INFORMATYKA EKONOMICZNA BUSINESS INFORMATICS 3(25) · 2012

ISSN 1507-3858

Michael Ashcroft
Uppsala University, Sweden
e-mail: mikeashcroft@inatas.com

USING BAYESIAN NETWORKS IN BUSINESS ANALYTICS: OVERVIEW AND SHORT CASE STUDY[1]

Abstract: Bayesian networks are a popular and powerful tool in artificial intelligence. They have many applications in commercial decision support. The point of this paper is to provide an overview of the techniques involved from this perspective. It presents a simplified mathematical overview of important background theory and examines an application of Bayesian networks to talent retention for international firms in China. In developing this case study, we examine the full process of utilizing this technology and the outputs that can be generated.

Keywords: Bayesian networks, decision assistance, business analytics, stochastic modeling.

1. Introduction

We begin by giving a simplified mathematical overview of what Bayesian networks are and the flavors they come in. We then look at how they can be created or learnt from data and the situations that lead to the use of ensemble models. Then we look at how an application of such a technology would proceed, using the human resources example of talent retention for international firms in China, examining the full process rather than technology specific elements. Finally, we look at the outputs that would be generated from such an application. This article is an emended version of the paper given at the AITM'2012 conference.

2. Bayesian networks

Recall from probability theory that two random variables, X and Y, are independent if and only if $P(X,Y) = P(X)P(Y)$. Analogously, X and Y are conditionally independent given a third random variable Z if and only if $P(X,Y \mid Z) = P(X \mid Z)P(Y \mid Z)$, which is equivalent to:

[1] Selected parts of this article were published under non-exclusive copyright in *Proceedings of the Federated Conference on Computer Science and Information Systems FedCSIS 2012* [Ashcroft 2012].

$$P(X \mid Z) = P(X \mid Y, Z). \tag{1}$$

Also recall that the chain rule for random variables says that for *n* random variables, $X_1, X_2, ...X_n$, defined on the same sample space *S*:

$$\begin{aligned} &P(X_1, X_2, \ldots X_n) = P(X_n \mid X_{n-1}, X_{n-2}, \ldots X_1) \\ &P(X_{n-1} \mid X_{n-2}, \ldots X_1) \\ &\ldots P(X_2 \mid X_1) P(X_1). \end{aligned} \tag{2}$$

Imagine we had five random variables: $\{A, B, C, D, E\}$. From the chain rule, we know that:

$$\begin{aligned} &P(A, B, C, D, E) = P(E \mid A, B, C, D) \\ &P(D \mid A, B, C) P(C \mid A, B) \\ &P(B \mid A) P(A). \end{aligned} \tag{3}$$

We can represent these five conditional independencies by means of a directed acyclic graph (DAG) and a set of conditional distributions, where:

– each random variable is mapped to a node of the DAG;
– each node has conditional distribution for its variable associated with this node;
– each node has incoming edges from the nodes associated with the variables on which the node's conditional distribution is conditional.

Such a representation is a Bayesian network. It satisfies the Markov Condition: A direct acyclic graph (DAG), *G*, with nodes N_G, a joint probability distribution, *P*, of random variables D_P, and a bijective mapping $f : D_P \Rightarrow N_G$ satisfies the Markov Condition if and only if for all $v \in D_P$, where $n = f(v)$, v is conditionally independent given *P* of the variables that are mapped to the non-descendants of *n* given the variables that are mapped to the parents *n*.

Table 1. Conditional independencies for the random variables of the DAG in Figure 1

Node	Conditional independencies
A	–
B	C and E, given A
C	B, given A
D	A and E, given B and C
E	A, B and D, given C

Source: own elaboration.

If we know no more than the decomposition given to us by the chain rule in equation 3, the associated Bayesian network's DAG will be complete (since each

variable is conditional on all those prior to it in the decomposition order). However, imagine that we knew that certain conditional independencies exist as specified in Table 1. From the definition of conditional independence, we know that:

– $P(C \mid B, A) = P(C \mid A)$;
– $P(D \mid C, B, A) = P(D \mid C, B)$;
– $P(E \mid D, C, B, A) = P(E \mid C)$;

accordingly:

$$P(A,B,C,D,E) = P(E \mid C)P(D \mid C,B) \\ P(C \mid A)P(B \mid A)P(A). \tag{3}$$

Whenever we simplify the conditional distributions in virtue of a known conditional independence relation, we remove an edge on the DAG of our Bayesian network representation. In this case, the resulting network is given by Figure 1.

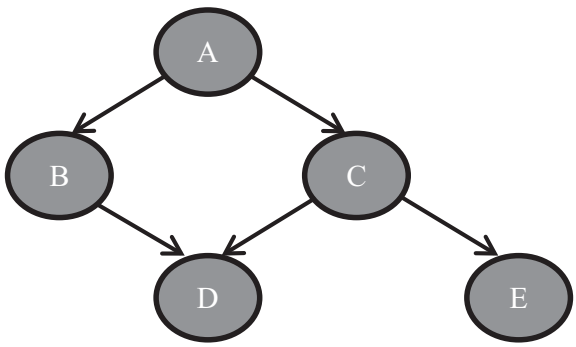

Figure 1. A DAG with five nodes

Source: own elaboration.

Loosely speaking, what we have done is to pull the joint probability distribution *P* apart by its conditional independencies. A Bayesian network is an encoding of these conditional independencies in the DAG topology coupled with the simplified conditional distributions. Note that the conditional independencies are encoded by the *absence* of edges.

The reason why Bayesian networks are useful is that this structure gives us a means of performing tractable calculations locally on the network whilst using all information of the joint distribution. It has been proven that every discrete probability distribution (and many continuous ones) can be represented by a Bayesian Network, and that every Bayesian network represents some probability distribution. Of course, if there are no conditional independencies in the joint probability distribution, representing it with a Bayesian network gains us nothing. But in practice, while

independence relationships between random variables in a system that we aim to model are rare (and assumptions regarding such independence are dangerous), conditional independencies are plentiful.

3. Discrete and continuous Bayesian networks

Bayesian networks come in a number of varieties according to the restrictions, if any, placed on the forms that the conditional probability distributions can take. We will concentrate on discrete Bayesian networks, where continuous variables are discretized during preprocessing. Discrete Bayesian networks:

a) deal with continuous variables by discretization into many arbitrarily sized intervals; various methods can be used for choosing the intervals, including automated clustering methods;

b) are not limited by linear and/or Gaussian noise assumptions;

c) are unrestricted by an *a priori* structure beyond that imposed by discretization, which is both good and bad:

– they follow the data when it leads;
– cases unencountered in the learning data will take the *a priori* distribution, which is generally uniform; there are situations where this is undesirable (e.g. where closely related cases all strongly evince a particular structure); methods exist that provide variance estimates which help indicate when dangerously novel cases are encountered;

d) permit efficient and accurate variance estimation on *a posteriori* probability distributions;

e) permit the use of exact inference algorithms;

f) permit, when combined with decision-theoretic extensions, the use of exact utility maximization algorithms for generating decision policies (including on meta-models);

g) can be used as the automated basis for the production of general Bayesian networks (see below).

Other common forms include Gaussian and hybrid discrete/Gaussian networks. Automated algorithms exist for the automatic learning of, and exact inference on, such networks. These, though, require Gaussian variables to be linear combinations of their parents with Gaussian noise (potentially conditional on the values taken by their discrete parents in hybrid networks).

General Bayesian networks, where any conditional probability distribution in the network can be of any type, are possible. Currently, no automated learning algorithms or exact inference algorithms are known for such networks, but sampling methods do exist for inference. When such networks are desired, it has been suggested that discretized variables be used for the structural learning process [Monti, Cooper 1997]. After the conditional independencies are discovered by this process, bespoke conditional distributions for each variable given its parents can be fitted to the non-discretized data given domain knowledge.

4. Learning

4.1. Learning a network from expert causal knowledge

Importantly, a causal network is a conditional independence encoding of the type described previously. Thus, if we have knowledge of the causal relationships pertaining between the variables we are modeling, then we can immediately produce the DAG structure of the Bayesian network. In such cases, domain experts may also directly specify the conditional distributions. Where this does not occur, we need to learn the conditional distributions from data. Discrete and Gaussian networks have efficient automated algorithms for parameter learning.

4.2. Learning a network from data

Where no expert knowledge is available, we can learn the conditional independencies encoded in the network from data. One strategy arises from the idea of searching the space of possible sets of conditional independencies for such an optimal set. The simplest method is to search the space of network tolopologies; however, since multiple topologies can encode the same set of conditional independencies, a more advanced algorithm is to search equivalence classes of topologies/conditional independence sets [Chickering 2002a]. Algorithms exist which guarantee that as the size of our learning data approaches infinity, the probability of learning the globally optimal model (with a single iteration of the algorithm) approaches 1 [Chickering 2002b].

A second strategy arises from Information Theory, and involves the maximization of the mutual information [Williamson 2000]. This approach is of interest for providing a principled method of according greater importance to particular variables (normally those to be predicted or which we desire decision policies for) in the learning process [Gruber, Ben-Gal 2012].

4.3. Meta-models

When a single network structure fails to dominate alternatives, we can collect multiple high scoring networks. We may, for example, collect all networks that are at least $\frac{1}{x}$ as probable as the best network for some x. These networks can be weighted by their relative probability and inference can be performed over the entire set. Effectively, we now reason using not just our best hypothesis of the system structure, but a set of plausible hypotheses, weighted for their plausibility. This can be a very powerful method, and all the inference algorithms discussed below can be run on such a meta-model.

4.4. Missing data

Structural learning can be performed with missing data items in the learning set. Common algorithms for dealing with this are the Gibbs Sampler and Expectation Maximization.

5. System analysis

5.1. Markov Blanket

The Markov Condition entails other conditional independencies. Because of the Markov Condition, these conditional independencies have a graph theoretic criterion called D-Separation (see [Gruber, Ben-Gal 2012] for a detailed definition). Accordingly, when one set of random variables, Γ, is conditionally independent of another, Δ, given a third, Θ, then we will say that the nodes representing the random variables in Γ are D-Separated from Δ by Θ.

The most important case of D-Separation/Conditional Independence is when a node is D-Separated of the entire graph given its parents, its children and the other parents of its children. Because of this, the parents, children and other parents of a node's children are called the "Markov Blanket" of the node.

This is important. Imagine we had a variable α, whose probability distribution we wish to predict and whose Markov Blanket is the set of nodes Γ. If we know the value of every node in Γ, then we know that there is no more information regarding the value taken by α. This can be generalized to look for the nodes that provide no additional information regarding the set of nodes we are interested in, given the variables we are certain, we will be always able to observe their values.

In this way, if we are confident that we can always establish the values of some of the variables that our network is modeling, we can often see that some of the remaining variables are superfluous, and we need not continue to include them in the network nor collect information on them. Since, in practice, collecting data on random variables can be costly, this can be very helpful.

5.2. Causal analysis

The connection between causality and conditional independence has led to the use of Bayesian networks in causal analysis, often in conjunction with manipulation tests. See [Neopolitan 2004] for details.

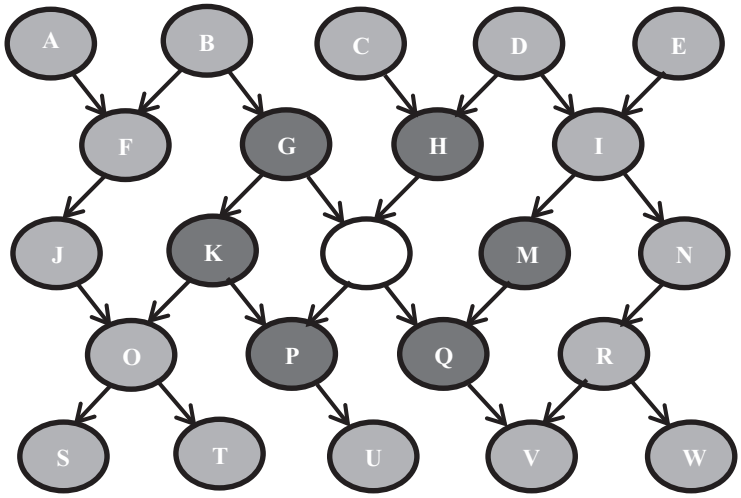

Figure 2. The Markov Blanket of node L

Source: own elaboration.

6. Inference

Inference is the practice of obtaining *a posteriori* probability distributions for variables of interest, given the available evidence. Once a Bayesian network has been created/learnt, we can use the network to calculate the *a posteriori* probability distributions for a subset of variables, Γ, given the observation that a second subset, Δ, has taken particular values. In the discrete and Gaussian cases, we are also able to obtain accurate estimations of the variance of such *a posteriori* distributions, permitting the calculation of "error bars" around the probability estimates.

Efficient exact algorithms exist for both the discrete and Gaussian cases. In the general case, or if a discrete network is sufficiently complex, exact inference algorithms are intractable. In such cases we turn to sampling techniques. The most important (largely for its extension in the application of particle filters in the case of general Bayes filters) is importance sampling.

7. Automated decision-making

Bayesian networks can be extended with utility, decision and information nodes to produce "Influence Diagrams". The utilities are entered by domain experts and specify the value to the user of the system being in particular states. Variables under the user's control are designated decision variables. Additionally, an information order is stipulated, which is based on the partial order in which the decisions must be

made as well as the specification of information variables, which are variables not under the user's control – if they are not currently known, they will be known before the performance of a particular set of decisions. Often this is because they will be known only after the performance of earlier decisions.

So extended, inference is performed on Junction Trees whose topology respects the information order and which have both probability and utility potentials associated with the clusters and intersection sets. Transmission of information through this structure now also includes a utility maximization procedure for each decision variable. The result is the output of decision policies which specify the value to which each (relevant) decision variable ought to be set to maximize the expected utility given the evidence that we will obtain at the time of the decision. Details of the algorithm can be found in [Jensen, Nielsen 2007].

8. An implementation example

Talent retention has become a significant issue for international firms in China. High quality local employees often quickly switch companies, and employees who are trained by their employers often seek better positions once their skill set has been enhanced. I am involved in efforts to assist a number of companies which systematically hire and retain quality employees. I will seek to explain the life cycle of such a project in terms of the following:

1) establishing variables of interest,
2) collecting data,
3) encoding domain knowledge,
4) creating a predictive model,
5) collaborative utility estimation,
6) test model,
7) implementation of access system,
8) post-implementation.

8.1. Establish tasks and variables of interest

The first task is to establish which variables *may* be of interest. The list of variables to be collected in our example might look like the following:

a) employee-related:

– employment history;
– education;
– language ability;
– age;
– sex;
– relationship status;
– demand for applicant's skills;

b) position-related:
– salary;
– professional training at years 1,2,3,+;
– language training at years 1,2,3,+;
– overseas opportunities at years 1,2,3,+;
– position type;
– hours;
– holidays;
– career path opportunities at years 1,2,3,+;

c) company-related:
– Chinese managerial viability (including head office);
– prestige;

d) outcome:
– hire,
– length of employment,
– average company satisfaction with employee over length of employment.

As such this model is to be used only for evaluating new hires, it makes sense to model it statically. If the task were, instead, to evaluate policies towards current staff over the coming six months, it would make sense to use a dynamic model. We must also establish which variables are under the control of the end-users (the participating companies).

8.2. Encode domain knowledge: pre-learning

Expert knowledge can be encoded in the network. This will take the form of specifying relationships between variables (edges in the network) that are required or prohibited, and concrete or defeasible parameters for the conditional probability distributions that relate the variables. In certain circumstances, the network will be entirely created from expert knowledge in this fashion. In our example, the current demand for the applicants' skills is something that is unlikely to have been tracked previously, so it is something that can only be incorporated into the intial model by encoding domain knowledge about its relationship to other variables. As data is collected, this initial specification will evolve.

In practice, where data exists, it is often better to leave the learning process unconstrained and add known relationships only insofar as they were too weak in the learning data for the learnt model to pick up.

When this is not the case, it is usually important to involve an expert in Bayesian networks to perform the encoding of expert knowledge. This is particularly true in the cases in which the entire network is not going to be created from expert knowledge, and where naive encoding using known causal relationships can hamper the efficiency of the learning algorithm and/or result in unnecessarily complex models.

8.3. Collect data

Where the network is not entirely based on expert knowledge, data will be required to be learnt from the model. Even when only using expert knowledge, it is useful to have data to test the validity of the model.

In our case, the data is internal to the company and it is likely that participating companies will need little assistance in collecting it. It will be essential, though, to ensure data security and confidentiality. Participating companies will not want their data to be viewed by other participants or outside entities, nor will individuals wish for their personal records to be reconstructible from the finished product.

8.4. Create network(s)

Where the network is not entirely based on expert knowledge, the model or meta-model will be learnt from the data collected. During this process, a number of methods permit us to test whether we have sufficient data. If we do not, we must obtain more (or switch to a less data-intensive method).

Further, during learning, redundant variables will be found and eliminated. As explained earlier, the topology of a Bayesian network indicates which variables provide no information regarding the state of the variables of our interest, given the variables which we can be certain of, always knowing their values. In our case we might find that the individual's gender is correlated with our variables of interest, but only insofar as it is related to a persons level in their company (perhaps men are more common in managerial positions). Given we know someone's position in the company, their sex contains no additional information regarding whether they will be a satisfactory employee nor whether they are likely to leave the company. As it is the employees of companies who will be using the models, the employees' position in their company can always be established and the gender of the employee is redundant.

We will also determine whether a single network dominates the possible hypotheses regarding the system being modeled, and thus whether we should utilize a meta-model made up of multiple networks as explained earlier.

8.5. Encode domain knowledge: post-learning

In certain circumstances, additional domain knowledge will be encoded in the model post-learning. This may include specifying utility values or transforming the network from a discrete network to a general Bayesian network.

In our case, the utilities involved will be different for different end-users and even for the same end-users in different circumstances. Accordingly, the utility values should be alterable and specifiableby the end-users.

8.6. Test model

The model should be used and evaluated in a pilot program where it will be incorporated into the access system (see below), encounter new real life cases and be used by real end-users. The guidance which the system offers should be useful (not trivial). End-users should find the access system easy to use, and the "reasoning" behind decisions understandable. Difficulties found here can be incorporated into access system revision or user training.

8.7. Implementation of access system

The access system is the software used to enter data into and query the finished model for decision assistance and predictions. It must be deployed and end-users must be trained in its use. For us, this means installing, in all participating companies, the finished software application, which would be a non-technical wrapper that permits all required interactions with the network, as well as providing training courses to end-users.

End-users must be able to enter new data, and the model must adapt to this data. In our case, data regarding new hires will occur automatically as end-users work with the program. But data regarding policies to employees, satisfaction with and of employees, and employee retention will need to be specifically entered. It will be necessary to ensure that the finished application permits such data-entry, and should alert designated individuals if such data-entry does not occur.

Bayesian networks can be set to automatically adapt the parameters of their conditional probability distributions to new data. On the assumption that the underlying system is stable, this generally suffices. If this assumption is questionable, an ongoing structural learning process should continue to model the relationships found in recent data to ensure that no abrupt alterations of the system have occurred and hence that the network remains valid for the domain. Our system is certainly subject to shocks from outside the variables we have included – for example the Chinese economy could enter a prolonged downturn – and so ideally such a process should be included as part of the final application and automatically operate as data becomes available. At the very least, it should be possible for humans to periodically implement such a process.

8.8. Post-implementation

The access system will require ongoing support and maintenance. As new individuals become end-users, they will need training. Finally, to ensure the model stays accurate in changing circumstances, ongoing data acquisition and collation will be required.

In our case, this means providing software support and training for novice end-users, such as new HR employees. The low level data acquisition program should be able to be implemented by participating companies.

9. Outputs

So what might we expect to obtain from such a system? Let us imagine a 26-year old, unmarried male without children. He has an engineering degree from a high-quality Chinese institution and is professionally competent in English. His last job was low-level management, and he specifies an expected salary of RMB7000/month. He has held three jobs in the last four years.

1. Should the prospective employee be hired?

2. If he is hired, what sort of contract and conditions should he be given to maximize the expected value of the hire, measured in terms of satisfaction with his contribution and retention whilst minimizing costs?

For example, he might be offered a number of attractions at a specified future time – perhaps an overseas placement opportunity after two years, or ongoing professional education paid for by the company from year 1 to 3, etc. It may be that he should *not* be offered additional language training, since this is unlikely to increase either his or the company's satisfaction but, if successful, this will greatly increase the risk that he will leave (because he was headhunted!).

A more sophisticated situation is where certain characteristics remain unknown (perhaps at a resume sifting stage). The company may decide not to trust applicants' claims about their language ability or plan to have applicants take additional tests. Decision policies will be given to specify whether an applicant should be hired and, if so, the details of the contract for each of the possible values of the unknown variables will be specified. For example, the applicant in question might be hired only if he performs outstandingly in the test (since there might be a high-risk of him leaving), but in such a case he could be offered a lucrative salary and numerous inducements to stay (since he would be a valuable employee if he could be retained). An example of a decision policy representing the above, and assuming that an appropriate test outcome variable was included, is given in Table 2.

Table 2. Decision policy for an applicant, given test result

Test score	Hire	Salary	Overseas placement	Professional training	Language training
<60	No	–	–	–	–
–70	No	–	–	–	–
–80	No	–	–	–	–
–90	No	–	–	–	–
–100	Yes	14,000	After two years	After three years	No

Source: own elaboration.

The network would also be able to produce *a posteriori* predictions for the variables of interest, given current knowledge and on the assumption that the decision

policies specified are followed. It could also be run with the decision variables treated as chance variables or set to other options to obtain *a posteriori* distributions given current knowledge, or to test alternative decision policies.

If we imagine the "average company satisfaction with employee over length of employment" variable takes values from 1 to 10 (representing some suitable function from yearly reviews) and the "length of employment" variable takes the values 1 to 5 and > 5 (representing the employee leaves the company before that many years), then we might be given the distributions represented in Tables 3 and 4.

Table 3. *A posteriori* probabilities for variable "average company satisfaction with employee over length of employment"

Value	*A priori* probability
1	≈0
2	≈0
3	.01
4	.02
5	.04
6	.08
7	.10
8	.43
9	.22
10	.10

Source: own elaboration.

Table 4. *A posteriori* probabilities for variable "length of employment"

Value	*A priori* probability
1	.08
2	.07
3	.05
4	.01
5	.01
>5	.78

Source: own elaboration.

10. Summary

Bayesian networks are a popular and powerful tool in artificial intelligence. They have many applications in commercial decision support. The point of this paper was to provide an overview of the techniques involved from this perspective. We

proceeded by giving a simplified mathematical overview of what Bayesian networks are and the flavors they come in. We then looked at how they could be created or learnt from data and the situations that led to the use of ensemble models. Then we looked at how an application of such a technology would proceed, using the human resources example of talent retention for international firms in China, examining the full process rather than technology specific elements. Finally, we looked at the outputs that would be generated from such an application.

References

Ashcroft M., Bayesian networks in business analytics, [in:] M. Ganzha, L. Maciaszek, M. Paprzycki (Eds.), *Proceedings of the Federated Conference on Computer Science and Information Systems FedCSIS 2012*, Polskie Towarzystwo Informatyczne, IEEE Computer Society Press, Warsaw, Los Alamitos, CA 2012, pp. 955–961.

Chickering D., Learning equivalence classes of Bayesiannetwork structures, *Journal of Machine Learning Research* 2002a, No. 2, pp. 445–498.

Chickering D., Optimal structure identification with greedy search, *Journal of Machine Learning Research* 2002b, No. 3, pp. 507–554.

Gruber A., Ben-Gal I., Efficient Bayesian network learning for system optimization in reliability engineering, *Quality Technology and Quantitative Management* 2012, Vol. 9, No. 1, pp. 7–114.

Jensen F.V., Nielsen T.D., *Bayesian Networks and Decision Graphs*, 2nd edition, Springer, 2007.

Monti S., Cooper G., *Technical Report: Learning Hybrid Bayesian Networks from Data*, 1997.

Neopolitan R., *Learning Bayesian Networks*, Pearson Prentice Hall, 2004.

Williamson J., Approximating discrete probability distributions with Bayesian networks, [in:] *Proceedings of the International Conference on Artificial Intelligence in Science and Technology*, December 2000.

WYKORZYSTANIE SIECI BAYESOWSKICH W ANALIZACH BIZNESOWYCH: PRZEGLĄD METOD ORAZ KRÓTKIE STUDIUM PRZYPADKU

Streszczenie: Sieci bayesowskie są popularnym i skutecznym narzędziem sztucznej inteligencji. Mają one wiele zastosowań we wspomaganiu decyzji biznesowych. Celem niniejszego artykułu jest dokonanie przeglądu techniki sieci bayesowskich z takiej właśnie perspektywy. W artykule przedstawiono zarys najważniejszych matematycznych podstaw teoretycznych sieci bayesowskich oraz omówiono ich zastosowanie do zatrzymywania utalentowanych pracowników w międzynarodowych firmach w Chinach. W ramach studium przypadku, przeanalizowano cały proces wykorzystywania sieci bayesowskich oraz omówiono dane, które mogą być wygenerowane przy ich użyciu.

Słowa kluczowe: sieci bayesowskie, wspomaganie decyzji, analizy biznesowe, modelowanie stochastyczne.