IDENTIFYING THE COMPETENCY MIX REQUIRED FOR THE ROMANIAN IT LABOR MARKET

Cristian Georgescu
cristian.georgescu@ugal.ro
„Dunarea de Jos" University of Galati, Romania

We are using data mining processes for this analysis with the aim of identifying the competency mix that offers the most chances for successfully getting hired within the target market. This analysis has been based on an association algorithm, similar to those used for shopping carts. This analysis leads us to a better allocation of our training efforts and generates a more accurate competency development plan.

Keywords: workforce market, IT competencies, data mining, association algorithm
JEL codes: M510, J240, R230

1. Introduction

During the "Informatics – a successful career" workshop – organized to mark the 10 year anniversary of the first graduates from Economic Informatics specialization – was made a questionnaire addressed to student participants. We got a series of apparently contradictory results, but which reveal important aspects at a closer analysis.

Table 1: Results questionnaire from "Informatics a successful career" workshop (Enache, 2012)

	In a small measure	**In a great measure**
Will specializing in Economic Informatics help you gain the necessary abilities to get hired in this domain?	24.07%	75.93%
Do you think you will have difficulties in finding a job corresponding to your training?	38.89%	61.11%
Do you see what you learned during faculty helping you in solving duties at your current / future workplace?	46.30%	53.70%
Do you think specializing in Economic Informatics will give you all necessary competencies in order to become a specialist?	57.41%	42.59%
Do you think specializing in Economic Informatics offers you competencies needed to understand related fields?	50.00%	50.00%
Do you think optional Informatics courses would be useful for perfecting yourselves, regardless of your domain of interest?	11.11%	88.89%

Correlating the question in which 75.93% of students consider that specializing in Economic Informatics would help them, in a great measure, to gain necessary abilities to get hired in this market, with the question in which 57.42% of students answer that specializing in Economic Informatics offers them, in a small measure, the needed competencies to become a specialist in this domain shows the superior positioning of this domain, on the workforce market, rather than an inability to gain needed competencies. These two answers can also be interpreted as an image capital these students have, as they graduate, that proves not to be perfectly viable when they get hired.

The slight majority (53.70%) of students that think what they learned during faculty will help them in solving duties from their workplace is not encouraging and the fact that a majority of students

does not consider gained abilities to be sufficient for getting hired in related domains represents a potential upset regarding this specialization.

All things considered, by far the most disappointing answers were those in which 61.11% of students consider they will have difficulties in finding a job according to their background – we eliminate the option of over-qualification – and the one in which 88.89% of students consider that optional courses are necessary for perfecting themselves. An informal, open questionnaire addressed to final year students later on, failed to prove relevant for identifying these competencies that almost 90% of students consider useful for supplementary development.

In order to have a correct answer that is able to guide our training and perfecting of future workforce from this domain, we have started collecting data regarding competencies required on the IT workforce market, using job postings displayed on websites of recruiting agencies.

The data for this analysis consists of approximately 1,200 job offers, from which over 130 competencies, which were required by employers, have been synthesized resulting in over 5,000 "job post – competency" combinations.

2. Analysis of competencies with Clustering Algorithm

Clustering is an unsupervised classification of elements, observations, patterns, etc. into groups, called clusters. In supervised classification the starting point is a set of labeled patterns (pre-classified) while in clustering, the unsupervised classification, as starting point you have a collection of unlabeled patterns that must be distributed in significant clusters (Jain, Murty, & Flynn, 1999).

From all segmentation and data partitioning algorithms applicable to the present issue, the ones available within SQL Server are K-means and Expectation-Maximization (EM) clustering algorithms. The main concept behind K-means algorithm is the centroid. The centroid of a set of tuples is the most representative tuple from the set, chosen by calculating the distance between it and the other elements through various methods (McCaffrey, 2013). The name of K-means algorithm comes from the division into k clusters of C_j where j = 1,…,k and their mean is c_j. The value of parameter k is set at the start of running the algorithm and the value c_j is the calculated weight centre of points from C_j. This algorithm is also called hard-clustering because one element may belong to a single cluster (Microsoft, 2013).

As opposed to K-means, which calculates distances, the EM algorithm (Dempstern, Laird, & Rdin, 1977) uses a probabilistic method for assigning elements into clusters. Each iteration consists of two steps (Jiawei, Kamber, & Pei, 2012): expectation step (E-step) and maximization step (M-step). In the first step missing data are estimated using observed data and estimates for the model's parameters. In the second step, the probability function is maximized assuming that missing data is known. This method is called soft-clustering because resulting clusters may overlap, they are not disjoint.

Using the EM clustering algorithm, where each job post is characterized by a number of competencies required by the employer, the parameter regarding the maximum number of clusters was set to five. After running there have resulted four clusters containing, each, around the same number of job posts (Figure 1). We chose the first ten competencies in order of probability from each cluster, competencies which, through their importance, set the profile of each cluster. Table 2 has resulted, containing competencies that have appeared among the top ten most significant competencies in at least a cluster (top 10). Probabilities attached to each competency are calculated as the number of job posts containing the competency from the first column (Variables), divided by the total number of job posts from that respective cluster. Within the rest of data, no competency from any clusters exceeds 5%.

We observe that some competencies from the "top 10" appear in a single cluster, while others are present in more than one. SQL[11] and JavaScript[12] appear in all clusters, while HTML[13], OOP[14] and Java[15] appear in three clusters, each.

[11] SQL (Structured Query Language) - programming language to query relational databases
[12] JavaScript – object oriented programming language run by the browser, used for building websites
[13] HTML (HyperText Markup Language) –code used for creating websites
[14] OOP - Object Oriented Programming
[15] Java - object oriented programming language

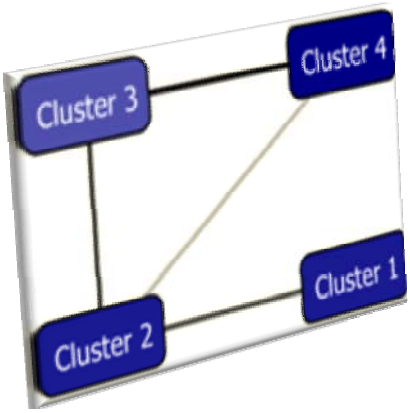

Figure 1: Cluster diagram

Table 2: Probabilities' distribution of competencies per clusters

Variables	Cluster 1	Cluster 2	Cluster 3	Cluster 4
(SQL)	**35.70%**	**47.80%**	**62.40%**	**32.00%**
(JavaScript)	**5.10%**	**27.50%**	**66.00%**	**81.00%**
(HTML)	0.30%	**17.30%**	**80.40%**	**83.00%**
(PHP)	4.10%	9.10%	**66.50%**	**69.40%**
(CSS)	0.00%	13.90%	**53.40%**	**81.70%**
(OOP)	2.90%	**34.20%**	**21.00%**	**33.60%**
(Windows)	***54.30%***	7.30%	3.00%	1.30%
(Java)	**6.00%**	**27.80%**	**27.80%**	2.70%
(C++)	**20.40%**	**31.00%**	**5.80%**	0.00%
(jQUERY)	0.00%	8.90%	**13.40%**	**52.50%**
(C#)	4.80%	***37.60%***	2.70%	9.20%
(AJAX)	0.50%	12.90%	8.90%	***42.00%***
(Microsoft Office)	***35.10%***	4.60%	2.00%	0.20%
(Oracle)	**21.70%**	**14.30%**	6.20%	0.30%
(Unix)	**22.40%**	8.60%	**9.20%**	2.10%
(C)	**14.60%**	**16.70%**	3.50%	2.50%
(linux)	***11.80%***	7.90%	2.20%	7.70%
(MySQL)	0.50%	4.00%	1.50%	***29.60%***
(ASP.NET)	0.90%	***18.50%***	0.10%	6.70%
(XML)	1.80%	5.80%	5.70%	***16.90%***
(Adobe Photoshop)	0.10%	1.60%	***10.50%***	7.70%

These results might be analyzed based on some resemblances between clusters represented by the links in Figure 1, but this type of analysis is aimed towards finding common characteristics between job posts present on the market, as they were grouped in clusters, and this is slightly different from our objective: identifying the competency mix that offers the greatest chances to get hired.

3. Analysis of competencies with Association Algorithm

In this respect, we continued our analysis by using an algorithm designed to identify rules of association. A rule of association is of the following form:

$$A_1, A_2, ..., A_m \rightarrow B_1, B_2, ..., B_n$$

, where A_i and B_j are items. The significance of this rule for data derived from a transactional system is: when articles A_1, A_2,..., A_m appear within a transaction, then with a certain probability articles B_1, B_2,..., B_n will also appear in the same transaction. The formal model of association rules (Agrawal, Imielinski, & Swami, 1993) was developed with the aim of analyzing databases containing commercial transactions.

Take $I = \{i_1, i_2,..., i_m\}$a set of elements, articles offered for sale (product catalog) and $D = \{t_1, t_2,..., t_n\}$ a set of transactions, where each transaction has a unique identifier and it contains a set of articles $t_i = \{i_{i1}, i_{i2},..., i_{ik}\}$ where $i_{ij} \in I$. An association rule has the following form: $X \Rightarrow Y$ where X and Y are sets of articles, item-sets, having the following properties: $X, Y \subseteq I$ and $X \cap Y = \varnothing$. X is called antecedent and Y the rule's consequence.

The frequency of an item-set is the total number of transactions containing that item-set.

The support of a rule X ⍰Y is the number of transactions containing X ⍰ Y and it is expressed in either absolute value or percentage of the group's total number of transactions.

s = number of transactions containing either X or Y / the total number of transactions

The significance of the *s* support shows that the association rule is valid in *s*% of all the transactions and it measures the frequency with which it appears in the database.

Confidence of an association rule $X \Rightarrow Y$ is the rapport between the number of transactions containing $X \cup Y$ and the total number of transactions containing X. The confidence value of a rule may be interpreted as being an estimate of the conditioned probability of finding the rule's consequence on other transactions that contain its antecedent, P(Y|X). In other words, if X and Y are in the same basket, which means bought by the same person, then Y will also be in other baskets where X is with a probability given by the confidence level. This measures the power of a rule.

The purpose of shopping basket analysis is to identify all $X \Rightarrow Y$ rules with a minimum support and a minimum trust. These values are given as input for the problem underpinning the selection of interesting rules.

The lift measure of a rule as it is proposed by (Brin, Motwanit, & Silverstein, 1997) is the rapport

$$P(X,Y) / P(X) P(Y)$$

in other words, it measures the correlation between the two sets X and Y, in case X and Y are independent.

When a rule has a higher frequency than another, within the same data set, it is more important than the other one. In certain circumstances, the frequency of generating rules may be affected. To correct these variations, the indicator called "importance" or "interesting score" can be calculated. First, calculate the rapport between the right side of the rule's probability conditioned by the left side of the rule and the right side of the rule's probability conditioned by the NON left side of the rule and then extract logarithm from this rapport (MacLennan, Tang, & Crivat, 2008).

$$\text{Importance } (X => Y) = \log (P(Y|X) / P(Y|\text{not } X))$$

Having a positive value signifies a direct correlation and a negative value means inverse correlation. The 0 value means that the two sets are independent.

Microsoft Association Algorithm (Microsoft, 2013), used in ulterior processing is an implementation of the Apriori algorithm (Agrawal & Srikant, 1994), the most renowned algorithm for discovering association rules.

This algorithm was developed for shopping cart analysis aiming to discover certain patterns in customers' behavior. From the first version of this algorithm, there were hundreds of tryouts aiming to develop it and make it applicable for a wider array of situations these contributions being systemized in the paper (Han, Cheng, & Xin, 2007). An improvement of the algorithm's performances was realized by adding the item-set's weight parameter as a supplementary step in identifying them (Lei, 2012).

In our context $I = \{i_1, i_2,..., i_m\}$ is the set of i competencies required by the employer and $D = \{t_1, t_2,..., t_n\}$ represents the set of t job posts. Rephrasing our problem into shopping cart terms, employers have certain tendencies towards consumption needing specific sets of competencies that they materialize by creating a vacancy and posting it along with the competencies required for that job. It's like going to the market, picking a set of competencies and putting them into their basket.

4. Dependencies between competencies on the Romanian IT market

The first trap in this analysis process appears when finding certain rules with maximum probability and importance. Undoubtedly there are correlations between specific competencies that are required by each other when getting hired, but they are less frequent considering the number of cases they appear in. Therefore, in this analysis it is important not to miss the support parameter for each of the rules. The association rule's probability, even if it is big, can be rendered insignificant should it be calculated based on a small number of appearances, in other words this refers to niche intercorrelations.

In the same way, if no limit is imposed regarding the maximum number of competencies within a rule, then we get situations with ten competencies as antecedent of the rule and probability 1.

The maximum value of a probability suggests there have been a small number of cases on which it is based rather than suggesting they are sure rules. To confirm this assumption we agreed to a minimum number of two sets on which to base the creation of a rule and have observed these rules have not been selected. This shows that these rules were built based on a single job post from the collected data, hence having probability of 1.

Data mining is a creative process, based on imagination as well as repeated attempts in which fine tuning variations are introduced for running parameters aiming to discover the truly important elements from all data.

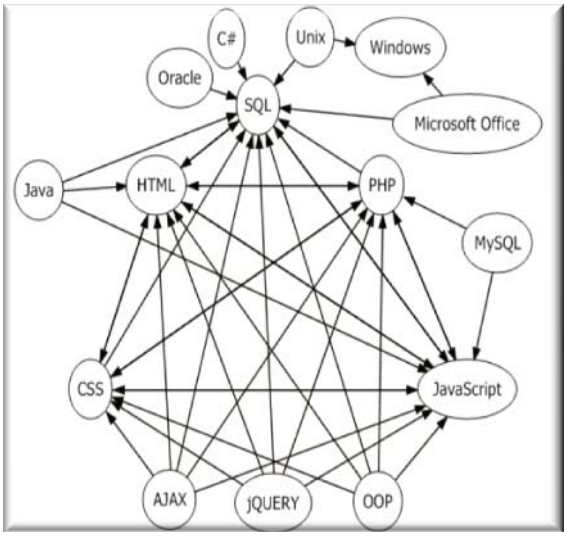

Figure 2: Dependency Network with 5% minimum number of sets

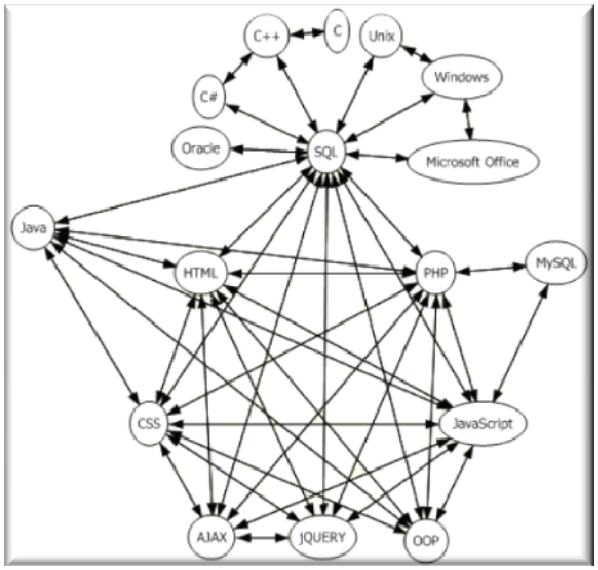

Figure 3: Dependency Network with 5% minimum number of sets and minimum probability of 10%

Thus, we raised the minimum number of sets of competencies that can base a rule to 5% of the total sets in order to eliminate accidental combinations. From running this algorithm we obtained 120 rules, out of which PHP appears in 49 of them, SQL in 34, HTML in 54, CSS[16] in 48 and JavaScript in 53, AJAX[17], jQUERY[18] and OOP are grouped from 17 to 19 rules and the rest in far less rules. In Figure 2 each node represents a competency and the arcs show probabilistic dependencies between them.A graphical analysis using Figure 2 shows a greater density of dependencies, corresponding to a greater number of association rules, within the area of SQL, HTML, PHP[19], CSS şi JavaScript competencies. These have been identified as main competencies both for the big number of rules in which they appear, but also for the high probabilities of these rules.

If the minimum number of sets remains at 5% and the minimum probability for a rule to count is decreased to 10%, supplementary dependencies appear, corresponding to rules having probabilities ranging between 10% and 40% (Figure 3).

Modifying minimum parameters from one stage to another was made in order to choose a certain granularity level that further allows highlighting of either more general aspects or details, according to present needs.

In Figure 3 the competency group that stands out is formed by jQUERY, AJAX, OOP and Java, which determine and are determined by the main competencies. Between Java and OOP, as well as between jQUERRY and AJAX, there are mutual dependencies that have not been highlighted so far.

We again observe what we stated earlier, that the Java, OOP, jQUERRY and Ajax competency group is in strong link with PHP, HTML, CSS and JavaScript, with the mention that for PHP and JavaScript the MySQL[20] competency is added. The arrows in the previous figures indicate dependencies between competencies, meaning the existence of association rules. Therefore, competencies like OOP, Java, jQUERRY and AJAX are added as equally important to the five main competencies.

This level of representation was used for highlighting and analyzing main competencies determined before. These competencies have already been observed in Figure 2 and Figure 3 but are less visible due to the extra number of dependencies. Modifying running parameters only made these competencies more obvious.

[16] CSS (Cascading Style Sheets) – language for describing presentations in a more accessible way for web documents users
[17] AJAX (Asynchronous JavaScript and XML) – programming technique for web applications creation
[18] jQUERY – JavaScript development platform
[19] PHP – programming language used in developing web pages and applications
[20] MySQL – relational database management system

5. Conclusions

From the educator's point of view, the common nucleus of competencies on the market is very important. At the main competencies level, we have SQL competency which is common for both database direction and the HTML, CSS and JavaScript group for web development.

Regarding competencies from fundamental preparation, competency patterns C, C++, C# and Unix, Windows, Microsoft Office requested in Romania highlight, along other common elements, a restructuring of preferences. In Romania two training guidelines are outlined, one based on programming language competencies from the C/C++ C# category and another, separated towards operating systems and office.

For the second level of competencies, those which appear in dependency with main competencies, we have the Oracle competency for databases and jQUERRY and AJAX competencies for web development.

A close analysis of Figure 3 and the set of associated rules helps identify three main groups of rules with common characteristics: the group of rules with negative importance, the group of rules with high importance which do not include main competencies and those in close connection with them and, finally, the group of rules with the main competencies and those depending on them.

These groups will be analyzed separately by filtering input data, obtained rules and by presenting output data in an appropriate manner so as to extract essential information.

References

1. *Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. Proceedings of the 20th VLDB Conference, (pp. 487-499). Santiago, Chile.*
2. *Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining Association Rules between Sets of Items in Large Databases. Proceedings of the ACM SIGMOD International Conference on Management of Data , 207-216.*
3. *Brin, S., Motwanit, R., & Silverstein, C. (1997). Beyond Market Baskets: Generalizing Association Rules to Correlations. Proceeding of the 1997 ACM-SIGMOD international conference on management (pp. 265–276). Tucson, AZ: ACM Inc.*
4. *Dempstern, A. P., Laird, M., & Rdin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological), Vol. 39, No. 1. , 1-36.*
5. *Enache, M. (2012, martie). Informatica - o carieră de succes. Retrieved iunie 2012, from www.edugal.ro: http://www.edugal.ro/www/index.php*
6. *Han, J., Cheng, H., & Xin, D. (2007). Frequent pattern mining: current status and future. Data Mining and Knowledge Discovery , 55-86.*
7. *Jain, A., Murty, M., & Flynn, P. J. (1999). Data Clustering: A Review. ACM Computing Surveys, Vol. 31(3) , 264-323 .*
8. *Jiawei, H., Kamber, M., & Pei, J. (2012). Data Mining Concepts and Techniques Third Edition. Waltham, MA 02451, USA: Elsevier Inc.*
9. *Lei, C. (2012). The Research of Data Mining Algorithm Based on Association Rules. The 2nd International Conference on Computer Application and System Modeling.*
10. *MacLennan, J., Tang, Z., & Crivat, B. (2008). Data Mining with SQL Server 2008. Indianapolis: Wiley Publishing, Inc.*
11. *McCaffrey, J. (2013, February). Detecting Abnormal Data Using k-Means Clustering. Retrieved April 2013, from MSDN Microsoft Corporation: http://msdn.microsoft.com/en-us/magazine/jj891054.aspx*
12. *Microsoft. (2013). Microsoft Association Algorithm Technical Reference. Retrieved June 2013, from MSDN.*
13. *Microsoft. (2013). Microsoft Clustering Algorithm Technical Reference. Retrieved June 2013, from MSDN: http://msdn.microsoft.com/en-us/library/cc280445.aspx*