Development of Simple Effort Estimation Model based on Fuzzy Logic using Bayesian Networks

Abou Bakar Nauman
Sarhad University of Science and Information Technology,
Peshawar, Pakistan

Romana Aziz
COMSATS Institute of Information Technology,
Islamabad, Pakistan

ABSTRACT
Intelligent software estimation models are need of the time. With increased development of Bayesian networks for software project management, one requires an explicit Bayesian Network (BN) to provide effort estimates based on historical data. This paper proposes a simple BN, based on classification approach. However the classes of ranges of size value, are distributed with help of fuzzification to distribute the probability of crisp value The model is simple and smaller, thus can easily be connected to static as well as dynamic Bayesian Networks.

General Terms
Software effort estimation.

Keywords
Bayesian Networks, Fuzzy logic.

1. INTRODUCTION:
Bayesian Network (BN) is a directed Acyclic Graph with nodes representing variables [11], and arcs represent conditional dependence. Let we have a graph with V as a node and π_v is parent node

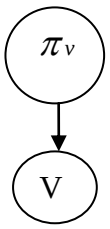

Fig 1: A Bayesian Network

The conditional probability is

$$P(V \mid \pi_v)$$

If there exists a set of variables (V1, V2, V3) in a space U then the joint probability distribution is product of all distributions by chain rule

$$P(U) = P(V_1).P(V_1 \mid V_2).P(V_2 \mid V_3) \quad \text{.....(1.1)}$$

and in case we have *n* variables the probability is

$$P(U) = \prod_{i=1}^{n} P(V_i \mid \pi_{vi}) \quad \text{.....(1.2)}$$

It is notable that probability $P(V_i \mid \pi_{vi})$ will be calculated by Bayesian rule. To estimate the parameter value, frequentist approach is used, e.g. mean of a distribution, however the inference mechanism is based on Bayesian approach [11]. There is a variety of Bayesian network models proposed by researchers in the area of software project management [2-11].

Software development project is a collection of efforts and resources in a defined time period to realize a software product which satisfies the requirements made by a client or agreed upon [12,13]. Project management focuses on suitable application of efforts and resources to achieve the constraints of Cost, Time and Quality. From very first day, the planning for efforts and resources is conducted based on estimates. Estimation is key to the planning and is made not only at the beginning but also at every single milestone. Current research in estimation is focused on issues like development of new models, metrics conversion, uncertainty, missing data, intelligent decision support and models for new life cycles [12-16].

In software development effort estimation, a large set of factors has been identified [13,18] which affects the final effort and the productivity of the organization. This set of factors reaches up-to 20 in some studies [18]. However incase of BN development we need to keep one critical issue i.e. size of model. The size of Bayesian network model increases the computational requirements [19], and keeping in mind the un-rolling (repeated-ness of a model in dynamic Bayesian Net) of the model in case of Dynamic Bayesian Nets, we need to select a minimal set of factors which represent the problem. Thus is this paper we try to develop a very small and simple model.

2. PROPOSED MODEL
This model looks the effort estimation as a simple classification problem. With continuous variable a Bayesian Net can be used to build the classification graph. This is relatively a simple method; however we tend to exploit some features of Bayesian network and its fusion with fuzzy functions. Effort mostly depends on size of software. Thus we also develop a small BBN (named and referred as M1 in this paper), in which effort is dependent on size.

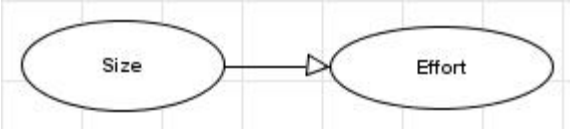

Fig 2. Model M1

2.1 Size Node:

Size is basic driver of effort. The effort required for developing software is directly proportional to the size. As size is itself estimated, so there exists a certain degree of error in estimation accuracy. There exist more than one units and metrics of size in software development; e.g. lines of code, function point and use case point. The most commonly used is function point; however other units can also be converted into function points, with applying some adjustment factors. To populate the NPT, of size node, data from ISBSG [20] repository is used. In the Data set sizing factor has attributes of count approach, function points, adjusted functional point. In this research Adjusted Function Point is used to populate the NPT. In the model M1, the size is classified in 07 groups with help of classification tree using SPSS.

2.2 Effort Node:

The estimation of required effort to develop software is essential for planning resources and right deployment of resources in cost effective manner. The effort is usually calculated as required staff hours to develop software. The unit can be adjusted in man-months or some weekly unit, however main theme behind this is to estimate total time required for a single human resource to develop software. Although there exists a wide range of factors, which determines required effort, but the major factor is size. In the data set, the effort is also provided with respect to different phases of project e.g. planning, design, and implementation; however this division is not provided for majority of projects.

2.3 Node Probability Table

Now issue is how to construct Node Probability Table (NPT) for this BN. If it is assumed that the effort and size have no linear relation, and software of a specific size required an effort which only depends on the size, whereas productivity of two different sizes can be significantly different; then we need to populate the NPT of effort with most likely effort required for a group/class of size. However as the effort is a random variable, the probability needs to be expressed in some probability distribution function e.g. normal distribution. To construct the NPT we run tree classification in SPSS and get 07 groups of size and their corresponding distribution of efforts. To simplify the NPT, we only used the records with size between 1 and 500.

Table 1: Effort of classes of size.

	Size Group/Class	Mean Effort	Std deviation
1	1-32	329	660
2	33-58	752	1380
3	59-82	1109	1849
4	83-108	1493	2465
5	109-177	1937	3309
6	178-285	2615	4343
7	285-500	3232	6200

The NPT of Effort node is now populated with Normal Distribution for each range

$Normal(m_i, \sigma_i) i = 1, 2...6$ Where i is group number.

By implementing the NPT at BN developed by using AgenaRisk Tool [21] is as under. By entering the size as observation, we get the resultant distribution for effort. The BN is implemented using the AgenaRisk Toolkit. This would help to understand the application of the model.

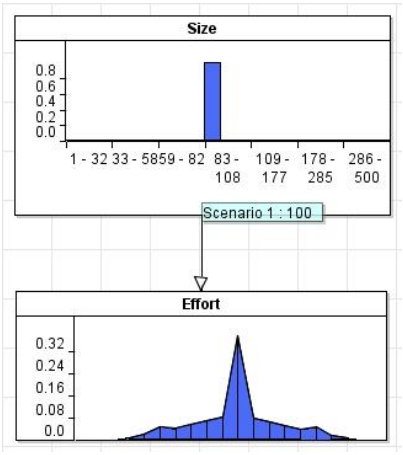

Fig 3: Model M1 implementation

2.4 Limitations of the proposed model:

However this BN has some limitations. As size is expressed in ranges/classes of values, a size e.g. 100 FP would be considered in range number 4, however it is on edge of range number 5 also. As there also exists possibility that size can also be miscalculated, it is thus very difficult to rely on this format of model. Secondly this BN would not be able to consider an input of single value; it would rather select the whole range. To solve this issue the size node is attached with two new nodes; measure and fuzziness.

3. FUZZY LOGIC

Fuzzy logic helps in situations where the uncertainty exists in the form of possibility. Fuzzy logic provides different fuzzy functions which can be used to map the uncertainty.

There had been long discussion about relationship of Fuzzy theory and probabilistic theory. The fuzzy logic expresses the events in terms of possibility of occurrence of event or possibility distribution. First difference between probability distribution and possibility distribution can be narrated as "the probability distribution P is defined on a sample space S and the sum of these probabilities should be equal to 1. Meanwhile, the possibility distribution is defined on an universal set X but there is not limit for the sum."

Fuzzifying function of crisp variable is a function which produces image of crisp domain in a fuzzy set. Fuzzifying function from X to Y is

the mapping of X in fuzzy power set $\widetilde{P}(Y)$.

$$\widetilde{f} : X \rightarrow \widetilde{P}(Y) \quad \text{.....(3.1)}$$

Thus its mapping from domain to set of ranges. This function is expressed as fuzzy relation R as:

$$\forall (x, y) \in X \times Y \quad \text{.....(3.2)}$$

$$\mu_{\widetilde{f}(x)}(y) = \mu_R(x, y) \quad \text{.....(3.3)}$$

The fuzziness of an event can be defined in terms of fuzzy rules,

e.g

R: If x is A, then y is B.

OR

R: A → B

Which can be written in terms of fuzzy implication function, as a fuzzy set with a two-dimensional membership function

$$\mu R(x, y) = f(\mu A(x), \mu B(y)) \quad \text{……(3.4)}$$

The problem in the proposed model is the same as the Crisp Value of Size has to be mapped to a range of possible values. Now the question is what are possible ranges of values?, to help this case there exist many fuzzification functions, e.g Triangular and box and bell shaped. The introduction of these functions is available at different resources and hence is out of scope of this paper. The main idea however is to apply the possibility distribution in such a way that the probability of size value corresponding to one range is distributed among more than one ranges.

Fuzzification

Fuzzy logic helps in situations where the uncertainty exists in the form of possibility. Fuzzy logic provides different fuzzy functions which can be used to map the uncertainty [22]. We use the symmetrical triangular membership function of fuzzy logic which provides a triangular possibility distribution.

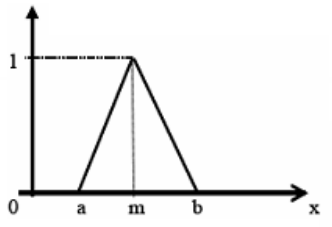

Fig 4: Triangular fuzzy function

$$\text{Fuzziness of TFN (F)} = \frac{b-a}{2m} \quad \text{.....(3.5)}$$

where m is the central value, a is lower limit and b is upper limit. The same triangular function is applied by Mittal in [22]. By taking k=1 in equations 11,12 in [22] we get the equations 3.6 and 3.7.

$$a = (1-F) * Size \quad \text{.....(3.6)}$$

$$b = (1+F) * Size \quad \text{……(3.7)}$$

$$where$$

$$0 < F < 1$$

$$taking$$

$$k = 1$$

Measure and Fuzzy Nodes:

Size is the value of size entered at measure node. The NPT of size node is hence used as triangular distribution where Triangular(Lower = a, Middle = Measure, Upper = b).

Table 2: Nodes of model M1.

Sr #	Node	Type	NPT
1	Measure	Continuous	Uniform(1-1000)
2	Fuzziness	Continuous	Uniform(0-0.4)
3	Size	Intervals	Triangular (Lower = a, Middle = Measure, Upper = b).
4	Effort	Continuous	Partitioned Expressions from table 1

The size now depends on measure and the fuzziness, as fuzziness is increased the probability of size being in neighboring groups increases. The effort node hence doesn't provide distribution of one range, but distribution of neighboring ranges also effect final estimation. This approach rectifies one of the limitations of the model. We can also learn fuzziness of a set of observations and then use the learned fuzziness for future estimates.

When a value of size is entered at measure node the possibility of the value of size, to be in more than one classes, is controlled by the triangular fuzziness. As the value of fuzziness is increased the probability of size being in more than one range is also increased. The effort node, now provide effort based on more than one ranges, which is more practical. For example the value of mean effort is estimated by this model for size value 100 fp is 1800 hrs. This value is only result of a Bayesian calculation rather than any simple arithmetic calculation. This estimation is performed by taking 0.3 as fuzziness value. One can see the degree of belief is divided among three classes of size.

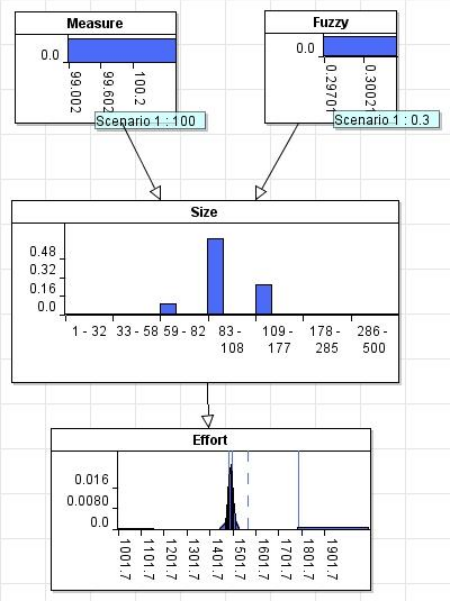

Fig 5: Revised implementation of Model M1

4. CONCLUSION

The model shows two specific achievements. First it is evident that a smaller Bayesian network can be developed to achieve intelligent effort estimates. Secondly the classifications of sizes can be managed with the help of fuzzy logic. The model M1 still has some limitations; first of all, as the effort node is populated with normal distributions of ranges of size, the effort depends largely on distribution of size among different ranges. The resultant effort for starting value and ending value of a range can be equal in lower values of fuzziness. Secondly we are not able to learn the productivity using this BN model.

The model can be further enhanced by introducing regression in place of classification. The model can also be enhanced by introducing more factors which affect the software development effort. It is also observed that the model needs to be tested for real project data. We are under process of its testing and would like to share the results with others soon.

5. REFERENCES

[1]. Jensen F.V.1996, "An Introduction to Bayesian Networks", UCL Press.

[2]. Fenton N.E., Paul Krause, Crossoak Lane and Martin Neil, 2001, "A Probabilistic Model for Software Defect Prediction", citeseer, manuscript available from the authors.

[3]. Pendharkar, P.C.; Subramanian, G.H.; Rodger, J.A. 2005, "A Probabilistic Model for Predicting Software Development Effort", IEEE Transactions on Software Engineering, Volume: 31 Issue: 7 Pages: 615-624.

[4]. Martin N., Fenton N.E., Nielson, Lars, 2000, "Building large-scale Bayesian networks", Journal of Knowledge engineering review, Volume: 15 Issue: 3

[5]. Bibi S., I. Stamelos.2004, "Software Process Modeling with Bayesian Belief Networks". IEEE Software Metrics 2004, On-line proceedings.

[6]. Azalia Shamsaei, 2005, M.Sc. Project report, Advanced Method in computer science at the University of London

[7]. Hearty P, Fenton NE, Marquez D, Neil M. 2009, "Predicting Project Velocity in XP using a Learning Dynamic Bayesian Network Model", IEEE Transactions on Software Engineering, Volume 35 , Issue 1,January.

[8]. Fenton N.E., William Marsh, Martin Neil, Patrick Cates, Simon Forey, and Manesh Tailor, 2004, "Making Resource Decisions for Software Projects", Proceedings of the 26th International Conference on Software Engineering (ICSE'04).

[9]. Fenton N.E., Neil, M.; Marsh, W.; Hearty, P.; Marquez, D.; Krause, P.; Mishra, R. 2007, "Predicting software defects in varying development lifecycles using Bayesian Nets", Information and Software Technology , Volume: 49 Issue: 1.

[10]. Khodakarami, V., Fenton, N., & Neil, M. 2009, "Project scheduling: Improved approach incorporating uncertainty using Bayesian networks", Project Management Journal.

[11]. Emilia Mendes (2007), "Predicting Web Development Effort Using a Bayesian Network" Proceedings of (EASE'07) 11th International Conference on Evaluation and Assessment in Software Engineering 2-3 April, pp. 83-93

[12]. C. Larman, "Agile and Iterative Development: A Manager's Guide", Addison Wesley, 2003

[13]. Bohem B. et al. 1995, "Cost models for future life cycle processes: COCOMO2.0", Annals of Software Engineering, Vol 1.

[14]. Jingzhou Li, Guenther Ruhe "Decision Support Analysis for Software Effort Estimation by Analogy", Third International Workshop on Predictor Models in Software Engineering (PROMISE'07)

[15]. Walker Royce, " Software Project Management, A Unified Frame work" Pearson Education, 2000

[16]. Mohammad Azzeh et al. "Software Effort Estimation Based on Weighted Fuzzy Grey Relational Analysis", ACM 2009

[17]. Andrew R. Gray, Stephen G. MacDonell, A comparison of techniques for developing predictive models of software metrics, Information and Software Technology 39 (1997) 425-437

[18]. Boehm B., C. Abts and S. Chulani 2000, "Software development cost estimation approaches—A survey", Annals of Software Engineering 10, pp. 177–205.

[19]. Kevin Murphy, "A Brief Introduction to Graphical Models and Bayesian Networks", 1998.

[20]. ISBSG data release 10, 2007, http://www.isbsg.org, accessed on 18-feb-2009.

[21]. Agena, Bayesian network and simulation software, http://www.agenarisk.com/ ,accessed on 18-feb-2009.

[22]. Anish Mittal, K. P., Harish Mittal (2010). "Software Cost Estimation Using Fuzzy Logic." ACM Software Cost Estimation Using Fuzzy Logic 35(1).