Chapter VI

Web Effort Estimation Using Case-Based Reasoning

Abstract

Software practitioners recognise the importance of realistic effort estimates to the successful management of software projects, the Web being no exception. Having realistic estimates at an early stage in a project’s life cycle allow project managers and development organisations to manage resources effectively. Several techniques have been proposed to date to help organisations estimate effort for new projects. One of these is a machine-learning technique called case-based reasoning. This chapter presents a case study that details step by step, using real data from completed industrial Web projects, how to obtain effort estimates using case-based reasoning, and how to assess the prediction accuracy of this technique. The reason to describe the use of case-based reasoning for effort estimation is motivated by its previous use with promising results in Web effort estimation studies.
**Introduction**

The effort estimation technique described in this chapter is called case-based reasoning (CBR). This technique, proposed by the machine-learning community, uses the following claim as its basis: Similar problems provide similar solutions.

CBR provides effort estimates for new projects by comparing the characteristics of the current project to be estimated against a library of historical data from completed projects with known effort (case base). It involves the following (Angelis & Stamelos, 2000).

1. **Characterising a new project** $p$ for which an effort estimate is required, with variables (features) common to those completed projects stored in the case base. In terms of Web and software effort estimation, features represent size measures and cost drivers, which have a bearing on effort. This means that if a Web company has stored data on past projects where, for example, the data represents the features effort, size, development team size, and tools used, the data used as input to obtaining an effort estimate will also need to include these same features.

2. **Use of this characterisation as a basis for finding similar (analogous) completed projects for which effort is known.** This process can be achieved by measuring the distance between two projects at a time (project $p$ and one finished project) based on the features’ values for all features ($k$) characterising these projects. Each finished project is compared to a project $p$, and the finished project presenting the shortest distance overall is the most similar project to project $p$. Although numerous techniques can be used to measure similarity, nearest-neighbour algorithms using the unweighted Euclidean distance measure have been the most widely used to date in Web and software engineering.

3. **Generation of an effort estimate for project $p$ based on the effort for those completed projects that are similar to $p$.** The number of similar projects to take into account to obtain an effort estimate will depend on the size of the case base. For small case bases (e.g., up to 90 cases), typical values are to use the most similar finished project, the two most similar finished projects, or the three most similar finished projects (one, two, and three closest neighbours or analogues). For larger case bases, no conclusions have been reached to date regarding the best number of similar projects to use. There are several choices to calculate estimated effort, such as the following:
   a. To use the same effort value as the closest neighbour.
   b. To use the mean effort for the two or more closest neighbours.
   c. To use the median effort for the two or more closest neighbours.
Related Content

Convergence Aspects of Autonomic Cooperative Networks
[www.igi-global.com/article/convergence-aspects-autonomic-cooperative-networks/65069?camid=4v1a](www.igi-global.com/article/convergence-aspects-autonomic-cooperative-networks/65069?camid=4v1a)

Server-Side Technologies
[www.igi-global.com/chapter/server-side-technologies/31193?camid=4v1a](www.igi-global.com/chapter/server-side-technologies/31193?camid=4v1a)

Quality of Service for Multimedia and Real-Time Services
[www.igi-global.com/chapter/quality-service-multimedia-real-time/61909?camid=4v1a](www.igi-global.com/chapter/quality-service-multimedia-real-time/61909?camid=4v1a)
Performance Evaluation of a Modern Web Architecture
www.igi-global.com/article/performance-evaluation-of-a-modern-web-architecture/85321?camid=4v1a