Asian Journal of Information Technology 15 (3): 647-650, 2016
ISSN: 1682-3915

A Panoramic Bayesian Analogy Based Method for Software Project Cost Estimation

Hussain Syed and Y. Prasanth
Department of Computer Science and Engineering, KL University, Andhra Pradesh, India

Abstract: Cost estimation is an imperative issue in venture administration. Taken cost estimation for programming undertaking is of specific significance as a lot of the product ventures experience the ill effects of genuine spending plan overwhelms. Going for exact cost estimation, a few systems have been proposed in the previous decades. Analogy based estimation which impersonates the procedure of undertaking supervisors settling on choices and acquires the formal articulations of case based thinking is a standout amongst the most every now and again contemplated techniques. Be that as it may, analogy based estimation is frequently condemned for its moderately poor prescient exactness, substantial computational cost and prejudice to unverifiable inputs. To reduce these disadvantages, this study is committed to enhance the Bayesian analogy based technique from three viewpoints: exactness, productivity and heartiness.

Key words: Analogy based estimation, Bayesian analogy based technique, standout, prjudice, heartiness

INTRODUCTION

There are several recorded programming cost estimation strategies, devices and models. The principal ones that we review are those that really assessed the product cost as a rate of the equipment cost. The yield of a large portion of them is an estimation of staff exertion from a wide assortment of inputs. Numerous utilization an investigative equation that takes as information parameters depictions of task qualities, framework size, many-sided quality and advancement system which are regularly called "cost drivers". You should make sure you comprehend what periods of the life cycle (Tan *et al*., 2014) are incorporated, since numerous models might cover just programming configuration, improvement and testing. The greater part of the procedures and models have critical confinements.

We order estimation strategies as: master conclusion. utilizing benchmark information; analogy, intermediary focuses, custom models, algorithmic models (Silva *et al*., 2015).

One reason for fizzled evaluations is a lacking foundation of estimators in the territory of programming estimation. Self-assertive choice and the visually impaired utilization of estimation techniques and instruments frequently prompt disillusioning results while the fundamental reasons stay vague. In examinations with corporate administration, it is not phenomenal to hear the expression "think about a number and duplicate by three". Deliberate choices in regards to the specific estimation technique and its proficient use require knowledge (Azzeh *et al*., 2014) into the standards of exertion estimation programming exertion estimation is regularly mistaken for task arranging and offering. These procedures, albeit identified with each other through covering exercises, contrast as for their targets and the yields they should convey.

With respect to any undertaking movement, there are various people that have intrigues in exertion estimation. We recognize a few parts connected with exertion estimation, contingent upon their stakes and correct contribution.

Estimation process owner: This part is in charge of presenting and keeping up estimation forms, techniques, models and information inside of an association. This individual has inside and out learning of the estimation strategy and oversees estimation activities and exercises inside of the association. The estimation process proprietor is normally a devoted full-time position inside of an association.

Estimator: This part utilizes estimation techniques and models existing inside of an association for evaluating specific advancement ventures. A venture director, who appraises and oversees ventures, regularly assumes the part of the estimator (Boehm, 2000).

Domain master: This part gives data to building an estimation model when estimation information are lost or lacking. Space specialists ought to be educated in undertaking exertion conditions inside of the setting for which the exertion estimation technique is connected.

Corresponding Author: Hussain Syed, Department of Computer Science and Engineering, KL University, Andhra Pradesh, India

Area specialists don't need to be proficient in exertion estimation. They ought to have a fundamental comprehension of estimation and know which considers and to what degree they impact exertion in the task, inside of this connection. On account of estimation taking into account human judgment, be that as it may, area specialists assume the part of estimators and in this manner ought to be moreover proficient in venture exertion estimation decision creator: this part speaks to other undertaking partners who are not straightforwardly included in the estimation handle but rather who have the ability to influence the task, including the execution of estimation. For instance, a venture proprietor (likewise alluded to as an undertaking patron) might put weight on estimators as for task assets and in this manner inclination gauges. A case of this circumstance from practice may be a Scrum arranging meeting amid which an intense item proprietor inclinations evaluations of the Scrum group. In the amazing case, it is sufficient that the item proprietor raises an eyebrow or shakes his/her head and the Scrum group then predisposition their appraisals keeping in mind the end goal to satisfy the item proprietor

MATERIALS AND METHODS

For the analogy method: pick a comparative framework, ordinarily comparative in structure, lients and environment, framework development and timetable and usage and operational environment. Dissect the distinctions express the distinctions as modifiers break out expenses for the full venture (e.g., prerequisites, venture administration, equipment) (Briand *et al*., 1992). Change base framework expenses to touch base at another assessment. The legitimacy of the evaluation is exceedingly reliant on the level of closeness of the frameworks (Fig. 1).

Data-driven effort estimation refers to methods that predict effort based solely on the quantitative analysis of historical project data. In the course of estimation, relationships between the actual project effort and project characteristics are explored based on the measurement data collected from already completed projects. The relationships found are then projected onto a new project in order to predict the expected effort.

Bayesian Belief Networks (BBN) is a hybrid estimation method (Chulani *et al*., 1999; Fenton and Neil, 1999). It represents a model-based, parametric1 estimation method that implements a define-your-own model approach. Actually, for the purpose of software effort estimation, the method adapts the concept of bayesian networks which has been evolving for many years in probability theory. The approach was recently adapted to

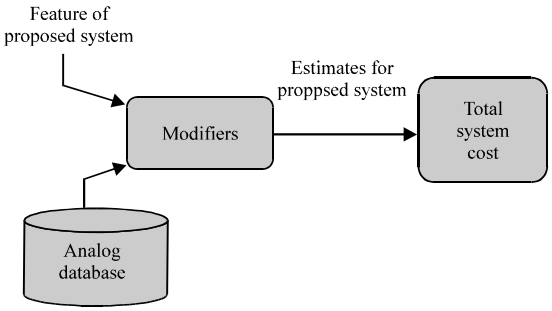

Fig. 1: Analogy method

software estimation due to its ability to combine knowledge based on quantitative measurement data and human judgment into intuitive graphical models with a sound theoretical basis. These applications include estimation of software development productivity (Hall, 1995; Jensen, 1996). This ability is particularly attractive in the software engineering context where measurement data are scarce and much knowledge is hidden in the heads of human experts.

Bayesian Belief Network (BBN) or simply belief network is a Directed Acyclic Graph (DAG) in which nodes represent random variables (discrete or continuous) and edges express probabilistic dependency among the linked variables. Note that although, BBNs are typically used to represent causal relationships, this does not need to always be the case; they can represent simple correlation. Each direct relationship is described by an edge starting from the influencing variable (parent node) and terminating on the influenced variable (child node) where "influence" is understood as statistical dependency. The root node "Development Productivity" has two child nodes: "Team Skills" and "Requirements volatility". These relationships represent the belief (typically based on empirical observations) that development productivity is influenced by the skills of the development team and the volatility of software requirements. In particular, the more skilled the team, the more productive it is and the more volatile the requirements are the less productive is development (e.g., because effort is spent on rework resulting from changing requirements). Finally, the "Requirements volatility" node has one child node "Customer Involvement". This association represents the belief that the more a customer is involved insoftware development (especially requirements specification), the more volatile are his or her requirements. BBNs represent knowledge about an uncertain domain and are mainly used in situations that require statistical inference that is situations where beliefs regarding the likelihood of the events that have not yet been observed need to be

updated in the light of other events that have actually been observed. In the context of BBNs, the events that have not yet been observed are called Hypotheses (H) whereas the observed events are called Evidence (E). BBNs use probability calculus and Bayes. Theorem for propagating the evidence throughout the belief network, thereby updating the strength of beliefs regarding the likelihood of the events that have not been observed yet. Strengths and weaknesses BBNs provide useful mechanisms for integrating quantitative techniques and expert judgment for the purpose of making predictions. In particular, BBNs are both mathematically rigorous and intuitively understandable. BBNs adopt probabilistic mechanisms for representing and operating on uncertain information. An inference mechanism in BBNs allows for testing various trade-off scenarios by introducing various evidence values. Yet, their practical application is still limited by sophisticated theoretical background, restrictions on the use of continuous and mixed (continuous and discrete) variables and exponential growth in complexity of acquiring conditional probability data and inference computations as the size of a causal model grows.

RESUTLS AND DISCUSSION

Bayesian analysis is a mode of inductive reasoning that has been used in many scientific disciplines. A distinctive feature of the Bayesian approach is that it permits the investigator to use both sample (data) and prior (expert-judgement) information in a logically consistent manner in making inferences. This is done by using bayes theorem to produce a "post-data or posterior distribution for the model parameters. Using Bayes theorem, prior (or initial) values are transformed to post-data views. This transformation can be viewed as a learning process. The posterior distribution is determined by the variances of the prior and sample information. If the variance of the prior information is smaller than the variance of the sampling information, then a higher weight is assigned to the prior information. On the other hand, if the variance of the sample information is smaller than the variance of the prior information, then a higher weight is assigned to the sample information causing the posterior estimate to be closer to the sample information (Fig. 2).

The change association alludes to the associations executing the product framework alterations, i.e., the gatherings included in the venture, for example, the client, the merchants, advisors and so forth. The change association variables in the system concerns the number of designers and engineers included in the venture.

Designers are the general population in the change venture who plan and adjust the engineering of the endeavor frameworks. Designers, then again are the ones composing and altering thesource code of the distinctive parts in the undertaking design. The planners and

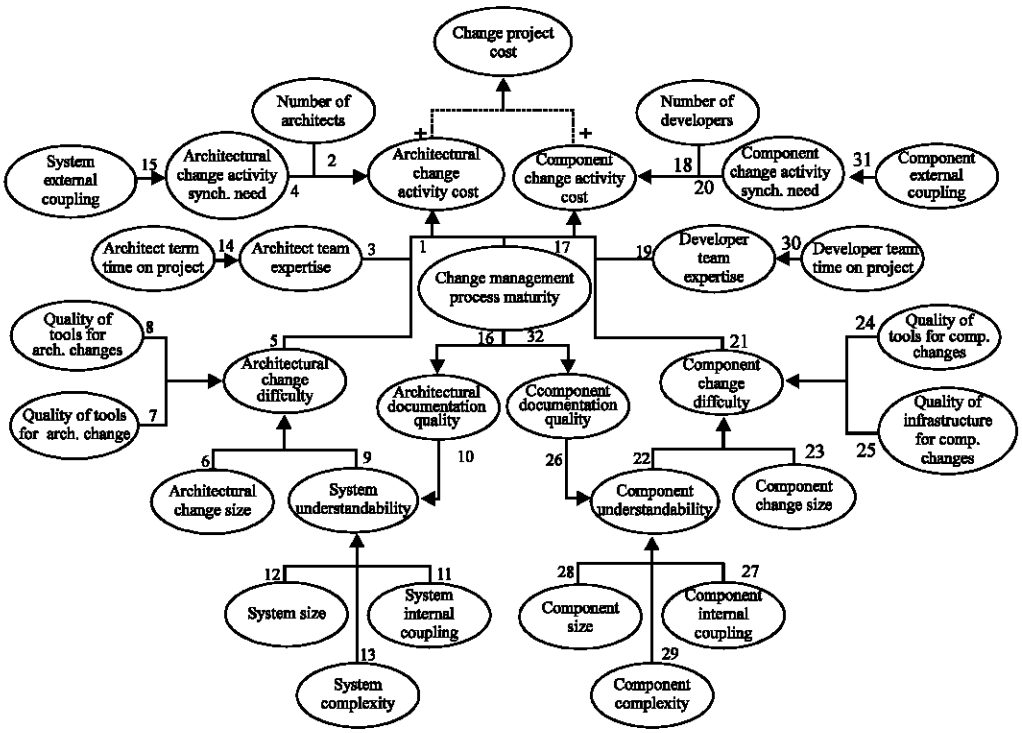

Fig. 2: The Bayesian network for software change project cost analysis

designers both have the variables ability and time on undertaking identified with them. Skill is measured regarding change venture experience, source code or plan dialect experience and framework experience. Time on task alludes to the measure of time a man spend in the task contrasted with other parallel work.

Framework documentation is restricted for planners and engineers to comprehend the frameworks, the parts, what's more, the earth. In this way, the design documentation and the segment documentation must be of high caliber, e.g., the documentation must be accessible, complete, precise, predictable and discernable. Commonplace archives could be framework justifications, prerequisites and configuration particulars, test arranges and information word references (Van Lamsweerde and Willemet, 1998; Lowry, 1992).

The framework environment and the segment environment contain devices. The accessible instruments have the aim of making the change work less demanding. Despite, the fact that this require the instruments to be of high caliber e.g., institutionalized, understood and simple to utilize. The framework and part environment additionally incorporates framework, for example, stages. Framework quality is measured as far as institutionalization level and accessibility. Change undertakings are separated into structural change exercises and part change exercises. Building change exercises are the exercises concerning alterations on a design level, i.e., including a few frameworks or parts. Part change exercises concern alterations to a solitary part. Both sorts of progress exercises have the variables cost, measured as number of worker hours and synchronization need. The more frameworks, parts individuals included and the higher the coupling between them, the higher the need of synchronization.

The change exercises perform specialized changes to the engineering and the segments. The specialized changes have two variables, change trouble and change size. Change size for engineering is measured in number of segments included and change size for segments is measured in number of lines of code included.

Specialized changes are executed in either a framework or a segment. There are five variables related to these, understandability, inside coupling, size, many-sided quality, and outer coupling. Segment outside coupling concern the relations to different segments, which is the same as the framework inside coupling. Framework outside coupling concerns the relations to different frameworks. Part size is measured in number of lines of code and framework size is a total of all the parts. Part inner coupling is the reliance inside of a part.

CONCLUSION

BBNs speak to information around an unverifiable area and are for the most part utilized as a part of circumstances that require measurable induction, that is, circumstances where convictions with respect to the probability of the occasions that have not yet been watched should be overhauled in the light of different occasions that have really been watched. In the connection of BBNs, the occasions that have not yet been watched are called speculations (H), though the watched occasions are called proof (E). BBNs use likelihood math and Bayes' theorem for spreading the confirmation all through the conviction system, in this manner redesigning the quality of convictions with respect to the probability of the occasions that have not been watched yet.

REFERENCES

Azzeh, M., Y. Elsheikh and M. Alseid, 2014. An optimized analogy-based project effort estimation. Int. J. Adv. Comput. Sci. Applic., 5: 6-11.

Boehm, B., 2000. Requirements that handle IKIWISI, COTS and rapid change. Computer, 33: 99-102.

Briand, L.C., V.R. Basili and W.M. Thomas, 1992. A pattern recognition approach for software engineering data analysis. IEEE Trans. Software Eng., 18: 931-942.

Chulani, S., B. Boehm and B. Steece, 1999. Bayesian analysis of empirical software engineering cost models. IEEE Trans. Software Eng., 25: 573-583.

Fenton, N. and M. Neil, 1999. A critique of software defect prediction models. IEEE Trans. Software Eng., 25: 675-689.

Hall, R.J., 1995. Systematic incremental validation of reactive systems via sound scenario generalization. Automated Software Eng., 2: 131-166.

Jensen, F.V., 1996. An Introduction To Bayesian Networks. Vol. 210, Taylor and Francis, London, ISBN: 9781857283327, Pages: 188.

Lowry, M.R., 1992. Software engineering in the twenty-first century. AI Magazine, 13: 71-87.

Silva, P., A.M. Moreno and L. Peters, 2015. Software project management: Learning from our mistakes [Voice of evidence]. IEEE Software, 32: 40-43.

Tan, C.H., K.S. Yap, H. Ishibuchi, Y. Nojima and H.J. Yap, 2014. Application of fuzzy inference rules to early semi-automatic estimation of activity duration in software project management. IEEE Trans. Hum. Mach. Syst., 44: 678-688.

Van Lamsweerde, A. and L. Willemet, 1998. Inferring declarative requirements specifications from operational scenarios. IEEE Trans. Software Eng., 24: 1089-1114.