w when applied to community 2.

Pearson Correlation between Degree and m-Degree for different w (Community 2)



(a) w = 45.

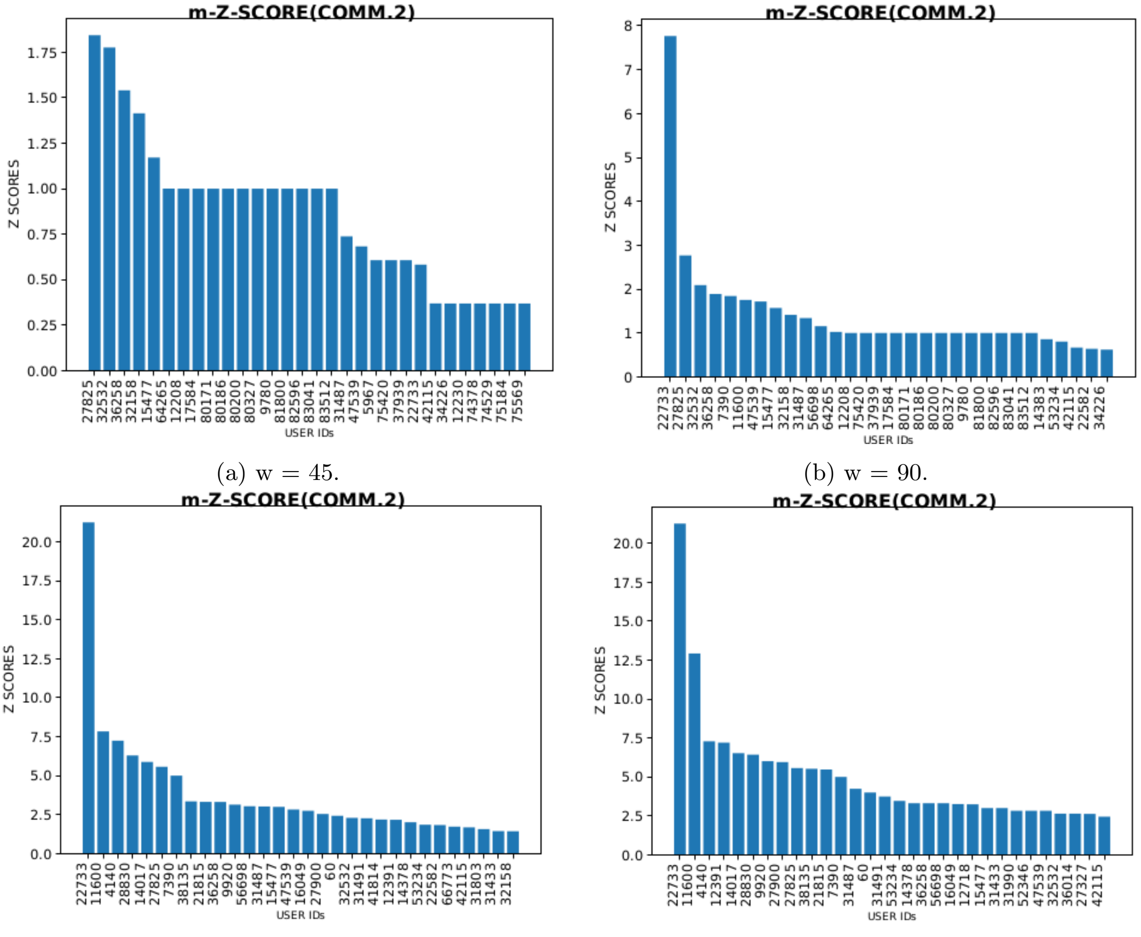
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.19: Pearson Correlation, Degree vs. m-Degree for different w when applied to Community 2.

Ranking of experts based on m-Z-Score by varying w (community 2)



(a) w = 45.

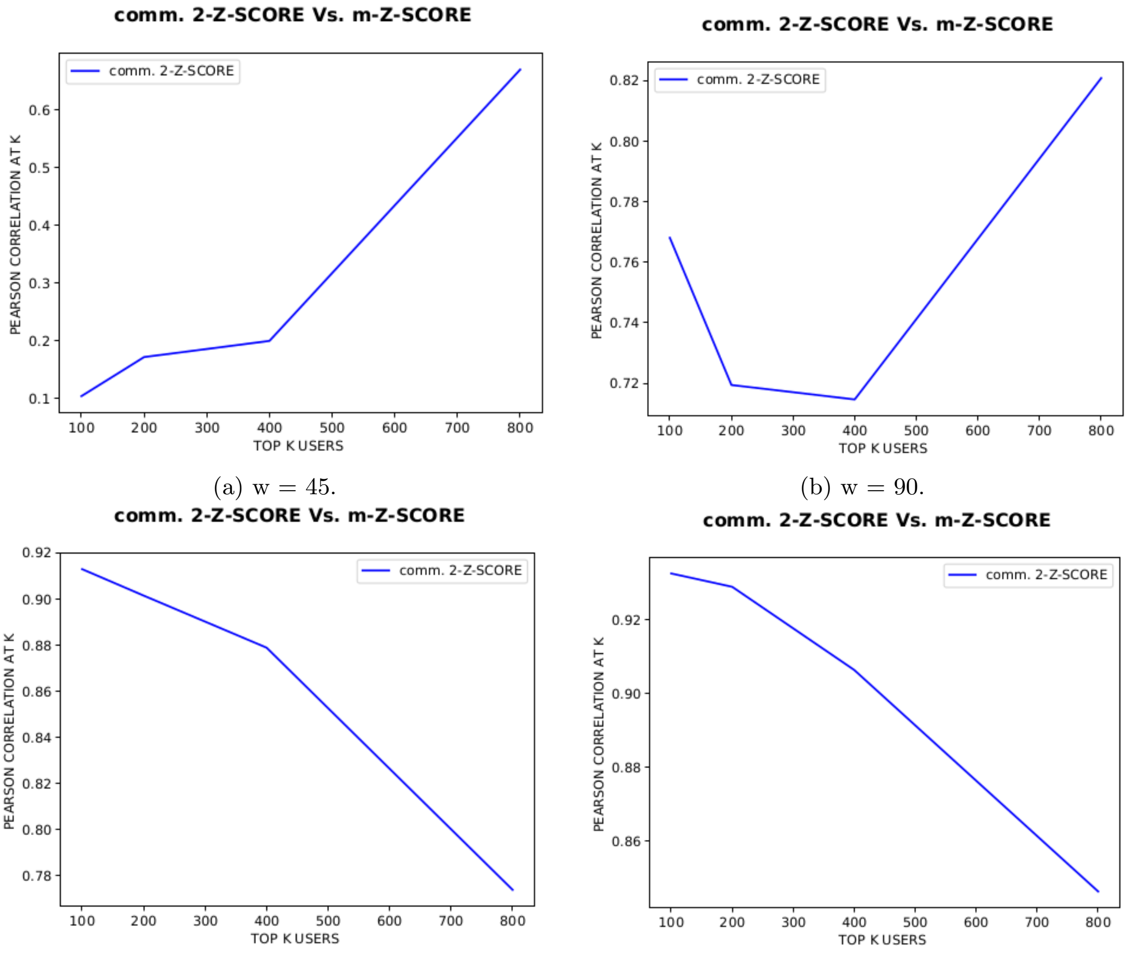
(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.20: Ranking of experts on the basis of m-Z-Score for different w when applied to community 2.

Pearson Correlation between Z-Score and m-Z-Score for different w (Community 2)



(a) w = 45.

(b) w = 90.

(c) w = 180.

(d) w = 360.

Figure 9.21: Pearson Correlation, Z-Score vs. m-Z-Score for different w when applied to Community 2.

Chapter 10

Conclusion

The distributed retention model correlates better against user feedback metrics as compared to the basic model.

The results obtained for the new model, M have not been compared against any base model as there doesn't exist any. However, We have compared the correlations of results obtained for M against the results obtained out of implementing the existing work. The correlation of the existing work vs. M shows expected behaviour. As we increase the window size, the correlation values inch towards +1. This is due to the fact that increasing the window size means we tend to move towards the entire community being taken as one window, which is exactly the case with the existing metrics.

Chapter 11

Anomalies Observed

The behaviour of the curve in Fig. 9.17(a) deviates from the expected behaviour. The correlation comes negative for the w = 45. This has been an unexpected case. Application of models to the communities and sub-communities have almost everytime resulted in the cases spanning figures from Fig. 9.10 - Fig. 9.15.

Chapter 12

Future Work

More experiments can be done on the models presented in this work by varying values for the parameters involved in the forgetting curve equation (6.1) A, B, and t. Similarly, values of the parameter β in the proposed m-Z-Score metric under the new model(M) can also be varied and experimented with. Comparative study of results thus obtained can be analysed and more refined conclusions can be put forward.

Anomalies described in Chapter 11 can also be studied and more analysis can be done.

Some work can also be done in order to come up with a base model in order to compare results for M. Comprehensive analysis can be drawn from such comparisons.

References

[1] A. Borodin,G.O. Roberts, J.S. Rosenthal and P. Tsaparas . Link Analysis Ranking Algorithms Theory And Experiments. *ACM Transactions on Internet Technology* (2005).

[2] T. H. Haveliwala. Topic-sensitive PageRank. *WWW* (2002).

[3] J.M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM* (1999).

[4] X. Liu, W. B. Croft and M. Koll. Finding experts in community-based question-answering services. *CIKM* (2005).

[5] P. Jurczyk and E. Agichtein. HITS on Question Answer Portals: An Exploration of Link Analysis for Author Ranking. *SIGIR* (2007).

[6] P. Jurczyk and E. Agichtein. Discovering authorities in question answer communities by using link analysis. *CIKM* (2007).

[7] Richard B. Anderson and Ryan D. Tweney. Artifactual power curves in forgetting. *Psychonomic Society, Inc.* (1997) 724–730.

[8] *https://en.wikipedia.org/wiki/Pearson_correlation_coefficient*

[9] *http://pi.math.cornell.edu/ mec/Winter2009/RalucaRemus/Lecture4/lecture4.html*