Fuzzy Analogy: A New Approach for Software Cost Estimation

Ali Idri, ENSIAS, Rabat, Morocco
Alain Abran, ETS, Montreal, Canada
Taghi M. Khoshgoftaar, FAU, Boca Raton, Florida

11th International Workshop on Software Measurement, Montréal, 28-29 August, 2001
Summary

- Introduction
- Software Cost Estimation Models
- Estimation by Analogy
- Fuzzy Logic
- Linguistic Quantifiers
- Fuzzy Analogy
- Conclusions and Future Work
Introduction

- Software cost estimation is one of the most critical activities in managing software projects.
- Estimation by Analogy is a promising technique to solve the software estimation problem.
- Critic: Estimation by Analogy cannot handle categorical data such as ‘very low’, ‘low’, ‘high’....
- However, software projects are always described by categorical rather than numerical data:
  - COCOMO’81: 16 out of 17 attributes are categorical
  - COCOMOII: 22 out of 24
  - Function Points: Evaluation of the complexity for Inputs, Outputs, Files, Inquiries. Evaluation of the TCF
Objective

A new approach for software cost estimation based on reasoning by **Analogy, Fuzzy Logic and Linguistic Quantifier**
Software Cost Estimation Models

History

- 1975, Halstead
- $\text{Effort} = \frac{\mu_1 N_2 \log(\mu_1 + \mu_2)}{2S\mu_2}$
- 1978, Putnam
- $CT = \left(\frac{\text{DSI}}{C}\right)^{\frac{9}{7}} D^{\frac{4}{7}}$
- 1981, Boehm, COCOMO
Software Cost Estimation Models

Classification

Algorithmic Models

- Regression simple/multiple, Interpolation, Bayesian, PCA, etc.

Advantages
- Easy to use
- Easy to develop

Critics
- They make assumption about the form of the prediction function $Effort = \alpha \times size^\beta$
- They need to be adjusted or calibrated to local circumstances
Software Cost Estimation Models

Classification

Non-algorithmic Models

- NN, CBR, Rule Induction, Regression Trees

Advantages

- Capabilities to adequately model the complex set of relationship between factors
- Learning
- Their behavior is easy to understand

Critics

- They are not easy to develop
- They need software tools to automate their process
Estimation by Analogy

- Estimation by Analogy is based on the affirmation:
  Similar software projects have similar costs

- Estimation by analogy is composed by:
  - Characterization of the projects by a set of attributes such as Reliability, Complexity, Analysts competence ...
  - Evaluation of the similarity between the candidate project and each project in the database
  - Adaptation

- Related Works: Vacninanza, Sheppered, Briand, Angelis,...
Shepperd et al. (1997)

\[ d(P_1, P_2, V) = \frac{1}{\sum_{v_j} d_{v_j}(P_1, P_2)} \]

\[ d_{v_j}(P_1, P_2) = \begin{cases} 
(V_j(P_1) - V_j(P_2))^2 & \text{si } V_j(P_1) \neq V_j(P_2) \\
0 & \text{si } V_j(P_1) = V_j(P_2)
\end{cases} \]

Imprecise and Uncertain Data

Low, High, Excellent ???

Classical Logic

Fuzzy Logic
The equality distance is not precise and can give great difference when estimating effort for two similar software projects described by Vagueness information.

We have proposed a set of similarity measures based on fuzzy logic.

We have validated by means of an axiomatic approach the proposed similarity measures.

We have improved the retained measures by using linguistic quantifier guided aggregations.
Fuzzy Logic

- **Values between ‘**TRUE**’ and ‘**FALSE**’?**

  ‘The main motivation of fuzzy logic is the desire to build up a formal, quantitative framework that captures the vagueness of human knowledge via natural languages’ Dubois and Prade 1991

- **1965, Zadeh : Fuzzy Set**

- **1994, Zadeh : Fuzzy Logic = Fuzzy Set Theory**

- **Fuzzy Set:** set with a membership function which takes values in the unit interval \([0, 1]\) rather than in the \(\{0, 1\}\) as in the classical logic
Linguistic quantifiers

- **Human discourse uses a large number of linguistic quantifiers**

- **Zadeh distinguishes between two classes:**
  - Absolute linguistic quantifiers *‘approximately 10’*
  - Proportional linguistic quantifiers *(most, few, at least, at most, …)*

- **Yager has distinguished three categories of proportional quantifiers:**
  - RIM quantifiers *(most, at least $\alpha$, …)*
  - RDM quantifiers *(few, at most $\alpha$, …)*
  - RUN quantifiers *(about $\alpha$)*
Fuzzy Analogy for Cost Estimation

- **Fuzzy Analogy is a fuzzification of the classical analogy procedure**
- **Fuzzy Analogy is composed of three steps:**
  - Identification of software projects
  - Evaluation of similarity between projects
  - Adaptation

- **Identification Step:**
  - The aim is to describe the software projects by a set of attributes that are:
    - Relevant
    - Independent
    - Comprehensive
    - Operational
Each selected attribute is measured either by numerical or categorical data.

Categorical data are represented by fuzzy sets rather than classical set.

Example: The factor DATA of the COCOMO model.

<table>
<thead>
<tr>
<th></th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very high</th>
</tr>
</thead>
<tbody>
<tr>
<td>D/P&lt;10</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>55</td>
</tr>
<tr>
<td>10&lt;=D/P&lt;100</td>
<td>10</td>
<td>100</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>100&lt;=D/P&lt;1000</td>
<td>55</td>
<td>550</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>D/P&gt;=1000</td>
<td>100</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>

Each selected attribute has a weight expressing its importance, $U_k$. 

© 2001 Software Engineering Management Research Laboratory
Evaluation of software projects similarity

\[ d_v(P_1, P_2) \]

\[ d_{vm}(P_1, P_2) \]

\[ d(P_1, P_2) \]
Individual similarities

\[
d_{v_j}(P_1, P_2) = \begin{cases} 
\max \min_{k} (\mu_{A_k}^{v_j}(P_1), \mu_{A_k}^{v_j}(P_2)) & \text{max-min aggregation} \\
\text{ou} & \\
\sum_{k} \mu_{A_k}^{v_j}(P_1) \times \mu_{A_k}^{v_j}(P_2) & \text{sum–product aggregation}
\end{cases}
\]

Overall similarity

\[
d(P_1, P_2) = \begin{cases} 
\text{all of } d_{v_j}(P_1, P_2) & \\
\text{most of } d_{v_j}(P_1, P_2) & \\
\text{many of } d_{v_j}(P_1, P_2) & \\
\text{at least four of } d_{v_j}(P_1, P_2) & \\
\text{...} & \\
\text{there exists of } d_{v_j}(P_1, P_2)
\end{cases}
\]

\[
d(P_1, P_2) = \sum_{j=1}^{M} w_j d_{v_j}(P_1, P_2)
\]

\[
w_j(P_1, P_2) = Q\left(\frac{\sum_{k=1}^{j} u_k}{T}\right) - Q\left(\frac{\sum_{k=1}^{j-1} u_k}{T}\right)
\]
Adaptation

Two questions

1- How many similar projects will be used in the adaptation?
2- How to adapt the chosen analogies in order to generate an estimate for the new project?

In the literature, there is no clear rule to guide the choice of the number of similar projects, $K$

In general $K=2$

Suppose that the first three similar projects to the new project $P$ have the following distances: 3.00, 4.00 and 4.01

When $K=2$, we consider only the two first projects

Why we have not take into account the third projects?
Solution

- What is ‘$P_i$ is closely similar to $P$’?
- The $d(P_i, P)$ is in the vicinity of 1

Adaptation formula:

$$\text{Effort}(P) = \frac{\sum_{i=1}^{N} \mu_{\text{vicinity of } 1}(d(P, P_i)) \times \text{Effort}(P_i)}{\sum_{i=1}^{N} \mu_{\text{vicinity of } 1}(d(P, P_i))}$$

If $\mu_{\text{vicinity of } 1}(x) = x$ then (3) is exactly the ordinary weighted average.
Conclusions and Future work

- We have propose a new approach for software cost estimation: Fuzzy Analogy when software projects are described by categorical data.

- Fuzzy Analogy is also applicable when the variables are numeric (no uncertainty).

- Advantages of Fuzzy Analogy
  - It can handle correctly the imprecision and the uncertainty when describing software project.
  - It can be easily adapted to the needs of each organization (RIM linguistic quantifiers, Vicinity of 1,...).
Empirical validation of Fuzzy Analogy is based on

- COCOMO’81 dataset
- **F_ANGEL**: A Software prototype based on Fuzzy Analogy
  (To be submitted at 8th IEEE Metrics, June, Ottawa, Canada)

Building prediction systems by analogy that satisfy **Soft Computing**:

- Tolerance of imprecision (Fuzzy Logic)
- Learning (Neural Networks)
- Uncertainty (Belief networks, genetic algorithms,...)