ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

A Novel Way of Cost Estimation in Software Project Development Based on Clustering Techniques

Swati Waghmode[1], Dr.Kishor Kolhe[2]

PG Student [IT], Dept. of Information Technology, MIT College of Engineering,PuneIndia[1]

Associate Professor, Dept. of Information Technology, MIT College of Engineering, Pune, India[2]

ABSTRACT: Software cost estimation is the very basic step to start any project. It provides us the overview of effort, resources and time required for a project in terms of cost.Project successes mostly depend on Cost estimation as it gives the initial idea of the path, challenges and risk involved in the project. It is very difficult to match approximately the estimated with the actual cost. Efficient estimate can help us make more reliable decisions in planning the project risk. Cost estimation has become very important as it may lead to adverse results if the predicted estimates are wrong. In this model, we have proposed a value shrinking technique based on multilayer feed-forward neural network, and auto-associative clustering. The Kernel component analysis is log-linear regression functions calibrated with large data set with ordinary least squares. We have showed that Kernel component analysis can improve the estimation model accuracy by shrinking the input variables into an equivalent pattern and removing irrelevant variable, based on the COCOMOII data set. We have showed that the models obtained by applying Kernel component analysis are more persistent, acceptable and dependable.

KEYWORDS: Software cost estimation, software effort estimation, ANN, KPCA, and COCOMOII.

I.INTRODUCTION

The procedure of calculating the plan, hard work, effort, size of the software solution, and overall cost associated with developing the software program application is referred to as software cost estimation.A cost estimation completed at the start of project can help decide which functions can be involved inside learning resource difficulties from the project. The risk of project is increases when the most important functions are involved at the end of the project. Thus, cost estimation may have a large effect on the life cycle and timetable for just a project. Which is among the most complicated work within managing as well as preserving software program, during the improvement method cost as well as time period evaluation performs an essential part inside software cost evaluation method.Cost evaluation to get a usual software project will start through first scoping as well as planning cycle of the project. This kind of earlier estimation connected with entire cost and implementation is extremely important because this specific estimation (both cost as well as time) is needed because feedback intended for primary uncooked affirmation as well as checking of the tasks overall improvement as well as health verify. After completion of all the work included, these kinds of estimations utilized with regard to project productivity review. Software cost estimation methods are divided in two categories:

A.ALGORITHMIC METHOD

Algorithmic models also called parametric models. These methods start using a formula to calculate the cost estimation. In this method costs are analyzed using mathematical formulae inputs with metrics to produce an estimated output.This method uses the mathematical equations to accomplish the application estimation. The exact equations use historical info or theory.

B.NON-ALGORITHMIC METHOD

In this method estimation process is done according to the analysis of previous datasets.Non-algorithmic methods usually do not use any formula to calculate the software cost estimation. This method makes comparison between previous dataset and existence dataset. After considering the categories of software cost estimation,we've proposed a novel concept of "Kernel Principle Component Analysis" (KPCA) which improves reliability and accuracy without applying the exhaustive procedures. This paper is organized in sections as listed below. Section I& II provides introduction and Literature review. Section III provides a brief overview of proposed method that is based on algorithmic and non-algorithmic methods. A brief introduction to the concept of KPCA that helps in increasing accuracy of software cost estimation without application of any exhaustive procedures is discussed in the same section. Section IV provides a brief overview of significance of proposed method. Section V includes analytical results of COCOMOII model. Conclusion and future scope is explained in sections VI & VII

II.LITEARTURE SURVEY

Software cost along with effort appraisal plays a crucial role inside software task management. Research of all the available literature shows there are many computer software cost appraisal methods readily available including gentle computing methods. Following section presents the review of the work done:"In this paper two methods with different technique used. In direct method size is measured in lines of code (LOC). In indirect method, size is represented as Function Points (FP).This technique improving the accuracy of software cost estimation model". Dr.N.Balaji et.al. [1]"This paper providesoverview of existing software cost estimation models and techniques. Cost estimation models are basically of two types: algorithmic and non-algorithmic. This paper presents the pros and cons of algorithmic and non-algorithmic method". SwetaKumari et al.[2]"In this paper, the author explores the use of Perceptron learning rule to implement COCOMO II for effort estimation, so that the estimated effort is more close to the actual effort. In this paper technique used COCOMOII, Neural network, Perceptron learning algo". RidhikaSharma et al. [3]"This paper uses new fuzzy logic method for improving the accuracy of software cost estimation model. The result of this model are compared with COCOMOII model.In this paper Technique used-Fuzzy Logic Model, COCOMOII Model".Zia Uddin, et al. [4]."In this paper the proposed neural networks model showed better software effort estimates as compared to traditional COCOMO." – Anupamkaushik, et al. [5]."In this paper several existing methods for software cost estimation are represented and all existing methods for software cost estimation and comparing their features. It is useful for selecting the Special method for each projectEstimation technique used: SLOC,Function point size estimates COCOMO,Analogy,Neural network."-Vahid, et al.[6]"This paper uses a new hybrid toolbox which is based on soft technique. Toolbox presenting a vital role it provide an efficient, flexible and user-friendly way of performing the effort estimation task"Ch.VMKet al. [7]."This paper introduces novel model using fuzzy logic to estimate the effort required in software development. This model improving the accuracy of software cost estimation model which present the better accuracy as compared to other methods."-J.N.V.R. Swarupkumar,et al. [8]It provides an overview of economic analysis techniques and their applicability to software engineering and management. It reviews the field of software cost estimation, including the major estimation techniques available, the state of the art in algorithmic cost models, and the outstanding research issues in software cost estimation."-Boehm B. W [15].

III.PROPOSED METHOD

Technology has become significant regions of organization improvement. A lot of the businesses rely upon technology such as computer systems & software. But also in another side organization likewise ponders the particular expenditure to be built on the software. A number of the businesses invest in brand new software whilst some of them acquire brand new software. In all of the these types of circumstances expenditure of your energy & money takes on a significant role so that it becomes necessary to help appraisal cost regarding software to be employed and also time period used for improvement.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

The recommended methodology is based on Algorithmic & Non-algorithmic methods for instance Function position size estimation, COCOMO &Artificial network. The combination of all these kind of methods assists in estimating cost in the software.
Proposed system follows specific steps in which the flow is maintained. The details of each stage are mentioned below.

A. INPUT ACTUAL DATASET COLLECTED AS PER PROJECT SPECIFICATION

Size: It is essential part for software cost estimation. For predicting the size of the project function point estimation is better than SLOC. This method includes number of inputs, number of outputs, number of inquiries, number of logical files, number of interfaces by using this parameters we can find the projects complexity like simple, average, complex [6]

Cost factors: Boehm introduced a set of 17 cost drivers in the Intermediate COCOMO that adds accuracy to the Basic COCOMO.
The cost drivers are grouped in four categories:-
Product factors: It is formulated with factors of product such as product complexity, reliability etc.
Computer factors: It depends on Execution time constraint, main storage constraint etc.
Personnel factors: It depends upon the ability of the programmer/analyst in development by using experience & knowledge.
Project factors: It depends upon project features such as milestone, deliverables etc.
Above cost factors depends upon the rating values corresponding to real number known as effort multipliers (EM). Rating values having six levels: Very low, Low, Nominal, High, Very high, Extra high.[6]

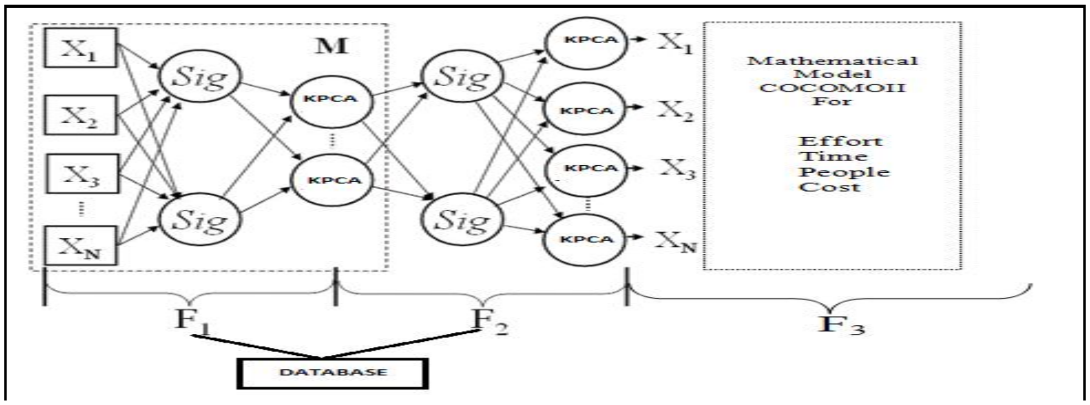


Figure 1: Proposed System Architecture Design for Software Cost Estimation

Scale factors:-COCOMO II depends on the five scale factors such as Precedentedness (PREC), Developing Flexibility (FLEX),Architecture / Risk Resolution (RESL), Team Cohesion (TEAM) and Process Maturity (PMAT) [6].

CLASSIFICATION USING KPCA

WhyKPCA is better than PCA

1. PCA does not supporting mean for multilayer neural network.
2. Large dataset like 0.000009 assume as non-linear value PCA does not take a nonlinear space value. KPCA allows us to identify the kernel principal directions in which the data varies with large variance

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

3. PCA support explicit mapping that's why PCA work with single layer neural network. In practice, a huge data set leads to a huge K, and storing K may become a problem. One way to deal with this is to perform clustering on your huge dataset, and populate the kernel with the means of those clusters.
4. KPCA support implicit mapping that's why KPCA work with multilayer neural network.

Steps for calculating number of kernel principal component are given below [11].

To understand the utility of KPCA, particularly for clustering, observe that, while *N* points cannot in general be linearly separated in dimensions, they can almost always be linearly separated in dimensions. That is, given *N* points, if we map them to an *N*-dimensional space with $\Phi(\mathbf{x}_i)$where $\Phi : \mathbb{R}^d \to \mathbb{R}^N$.

in kernel PCA, a non-trivial, arbitrary Φfunction is 'chosen' that is never computed explicitly, allowing the possibility to use very high dimensional Φ's if we never have to actually evaluate the data in that space. Since we generally try to avoid working in the Φ-space, which we will call the 'feature space', we can create the N-by-N kernel

$$K = k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}), \Phi(\mathbf{y})) = \Phi(\mathbf{x})^T \Phi(\mathbf{y})$$

We note that $\Phi(\mathbf{x_i})^T \Phi(\mathbf{x})$denotes dot product, which is simply the elements of the kernel K. It seems all that's left is to calculate and normalize the $\mathbf{a_i}^k$, which can be done by solving the eigenvector equation

$$N\lambda\mathbf{a} = K\mathbf{a}$$

Where N is the number of data points in the set $\lambda$and $\mathbf{a}$ is the eigenvalues and eigenvectors of K. Then to normalize the eigenvectors $\mathbf{a}^k$'s, we require that

$$1 = (\mathbf{a}^k)^T \mathbf{a}^k$$

Care must be taken regarding the fact that, whether or not $x$has zero-mean in its original space, it is not guaranteed to be centered in the feature space (which we never compute explicitly). Since centered data is required to perform an effective kernel principal component analysis, we centralize K to become K'

$$K' = K - \mathbf{1_N} K - K \mathbf{1_N} + \mathbf{1_N} K \mathbf{1_N}$$

Where $\mathbf{1_N}$denotes an N-by-N matrix for which each element takes value $1/N$. We use $K'$to perform the KPCA algorithmMore important data in the project estimation is collected in the sample data set which consists of many important factors for effort estimating such as software size, effort, productivity, development progress of project, project attributes, platform attributes, scale attribute, architecture etc. Then quantify the data before processing the data. The sample data set should be preprocessed, because some data might get missed, in the mean time for calculating the eigenvalue we should compute correlation coefficient matrix by using KPCA and input values and finally we should determine the number of kernel principal components based on size, cost factors and scale factors.

B. ARTIFICIAL NEURAL NETWORK

Artificial Neural Network is used in cost estimation due to its ability to learn from existingdataset.A basic neural network includes a number of inputs that are applied by some weights which are combined together to give an output. The steps used for cost estimation by using Delta feed-forward multi-layer ANN are summarized as follows [1]:

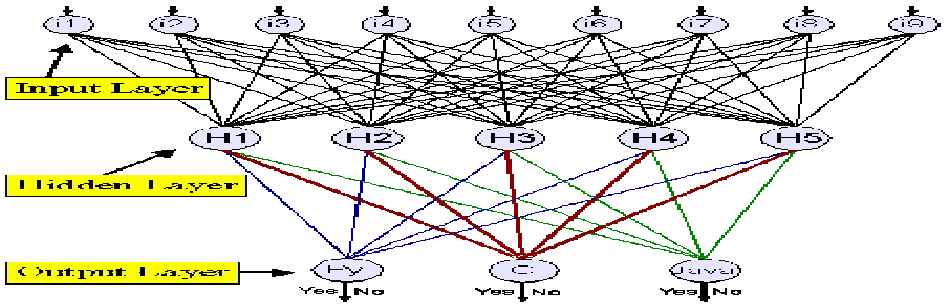


Figure 2: Architecture of Delta feed-forward neural network

Step1: The input layer receives input signal i.e. principal components and sends it to the hidden layer.
Step 2: It includes data training.
Training algorithm includes following steps:

a. Choose the training sample i.e. kernelprincipal components and train it with sample dataset in matrix form.
b. Determine error in hidden layer but there is less chances of error because KPCA provides exact eigenvalue.
c. If error occurs then update the neural network weights.
d. Repeat until the neural networks error is sufficiently small after an epoch is complete

Step3: Output layer sends size, effort multiplier and scale factor rating values to COCOMOII by using activation function as shown in figure 2.

C.COCOMO II

COCOMOII is the latest version of COCOMO.COCOMO dataset includes 63 historical projects and COCOMOII dataset includes 161 historical projects. The estimated effort in person-months (PM) for the COCOMOII is given as:

$Effort = A\times [SIZE]^{E} \times {}_{i=1}\prod^{17} EM_i$ *[1]*

$Where\ E=B+0.01\times{}_{j=1}\sum^{5} SF_j \qquad A=2.94, B=0.91$

$Time=C\times (Effort)^{F}\ Where\ F=D+0.2\times 0.01\times{}_{j=1}\sum^{5} SF_j\ C=3.67, D=0.28$ *[2]*

People= Effort/Time [3]

COCOMOII uses function point size estimate for calculating the size of the software and composes of 17 Effort multipliers and 5 scale factors (SF) [4]

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

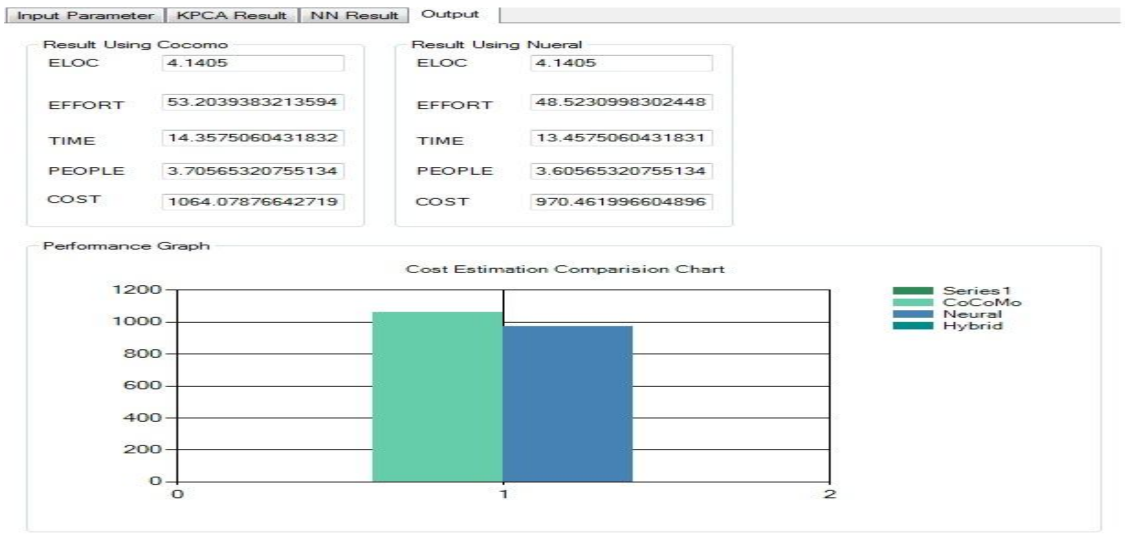
(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

*Cost = Effort * Average salary per unit time + Other expenses [4]*

V RESULT AND DISCUSSION

We will try to get results using formulae's mentioned below for COCOMO model and using trained dataset we will get the neural network result. Figure 3 displays the variation of estimated effort of COCOMO II, ANN. In the ANN, estimated effort in terms of ELOC, EFFORT, TIME, PEOPLE, and COST which is better as compared to COCOMOII model. In this way, Hybrid model will helps in improving the accuracy of software cost estimation.



Result Using Cocomo

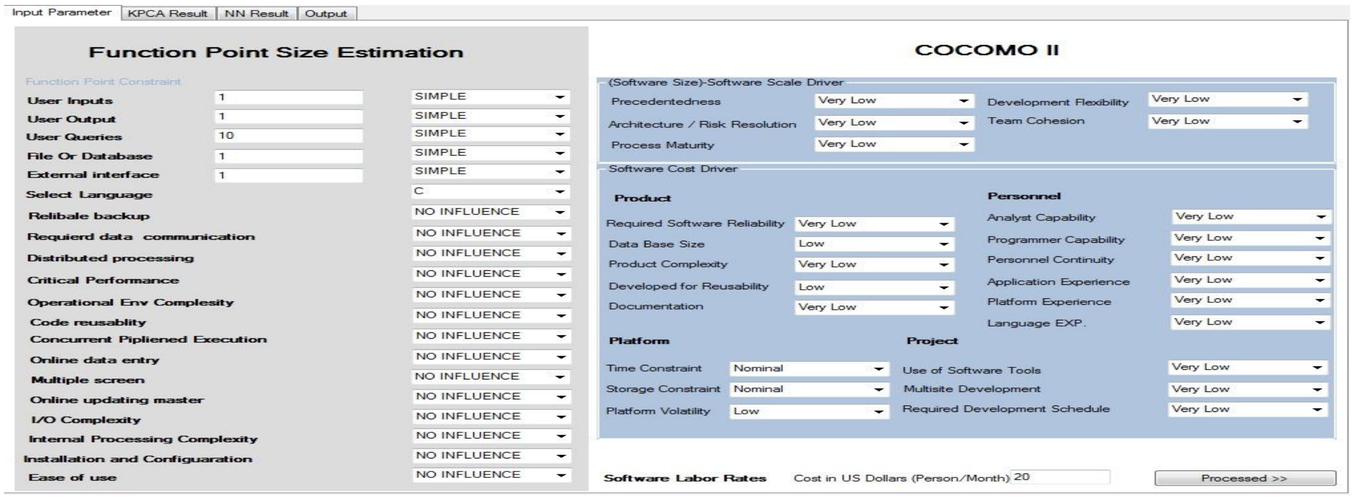
ELOC	4.1405
EFFORT	53.2039383213594
TIME	14.3575060431832
PEOPLE	3.70565320755134
COST	1064.07876642719

Result Using Nueral

ELOC	4.1405
EFFORT	48.5230998302448
TIME	13.4575060431831
PEOPLE	3.60565320755134
COST	970.461996604896

Figure3: Result and graph of COCOMO model and Neural network

Figure 4 present Input Parameter of COCOMOII model, COCOMO II uses function point size estimation method for calculating the size of the software. It is composed of seventeen effort multipliers, five scale factors and four cost factors.



Input Parameter | KPCA Result | NN Result | Output

Function Point Size Estimation

Function Point Constraint

User Inputs	1	SIMPLE
User Output	1	SIMPLE
User Queries	10	SIMPLE
File Or Database	1	SIMPLE
External interface	1	SIMPLE
Select Language		C
Relibale backup		NO INFLUENCE
Requierd data communication		NO INFLUENCE
Distributed processing		NO INFLUENCE
Critical Performance		NO INFLUENCE
Operational Env Complesity		NO INFLUENCE
Code reusablity		NO INFLUENCE
Concurrent Pipliened Execution		NO INFLUENCE
Online data entry		NO INFLUENCE
Multiple screen		NO INFLUENCE
Online updating master		NO INFLUENCE
I/O Complexity		NO INFLUENCE
Internal Processing Complexity		NO INFLUENCE
Installation and Configuaration		NO INFLUENCE
Ease of use		NO INFLUENCE

COCOMO II

(Software Size)-Software Scale Driver

Precedentedness	Very Low	Development Flexibility	Very Low
Architecture / Risk Resolution	Very Low	Team Cohesion	Very Low
Process Maturity	Very Low		

Software Cost Driver

Product

Required Software Reliability	Very Low
Data Base Size	Low
Product Complexity	Very Low
Developed for Reusability	Low
Documentation	Very Low

Personnel

Analyst Capability	Very Low
Programmer Capability	Very Low
Personnel Continuity	Very Low
Application Experience	Very Low
Platform Experience	Very Low
Language EXP.	Very Low

Platform

Time Constraint	Nominal
Storage Constraint	Nominal
Platform Volatility	Low

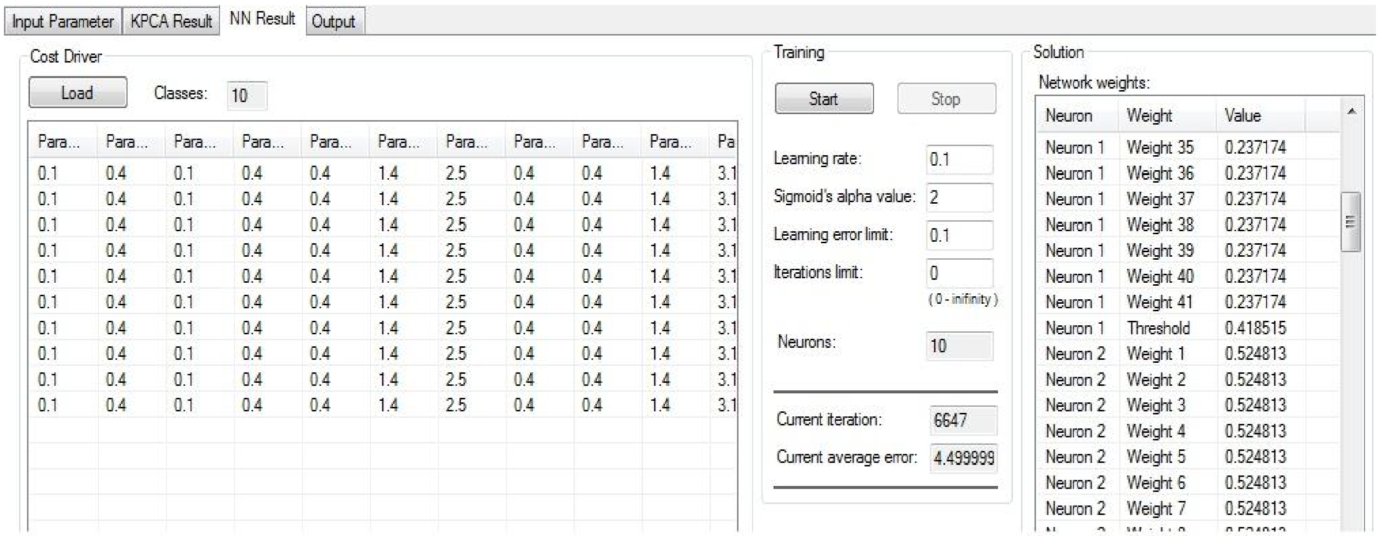
Project

Use of Software Tools	Very Low
Multisite Development	Very Low
Required Development Schedule	Very Low

Software Labor Rates Cost in US Dollars (Person/Month) 20

Figure 4:-Input Parameter of COCOMOII model

Figure 5 displays result of neural network using training dataset and cost driver.



Input Parameter | KPCA Result | NN Result | Output

Cost Driver

Classes: 10

Para...	Para...	Para...	Para...	Para...	Para...	Para...	Para...	Para...	Para...	Pa
0.1	0.4	0.1	0.4	0.4	1.4	2.5	0.4	0.4	1.4	3.1
0.1	0.4	0.1	0.4	0.4	1.4	2.5	0.4	0.4	1.4	3.1
0.1	0.4	0.1	0.4	0.4	1.4	2.5	0.4	0.4	1.4	3.1
0.1	0.4	0.1	0.4	0.4	1.4	2.5	0.4	0.4	1.4	3.1
0.1	0.4	0.1	0.4	0.4	1.4	2.5	0.4	0.4	1.4	3.1
0.1	0.4	0.1	0.4	0.4	1.4	2.5	0.4	0.4	1.4	3.1
0.1	0.4	0.1	0.4	0.4	1.4	2.5	0.4	0.4	1.4	3.1
0.1	0.4	0.1	0.4	0.4	1.4	2.5	0.4	0.4	1.4	3.1
0.1	0.4	0.1	0.4	0.4	1.4	2.5	0.4	0.4	1.4	3.1
0.1	0.4	0.1	0.4	0.4	1.4	2.5	0.4	0.4	1.4	3.1

Training

Learning rate: 0.1
Sigmoid's alpha value: 2
Learning error limit: 0.1
Iterations limit: 0 (0 - inifinity)
Neurons: 10

Current iteration: 6647
Current average error: 4.499999

Solution

Network weights:

Neuron	Weight	Value
Neuron 1	Weight 35	0.237174
Neuron 1	Weight 36	0.237174
Neuron 1	Weight 37	0.237174
Neuron 1	Weight 38	0.237174
Neuron 1	Weight 39	0.237174
Neuron 1	Weight 40	0.237174
Neuron 1	Weight 41	0.237174
Neuron 1	Threshold	0.418515
Neuron 2	Weight 1	0.524813
Neuron 2	Weight 2	0.524813
Neuron 2	Weight 3	0.524813
Neuron 2	Weight 4	0.524813
Neuron 2	Weight 5	0.524813
Neuron 2	Weight 6	0.524813
Neuron 2	Weight 7	0.524813

Figure 5:- Neural network result

POTENTIAL SIGNIFICANCE OF THE PROPOSED MODEL

a.Support large domain space.

b.Learning using standard database: It uses standard rating values stored in sample data set which is provided by International
software benchmarking standard group.

c. Very profound information is easily available.

IV CONCLUSION

By using proposed model the accuracy of cost estimation will be improved.Estimated cost can be very close to the actual cost.COCOMOII model used for calculating the effort,duration cost of the software.Neural network used for cost estimation due to its ability to learn from existing dataset.If KPCA is applied then it's estimates is more accurate than COCOMOII and Neural network.KPCA has the advantage that reconstructs the PCA using kernel for assuming very small or large nonlinear value to construct the cluster for reduction for better feature extraction rate.

V FUTURE SCOPE

Presently software program design experts are becoming aware about effectively forecasting the cost and excellent with the application. Software program development has turned into a essential and important investment decision for many organizations. We can do comparison based study on all SVM model as ICA, PCA, KPCA, IKPCA.

REFERENCES

1.Dr.N.Balaji,Year2013,"*SoftwareCostEstimation using Function Point with Non Algorithmic Approach"*, Global Journal of Computer Science and Technology Software & Data Engineering(USA),Volume 13 Issue 8 Version 1.0,Online ISSN: 0975-4172 & Print ISSN: 0975-4350
2.SwetaKumari,July2013"*PerformanceAnalysisoftheSoftware Cost EstimationMethods:Review"*,IJARCSSE,,Volume3,Issue7,ISSN:2277 128X
3.Radhika Sharma, June 2013, *"COCOMO II Implementation Using Perceptron Learning Rule"*, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 2 Issue 6.
4. Ziauddin, March, 2013, *"A Fuzzy Logic Based Software cost Estimation Models***"**, International Journal of Software Engineering. Vol. 7, No. 2,
5. Anupam Kaushik, August2011,*"COCOMO Estimates Using Neural Networks"*I.J. Intelligent Systems and Applications, 2012, 9, 22-28 Published Online in MECS.
6. VahidKhatibi, 2011,*"Software Cost Estimation Methods: A Review"*,Journal of Emerging Trends in Computing and Information SciencesVol. 2 No. 1
7. Hari, October 2011 *"SEEPC: A Toolbox for Software Effort Estimation using SoftComputingTechniques"*International Journal of Computer Applications (0975 – 8887) Volume 31
8.JNVR Swarupkumar,Dec 2011,"*Fuzzy logic for Software Effort Estimation Using Polynomial Regression as Firing Interval"*,International journal of software engineering and its application
9.Ch.Satyananda Reddy,January2010,"An Neural Network Model for Software Effort Estimation" Int.J.of Software Engineering, IJSE Vol.3 No.1
10Ch. Satyananda Reddy," May 2009, *"A Concise Neural Network Model for Estimating Software Effort*"," International Journal of Recent Trends in Engineering, Issue. 1, Vol. 1,
11Sikka.Kaur, 2010 "Estimating function points: Using machine learning and regression models",(ICETC), 2nd International Conference.
12.Iman Attarzadeh 2012" *Proposing an Enhanced Artificial Neural Network Prediction Model to Improve Accuracy in Software Effort Estimation*" Fourth International Conference on Computational Intelligence,
13.Max Welling,2003,"*Kernel Principal Components Analysis"*,Department of Computer Science, University of Toronto, 10 King's College Road,Toronto, M5S 3G5 Canada
14.Z.Zia,2011"*Software cost estimation for component based fourth-generation- language software applications"*, I T Software, vol. 5, no. 1, (2011), pp. 103-11015.BoehmB.W.,1981*"SoftwareEngineering Economics"*, Englewood Cliffs, NJ,Prentice-Hall, 1981.

BIOGRAPHY

SWATI D.WAGHMODE She is ME Second Year student from Information Technology Department at MAEER's MIT College of Engineering, Pune, India. Holds B.E. Degree in Information technology from SKNCOE 2006(Sinhgad Technical Education Society). She has 2 years teaching experience from VenutaiChavan Diploma College (Sinhgad Technical Education Society, from 2007-2009). Her main area of research interest and domain of expertise is in Software Engineering.

DR. KISHOR R. KOLHEobtained PhD from Shri JJT University, Jhunjhunu (Rajasthan), India in 2013, M. Tech. in Information Technology from the Bharati Vidyapeeth Deemed University, Pune [M.S.] in year 2010 and B.E (Hons)Electronics Engineering from S.G.G.S. Institute of Engineering & Technology,Nanded [M.S.] in year 1996.

He is currently working as Associate Professor in Information Technology Department at MAEER's MIT College of Engineering, Pune, India. He has more than7 years teaching and 10.5 years of industry experience. His areas of interest are Software Engineering, Computer Network and Artificial Intelligence.He has published more than 15 research papers in journals and conferences. He has also guided fifteen undergraduatestudent and two postgraduate.