Proceedings of I-KNOW '05
Graz, Austria, June 29 - July 1, 2005

Learning Skills from Data Based on XML Structured Qualification Profiles

Alexander Holland
(University of Dortmund, Germany
Alexander.Holland@udo.edu)

Klaus Moritz Peitzsch
(PRO DV Software AG, Germany
Klaus.Peitzsch@prodv.de)

Abstract: In this paper we address and discuss the approach of learning employee skills from data based on XML structured profiles and their representation as a Bayesian network. For extracting new information we use a dependency analysis approach. Many enterprise resource management systems (ERP) come along with integrated modules for Human Resource Management (HRM). One main task of HRM is to manage, improve and deploy the right skills at the right time. These processes are well known as skill management. Furthermore the problem of finding hidden or implicit dependencies between employee skills is considered. Using an information theoretical approach to construct a powerful skill representation as graphical model is recommendable. To demonstrate the achievement of the learned network structure, a test scenario is given, where historical reference project data is used.

Keywords: Skill Management, Information Theory, Uncertainty, XML, Qualification Profiles, Balanced Scorecard, Bayesian Network

Category: I.2.4, I.2.6, G.3

1 Introduction

A Bayesian network, as it is a formal graphical language, is a powerful instrument to represent knowledge especially when methods like reasoning under uncertainty are taken into consideration for further analysis. Many articles describe the practical deployment for real life scenarios like logistic or medical diagnosis. Bayesian structures exist to model uncertainty scenarios in network balanced scorecards [Holland 2004] as extensions of the traditional balanced scorecard method [Kaplan and Norton 1997] and to use balanced scorecards to measure supply chain performance [Brewer and Speh 2000]. In this article we will not use the available skill data to represent and calculate with them for the matching of skill profiles to minimize the gap between desired and available skills (e.g. computer language skills regarding C++ for an IT project). Instead our focus lies on applying an algorithm for Bayesian network learning without node ordering to discover dependencies [Holland and Peitzsch 2004] between social, technical, methodical and personal skills automatically. We will benefit from XML as a standard data format and use it to represent the qualification profiles.

The advantage of this approach is to detect the main influencing skills while simultaneously the amount of nodes within the skill data map is reduced by eliminating skills with less impact. We apply the formerly discussed approach in a test scenario introduced in chapter 4.

2 Probabilistic Models and Bayesian Inference

Bayesian networks are graphical models to represent knowledge under conditions of uncertainty. The use of such probabilistic models is based on direct acyclic graphs (DAG) with a probability table for each node. The nodes ς in a Bayesian network represent propositional variables in a domain, the edges E between the nodes represent the dependency relationship among the variables. Each node has a conditional probability table $P(X \mid X_1,\ldots,X_n)$ attached that quantifies the effects that the parents $X_1,\ldots,X_n$ have on the node. We could say that the conditional probabilities encode the strength of dependencies among the variables. For each X ε ς a conditional probability distribution is defined that specifies the probabilities of ς given the values of the parents of X. For instance, the recruitment and personal development of qualified employees can be represented using graphical models that facilitate a decision process consistent with the company's strategic planning. Based on the gathered skills of the employees a modern decision maker has to integrate this knowledge into the decision making process. The decision maker reaches decisions by combining his own knowledge, experience and intuition with that available from other sources. Given a learned network structure like Bayesian networks the decision maker can implement additional information by applying an inference algorithm. We use the learned Bayesian network to calculate new probabilities based on incoming incomplete data. For instance let A have n states with $P(A) = (x_1, ..., x_n)$ and assume that we get the information e that A can only be in state i or j. This statement expresses that all states except i and j are impossible, so next we can illustrate the probability distribution as $P(A, e) = (0,...,0, x_i, 0,...,0, x_j ,0, ...,0)$. Assume a joint probability table $P(U)$ where $\underline{e}$ is the preceding finding (n-dimensional table of zeros and ones). Using the chain rule for Bayesian networks [Russel and Norvig 2003] we can express the following

$$P(U, e) = \Pi \text{ A}\varepsilon\text{U } P(A \mid \text{parents}(A)) \cdot \Pi\ i\ \underline{e}_i \qquad (2.1)$$

and for A ε U we have

$$\boldsymbol{P(A,e)} = \frac{\sum_{U \setminus \{A\}} \boldsymbol{P(U,e)}}{\boldsymbol{P(e)}} \qquad (2.2).$$

On constructing Bayesian networks from skill data we use nodes to represent database attributes. Different Bayesian network structure learning algorithms have been developed. A good overview demonstrating general approaches to graphical probabilistic model learning from data is introduced by [Krause 1996], [Heckerman 1995] and [Borgelt and Kruse 2002]. In general we can distinguish between the search and score methods and the dependency analysis approach. In the first case the algorithm interprets the skill learning problem as searching for a structure that can fit the data best. The methods start as graphical representation without any edges, and then use some search method to add an edge to the representation. In the next step they can use score methods to compare the new with the older structure. The main problem to learn Bayesian networks using search and scoring methods is the NP-hard complexity. Representative algorithms belonging to the search and scoring method are polytree construction algorithms, the K2 algorithm applying a Bayesian scoring method or the Lam-Bacchus algorithm applying the minimal description length principle. Using the second dependency analysis method is a different approach. The algorithms try to discover the interdependencies of the data and then use these dependencies to infer the structure. Our approach introduced in the following chapter 3 belongs to the dependency analysis method without node ordering as extension. One representative of this second probabilistic model learning algorithms is the boundary DAG algorithm introduced by [Pearl 1988].

3 Learning Network Structures Based on Skill Data

The approach initiated here is based on dependency analysis. The conditional independence relationships play a fundamental role while using skill information. When learning networks from skill data, we can apply information theoretic measures to detect conditional independence relations and then use the d-separation concept [Pearl 1988] to infer the structures of networks. We measure the volume of the information flow between two nodes to see if a group of valves corresponding to a condition set can reduce and eventually block the information flow. In Bayesian networks we can determine information about the value of a node if we know the value of the other node and both nodes are dependent. The mutual information between two nodes can therefore provide us with information if two nodes are dependent and (also important) the degree of their relationship. The mutual information of two nodes X_i and X_j is defined as

$$\boldsymbol{Inf(X_i, X_j) = \sum_{x_i, x_j} P(x_i, x_j) \log \frac{P(x_i, x_j)}{P(x_i)P(x_j)}} \tag{3.1}$$

and the conditional mutual information is defined as

$$\boldsymbol{Inf(X_i, X_j \mid C) = \sum_{x_i, x_j, c} P(x_i, x_j, c) \ \log \frac{P(x_i, x_j \mid c)}{P(x_i \mid c)P(x_j \mid c)}} \qquad (3.2).$$

Based on the model complexity and evaluation measures as conditional mutual information we built up parameterised scoring functions through β as

$$f_1(m,D) = (\sum_{i} \sum_{x_i \in dom(X_i)} \sum_{Y_j \in pa(X_i)} \sum_{y_j \in dom(Y_j)} P(X_i = x_i, Y_j = y_j) \log_2 \frac{P(X_i = x_i, Y_j = y_j)}{P(X_i = x_i) \cdot P(Y_j = y_j)}) - \beta \cdot \sum_{i} (|X_i| - 1) \cdot \prod_{X \in pa(x_i)} |X| \qquad (3.3).$$

The formula (3.3) takes the following form using maximum likelihood

$$f_2(m,D) = (N \cdot \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \frac{N_{ijk}}{N} \log_2 \frac{N_{ijk}}{N_{ij}}) - \beta \cdot |\theta_m| \qquad (3.4).$$

where *m* represents the network structure, *D* the training data, $r_i\ (1 \le i \le n)$ the cardinality of the random variables X_i, $q_i\ (1 \le i \le n)$ the number of states of all parent nodes and $|\theta_m|$ stands for the number of independent network parameters, expressed also as $|\theta_m| = \sum_{i=1}^{n} (r_i - 1) \cdot q_i$. In special cases regarding the scoring function known quality measures may be deduced like $\beta = 1$ (AIC metric or "akaike" information criterion metric) or $\beta = \frac{1}{2} \ln N$ (minimum description length metric). Then the algorithm described uses Bayesian network learning when node ordering is not necessary. This algorithm takes a data table as input and constructs a network structure as output (see also Figure 1). The problems to solve are how to determine the conditional independence of two nodes and how to orient the edges in the learned graph. We have to use quantitative conditional independence tests based on conditional mutual information. We can avoid the second problem by using an edge orientation based on collided identification. The phases of the algorithm are in detail the following:

Phase 1: We compute mutual information of each pair of nodes as a measure of closeness and next create a draft based on this information.

Phase 2: We add edges to the current graph when the pairs of nodes can not be separated by using conditional independence tests. The result contains all the edges of the underlying dependency probability model given so that the underlying model is monotone direct acyclic graph faithful.

Phase 3: We examine each edge and remove them if the two nodes of the edge are conditionally independent. The result contains the same edges as those in the underlying model. We can also orient the edges of the learned graph. The resulting model is a monotone faithful direct acyclic graph.

4 Experimental Results on XML Structured Qualification Profiles

We can name different reasons to learn employee skills from data. Practical reasons are to identify and discover knowledge potential, increase transparency, derive market trends and then follow them to train employees in the main (independent) profile skills or the combination of soft (e.g. social) and hard (e.g technical) facts in an adaptive skill model. [Peitzsch 2004] has introduced an extended knowledge map where we build up and categorize skills. It is possible to separate employee skills in categories like personal data, professional competence, methodical competence, social competence and additional information based on reference projects. Underlying subcategories can be installed to refine the information concerning the professional competence in personal (e.g. education), technical (e.g. operating system skills or program languages), expert (e.g. CRM, ERP) and line of business (e.g. telecommunications) competence. A qualification profile consists of information structured in aspects like technical, methodical (e.g. presentation techniques) or social skills (e.g. communication, management by motivation, thinking in business processes), completed by project references proving the experience of an employee [Peitzsch 2004]. When defining a structured qualification profile based on the extended knowledge map [Peitzsch 2004] we utilize XML [Garro und Palopoli 2003] as structured mark-up language. We have an instrument that provides a flexible format for expressing skill data for different approaches, whether as a wire format for sending data between client and server, providing a transfer format for sharing skill data between applications or applying as a storage format for databases. We can build up the hierarchical knowledge structure and use the learned network with nodes representing conditional independent skills as input information for skill management systems and skill applications (e.g. e-learning). For instance to manage skill maps for organizations or to train employees in adapting their competence gap between available and necessary skills related to their roles.

As shown in Figure 1, we can graphically represent the different types and occurrences of skills as a tree. Different positions for example imply different skills. These information is not necessarily represented in the skill profile. Using information about the implied skills together with the given skill profile allows us to derive additional information for each employee.

When we analyse necessary skills for a given job profile, we can use the same proceeding. We know the position and we can extract further required skills from the job description. Putting this information together, we are again able to derive additional required skills.

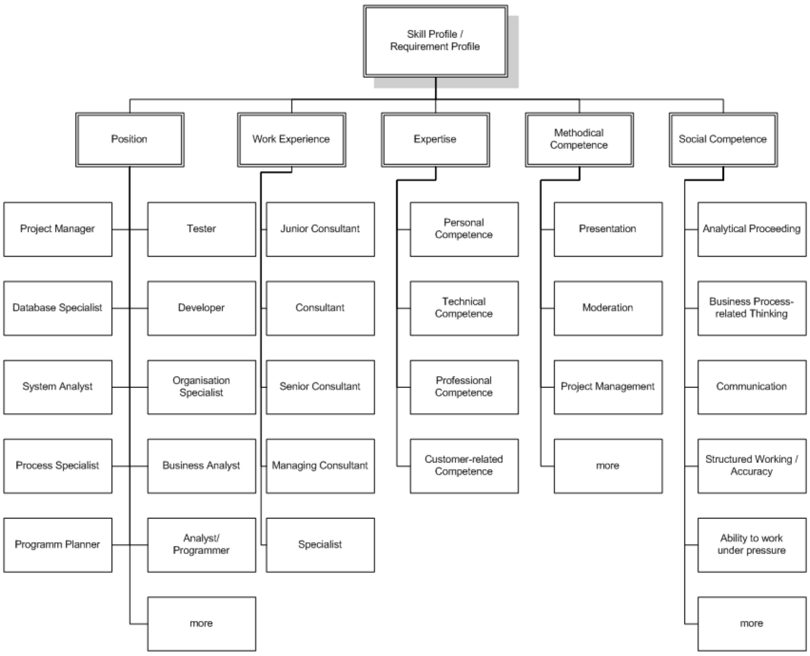

Figure 1: Example of a top-level skill matrix [Peitzsch 2004]

4.1 Scenario and Mathematical Representation

Let us assume a consulting company with access to historical reference project skill data stored in databases regarding *1000* entries. Define a skill-project matrix A as A= ($A_{i,j}$) with $i=1,\ldots,m$ and $j=1,\ldots,n$. $A_{i,j}$ represents the located skills from employee *i* regarding the knowledge map entry *j* with $0 \leq A_{i,j} \leq 1 \varepsilon$ P as visualized in table 1:

	Java	*SQL*	*OOD*	*ERP*	*Fuzzy logic*	*analytical*	…
Employee i	0.2	0.5	0.05	0.8	0.9	0.76	…
Employee i+1	0.7	0.28	0.62	0.32	0.36	0.44	…
Employee i+2	0.63	0.94	0.16	0.54	0.2	0.31	…
…	…	…	…	…	…	…	…
Employee m-1	0.84	0.08	0.16	0.6	0.5	0.6	…
Employee m	0.4	0.3	0.83	0.68	0.19	0.57	…

Table 1: Skill Matrix

The learning algorithm without given node ordering calculated is based on A, the under-lying dependency model where we can compute mutual information using equation (3.1) and (3.2).

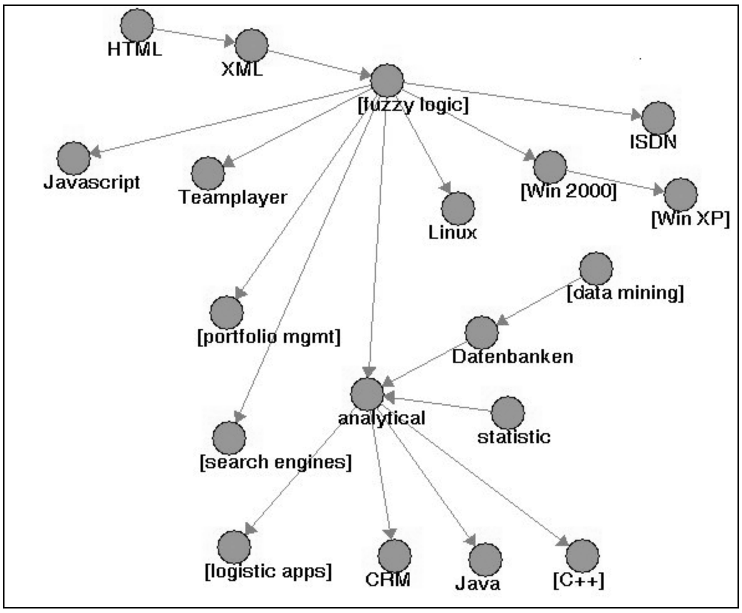

Figure 2 : Learned Network Structure from Skill Data

Figure 2 shows the combinatorial representation of hard and soft skills in a learned Bayesian network structure. Simultaneously the multi-connected network is shown where analytical skills are the fundamental knowledge role for software developers and also in the focus of expert knowledge (CRM) as well as in the line of business. Analytical knowledge itself conditionally depends on database and data mining skills, statistical knowledge and fuzzy logic orientation based on the test data extracted from the underlying matrix A. Conditional dependent employee skills can merge through node aggregation under the fusion aspect first introduced by [Maynard-Reid and Chajewska 2001].

5 Conclusions

In this article we describe the continuous discovery of skill management and set our focus on skill profile data which is used for Bayesian network learning when node ordering is not given. We applied an extended algorithm for Bayesian network learning based on information theoretical aspects to detect information dependencies. In this field we were able to demonstrate the conditional dependence of different employee skills not visible or directly obvious when manual handling *n* skills for exactly 1000 employees as in the test scenario. The output Bayesian structure can be used for future skill management applications like assisting skill processes in Decision Support Systems, managing skill maps of organisations and updating it according to the learning improvements or the measuring of human resource competence gaps. For the future new skill management aspects regarding skill profiles are also interesting. For instance the integration of knowledge from different sources or the availability of partial information concerning the underlying network structure. Another aspect is the development of graphical models by merging information about experts working in distributed teams (fusion). We can distinguish in the last case competitive, complementary and cooperative fusion scenarios. The underlying diverse network structures and conditional probability tables should well integrate in the first case. The complementary scenario is based on uncertain knowledge of each expert involved. The appropriate algorithm consolidates disjunctive network structures as aggregation of case databases. Cooperation exists in the case of dependent conditional probability table parameters and therefore also in the underlying network structure.

References

Borgelt and Kruse 2002] Borgelt, C.; Kruse, R.: *Graphical models. Methods for data analysis and mining.* John Wiley & Sons, Wiley Press, 2002.

[Brewer and Speh 2000] Brewer, P.; Speh, T.: *Using the balanced scorecard to measure supply chain performance.* Journal of Business Logistics, Vol. 21 (2000), No.1, pp. 75-93.

[Heckerman 1995] Heckerman, D.: *A tutorial on learning bayesian networks.* Technical Report MSR-TR-95-06, Microsoft Research, 1995.

[Garro and Palopoli 2003] Garro, A.; Palopoli, L.: *An XML multi-agent system for e-learning and skill management.* In Agent Technologies, Infrastructures, Tools and Applications for E-Services, LCNS volume 2592, p. 283-294, Springer, 2003.

[Holland 2004] Holland, A.: *A bayesian approach to model uncertainty in network balanced scorecards*. In: Matousek, R.; Osmera, P. (Editors): Proceedings of the 10th International Conference on Soft Computing (Mendel 2004), Brno, Czech Republic, June 16-18, 2004, p. 134-138.

[Holland and Peitzsch 2004] Holland, A.; Peitzsch, K.M.: *Evaluating skill management applying knowledge management in decision support systems.* In: Tochtermann, K.; Maurer, H. (Editors): Proceedings of the 4th International Conference on Knowledge Management (I-Know ´04), Graz, Austria, June 30 - July 2, J.UCS, Know Center Graz, 2004.

[Jensen 2001] Jensen, F.V.: *Bayesian networks and decision graphs*. Statistics for Engineering and Information Science, Springer, Heidelberg, 2001.

[Kaplan and Norton 1997] Kaplan, R.; Norton, D.: *The balanced scorecard - Translating strategy into action.* Harvard Business School Press, Vol. 21, 1997.

[Krause 1996] Krause, P.: *Learning probabilistic networks.* Technical Report, Philips Research Laboratories, UK, 1996.

[Maynard-Reid and Chajewska 2001] Maynard-Reid, P.; Chajewska, U.: *Aggregating Learned Probabilistic Beliefs.* Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence, pp. 354-361, Seattle, Washington, U.S.A., 2001.

[Pearl 1988] Pearl, J.: *Probabilistic reasoning in intelligent systems.* Morgan Kaufmann, 1988.

[Peitzsch 2004] Peitzsch, K.M.: *Bewertung von Wissensmanagement-Systemen mit Hilfe von Kennzahlen am Beispiel eines Unternehmensportals für Skill Management.* Diploma thesis (German), University of Dortmund , 09/2004.

[Russel and Norvig 2003] Russel, S., Norvig, P.: *Artificial intelligence - A modern approach.* Prentice Hall, New Jersey, 2003.