SOFTWARE EFFORT ESTIMATION FRAMEWORK TO IMPROVE ORGANIZATION PRODUCTIVITY USING EMOTION RECOGNITION OF SOFTWARE ENGINEERS IN SPONTANEOUS SPEECH

B.V.A.N.S.S. Prabhakar Rao¹ and P. Seetha Ramaiah²

¹School of Computing Science and Engineering, VIT University, Chennai Campus, India
E-mail: prabhakarrao@vit.ac.in

²Department of Computer Science and Systems Engineering, Andhra University, India
E-mail: psrama@gmail.com

Abstract

Productivity is a very important part of any organisation in general and software industry in particular. Now a day’s Software Effort estimation is a challenging task. Both Effort and Productivity are inter-related to each other. This can be achieved from the employee’s of the organization. Every organisation requires emotionally stable employees in their firm for seamless and progressive working. Of course, in other industries this may be achieved without man power. But, software project development is labour intensive activity. Each line of code should be delivered from software engineer. Tools and techniques may helpful and act as aid or supplementary. Whatever be the reason software industry has been suffering with success rate. Software industry is facing lot of problems in delivering the project on time and within the estimated budget limit. If we want to estimate the required effort of the project it is significant to know the emotional state of the team member. The responsibility of ensuring emotional contentment falls on the human resource department and the department can deploy a series of systems to carry out its survey. This analysis can be done using a variety of tools, one such, is through study of emotion recognition. The data needed for this is readily available and collectable and can be an excellent source for the feedback systems. The challenge of recognition of emotion in speech is convoluted primarily due to the noisy recording condition, the variations in sentiment in sample space and exhibition of multiple emotions in a single sentence. The ambiguity in the labels of training set also increases the complexity of problem addressed. The existing models using probabilistic models have dominated the study but present a flaw in scalability due to statistical inefficiency. The problem of sentiment prediction in spontaneous speech can thus be addressed using a hybrid system comprising of a Convolution Neural Network and Hidden Markov Model.

Keywords:
Estimation, Productivity, Deep Neural Networks, CNN, Sentiment Prediction, Spontaneous Speech, HMM

1. INTRODUCTION

1.1 PROBLEM

Right from the inception, software industry is facing lot of trouble in releasing the product on time and within the budget limit. Of course, good estimation tools are available in the market at an affordable price with many features. But few of the well suited for many application. Whatever be the reason the software product development success rate is low and failure rate is high. Software Engineering is a systematic approach to the development, operation, maintenance and retirement of software product. Software is intangible product. Based on the contract type after discussion with all the stakeholders customer’s organization has to estimate the required effort, schedule and cost of the product with delivery date with in the budget limit [26]. Sometimes customer will fix the delivery date then developing team has to work with that dead line. This research is mainly focused on only emotion of the software engineers. The approach is so simply. In this paper we will focus on the estimation of a particular product with the concern and willingness of the developing team member, the so called software engineer, who is going to take part of the project with their present emotional state. For suppose if we want to estimate the travel time from source to destination. We need to consider the parameters like vehicle condition, driver potential and road condition including traffic. So, Software project development in industry is labour intensive, nothing wrong in considering software engineers emotional conditions. Because their priorities like work culture, shifts, location, domain, platform, team, onsite, work at home, etc., may change from time to time.

Hence, these study suggesting the estimator to consider emotional level of each team member and confirm your plan of execution. Of course some researchers arguing that in the field of estimation, team formation and other things are not so important. But, this paper is proposing whatever the work nothing wrong in knowing their situation before assigning the work task. The main challenge is assign the right task to right employee may lead to minimize the burden and in turn that will improve organization productivity.

There is a common myth in industry that when we nearing due date, increase the number of programmers so that product can be delivered on time. Because of complexity and other issues this is not a suggested one [28-30].

1.2 BACKGROUND

The feedback system can be used as a continuous assessment tool for gauging the emotions of the employees on a regular basis. This system must be implemented such that we can accurately determine and hence classify the emotions of speech that is the source of expression for all employees. The technology age is ushering in a need for a lot of abstract level data analysis and predictive models that learn from user experience. Sentiment prediction hence plays a vital role in identification of feedbacks and assessments from end users. Deep Neural network is a new sub branch of machine learning algorithms that are used extensively in the modeling of audio/visual signals using an array of architectures such as convolution neural networks and deep belief networks.
1.3 THE MOTIVATION FOR THIS WORK

According to Banker and Kauffman, Software productivity is the ratio between the functional values of software produced (size of the application developed) to the labour and expense of producing it. The present day tools to measure software productivity take into account the functionality delivered to the customer, the complexity of the program being developed, and the time and effort involved. In addition to that measurement we take into account the willingness of the team member to be associated with developing software. The reason is so simply, for suppose the father may dream on his son career and he may fix that he should become a software engineer. But son might interest in other field. By forcing the father ideas on the son finally it may leads to spoil the life of the son. So, this work helps the estimator to know about team sentiment on the assigned project.

Most current sentiment prediction models are text-computation intensive. Moreover a lot of information is lost as important characteristics like voice modulation, pitch are ignored in text analysis. How a person speaks is a big indication of what is intended in speech, and spontaneity is better observed in speech than in written material. Our proposed system performs sentiment prediction on immediate speech samples, which can be applied for study of one-to-one interactions.

2. EMPIRICAL EVIDENCE

In software cost estimation, effort allocation is an important and usually challenging task for project management. Project managers often find it difficult to plan for staffing and other team resources. This often leads to risky decisions to assign too few or too many people to complete software lifecycle activities. As a result, projects with inaccurate resource allocation will generally experience serious schedule delay or cost overrun. A fuzzy logic approach is used to calibrate the productivity factor in the regression model. Moreover, a multilayer perceptron neural network model was developed to predict software effort based on the software size and team productivity [27, 33].

2.1 DEEP NEURAL NETWORK

The conventional models used for event detection in speech uses models such as Gaussian Mixture model [8], Dynamic Bayesian Network, Conditional Random Fields [5,6], and Support Vector Machines[7]. Even though above mentioned models have achieved good results, they lack the expressive power to capture high-level semantics. To address this problem, deep architectures have been employed in the field of vision and speech recognition. This work focuses on Non-linear architectures such as Deep Boltzmann Machines [2] and Convolutional Neural Net [4].

2.1.1 Deep Belief Network:

It consists of a stack of restricted Boltzmann machines trained greedily layer by layer [9]. RBM are complete bipartite graphs that comprises of two sets of hidden and visible units. Deep belief network is per-trained to maximize data likelihood and then fine tune as an artificial neural network. The weights initialized by per-training helps the model to avoid bad local minima. The algorithm used for training is contrastive divergence, which enables these networks to learn complex representation [16].

2.1.2 Hidden Markov Model:

An HMM is a doubly stochastic process with an underlining process that is not observable i.e. hidden but can only be observed through another set of process that produce a sequence of observed symbols. ‘An HMM is considered as the simplest dynamic Bayesian network’ [10]. This probabilistic model has been extensively used in speech recognition (emotion recognition) to discriminate model [20, 21].

2.1.3 Deep learning in Emotion Recognition:


3. PROPOSED ESTIMATION MODEL

3.1 SIZE, EFFORT, AND PRODUCTIVITY

Sizing development effort using a well-established estimation method like FPA can be a very powerful yardstick that can use effectively estimate [25]. A key metric used to determine

\[
\text{Effort} = (\text{system size}) \times (\text{productivity})
\]

System size might be in the form “thousand of line of code” (KLOC).

Software project effort and cost:

\[
(\text{effort, cost}) = f(\text{size, factors})
\]

- time to develop
- staff
- quality
- productivity

Through well-defined sizing and costing methods, IT groups within organizations can derive a variety of benefits in various activities that include software contracts, project management, cost of ownership; IT budgets, outsourcing costs, and more [31-32].

1077
Fig.1. Software Estimation with productivity

3.2 HYBRID DBN HMM

In [4] the work done on this modelling uses dynamic frame-level with DBN based acoustic models. Their work involved model as a left-to-right HMM with 1, 3, or 5 states and 16 Gaussian mixture components. An additional HMM is used to get background noise at two positions, beginning and end for each utterance. After training, the HMMs create mappings and improve the networks. After this the probabilities do not change.

They achieved an accuracy of 45.08% UAR with 1-state HMMs and 37-frame input windows and an accuracy of 45.60% UAR wherein 57 frame input windows and different HMMs were used.

4. IMPLEMENTATION

Software productivity is a simple concept. Although its earliest measurement was in lines of code per man-hours worked, a better definition is the ratio between the functional value of software produced to the labour and expense of producing it. There are several ways to measure software productivity, including Function Point Analysis, Cost Component Modelling, Cyclomatic Complexity, and program performance metrics that take into account the costs of running and maintaining the software.

CNN principles are applicable along the frequency access of speech. The temporal modelling is handled by the HMM model and the dependency between adjacent speech frames is dealt with long time context window. Each input of neural network is a stack of consecutive feature frames centering at time t while the target output is the probability of the centered frame belonging to each HMM state at time t [17-18].

4.1 SAMPLE DATA SET

A practitioner investigating the SEE literature is likely to find a dauntingly large space of studies, conducted on different data sets and employing different SMs and sometimes with contradictory recommendations [34]. Software size estimate is one of the most popular inputs for software effort prediction models. Accordingly, providing a size estimate with good accuracy early in the lifecycle is very important; it is equally challenging too [35], [36]. This study aims at quantitatively analyzing and effectively handling local bias associated with historical cross-company data [24], thus improves the usability of cross-company datasets for calibrating and maintaining parametric estimation models [37], [38]. For these kinds of work the origin database was made as a part of the DFG funded research project SE462/3-1 in 1997. The recordings took place in the anechoic chamber of the Technical University Berlin, department of Technical Acoustics. There are sample emotions classes in the set along with neutral version. The recording includes samples from both male and female. The audio files are in .wav format.

Table.1. Sample code of emotions

<table>
<thead>
<tr>
<th>Letter (English)</th>
<th>Emotion (English)</th>
<th>Letter (German)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Anger</td>
<td>W</td>
</tr>
<tr>
<td>B</td>
<td>Boredom</td>
<td>L</td>
</tr>
<tr>
<td>D</td>
<td>Disgust</td>
<td>E</td>
</tr>
<tr>
<td>F</td>
<td>Anxiety/fear</td>
<td>A</td>
</tr>
<tr>
<td>H</td>
<td>Happiness</td>
<td>F</td>
</tr>
<tr>
<td>S</td>
<td>Sadness</td>
<td>T</td>
</tr>
<tr>
<td>N</td>
<td>Neutral version</td>
<td></td>
</tr>
</tbody>
</table>

1078
Table 2. Sample information about speakers

<table>
<thead>
<tr>
<th>SCode</th>
<th>Gender</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE003</td>
<td>Male</td>
<td>31</td>
</tr>
<tr>
<td>SA008</td>
<td>Female</td>
<td>34</td>
</tr>
<tr>
<td>ST009</td>
<td>Female</td>
<td>21</td>
</tr>
<tr>
<td>SD010</td>
<td>Male</td>
<td>32</td>
</tr>
<tr>
<td>SC011</td>
<td>Male</td>
<td>26</td>
</tr>
<tr>
<td>SD011</td>
<td>Female</td>
<td>35</td>
</tr>
<tr>
<td>SC013</td>
<td>Female</td>
<td>25</td>
</tr>
<tr>
<td>SC014</td>
<td>Female</td>
<td>32</td>
</tr>
<tr>
<td>SC015</td>
<td>Male</td>
<td>31</td>
</tr>
</tbody>
</table>

As discussed in the Fig. 2 the estimator will follow this approach at the beginning. May come to a decision that the team member is well-matched with the proposed application or not. If not, immediately they can take the remedial action on the team formation.

4.2 LOCALITY

Local frequency structures are modelled by convolution layer to receive input only from limited bandwidth of the speech spectrum. The speech inputs in a frequency scale can be divided into a number of local bands. Thus standard MFCC features are not suitable in this scenario. However linear spectrum mel-scale spectrums are perfect for local filtering.

4.3 MAX POOLING

Max pooling is added at the top of each convolution layer and is divided into m bands. Each band receives input from our convolution layer neighboring bands to generate j values, representing the maximum activation received from j convolution filters within r bands. The max pooling usually generates a lower resolution version of the convolution layer by doing a maximization operation of every n bands.

4.4 WORKING PROCEDURE

In any neural network oriented training we need use threshold value in order to adjust weight as per the constraints. This entire process will continue till the desired level. That means the estimator may decide the threshold for training or may stop after fulfillment.

The CNN consists of one or more pairs of convolution and max-pooling layers, where the lowest layers process a small number of input frequency bands independently to generate higher level representation with lower frequency resolution. The number of bands decreases in higher layers. The input to each convolution layer can be padded to ensure that the first and last input bands are processed by a suitable number of filters in the convolution layer. In this work, each input is padded by adding half of filter size of dummy bands before and after the first and last bands so that the number of bands stays the same in both the input and convolution layers. Usually the top layers in CNN are fully connected just like that of a normal forward-feeding NN. These fully connected top layers are expected to combine different local structures extracted in the lower layers for the final recognition purpose [19].

When it is used in a hybrid NN-HMM model for speech recognition, posterior probabilities of HMM states are computed using a top softmax layer. The CNN processes each input speech utterance by generating all HMM state probabilities for each frame. Then a Viterbi decoder is used to get the sequence of labels corresponding to the input utterance [23].

In training stage, CNN is estimated using the standard backpropagation algorithm to minimize cross entropy of targets and output layer activations. For a max-pooling layer, the error signal is backpropagated only to the convolution layer node that generates the maximum activation within the pooled nodes. The training targets are obtained from forced alignments generated from a trained HMM model [22].
5. ANALYSIS OF RESULTS

Of the four models used to train for prediction, the most efficient was the Multi Level perceptron with 5 layers. Its Validation misclassification error rate was as low as 35.26% with a test case misclassification error of 43.33% as shown in Table.3.

Table.3. Table showing training phase results of different models with Misclassification Error

<table>
<thead>
<tr>
<th>Model</th>
<th>Validation Error</th>
<th>Test Case Error</th>
<th>Training Time seconds/epoch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maxout Network</td>
<td>75.57%</td>
<td>76.21%</td>
<td>120</td>
</tr>
<tr>
<td>Multi Level Perceptron</td>
<td>35.27%</td>
<td>43.41%</td>
<td>103</td>
</tr>
<tr>
<td>Convolution Network</td>
<td>76%</td>
<td>76.26%</td>
<td>500</td>
</tr>
<tr>
<td>Multi Level Perceptron with more layers</td>
<td>35.26%</td>
<td>43.33%</td>
<td>153.8</td>
</tr>
</tbody>
</table>

Fig.3. Mel spectrogram based on different categories of emotion

Fig.4. Graph output analysis for Multi layer perceptron 4 layers

This plot represents the convergence of MLP (4 layer) after 400 iterations. The x-axis is the number of examples that were observed by the algorithm during testing and validation phase. The y-axis represents the misclassification error. Thus the objective is to minimize the misclassification error.

Fig.5. Graph output analysis for Maxout network Model
The accuracy of the implemented system can be improved further and deeply integrated system software can be developed that uses sentiment prediction in its filtering process with all the constraints in different applications. The current model takes only speech as an input, the scope of the model can be extended to incorporate visual expression also as an input and generalize over an audio-visual input.

REFERENCES


