IMPLEMENTATION OF DBRAIN SEARCH ALGORITHM ON PAGE CLUSTERING

V. Syamasudha¹, G. Hari Prasad²

¹Assistant Professor Dept of CSE
Annamacharya Institute of Technology & Sciences, Tirupati, A.P, India

²Assistant Professor Dept of CSE
Sree Rama Engineering College, Tirupati, A.P, India

Abstract: This paper implements an DBRAIN of a clustering algorithm based on gravitational forces to the problem of Web Page Clustering in a dynamic environment. Web pages are attributed with weights in a mathematical space. Computing the gravity between the pages in the mathematical space is expensive, since the definition of mathematical space is very expensive (time, space). Clustering without using the notion of the space is used in this paper Experiments with data representing real URL’s and sessions are performed, and a comparison with the incremental connected components algorithm, which has been previously used to solve this problem, is done.

Keywords: Data mining, page clustering, Web usage/structure mining, Web page clustering

1. INTRODUCTION

Data mining is the extraction of the hidden predictive information from large databases and it is a powerful new technology with great potential to analyze important information in the data warehouse. The term 'Data mining' refers to the finding of relevant and useful information from databases. Data mining and Knowledge discovery in the databases is a new interdisciplinary field, merging ideas from statistics, machine learning, databases and parallel computing. Researchers have defined the term ‘data mining’ in many ways:

Data mining or Knowledge discovery in databases, as it is also known, is the non-trivial extraction of implicit, previously unknown and potentially useful information from the data. This encompasses a number of technical approaches, such as clustering, data summarization, classification, finding dependency networks, analyzing changes, and anomalies.

Data mining is the search for the relationships and global patterns that exist in large databases but are hidden among vast amounts of data, such as the relationship between patient data and their medical diagnosis. This relationship represents valuable knowledge about the database, and the objects in the database, if the database is a faithful mirror of the real world registered by the database.

Data mining is the process of discovering meaningful, new correlation patterns and trends by shifting through large amount of data stored in repositories, using pattern recognition techniques as well as statistical and mathematical techniques.

Clustering is an unsupervised learning technique, which is similar to classifications in that data are grouped, which are not predefined. Clustering is a collection of data objects in which data objects are similar to one another with in the same cluster and dissimilar to the objects in the other clusters. Clustering is an unsupervised learning technique that takes unlabeled points and assigns each of them into a group so that points of a same cluster should have a high similarity measure (or low distance) and elements of two different clusters have low similarity.

Several techniques have been proposed for clustering and most of them take the number of clusters as a parameter. When the number of clusters is not provided and found by the algorithm itself, the techniques are called unsupervised. On other hand, page clustering represents a particular problem for soft computing techniques. Clustering methods group web pages according to different measures.

In this sense, the most common approach is to use information gathered from network users so that they themselves define the clusters of the pages. Then, the problem of page clustering will be reduced finally to find the induced similarity and then to classify the pages according to it.

The kind of approach used for web mining is usually called “Relational Clustering” because the absence of a space where points can be placed. The information generated by those techniques is important for discovering the site user’s tendencies.

There are basically two designs for modules that perform clustering and suggestion procedures: The first one uses

1. off-line clustering and
2. on-line suggestions generation when the users are surfing the web.

The second one is designed as a tool capable to perform in

1. one step clustering
2. suggestions on-line

This tool uses incremental connected components algorithm.

2. PROBLEM STATEMENT

2.1 Definition:
- Web pages are attributed with weights in a mathematical space.
- Computing the gravity between the pages in the mathematical space is expensive, since the definition of mathematical space is very expensive (time, space).
- Clustering without using the notion of the space is used in this paper

2.2 Description:
The rapidly changing structure and content of today’s web sites causes difficult problems to the information management and retrieval processes because new users are arriving and clusters change over time. As a consequence, web data mining has been used to acquire information of web site usage in order to modify the sites behavior in a personalized manner.

These changes are done according to the knowledge found. Some of the content problems of web sites arise because visitors often have different interest and therefore they behave in a different way when surfing on one’s web site pages. In those cases, classification and clustering techniques represent natural approaches in finding solutions.

Randomized Gravitational Clustering, for an n-dimensional data set with N data points, each data point is considered as an object in the n-dimensional space with mass equal to 1. Each point in the data set is moved according to a simplified version of the gravitational Law using the Second Newton’s Motion Law. The basic ideas behind applying the gravitational law are:

A data point in some cluster exerts a higher gravitational force on a data point in the same cluster than on a data point that is not in the cluster. Then, points in a cluster move in the direction of the center of the cluster. In this way, the proposed technique will determine automatically the clusters in the data set.

If some point is a noise point, i.e., does not belong to any cluster, then the gravitational force exerted on it from other points is so small that the point is almost immobile. Therefore, noise points will not be assigned to any cluster. In order to reduce the amount of memory and time expended in moving a data point according to the gravitational field generated by another point (y), we use the following simplified equation:

$$ x(t+1) = x(t) + \frac{G}{\|x-y\|^2} $$

where, $\frac{1}{d} = \frac{1}{x-y}$, and the gravitational constant $G$.

In order to eliminate this limit effect, the gravitational constant $G$ is reduced each iteration in a constant proportion (the decay term $\Delta G$). Algorithm 1 shows the randomized gravitational clustering algorithm.

**Algorithm 1 Randomized Gravitational Clustering**

```
RGC(x, G, \Delta G, N, M, e)
1: for i = 1 to N do
2: MAKE(i)
3: end for
4: for j = 1 to M do
5: for k = 1 to N do
6: if dist(x_j, x_k) \leq e then
7: MOVE(x_j, x_k) (see Eq (1)) //Move both points
8: if dist(x_j, x_k) \leq e then
9: UNION(j, k)
10: end if
11: end for
12: end for
13: G = (1 - \Delta G)G
14: end for
15: for i = 1 to N do
16: FIND(i)
17: end for
18: return disjoint-sets
```

Function MOVE (line 7), moves both points $x_j$ and $x_k$ using (1) taking into consideration that both points cannot move further than half of the distance between them. In each iteration, RGC creates a set of clusters by using an optimal disjoint set union-find structure and the distance between objects (after moving data points according to the gravitational force). When two points are merged, both of them are kept in the system while the associated set structure is modified.

In order to determine the new position of each data point, the proposed algorithm only selects another data point in a random way and move both of them according to (1) (MOVE function). RGC returns the sets stored in the disjoint set union-find structure.

**DBRAIN**

In order to achieve the clustering phase without generate the Euclidean space (or any other) some modifications were performed to the original RAIN algorithm. This version of the algorithm works for problems where the positions of the points in a space (for example an Euclidean space) cannot be defined but the distance between each pair of points is known.

The modification contained in DBRAIN (Database Research on Algorithms and Incentives in Networks) resides in the MOVE function. In RAIN the new position of a point in the space was computed using (1). In DBRAIN we do not have the concept of space so this is done according to the following procedure:

1. Compute the new distance of the selected points.
2. Update the distance of the selected points to the rest of the points.
3. Keep the distance properties of the distance matrix.
When two points cross each other by the gravity force, DBRAIN considers that a collision has happened and sets the distance to zero, \( d_{t+1}(i, j) = 0 \).

\[
d^t(k, \sum_{i=1}^{n} w_i x_i) = \sum_{i=1}^{n} w_i d^t(h, x_i) - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j d^t(x_i, x_j) \quad (3)
\]

\[
x_1 = 1 \quad (4)
\]

\[
w_1 = \frac{d(x_i, j) - d(x_i, i)}{2d(x_i, j)} \quad (5)
\]

\[
w_2 = 1 - w_1 \quad (6)
\]

In the final step we have to consider some distance properties. As it is well know these conditions are the following:

\[
d(i, i) = 0 \quad (7)
\]

\[
d(i, j) = d(j, i) \quad (8)
\]

\[
d(i, k) \leq d(i, j) + d(j, k) \quad (9)
\]

Finally the function MOVE became an algorithm shown in algorithm 3.

**Algorithm 3 MOVE function**

MOVE \( i, j \) (\( d_{t+1}(i, j) \))

1: compute \( d_{t+1}(i, j) \) according to (2)
2: if \( d_{t+1}(i, j) < 0 \) then
3: \( d_{t+1}(i, j) = 0 \)
4: end if
5: compute \( w_1 \) and \( w_2 \) according to (5) and (6)
6: for \( k = 1 \) to \( N_j \), \( k \neq i, k \neq j \) do
7: update \( dMatrix(k, i) \) according to (3)
8: update \( dMatrix(k, j) \) according to (3)
9: end for

3. PROPOSED MODEL

In order to generate a model which can run in real time, it has to generate clusters when new sessions appear. We used an approach similar to the proposed by Silvestri et al but using the DBRAIN algorithm presented here. In this way, we used the same input functions as the co-occurrence matrix because such matrix gives us a similarity measure in the interval \([0,8]\) and it can be easily transformed into a distance matrix using (10). The factor \( d \) is used to maintain points separation so we can avoid big crunch (in the interval \([0, 1]\) the gravitational force is very strong).

\[
d(i, j) = \begin{cases} 
\exp(Matrix(i, j)) & \text{if } i \neq j \\
0 & \text{otherwise} 
\end{cases} \quad (10)
\]

In order to test the quality of the clusters we also use the same algorithm as the proposed in SUGGEST.

The correlation matrix is probably the most important structure on the SUGGEST architecture. At the same time this is the center of the data we use. This matrix stores the co-occurrence records for the pages of the site as the number of sessions containing both pages. The diagonal of this matrix stores the total number of visits to a page. There are some details in the setting of the matrix we think are important to clarify:

1. The co occurrence matrix stores only one entry for multiple apparitions of the same pair of pages in a session. Multiple visits to the same page within a session are not registered.

2. The position \( i, j \) of the matrix is the sum of every different couple of sessions having the \( i \) and \( j \) pages divided by the maximum number of sessions containing only one of the pages. This is done to reduce importance to index pages, which are visited with all others frequently. i.e. if \( N_j \) is the total number of times that both pages appear then \( W_{ij} \) is the normalized similarity. The whole expression can be found in 11.

\[
W_{ij} = \frac{N_{ij}}{\max{[N_j, N_i]}} \quad (11)
\]

3. The matrix is symmetric making it possible to store the upper triangle and the diagonal only, saving memory to enable to handle bigger sites.

**Clustering Phase**

As mentioned before this is the most important part of this work. Clustering phase consists in a run of the DBRAIN algorithm. Notice that, the clustering phase does not run every time the similarity matrix is updated but it is run when \( n \) sessions are finished, \( n \) is a parameter the user can fix.

The clustering phase begins with a transformation of the similarity matrix into a distance one. Once the distance matrix is generated it is used as a parameter to the algorithm and the cluster set is updated. Finally the counter is reseted to zero so when \( n \) new sessions have passed the algorithm will run again.

**Generation of Suggestions**

Once the clustering phase is performed it is necessary to present to the users the \( m \) suggestions for the actual page (note that \( m \) is a parameter for this algorithm). To find these suggestions the \( n \) previous pages visited for the actual session are used. This \( n \) is called the page window. The suggestions set is built finding the cluster with the largest intersection with the page window. Once the cluster with the largest intersection is found we look into it for the \( m \) most relevant (closest) pages to the actual page.
**Description:**
The above diagram shows the total working of the paper. Whenever the client makes a request to the web application, the server log will be generated based on the attributes of the web pages through proxy server. The proxy server will generate the web server log based on the access behavior of the web pages for a web site. The client will access the web pages from web application. For each request from the user the proxy will duplicates the attributes of the accessed web page and it will generate the web server log. Server log contains the records of the accessed web pages. Using that web server log the correlation matrix will be generated.

The correlation matrix is probably the most important structure on the SUGGEST architecture. At the same time this is the center of the data we use. This matrix stores the co occurrence records for the pages of the site as the number of sessions contain both pages. The diagonal of this matrix stores the total number of visits to a page.

The similarity matrix will be generated based on the correlation matrix. The clustering phase begins with a transformation of the similarity matrix into a distance one. Once the distance matrix is generated it is used as a parameter to the algorithm and the cluster set is updated. Finally the counter is reseted to zero so when new sessions have passed the algorithm will run again.

Once the clustering phase is performed it is necessary to present to the users the suggestions for the actual page (note that $m$ is a parameter for this algorithm). To find these suggestions the $n$ previous pages visited for the actual session are used. This $n$ is called the page window. The suggestion set is built finding the cluster with the largest intersection with the page window. Once the cluster with the largest intersection is found we look into it for the $m$ most relevant (closest) pages to the actual page.

Using the generated suggestion set, the web designer will understand the web user’s tendencies. The web designer restructures the web site in personalized manner.

**4. CONCLUSIONS**
Our results show the DBRAIN algorithm opens a promising research path in applications using it and in future improvements to the algorithm itself. The
combination of the distance based clustering and the gravitational algorithm suits well for the Web environment as any similarity measure can be used to build the clusters according to particular research interests.

The results obtained for our link suggestion prototype based on the SUGGESTOL module are encouraging in terms of suggestion quality, given the algorithm is an early design and development stage. One of the paths of research would be to find a version of the algorithm that takes advantage of the clusters previously generated. The current prototype does not include this capability yet.

REFERENCES