Journal of International Technology and Information Management

Volume 25 | Issue 3 Article 6

10-1-2016

Prediction And Recommendations On The It Leaners' Learning Path As A Collective Intelligence Using A Data Mining Technique

Seong-Yong Hong

Juyun Cho

Yonghyun Hwang

PREDICTION and RECOMMENDATIONS on the IT LEANERS' LEARNING PATH as a COLLECTIVE INTELLIGENCE USING a DATA MINING TECHNIIQUE

Seong-Yong Hong
Korea Advanced Institute of Science and Technology (KAIST), South Korea,

Juyun Cho
Colorado State University, Pueblo, CO, USA

Yonghyun Hwang
University of California, Irvine, CA, USA

ABSTRACT

With the recent advances in computer technology along with pervasive internet accesses, data analytics is getting more attention than ever before. In addition, research areas on data analysis are diverging and integrating lots of different fields such as a business and social sector. Especially, recent researches focus on the data analysis for a better intelligent decision making and prediction system. This paper analyzes data collected from current IT learners who have already studied various IT subjects to find the IT learners' learning patterns. The most popular learning patterns are identified through an association rule data mining using an arules package running under R studio. Experimental results are used to recommend the IT learning path to rudimentary IT learners. It is expected that our research promotes IT learning field and results in a platform of IT learning helpful to IT learners.

KEYWORDS: IT learners' learning path, data mining, association rule, R package

INTRODUCTION

Big data-based convergence research has been actively formed recently and considered as an important research in many areas such as a defense, medical, education, and business field, which requires a new discovery and the creation of a new knowledge. The advancement in the IT (Information Technology) has the biggest influence on the changes of a human society, and the amount of data generated in our daily lives contains enormous amount of information unimaginable. According to the IDC (International Data Corporation), the total amount of digital information created in the world in 2011 was about 1.8 Zetta bytes (10^{12} Giga bytes). IDC predicted that the total amount of digital information would be reaching to 35.2 Zetta bytes in 2020. The characteristics of big data can be summarized as 5 V's - 1) Volume: data size is big, 2) Variety: various types of data exists such as text, audio, video, and graph, 3) Velocity: data grows continuously, 4) Variability: the structure of data changes, and 5) Value: organization can gain business value from big data (Abbasi, Sarker, & Chiang, 2016; Fan & Bifet, 2013). Due to the nature of big data, data storage and processing structure should be provided at the same time.

The main purpose of processing and mining big data is to create a new value. Interpreting new knowledge and meaning obtained from big data can be applied to various areas such as organizations' decision-making as well as customized services for individuals. In the age of big data, using a data mining technique, our paper conducts a research and proposes the way to predict IT learners' learning pattern and then recommend the learning path later.

RELATED RESEARCH

There are a lot of research field related to big data. Due to the idiosyncrasy of big data mentioned in the introduction section, both data processing and analysis must be applied to big data in real-time. Some of big data mining researches pursue a human-centered computing technology such as creation of new knowledge, prediction, or recommendation through a machine learning or data mining algorithm.

BIG DATA PLATFORM TECHNOLOGY

Big data platform technology includes handling, collecting, processing, and analyzing various types of data (McKinsey Global Institute, 2011). Many storage systems and databases are used in data analytics including VoltDB (Buckle, 2012; Lasota, Deniziak, & Chrobot, 2016; Stonebraker & Weisberg, 2013; Technical Overview, 2016), SAP HANA (Färber et al., 2012), Vertica (Lamb et al., 2012), Greenplum (Waas, 2009), and IBM Netezza (Atriwal, Nagar, Tayal, & Gupta, 2016; Singh & Leonhardi, 2011). The most widely known technology that helps to handle large-data would be a distribution data process framework of the Map-Reduce method, such as Apache Hadoop (Borthakur et al, 2011). Dryad is a framework that enables processing parallel data by forming the data channel between programs in a graph type. Developers who use Map/reduce framework should write Map and Reduce functions; thus, if using Dryad, they need to make a graph which processes the corresponding data (Isard, et al., 2007). Apache Pig provides a high-standard data processing structure, allowing large data processing through combination

(Olston et al., 2011; Sharma & Singh, 2016). Apache Hive helps to analyze large data by using the query language called HiveQL for data source, such as HDFS or HBase. Architecture is divided into Map-Reduce-oriented execution, metadata information for a data storage, and an execution part that receives a query from user or applications for execution (Thusoo et al., 2009).

Big data analysis technology can be classified into the following five areas. 1) Data collection and integration technology: Technology that collects existing data on the distributed web or newly generated data in real-time and that integrates/stores data later. 2) Data preprocessing technology: Technology that changes heterogeneous formatted data or unstructured data into structured data or refined data by filtering unnecessary data. 3) Data storage and management technology: Technology that stores huge amount of data in real-time generated in various ways and that supports and manages distributed storage structure through computer networks. 5) Data analysis technology: Technology that extracts, predicts, and recommends new knowledge using algorithms such as data mining, learning machine, artificial intelligence, and statistical process for massive data. 6) Data visualization technology: UX (User Experience)-based Info-Graphics technology that helps recognize analyzed data easily and quickly.

BIG DATA INTEGRATED SOLUTION

The world class corporations such as Intel, Oracle, IBM, Microsoft, HP, SAS, SAP, Google, and EMC have been trying to occupy the big data market through a big data analysis platform. Hadoop, which is an open software framework developed by Doug Cutting and Mike Cafarella in 2005 to implement Google's MapReduce algorithm, has been researched and used more than any other platforms. The big data analysis platform technology based on Hadoop has been improved and is moving to a big data infra technology with the aid of cloud computing infrastructure and is continued to grow quickly together with an analysis package 'R'. Hadoop provides various features including 1) a massive distributed computing support framework for massive data processing and analysis, 2) a distributed database, 'HBase', 3) 'Hive' for Hadoop data query language, 4) an interactive script environment 'Pig' for data processing, 5) a clustering and cooperation filtering, 6) a machine learning library, 'Mahout', which provides an algorithm to recognize similarities, 7) 'SQOOP' for the function of exchanging input/output through RDBMS, 8) a workflow environment, 'Oozie' for adjusting complex data processing tasks, and finally 9) a document-oriented database, 'Cassandra'. As shown, Hadoop is becoming an integrated solution for big data processing and management.

COLLECTIVE INTELLIGENCE

Collective Intelligence is an active field of research that predates the web (O'reilly, 2007). Web 2.0 is a term that has generated passionate emotions, ranging from being dismissed as marketing jargon to being anointed as the new or next generation of the Internet. It is widely regarded that harnessing collective intelligence is the key or core component to Web 2.0 applications (Karahan & Roehrig, 2016; Rogers et al., 2007). Scientists from the fields of sociology, mass behavior, and computer science have made importance contributions to this field. When a group of individuals collaborate or compete with each other, intelligence or behavior that otherwise did not exist suddenly emerges; this is commonly known as collective intelligence (Mataric, 1993; Wolpert & Tumere, 1999). As users interact on the web and express their opinions, they influence others. Their initial circle of influence is the group of individuals that most interact with. Because the web is a highly connected network of sites, this circle of influence grows and may shape the thoughts of everybody in group (Oyalabu, 2012).

YouTube is a good example that shows how a collective intelligence is utilized. YouTube. In October 2006, Google bought YouTube for $1.65 billion. In its 20 months of existence, YouTube had grown to be one of the busiest sites on the Internet, dishing out 100 million video views a day (Falch, Henten, Tadayoni, & Windekilde, 2009). It ramped from zero to more than 20 million unique user visits a day, with mainly viral marketing spread from person to person; similar to the way the pandemic flu spreads. In YouTude's case, each time a user uploaded a new video, she was easily able to invite others to view this video. As those others viewed this video, other related videos popped up as recommendations, keeping the user further engaged. Ultimately, many of these views also became submitters and uploaded their own videos as well. As the number of videos increased, the site became more and more attractive for new users to visit (Gill et al., 2007).

A recommendation made by a friend or person of influence can have a big impact on other users within the same group. Moreover, a review or comments about a user's experience with a particular provider or service is contextually relevant for the other users inquiring about that topic, especially if it's within the context of similar use (Ma et al., 2008). Recommender systems attempt to suggest items (books, movies, music, news, images, peer friend, learning contents, etc.) that are likely to interest the users. Typically, recommender systems are based on collaborative filtering, which is a technique that automatically predicts the interest of an active user by collecting rating information from other similar users or items (Jiang et al., 2012; Ma et al., 2011). Adding the ability for users to add tags, keywords or labels provided by a user to classify items of interest such as articles, items being sold, pictures, videos, podcasts, and so on is a powerful technique to solicit information from the user. Tags can also be generated by professional editors or by an automated algorithm that analyzes content (Eck et al., 2008; Suchanek & Gunawardena, 2008). These tags are used to classify data, bookmark sites, connect people with each other, aid users in their searches, and build dynamic navigation in your application, of which a tag cloud is one example (Caldarola & Rinaldi, 2016; Kaser & Lemire, 2007). Figure 1 shows a tag cloud showing popular tags at www.flicker.com, a popular photography site. In a tag cloud, tags are displayed alphabetically, with the size of the font representing the frequency of occurrence. The larger the font of the tag, the more frequently it occurs (Bateman et al., 2008).

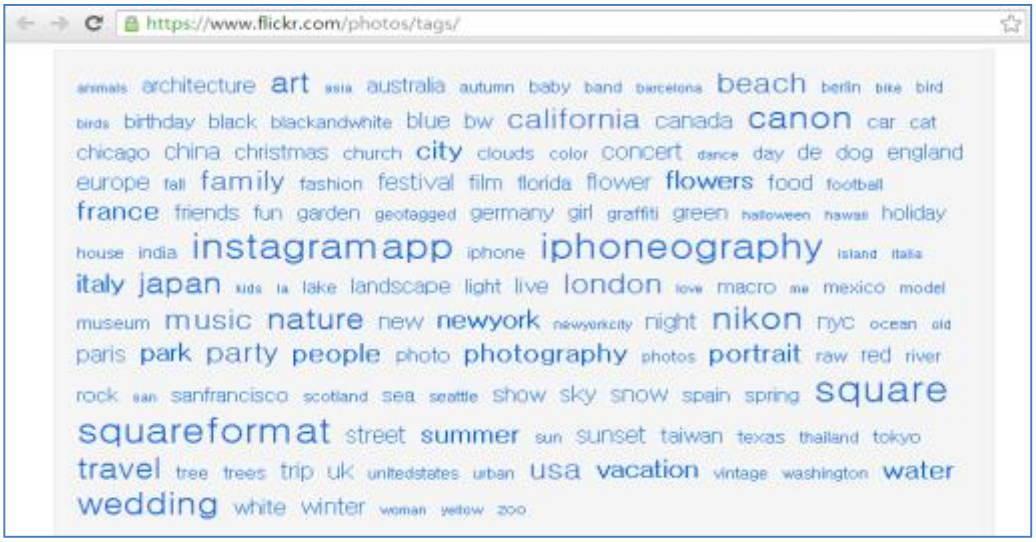

Figure 1: Tag cloud shows popular tags.

R-BASED DATA ANALYSIS

Various programs have been used for big data analysis. Although programs can be written using a general programming language for big data analysis, an objected-oriented language or function-based language is widely used due to the recent advancement of software. Instead of creating directly a complex algorithm such as Statistics and data mining, various programs that provide existing algorithms or data structure as a form of functions, packages, or libraries that can be used easily and quickly are being developed (Alzola & Harrell, 2002). In this paper, we employed a script-based open source 'R', which is generally used for statistical data analysis and processing. 'R' was developed under the influence of a statistical language 'S' developed by AT & T and is broadly used for big data mining (Ihaka, 1998). 'R' provides various analysis algorithms, graphic functions, and a GUI environment. 'R' is also flexible with program languages such as Java, C, and Python and works well with commercial databases for data management.

PREDICTION OF IT LEARNER'S LEARNING PATH

The ultimate goals of processing and utilizing big data are 1) to extract meaningful information from raw data, 2) to predict changes of information, and 3) to recommend users to make a good decision. In this research, we studied a way to predict how IT learners change their learning path. For this purpose, we used an associative analysis technique with the collective intelligence viewpoint.

THE STRUCTURE OF IT LEARNERS' DATA

According to the 2016 Google search trends (http://www.google.com/trends), the top 10 most searched IT terms are: 'SQL', 'HTML', 'Java', 'CSS', 'JavaScript', 'Python', 'PHP', '.NET Framework', 'C#' and 'C'. The top 10 keywords were utilized as IT learner's subjects in our research. Figure 2 illustrates interest changes of keyword search on the five software technologies (SQL, Java, Python, HTML, PHP) from 2004 to 2016.

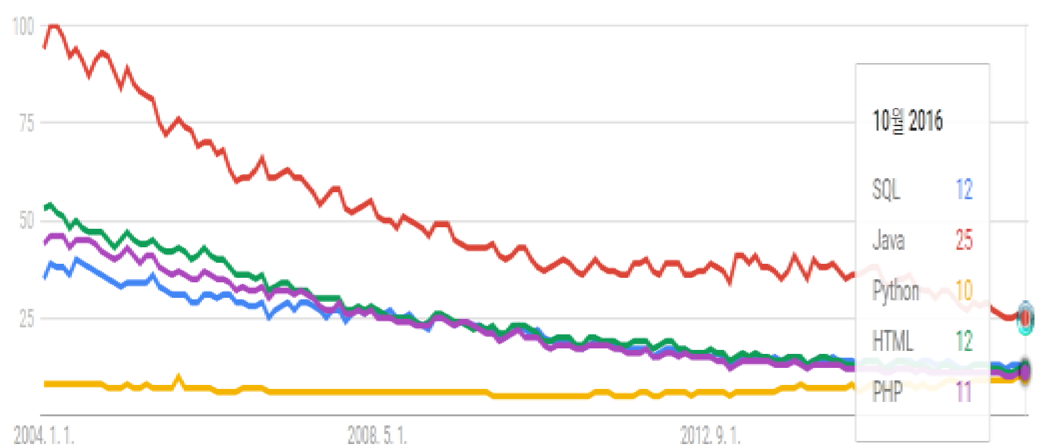

Figure 2: Interest changes of search keyword on the software technology areas

The numbers insie the box on the right side represent search interest relative to the highest point on the chart for the given region and time. For example, a value of 100 is the peak popularity for the term and a value of 50 means that the term is half as popular. Google Trends provides a time series index of the volume of queries users enter into Google in a given Geographic area. The query index is based on query share: the total query volume for the search term in question within a particular geographic region divide by the total number of queries in that region during the time period being examined. The maximum query share in the time period specified is normalized to be 100, and the query share at the initial data being examined is normalized to be zero (Carneria & Mylonakis, 2009; Choi & Varian, 2012).

For an association rule mining, we can define every subject item as a set of I = {i_1, i_2, i_3 ... i_m} and all of learners' learning transactions as a set of T = {t_1, t_2, t_3 ... t_n} as shown in table 1. A rule is defined as an implication of the form X → Y where X and Y belongs to I, and X ∩ Y is an empty set. The binary number 1 in the table means that the learning happens and zero means the leaning was not happened. If we apply an association rule mining to Table 1 for an association analysis, we can easily find the association from one subject to another. For example, the result of association from 'HTML' to 'CSS' (HTML→CSS) allows us to predict that likelihood of 'HTML learners will learn 'CSS'

T	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
t_1	1	0	1	0	1	1	0	0	1	1
t_2	0	1	1	0	0	0	1	0	1	0
t_3	1	1	0	1	1	1	1	0	0	0
t_4	1	1	1	0	0	1	0	1	1	1
t_5	1	1	1	0	1	0	0	1	0	1
……										
t_n	*bit*	*bit*	*bit*	*bit*	*bit*	*bit*	*bit*	*bit*	*bit*	*bit*

Table 1: Learning transaction storage structure

To explain the association between subjects represented as a binary number form, we can use support and confidence as shown in formula 1 and 2 respectively, where N is the total number of transactions, n(X U Y) is the number of occurrence of X or Y, n(X) is the number of occurrence of X, and n(Y) is the number of occurrence of Y.

Support: S (X → Y) = n(X U Y)/N (1)
Confidence: C (X → Y) = n(X ∩ Y)/n(X) (2)

As shown in the formula, the support is the ratio of item X or Y over all transactions, and the confidence is the ratio of item X and Y over the number of items X. For example, if the number of IT learners who studied HTML or CSS is 400 and the number of total transactions is 1000, the support S (HTML → CSS) is 400/1000 = 0.40. This means that the probability that IT learners will study either HTML or CSS is 0.40. Likewise, if the number of IT leaners who studied HTML and CSS is 200 and the number of IT learners who studied HTML only is 300 then the confidence C (HTML → CSS) is 200/300 = 0.66. This means the probability that IT learners will study the CSS is 0.66 given that IT learners have already studied HTML. The higher confidence in a given association rule X → Y, means more possibility that a transaction, which contains item X will contain item Y.

The lift is a calculated value of dividing the confidence (C (X → Y)) by the support, S (Y).

$$\text{Lift}(X, Y) = C(X \rightarrow Y)/S(Y) \qquad (3)$$

That is, the lift measures the improvement rate of confidence over item Y. Therefore, lift value greater than 1 indicates that the occurrence of X has a positive effect on Y or that X is positively correlated with Y. Lift value less than 1 indicates that the occurrence of X has a negative effect on the occurrence of Y or that X is negatively correlated with Y. A lift value near 1 or equals to 1 indicates that the occurrence of X has almost no effect on the occurrence of Y or that X and Y have zero correlation.

ASSOCIATION RULE DATA MINING BASED ON R

This research analyzed and experimented IT learners' data by utilizing an association rule mining technique, which is provided by R analysis tool. Apriori algorithm, which is a widely used search algorithm for association rules was used to find frequent item sets without calculating the Support for all item sets (Agrawal & Srikant, 1994; Srikant & Agrawal, 1995). Figure 3 explains an UML class diagram, which implements the Apriori algorithm as an arules package included in Hahsler, Grün, Hornik, and Buchta (2009) described the detail of this diagram.

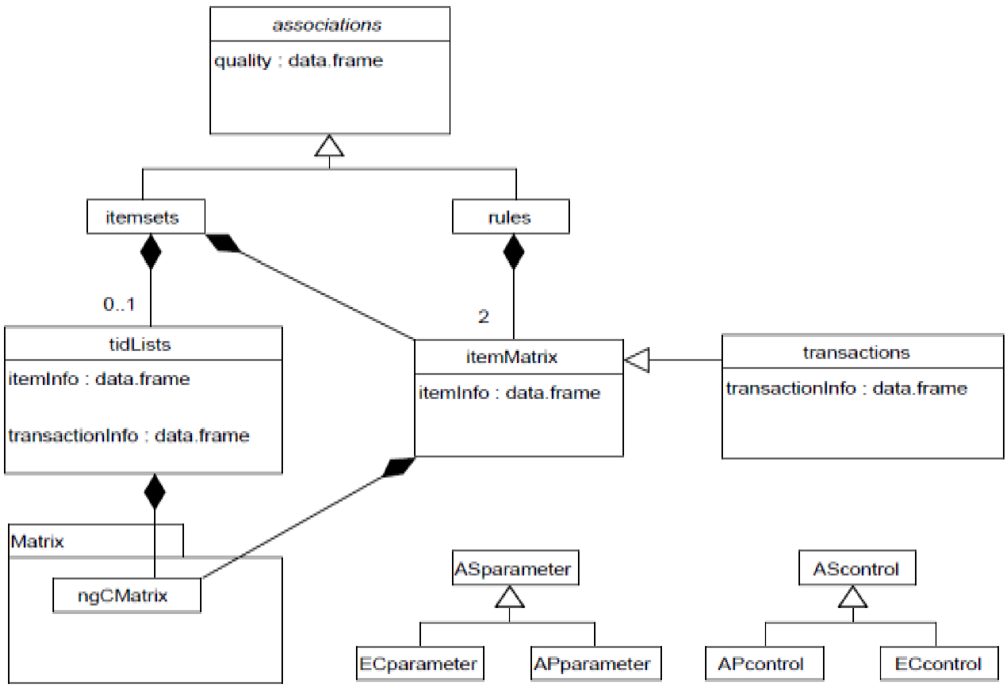

Figure 3: Arules package UML class diagram (Source: Hahsler, Grün, Hornik, & Buchta, 2009)

The arules package implements a transaction data association rule mining algorithm as a functional unit. This package was introduced by Agrawel, Imielinski, and Swami in 1993 and provides various functions for association rule mining analysis using Apriori function as a main one. The arules package also provides a R-extension package, arulesViz, for the UX-based various visualizations of association rule mining results (Hahsler & Chelluboina, 2011). The arulesViz package can illustrate mining results as different graph forms such as Scatter plot, Matrix-based plot, Grouped matrix-based plot, Network graph-based plot, Parallel coordinates plot, and Double decker plot. Utilizing these graph functions has been an active research topic.

EXPERIMENT RESULT AND DISCUSSION

We conducted a survey to ask IT learners which subjects they have studied among the most searched top 10 IT courses (‘SQL’, ‘HTML’, ‘Java’, ‘CSS’, ‘JavaScript’, ‘Python’, ‘PHP’, ‘.NET Framework’, ‘C#’ and ‘C’). Survey participants must either take a college course of a subject or familiar with the subject through the self-study to claim that they have studied the subject. The survey was conducted for 6 months until we collected data from 100 IT learners. Table 2 shows the first 20 IT learners survey results.

IT Learners	IT Courses Studied
1	{SQL, HTML, JavaScript}
2	{SQL, HTML, Java, CSS, JavaScript, Python, PHP, C#}
3	{SQL, CSS, PHP, C}
4	{ SQL, Java, JavaScript, PHP, .NET Framework, C#}
5	{SQL, HTML, CSS, JavaScript, PHP, .NET Framework, C#}
6	{SQL, CSS, JavaScript, Python, PHP, .NET Framework, C#}
7	{HTML}
8	{HTML, CSS, .NET Framework, C#}
9	{SQL, HTML, .NET Framework}
10	{HTML, CSS, JavaScript, .NET Framework, C#}
11	{SQL, HTML, Java, PHP, C#, }
12	{ SQL, HTML, , Python, PHP, .NET Framework, C#}
13	{Java, JavaScript, Python, PHP, C#}
14	{Java , Python, PHP, .NET Framework, C#, C }
15	{SQL, HTML, Java, CSS, JavaScript, Python, PHP, C#}
16	{JavaScript, .NET Framework, C}
17	{HTML, CSS, Python, PHP, .NET Framework}
18	{SQL, .Python, NET Framework}
19	{SQL, Java, CSS, C#}
20	{HTML, CSS, JavaScript, Python, .NET Framework, C#}

Table 2: IT learner's data items

We converted the IT learners' transaction data into binary mapping data. Table 3 shows a corresponding conversion result of the table 2 which contains columns for the top 10 most searched subjects and rows for individual IT learners. The binary mapping data were analyzed by arules package in R to find association rules with the apriori algorithm. 460 association rules were generated, and table 4 describes the summary of quality measures. The first 2 pages and the last page of 460 association rules are described in the Appendix A. According to the table 4, an association rule {.NET, C} => {HTML} has the minimum support (0.1020) which means the probability that IT learners, study {.NET Framework and C language} or HTML is low. Whereas, an association rule {} => {Python} has the maximum support (0.5300) which means the probability that IT leaners who study Python is high, i.e., Python is the most popular subject to IT leaners among 10 subjects.

IT Learners	SQL	HTML	Java	CSS	JavaScript	Python	PHP	.NET Framework	C#	C
1	1	1	0	0	1	0	0	0	0	0
2	1	1	1	1	1	1	1	0	1	0
3	1	0	0	1	0	0	1	0	0	1
4	1	0	1	0	1	0	1	1	1	0
5	1	1	0	1	1	0	1	1	1	0
6	1	0	0	1	1	1	1	1	1	0
7	0	1	0	0	0	0	0	0	0	0
8	0	1	0	1	0	0	0	1	1	0
9	1	1	0	0	0	0	0	0	1	0
10	0	1	0	1	1	0	0	0	1	1
11	1	1	1	0	0	0	1	0	1	0
12	1	1	0	0	0	1	1	1	1	0
13	0	0	1	0	1	1	1	0	1	0

14	0	0	1	0	0	0	1	1	1	1
15	1	1	1	1	1	1	1	0	1	0
16	0	0	0	0	1	0	0	1	0	1
17	0	1	0	1	0	1	1	1	0	0
18	1	0	0	0	0	1	0	1	0	0
19	1	0	1	1	0	0	0	0	1	1
20	0	1	0	1	1	1	0	1	1	0

Table 3 Binary mapping data for IT learners' data items

Also, an association rule {SQL} => {Python} or vice versa has a 0.28 support which is the highest value among non-empty left-hand side association rules. This indicates that the likelihood that IT learners who study either SQL or Python is high.

Measures	Support	Confidence	Lift
Min	0.1020 {.NET, C} => {HTML}	0.4126 {HTML, JavaScript} => {.NET}	0.8453 {.NET, C} => {HTML}
1st. Quarter	0.1200	0.4805	0.9615
Median	0.1290	0.5000	0.9899
Mean	0.1583	0.4986	0.9902
3rd Quarter	0.1432	0.5146	1.0199
Max	0.5300 {} => {Python}	0.5942 {SQL, .NET} => {Python}	1.1567 {JavaScript, C#} => {CSS}

Table 4: Summary of quality measures

An association rule {HTML, JavaScript} => {.NET} has a minimum confidence

(0.4126) which means probability that IT learners study .Net Framework is low given that the IT learners have studied HTML and JavaScript. Whereas, an association rule {SQL, .NET} => {Python} has the maximum confidence (0.5942) which means the probability that IT learners study Python is high given that the IT learners have studied SQL and .NET. An association rule {.NET, C} => {HTML} has the minimum lift (0.8453). This means studying .NET Framework and C language has negatively correlated with studying HTML. There is no wonder why this association rule also has a minimum support (0.1020) as shown in the table 4. Whereas, an association rule {JavaScript, C#} => {CSS} has a highest lift value (1.1567) which means left hand side and right hand side are positively related, more specifically the probability that IT learners study CSS when they study JavaScript and C# is increased by 15.67.

According to the table 5, the order from the most studied subject to the least studied subject is as follows: Python, JavaScript, SQL, HTML, C#, C, PHP, CSS, Java, .NET. This table also indicates that 53.0% of the participants studied Python, 51.6% of the participants studied JavaScript, 51.4% of the participant studied SQL and so on. Another indication is that differences of the Support between subjects are very minimal. The difference between rule number 1 and 10 is only 0.048 (0.53 - 0.482 = 0.048), and the average Support is 0.5035 which indicates that in average, more than 50% of IT learners studied all 10 subjects. This proves that all subjects we selected are popular to IT learners.

#	Association Rule	Support	Confidence	Lift
1	{} => {Python}	0.53	0.53	1
2	{} => {JavaScript}	0.516	0.516	1
3	{} => {SQL}	0.514	0.514	1
4	{} => {HTML}	0.507	0.507	1
5	{} => {C#}	0.506	0.506	1
6	{} => {C}	0.504	0.504	1
7	{} => {PHP}	0.497	0.497	1
8	{} => {CSS}	0.496	0.496	1
9	{} => {Java}	0.483	0.483	1
10	{} => {.NET}	0.482	0.482	1

Table 5: Order of popularity

We identified 38 association rules that left-hand side has a positive effect on the right-hand side (Lift > 1.0) among the total of 99 rules that has an only one subject on the left side. These rules are shown in Table 6. First 4 rules indicate that the likelihood of IT learners study SQL (0.5283), HTML (0.5113), CSS (0.4981) and .NET Framework (0.4962) is high in this order given that IT leaners have already studied Python. Based on this information, we can recommend IT learners who have studied Python the next good subjects to study are SQL, HTML, CSS and .NET. From the rule number 5 and 6, we can recommend IT learners who have already studies JavaScript, to study CSS and HTML as next subjects. We can draw an IT learners' learning path as shown in figure 4 using the information given in table 6. Each learning path indicates the next most to least preferable subject from top to bottom, clock-wise in that order. We notice that CSS is positively correlated with 8 other subjects (JavaScript, Python, HTML, SQL, C#, C, PHP, and .Net Framework). This implies that if IT learners study CSS they can be recommended to study variety of subjects. PHP is also positively correlated with 6 subjects (SQL, HTML, C, C#, CSS, and .NET Framework). Another noticeable item is that Java has a positive effect on only one subject, .Net Framework. C and C# have a positive effect on 2 (CSS and PHP) and 3 subjects (HTML, PHP, and CSS) respectively.

#	Association Rule	Support	Confidence	Lift
1	{Python} => {SQL}	0.280	0.5283	1.0278
2	{Python} => {HTML}	0.271	0.5113	1.0085
3	{Python} => {CSS}	0.264	0.4981	1.0040
4	{Python} => {.NET}	0.263	0.4962	1.0295
5	{JavaScript} => {CSS}	0.276	0.5349	1.0783
6	{JavaScript} => {HTML}	0.269	0.5213	1.0282
7	{SQL} => {Python}	0.280	0.5447	1.0278
8	{SQL} => {PHP}	0.264	0.5136	1.0334
9	{SQL} => {CSS}	0.255	0.4961	1.0002
10	{HTML} => {Python}	0.271	0.5345	1.0085
11	{HTML} =>	0.269	0.5306	1.0282

	{JavaScript}			
12	{HTML} => {PHP}	0.262	0.5168	1.0398
13	{HTML} => {C#}	0.261	0.5148	1.0174
14	{HTML} => {CSS}	0.255	0.5030	1.0140
15	{C#} => {HTML}	0.261	0.5158	1.0174
16	{C#} => {PHP}	0.256	0.5079	1.0220
17	{C#} => {CSS}	0.255	0.5060	1.0200
18	{C} => {CSS}	0.255	0.5040	1.0160
19	{C} => {PHP}	0.253	0.5000	1.0060
20	{PHP} => {SQL}	0.264	0.5312	1.0334
21	{PHP} => {HTML}	0.262	0.5271	1.0398
22	{PHP} => {C}	0.256	0.5151	1.0220
23	{PHP} => {C# }	0.253	0.5091	1.0060
24	{PHP} => {CSS}	0.251	0.5050	1.0182
25	{PHP} => {.NET}	0.241	0.4849	1.0060
26	{CSS} => {JavaScript}	0.276	0.5565	1.0784
27	{CSS} => {Python}	0.264	0.5323	1.0042
28	{CSS} => {HTML}	0.255	0.5141	1.0160
29	{CSS} => {SQL}	0.255	0.5141	1.0002
30	{CSS} => {C#}	0.255	0.5140	1.0160
31	{CSS} => {C}	0.255	0.5141	1.0201
32	{CSS} => {PHP}	0.251	0.5060	1.0182
33	{CSS} => {.NET}	0.240	0.4829	1.0039
34	{Java} => {.NET}	0.235	0.4865	1.0094

35	{.NET} => {Python}	0.263	0.5456	1.0295
36	{.NET} => {PHP}	0.241	0.5000	1.0060
37	{.NET} => {CSS}	0.240	0.4980	1.0039
38	{.NET} => {Java}	0.235	0.4876	1.0094

Table 6: Association rules that have a positive effect

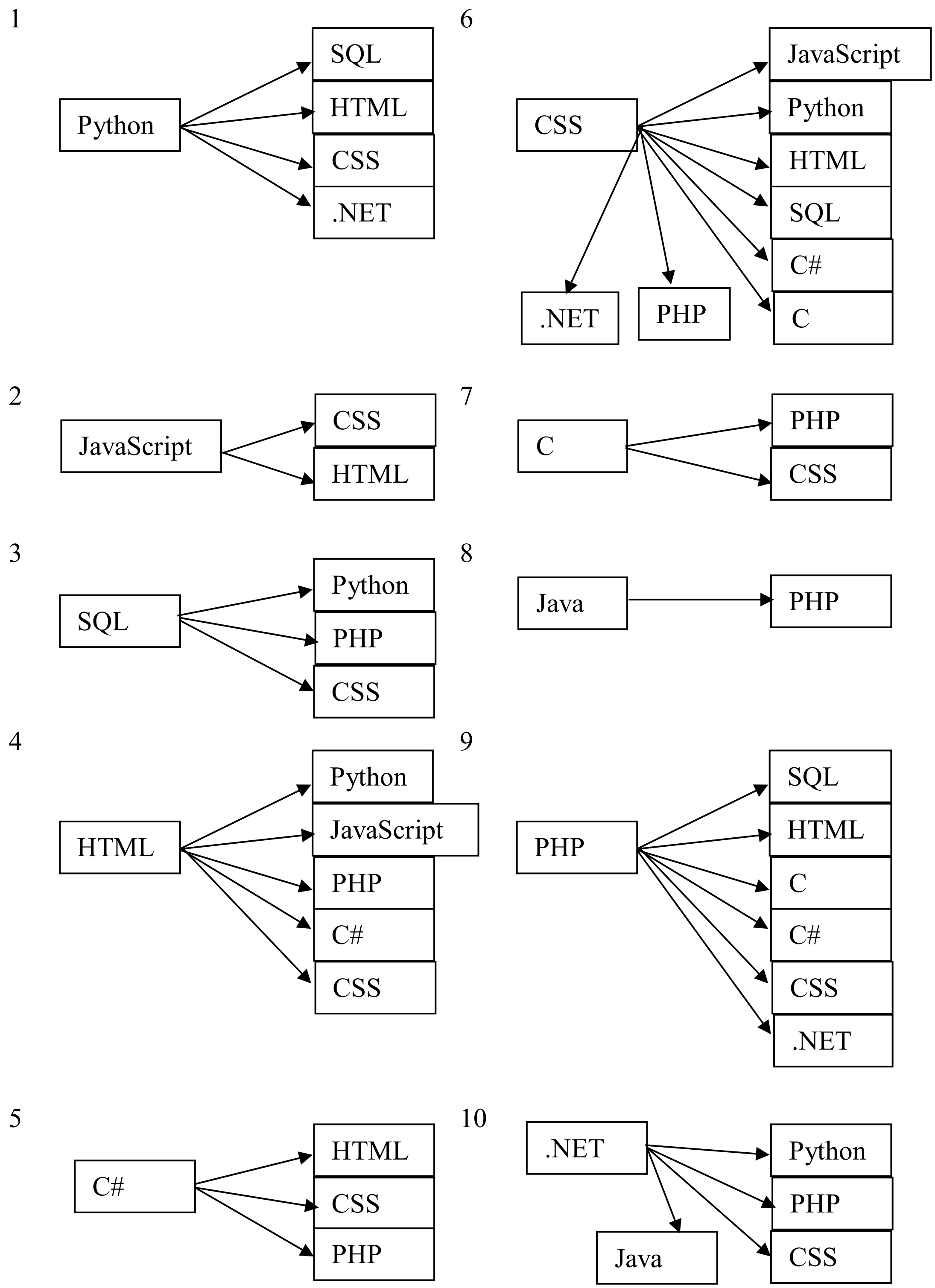

Figure 4: IT Learners' learning path

Figure 5 shows a multi-level IT learners' learning path which enables us to recommend the sequence of subjects that rudimentary IT learners can study. Based on this information, we create table 7 that shows some possible study sequence of subjects. Every recommended sequence starts with Python because Python has a highest Support value (0.53) among 10 subjects.

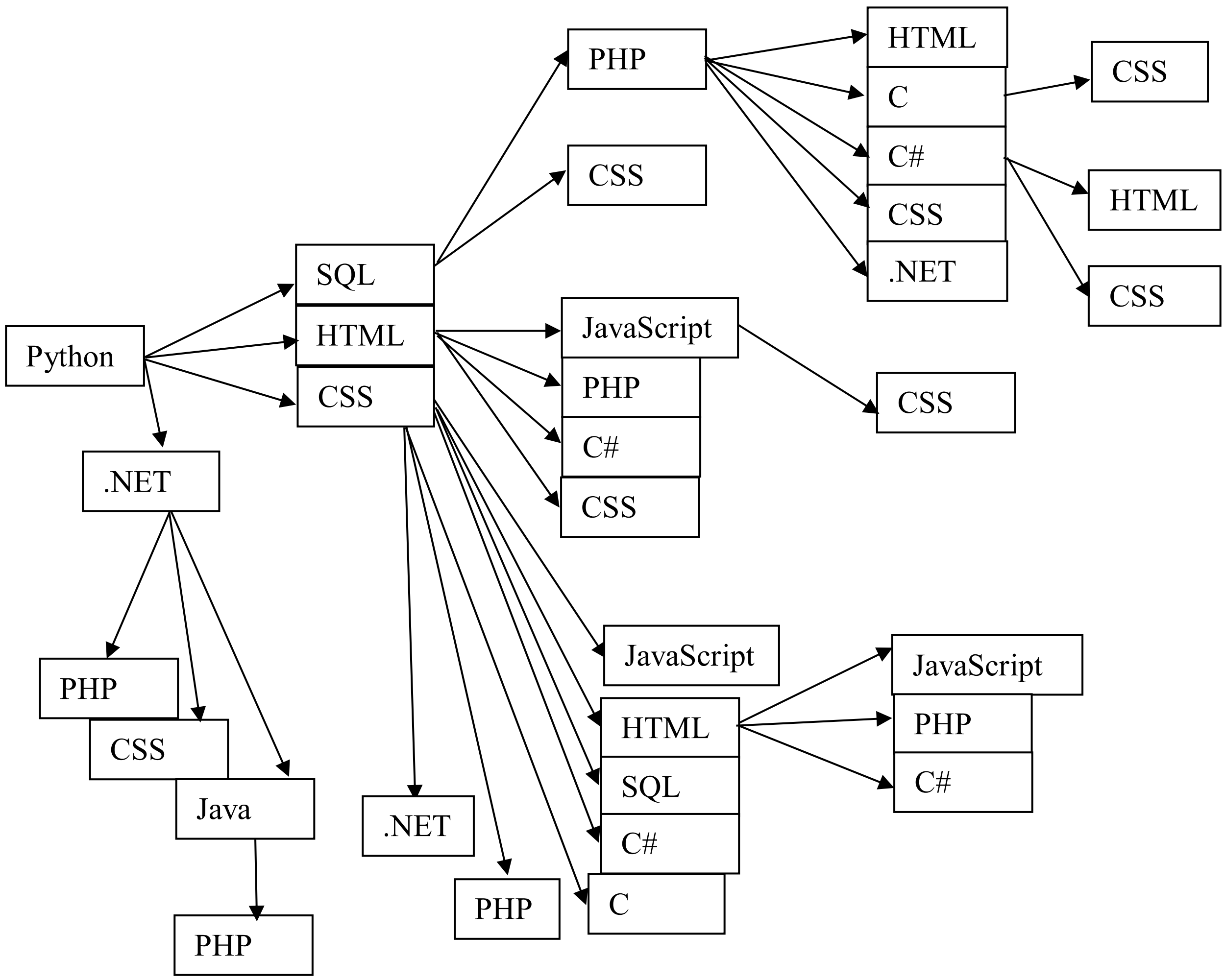

Figure 5: IT Learners' multi-level learning path

#	Recommended Possible Learning Path
1	Python => SQL => PHP => HTML => JavaScript => CSS => C#; .NET => Java; C
2	Python => SQL => PHP => HTML => JavaScript => CSS => C; .NET => Java; C#
3	Python => SQL => PHP => HTML => JavaScript => CSS => .NET => Java; C#; C
4	Python => SQL => PHP => C => CSS => HTML; JavaScript; .NET => Java; C#;
5	Python => SQL => PHP => C => CSS => C# => JavaScript; .NET => Java; C#;
6	Python => SQL => PHP => C => CSS =>.NET => Java; C#; HTML=>JavaScript
7	Python => SQL => PHP => C# => JavaScript => CSS; .NET => Java; C; HTML
8	Python => HTML => JavaScript => CSS => SQL => PHP => C; .NET => Java; C#
9	Python => HTML => PHP => C => CSS => JavaScript; .NET => Java; C#; SQL
10	Python => HTML => C# => CSS => .NET => Java; JavaScript; C#; SQL; PHP
11	Python => HTML => CSS => PHP => .NET => Java; JavaScript; C; C#; SQL; HTML
12	Python => CSS => JavaScript; HTML => PHP => C; .NET => Java; C#; SQL
13	Python => CSS => HTML => JavaScript; SQL => PHP => C; .NET => Java; C#
14	Python => CSS => SQL => PHP => HTML => JavaScript; C; .C#; NET => Java

15	Python => CSS => C# => HTML => PHP => C; JavaScript; SQL; .NET => Java
16	Python => CSS => PHP => HTML => .NET => Java; C; JavaScript; SQL; C#
17	Python => .NET => PHP => C => CSS => HTML => JavaScript; SQL; C#; Java
18	Python => .NET => CSS => C# => HTML => PHP => C; SQL; JavaScript; Java
19	Python => .NET => Java => PHP => C => CSS; SQL; JavaScript; C# => HTML

Table 7: Recommended learning path

Figure 6 illustrates a grouped matrix for 460 association rules. The bigger the circle is, the higher correlation is represented between left-hand side and right-hand side subjects. Figure 7 shows a grouped matrix for 85 association rules. The 85 association rules were generated by a pruning algorithm provided by R, which removes some of associations that provide little power. A pruning process reduces the complexity and improves predictive accuracy.

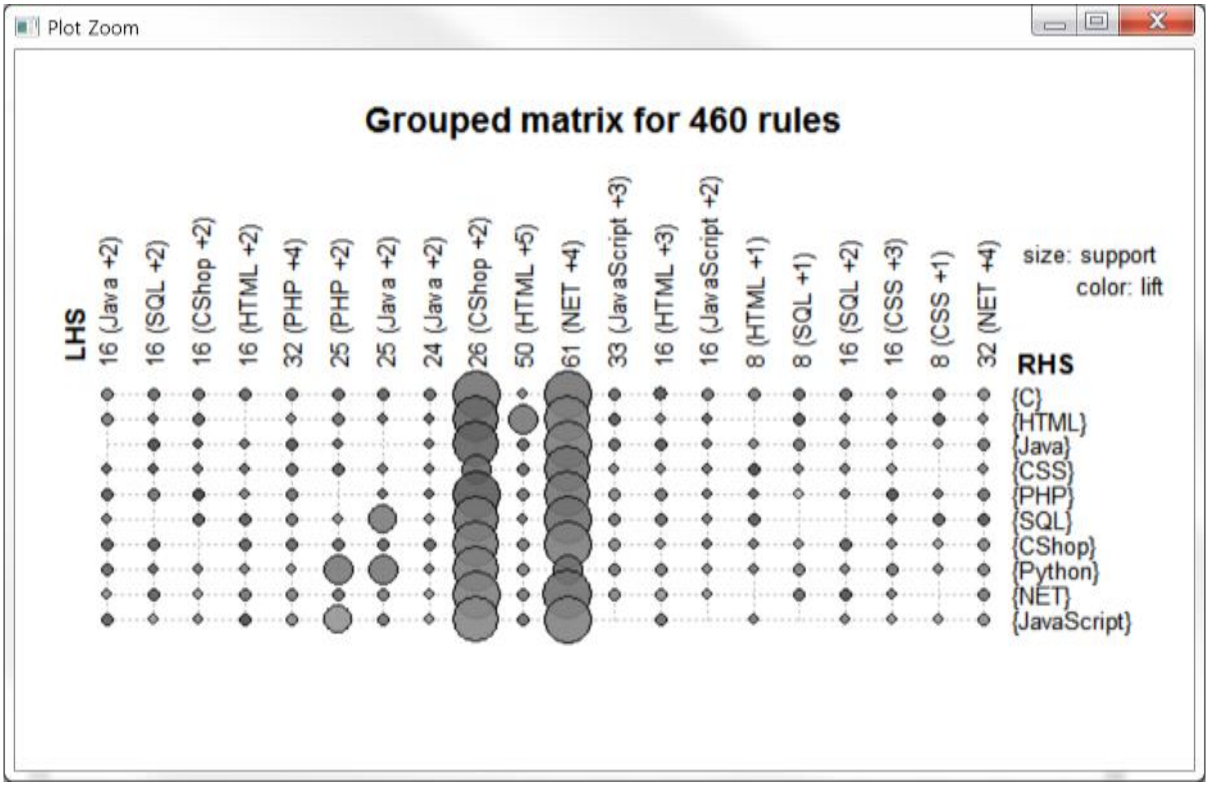

Figure 6: Grouped matrix before pruning

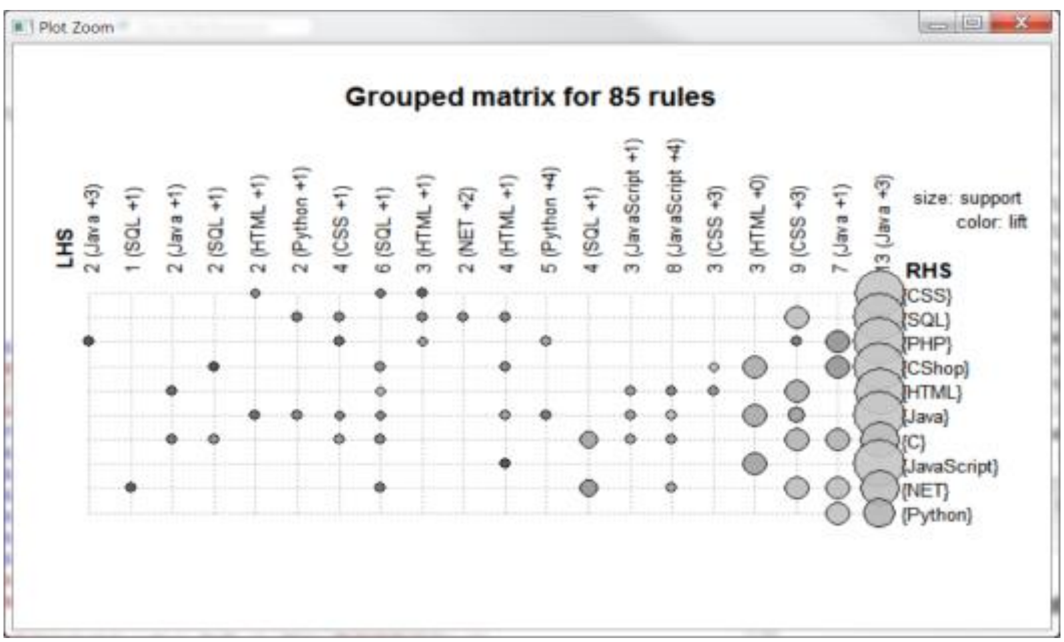

Figure 7: Group matrix after pruning

Also, we visualized the 85 association rules using arulesViz as shown in figure 8.

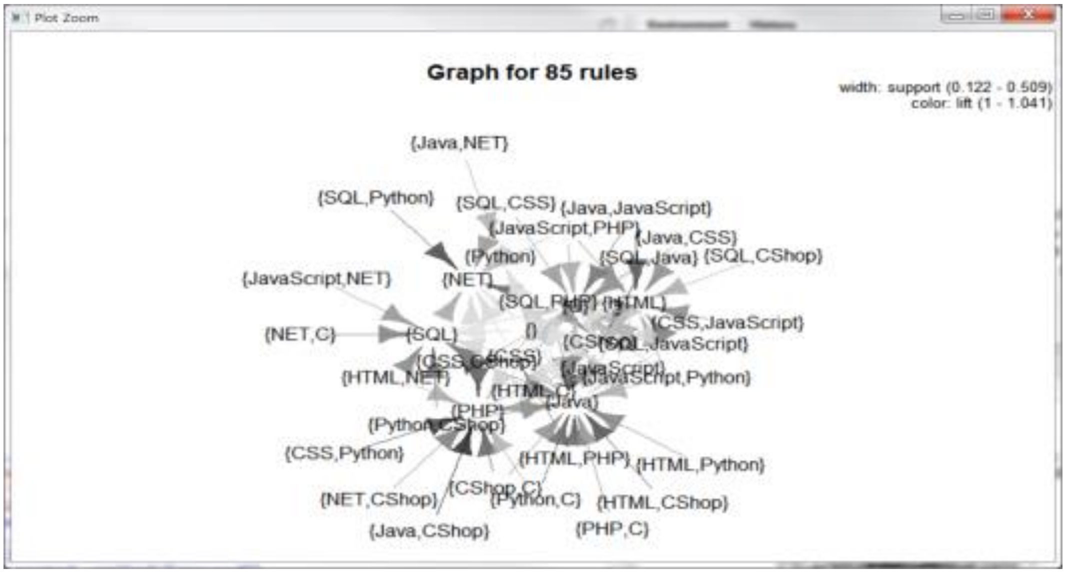

Figure 8: IT course association graph

CONCLUSION

Using the top 10 most searched IT subjects obtained from Google search trends, we collected and stored IT learners' learning propensity as a binary mapping data, We analyzed the binary data using a data mining technique with an association rule algorithm implemented in R package to suggest a way to predict and recommend IT learner's learning path from a collective intelligence viewpoint. To find out the learning path we utilized the Support, Confidence and Lift values. We found that all of the 10 subjects were studied by more than 50% of IT leaners with a very minimal variance in a Support value. Among the 10 subjects, Python was the most studied by IT learners, followed by JavaScript, SQL, HTML, C#, PHP, CSS, Java, and .NET. We identified 38 association rules that tell us the probability that IT learners study a subject given that they have already studied a particular subject. By utilizing these rules, we came up with IT learner's learning path and recommended some possible sequences of subjects for rudimentary IT learners. As a result of this research, it is expected that more useful knowledge and a better decision making on the learning process would be provided to IT learners.

We think further studies on the changes of popularity of the top 10 most searched IT subjects would be interesting. In particular, finding which subjects still remains popular and which subjects become unpopular, and what are the factors that make these changes might also be an interesting research topic. In addition, finding IT learners' learning propensity in developed, developing, and under developed countries, and comparing them each other might be a good future research topic.

REFERENCES

Abbasi, A., Sarker, S., & Chiang, R. (2016). Big data research in information systems: Toward inclusive research agenda. *Journal of the Association for Information Systems, 17*(2), i-xxxii.

Agrawal, R., and Srikant, R. (1994). Fast Algorithms for mining association rules. *VLDB,* 487-499.

Alzola, C. F., & Harrell, F. E. (2002). An introduction to S and the HMISC and design libraries. [http://cran.r-project.org/doc/contrib/Alzola+Harrell-Hmisc-Design-Intro.pdf]

Apache Hadoop, http://hadoop.apache.org

Atriwal, L., Nagar, P., Tayal, S., & Gupta, V. (2016). Business Intelligence Tools for Big Data. *Journal of Basic and Applied Engineering Research, 3*(6), 505-509.

Bateman, S., Gutwin, C., & Nacenta, M. (2008). Seeing things in the clouds: the effect of visual features on tag cloud selections. In Proceedings of the nineteenth ACM conference on Hypertext and hypermedia. *ACM,* 193-202.

Borthakur, D., Gray, J., Sarma, J. S., Muthukkaruppan, K., Spiegelberg, N., Kuang, H., & Aiyer, A. (2011). Apache hadoop goes realtime at facebook. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data. *ACM*, 1071-1080.

Buckle, S. (2012). Introduction to VoltDB, retrieved April 08, 2016 from [http://www.ibm.com/developerworks/opensource/library/os-voltdb/os-voltdb-pdf.pdf]

Caldarola, E. G., & Rinaldi A. M. (2016). Improving the visualization of WordNet large lexical database through semantic tag cloud. *IEEE International Congress on Big Data (BigData Congress),* 34-41.

Carneiro, H. A., & Mylonakis, E. (2009). Google trends: a web-based tool for real-time surveillance of disease outbreaks. *Clinical Infectious Diseases*, *49*(10), 1557-1564.

Choi, H., & Varian, H. (2012). Predicting the present with google trends. *Economic Record,* 88(s1), 2-9.

Eck, D., Lamere, P., Bertin-Mahieux, T., & Green, S. (2008). Automatic generation of social tags for music recommendation. *In Advances In Neural Information Processing Systems*, 385-392.

Falch, M., Henten, A., Tadayoni, R., & Windekilde, I. (2009). Business models in social networking. *In CMI Int. Conf. on Social Networking and Communities.*

Fan, W., & Bifet, A. (2013). Mining Big Data: Current Status, and Forecast to the Future. *ACM SIGKDD Explorations Newsletter, 14*(2), pp.1-5.

Färber, F., Cha, S. K., Primsch, J., Bornhövd, C., Sigg, S., & Lehner, W. (2012). SAP HANA database: data management for modern business applications. *ACM Sigmod Record*, *40*(4), 45-51.

Gill, P., Arlitt, M., Li, Z., & Mahanti, A. (2007). Youtube traffic characterization: a view from the edge. In Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. *ACM, 15-28*

Hahsler, M., Grün, B., Hornik, K., & Buchta, C. (2009). Introduction to arules–A computational environment for mining association rules and frequent item sets. *The Comprehensive R Archive Network*. [http://cran.r-project.org/web/packages/arules/vignettes/arules.pdf]

Hahsler, M., & Chelluboina, S. (2011). Visualizing association rules: Introduction to the r-extension package arulesviz. *R Project Module*. [http://cran.r-project.org/web/packages/arulesViz/vignettes/arulesViz.pdf

Ihaka, R. (1998). R: Past and future history. *Computing Science And Statistics*, 392-396. [http://www.stat.auckland.ac.nz/~ihaka/downloads/Interface98.pdf]

Isard, M., Budiu, M., Yu, Y., Birrell, A., & Fetterly, D. (2007). Dryad: distributed data-parallel programs from sequential building blocks. *In ACM SIGOPS Operating Systems Review*, *41*(3), 59-72.

Jiang, M., Cui, P., Liu, R., Yang, Q., Wang, F., Zhu, W., & Yang, S. (2012, October). Social contextual recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 45-54). ACM.

Karahan, E., & Roehrig, G. (2016). Use of web 2.0 technologies to enhance learning experiences in alternative school settings. *International Journal of Education in Mathematics, Science and Technology. 4*(4), 272-283.

Kaser, O., & Lemire, D. (2007). Tag-cloud drawing: Algorithms for cloud visualization. arXiv preprint cs/0703109.

Lamb, A., Fuller, M., Varadarajan, R., Tran, N., Vandiver, B., Doshi, L., & Bear, C. (2012). The vertica analytic database: C-store 7 years later. *Proceedings of the VLDB Endowment, 5*(12), 1790-1801.

Lasota, M., Deniziak, S., & Chrobot, A. (2016). An SDDS-Based architecture for a real-time data store. *I.J. Information Engineering and Electronic Business, 1*, 21-28.

Ma, H., Yang, H., Lyu, M. R., & King, I. (2008). Sorec: social recommendation using probabilistic matrix factorization. In Proceedings of the 17th ACM conference on Information and knowledge management. *ACM,* 931-940.

Ma, H., Zhou, D., Liu, C., Lyu, M. R., & King, I. (2011). Recommender systems with social regularization. In Proceedings of the fourth ACM international conference on Web search and data mining. *ACM,* 287-296.

Mataric, M. J. (1993). Designing emergent behaviors: From local interactions to collective intelligence. In Proceedings of the Second International Conference on Simulation of Adaptive Behavior, 432-441.

McKinsey Global Institute. (2011). Big Data :The next frontier for innovation, competition and Productivity. *McKinsey and Company.*

Olston, C., Chiou, G., Chitnis, L., Liu, F., Han, Y., Larsson, M., & Wang, X. (2011). Nova: continuous pig/hadoop workflows. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data. *ACM,* 1081-1090.

O'reilly, T. (2007). What is Web 2.0: Design patterns and business models for the next generation of software. *Communications and Strategies*, *65*(1), 17-37.

Oyalabu, A. O. (2012). Twitter: Businesses Increasing Their Revenues 140 Characters at a Time.

Rogers, P. C., Liddle, S. W., Chan, P., Doxey, A., & Isom, B. (2007). Web 2.0 learning platform: Harnessing collective intelligence. *Online Submission, 8*(3), 16-33.

Sharma, S., & Singh, S. (2016). Evolving an algorithm to generate sparse inverted index using Hadoop and Pig. *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems*, *2*, 499-508.

Singh, M., & Leonhardi, B. (2011). Introduction to the ibm netezza warehouse appliance. In Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research. *IBM Corp,* 385-386.

Srikant, R., & Agrawal, R. (1995). Mining generalized association rules. *VLDB,* 407-419.

Stonebraker, M., & Weisberg, A. (2013). The voltdb main memory dbms. *IEEE Data Eng. Bull., 36*(2), 21-27.

Suchanek, F. M., Vojnovic, M., & Gunawardena, D. (2008). Social tags: meaning and suggestions. In Proceedings of the 17th ACM conference on Information and knowledge management. *ACM,* 223-232.

Technical Overview, High performance, scalable, RDBMS for big data and real-time analytics, retrieved April 11, 2016 from http://voltdb.com/downloads/datasheets_collateral/technical_overview.pdf

Thusoo, A., Sarma, J. S., Jain, N., Shao, Z., Chakka, P., Anthony, S., & Murthy, R. (2009). Hive: a warehousing solution over a map-reduce framework. Proceedings of the *VLDB Endowment, 2*(2), 1626-1629.

Waas, F. M. (2009). Beyond conventional data warehousing—massively parallel data processing with Greenplum database. *In Business Intelligence for the Real-Time Enterprise*, 89-96. Springer Berlin Heidelberg.

Wolpert, D. H., & Tumer, K. (1999). An introduction to collective intelligence. arXiv preprint cs/9908014.

Appendix A

460 ASSOCIATION RULES

1	{}	=>	{Python}	0.53	0.53	1
2	{}	=>	{JavaScript}	0.516	0.516	1
3	{}	=>	{SQL}	0.514	0.514	1
4	{}	=>	{HTML}	0.507	0.507	1
5	{}	=>	{CSharp}	0.506	0.506	1
6	{}	=>	{C}	0.504	0.504	1
7	{}	=>	{PHP}	0.497	0.497	1
8	{}	=>	{CSS}	0.496	0.496	1
9	{}	=>	{Java}	0.483	0.483	1
10	{}	=>	{NET}	0.482	0.482	1
11	{SQL}	=>	{Python}	0.28	0.544747	1.027825
12	{Python}	=>	{SQL}	0.28	0.528302	1.027825
13	{CSS}	=>	{JavaScript}	0.276	0.556452	1.078395
14	{JavaScript}	=>	{CSS}	0.276	0.534884	1.078395
15	{HTML}	=>	{Python}	0.271	0.534517	1.008522
16	{Python}	=>	{HTML}	0.271	0.511321	1.008522
17	{HTML}	=>	{JavaScript}	0.269	0.530572	1.02824
18	{JavaScript}	=>	{HTML}	0.269	0.521318	1.02824
19	{C}	=>	{Python}	0.264	0.52381	0.98832

20	{Python}	=>	{C}	0.264	0.498113	0.98832
21	{PHP}	=>	{SQL}	0.264	0.531187	1.033438
22	{SQL}	=>	{PHP}	0.264	0.513619	1.033438
23	{CSS}	=>	{Python}	0.264	0.532258	1.004261
24	{Python}	=>	{CSS}	0.264	0.498113	1.004261
25	{JavaScript}	=>	{SQL}	0.264	0.511628	0.995385
26	{SQL}	=>	{JavaScript}	0.264	0.513619	0.995385
27	{NET}	=>	{Python}	0.263	0.545643	1.029515
28	{Python}	=>	{NET}	0.263	0.496226	1.029515
29	{JavaScript}	=>	{Python}	0.263	0.50969	0.961679
30	{Python}	=>	{JavaScript}	0.263	0.496226	0.961679
31	{PHP}	=>	{HTML}	0.262	0.527163	1.039769
32	{HTML}	=>	{PHP}	0.262	0.516765	1.039769
33	{PHP}	=>	{Python}	0.262	0.527163	0.994647
34	{Python}	=>	{PHP}	0.262	0.49434	0.994647
35	{CSharp}	=>	{HTML}	0.261	0.515813	1.017377
36	{HTML}	=>	{CSharp}	0.261	0.514793	1.017377
37	{CSharp}	=>	{SQL}	0.258	0.509881	0.991987
38	{SQL}	=>	{CSharp}	0.258	0.501946	0.991987
39	{C}	=>	{PHP}	0.256	0.507937	1.022005

40	{PHP}	=>	{C}	0.256	0.515091	1.022005
41	{C}	=>	{SQL}	0.256	0.507937	0.988203
42	{SQL}	=>	{C}	0.256	0.498055	0.988203
43	{C}	=>	{CSS}	0.255	0.505952	1.020065
44	{CSS}	=>	{C}	0.255	0.514113	1.020065
45	{CSharp}	=>	{CSS}	0.255	0.503952	1.016033
46	{CSS}	=>	{CSharp}	0.255	0.514112	1.016033
47	{CSS}	=>	{HTML}	0.255	0.514113	1.014029
48	{HTML}	=>	{CSS}	0.255	0.502959	1.014029
49	{CSS}	=>	{SQL}	0.255	0.514113	1.00022
50	{SQL}	=>	{CSS}	0.255	0.496109	1.00022
51	{HTML}	=>	{SQL}	0.255	0.502959	0.978519
52	{SQL}	=>	{HTML}	0.255	0.496109	0.978519
53	{CSharp}	=>	{PHP}	0.253	0.5	1.006036
54	{PHP}	=>	{CSharp}	0.253	0.509054	1.006036
55	{C}	=>	{JavaScript}	0.252	0.5	0.968992
56	{JavaScript}	=>	{C}	0.252	0.488372	0.968992
57	{PHP}	=>	{JavaScript}	0.252	0.507042	0.98264
58	{JavaScript}	=>	{PHP}	0.252	0.488372	0.98264
59	{CSharp}	=>	{JavaScript}	0.251	0.496047	0.961332

60	{JavaScript}	=>	{CSharp}	0.251	0.486434	0.961332
61	{PHP}	=>	{CSS}	0.251	0.50503	1.018206
62	{CSS}	=>	{PHP}	0.251	0.506048	1.018206
63	{C}	=>	{CSharp}	0.25	0.496031	0.9803
64	{CSharp}	=>	{C}	0.25	0.494071	0.9803
65	{CSharp}	=>	{Python}	0.249	0.492095	0.928481
66	{Python}	=>	{CSharp}	0.249	0.469811	0.928481
67	{Java}	=>	{Python}	0.248	0.513458	0.968788
68	{Python}	=>	{Java}	0.248	0.467925	0.968788
69	{NET}	=>	{SQL}	0.244	0.506224	0.984872
70	{SQL}	=>	{NET}	0.244	0.474708	0.984872
71	{Java}	=>	{HTML}	0.244	0.505176	0.996402
72	{HTML}	=>	{Java}	0.244	0.481262	0.996402
73	{Java}	=>	{JavaScript}	0.244	0.505176	0.979023
74	{JavaScript}	=>	{Java}	0.244	0.472868	0.979023
75	{Java}	=>	{C}	0.243	0.503106	0.998225
76	{C}	=>	{Java}	0.243	0.482143	0.998225
77	{Java}	=>	{CSharp}	0.243	0.503106	0.99428
78	{CSharp}	=>	{Java}	0.243	0.480237	0.99428
79	{C}	=>	{HTML}	0.243	0.482143	0.950972

		=		0.24		0.95097
80	{HTML}	>	{C}	3	0.47929	2
		=		0.24	0.50207	0.97301
81	{NET}	>	{JavaScript}	2	5	3
		=		0.24	0.46899	0.97301
82	{JavaScript}	>	{NET}	2	2	3
		=		0.24		1.00603
83	{NET}	>	{PHP}	1	0.5	6
		=		0.24		1.00603
84	{PHP}	>	{NET}	1	0.48491	6
		=			0.49792	1.00388
85	{NET}	>	{CSS}	0.24	5	2
42		=		0.11	0.46721	0.94006
0	{Java, JavaScript}	>	{PHP}	4	3	7
42		=		0.11	0.45238	0.93660
1	{JavaScript, PHP}	>	{Java}	4	1	7
42		=		0.11	0.47280	0.91628
2	{NET, CSharp}	>	{JavaScript}	3	3	6
42		=		0.11	0.46694	0.92281
3	{JavaScript, NET}	>	{CSharp}	3	2	1
42	{JavaScript,	=		0.11	0.45019	0.93402
4	CSharp}	>	{NET}	3	9	3
42		=		0.11		0.92481
5	{PHP, NET}	>	{HTML}	3	0.46888	2
42		=		0.11	0.52073	1.04776
6	{HTML, NET}	>	{PHP}	3	7	1
42		=		0.11	0.43129	0.89480
7	{HTML, PHP}	>	{NET}	3	8	9
42		=		0.11	0.46311	0.90100
8	{HTML, Java}	>	{SQL}	3	5	2
42		=		0.11	0.47083	0.92866
9	{SQL, Java}	>	{HTML}	3	3	5
43		=		0.11	0.44313	0.91746
0	{SQL, HTML}	>	{Java}	3	7	8
43		=		0.11	0.46861	0.92429
1	{NET, CSharp}	>	{HTML}	2	9	8
43		=		0.11	0.51612	1.02001
2	{HTML, NET}	>	{CSharp}	2	9	8
43		=		0.11	0.42911	0.89028
3	{HTML, CSharp}	>	{NET}	2	9	8

434	{Java, CSS}	=>	{SQL}	0.112	0.474576	0.9233
435	{SQL, Java}	=>	{CSS}	0.112	0.466667	0.94086
436	{SQL, CSS}	=>	{Java}	0.112	0.439216	0.909349
437	{NET, C}	=>	{JavaScript}	0.111	0.466387	0.90385
438	{JavaScript, NET}	=>	{C}	0.111	0.458678	0.910075
439	{JavaScript, C}	=>	{NET}	0.111	0.440476	0.913851
440	{HTML, NET}	=>	{JavaScript}	0.111	0.511521	0.991319
441	{JavaScript, NET}	=>	{HTML}	0.111	0.458678	0.90469
442	{HTML, JavaScript}	=>	{NET}	0.111	0.412639	0.856098
443	{Java, CSharp}	=>	{Python}	0.111	0.45679	0.861868
444	{Java, Python}	=>	{CSharp}	0.111	0.447581	0.884547
445	{Python, CSharp}	=>	{Java}	0.111	0.445783	0.922946
446	{Java, NET}	=>	{CSharp}	0.109	0.46383	0.91666
447	{NET, CSharp}	=>	{Java}	0.109	0.456067	0.944238
448	{Java, CSharp}	=>	{NET}	0.109	0.44856	0.930622
449	{HTML, NET}	=>	{SQL}	0.107	0.493088	0.959314
450	{SQL, NET}	=>	{HTML}	0.107	0.438525	0.86494
451	{SQL, HTML}	=>	{NET}	0.107	0.419608	0.870556
452	{Java, NET}	=>	{PHP}	0.105	0.446809	0.899011
453	{PHP, NET}	=>	{Java}	0.105	0.435685	0.902039

45		=		0.10	0.45652	0.94714
4	{Java, PHP}	>	{NET}	5	2	1
45		=		0.10	0.43829	0.86449
5	{Java, NET}	>	{HTML}	3	8	3
45		=		0.10	0.47465	0.98272
6	{HTML, NET}	>	{Java}	3	4	1
45		=		0.10	0.42213	0.87579
7	{HTML, Java}	>	{NET}	3	1	1
45		=		0.10	0.42857	0.84530
8	{NET, C}	>	{HTML}	2	1	9
45		=		0.10	0.47004	0.93263
9	{HTML, NET}	>	{C}	2	6	1
46		=		0.10	0.41975	0.87085
0	{HTML, C}	>	{NET}	2	3	7