Feature Weighting Techniques for CBR in Software Effort Estimation Studies: A Review and Empirical Evaluation

Boyce Sigweni Martin Shepperd
Department of Computer Science
Brunel University
London, UB8 3PH,United Kingdom
{boyce.sigweni,martin.shepperd}@brunel.ac.uk

ABSTRACT

Context: Software effort estimation is one of the most important activities in the software development process. Unfortunately, estimates are often substantially wrong. Numerous estimation methods have been proposed including Case-based Reasoning (CBR). In order to improve CBR estimation accuracy, many researchers have proposed feature weighting techniques (FWT).
Objective: Our purpose is to systematically review the empirical evidence to determine whether FWT leads to improved predictions. In addition we evaluate these techniques from the perspectives of (i) approach (ii) strengths and weaknesses (iii) performance and (iv) experimental evaluation approach including the data sets used.
Method: We conducted a systematic literature review of published, refereed primary studies on FWT (2000-2014).
Results: We identified 19 relevant primary studies. These reported a range of different techniques. 17 out of 19 make benchmark comparisons with standard CBR and 16 out of 17 studies report improved accuracy. Using a one-sample sign test this positive impact is significant ($p = 0.0003$).
Conclusion: The actionable conclusion from this study is that our review of all relevant empirical evidence supports the use of FWTs and we recommend that researchers and practitioners give serious consideration to their adoption.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management—*Cost estimation*; I.2.6 [**Learning**]: Analogies

General Terms

Systematic literature review, Meta-analysis

Keywords

Software effort estimation, Case-based reasoning, Feature weighting, Feature subset selection

PROMISE '14, September 17 2014, Torino, Italy
Copyright 2014 ACM 978-1-4503-2898-2/14/09 ...$15.00.
http://dx.doi.org/10.1145/2639490.2639508

1. INTRODUCTION

Given the importance of timely and accurate software cost prediction it is unsurprising that there is a large body of published research work, the majority of which has focused on effort prediction since effort is normally the dominant and hardest to predict component of overall cost. For an overview of the extent of this research see the mapping study by Jørgensen and Shepperd [20] and more recently the review of machine learning based studies by Wen et al. [45].

Although many approaches have been proposed, a widely used technique is based upon case-based reasoning (CBR) [29] and is usually referred to as estimation by analogy (EBA) [40, 35]. CBR essentially proceeds by using knowledge of past episodes of interest called cases that are encoded as vectors of features that describe the case state and the case solution. New, or target cases, are solved by utilising solutions from past cases that exhibit similarity, i.e., are proximal in the feature space.

For effort prediction a case is usually a project, the case state will be features such as size, development environment, client experience, etc., and the solution is the actual amount of effort utilised. Thus the case-base is conceptually an $n \times p$ matrix where there are n cases and p features. Often the features that are included in the case-base are more due to happenstance and availability rather than because it is known that there are well defined relationships with the case solution. Moreover, there may exist multicollinearities amongst these features. Consequently, as is common with the majority of machine learning techniques, it is widely acknowledged that not all features are of equal importance [42, 14]. Thus CBR systems will benefit from optimisation of the feature sets.

Feature set optimisation can be accomplished by means of feature weighting which has the effect of stretching or compressing the feature space thus impacting the proximity of cases (projects) and thus modifying the set of neighbour projects that are used to donate solutions. Such problems are NP-hard and for non trivial numbers of cases and feature sets present significant computational challenges. A slightly less daunting approach, although still NP-hard, is feature subset selection where features are assigned weights of $\{0, 1\}$. Until recently this has been the dominant approach within software effort prediction.

So although there is widespread consensus that some form of feature weighting technique is beneficial there has been no systematic review of all relevant primary studies, Nor has there been an analysis of the extent of FWTs, how they have been experimentally evaluated and the interaction between performance and data set.

Our systematic literature review (SLR) aims to identify and empirically evaluate existing feature weighting techniques used in analogy-based software effort estimation studies published between January 2000 and April 2014. The SLR characteristics are summarised in Table 1 and described in more detail in subsequent sections. Our results show that feature subset selection (FSS) is an important aspect of CBR, however feature weighting is still under explored as evidenced by the small number of articles, proposing many techniques. There are many published feature weighting techniques which vary and are complex. No up-to-date, comprehensive picture of the current state of FWTs in CBR exists, the closest being the review by Wettschereck et al. [47] which we utilise to provide a framework for comparison.

The remainder of the paper is organised as follows. The next section summarises the main underlying ideas of feature weighting for case-based reasoning software project prediction. This is followed by a description of our systematic literature review (SLR) procedure and the protocol which is summarised by Table 1. We also set out the four research questions we intend the SLR to address. Section 4 presents our results organised by research question and the paper concludes with a discussion of the implications of this meta-analysis for researchers and practitioners.

2. BRIEF BACKGROUND TO THE REVIEW TOPIC

The feature weighting techniques explored in this systematic literature review are embedded in CBR algorithms employing different variants of the similarity distance function shown in Equation 1.

$$Similarity(T, S) = \sum_{k=1}^{p} f(T_k, S_k) \times w_k \qquad (1)$$

where T is the target case, S is the source case, f is the similarity function, p is the number of features and w_k is the k^{th} feature weight where $1 \leqslant k \leqslant p$. Typically, but not necessarily, f is some variant of Euclidean distance.

As previously indicated FSS is a special case where only two possible weights for $w_k, \{0, 1\}$ provide for inclusion or exclusion. The more general case is $w_k, [0, 1]$. While the basic CBR algorithms assign a constant scalar to w_k, these feature weighting techniques allow different weights to be assigned to features. These techniques have been frequently studied and used in CBR, but it is yet to be established how they all relate or how effective such weighting schemes actually are.

There are two general approaches to finding feature weights for a particular case base, namely filters and wrappers. Filters use statistical or other general information that can be extracted from the data set alone to attempt to determine the important features which is then reflected in the feature weights. An example would be to use principal components analysis. The alternative, but far more computationally demanding, method is to repeatedly apply the CBR system to different sets of feature weights with a view to searching for a more effective set. Generally wrappers are found to perform better than filters (see Kohavi and John [28]).

The earliest work on feature weighting in the domain of software effort estimation, that we are aware of, was by Mair et al. [33] in which an exhaustive search combined with a wrapper was applied to the Desharnais data set. Clearly the disadvantage of an exhaustive search is the computational cost, particularly with regard to p, the number of features such that the cost approximates to 2^p but is linear with regard to the number of cases n. The approach was extended Kirsopp et al. [23] based upon the use of metaheuristic search to find good feature subsets. The authors reported substantial improvement on predictive performance over standard EBA.

Subsequently the basic ideas have been extended by a number of different research groups including extension of metaheuristic search to explore population based techniques such as genetic algorithms [18, 32] and more statistically based procedures such as Mantel's matrix correlation [21]. Other research groups have worked on hybrid techniques, for example Wu et al. [48].

Whilst many of these studies have individually claimed good results, it is unclear what is the big picture nor the overall level of support for feature weighting techniques either in general or in particular. Nevertheless the informal picture would seem to be quite encouraging. This then is the motivation for our systematic literature review.

3. METHOD

The notion of evidence based software engineering was first advocated by Kitchenham et al. [25]. Our approach is strongly influenced by the methodological work of Kitchenham and Charters [24] who have adapted many of the general principles of conducting systematic literature reviews to the specific setting of software engineering. However, we deviated in the following respects principally due to the fact that our review was very focused and the number of papers involved was very small:

- We didn't solicit external review of our protocol.
- The study quality was only addressed in terms of peer review of the candidate studies, so we had no formal quality instrument.
- We didn't contact the authors of the papers we located in this review. The rationale was that both authors were involved in updating the Jørgensen and Shepperd cost estimation mapping study [20] and during that process we had emailed the same authors for papers that we may have missed or were accepted therefore in-press. The updated mapping study is a superset and therefore subsumes all papers in the FWT SLR described by this paper.

3.1 Research questions

The principal aim of this SLR is to answer the question of whether feature weighting techniques improve the predictive performance of CBR prediction systems for software effort? In doing so we need to review the range of FWTs that have been proposed, consider how they have been empirically evaluated and using which data sets. This will enable us to provide guidance for researchers and

Table 1: Systematic Literature Review Summary

Characteristic	Value
Review type	Systematic literature review
Research question(s)	Do FWTs improve predictive performance (RQ3)? Related questions are the range FWTs (RQ1), their strengths and weaknesses (RQ2) and the various approaches to the primary study experimental design (RQ4).
Purpose	Guide researchers and practitioners using estimation by analogy techniques.
Audience	Researchers
Search method	Automated and hand search, citation analysis, previously known articles plus approached authors.
Databases used	BESTweb, IEEE Xplore, ScienceDirect, Google scholar, ACM digital library, Springer
Population	Empirical studies relating to FWT in software effort estimation
Setting	Commercial software projects
Study types	Experiments, case studies, observational studies, simulation.
Inclusion criteria	(i) Refereed paper (journal or conference) (ii) Empirical study (iii) Copy of the article available
Language	English language only
Search date	April 2014
Article dates	2000-2014 plus in press articles

practitioners and identify areas for further research in order to improve the performances of current CBR models. To achieve this objective four research questions are formulated and presented below.

> ***RQ1***: What is the range and diversity of feature weighting techniques used for software development effort estimation? In answering this question we characterise them using the dimensions from a previous review by Wetterschereck [47]
> *Rationale*: Practitioners can take the identified feature weighting techniques as candidate solutions in their practice. For feature weighting techniques that have not yet been employed in CBR, researchers can explore the possibility of using them as potential feasible solutions.
>
> ***RQ2***: What are the strengths and weaknesses of existing feature weighting techniques?
> *Rationale*: This will be helpful for practitioners to better understand the practical issues around deployment.
>
> ***RQ3***: What is the estimation accuracy of each FWT and how do they compare?
> *Rationale*: to enable us to compare techniques and determine which support the most accurate cost predictions.
>
> ***RQ4***: How has the experimental evaluation been conducted e.g., which performance metrics are used?
> *Rationale*: This helps determine the importance we attach to the evidence e.g., some performance metrics such as Mean Magnitude of Relative Error (MMRE) have been shown to be biased by studies such as [26, 12].

3.2 Search strategy

Previous systematic reviews e.g. [20] have reported that automated article searches via on-line databases may lead to low recall rates, may not be thorough and also with the likelihood of much additional work arising from low precision rates. However, since our search is relatively focused there is little danger of that manifesting in significant proportion. The following sections discuss how we: defined our search terms, selected appropriate literature sources and the search process used.

3.2.1 *Search terms*

We employed the following steps to build the search terms based on [1]:

1. Derive major terms from the research questions.
2. Identify synonyms and alternative spellings for major terms.
3. Using the Boolean OR and AND to incorporate and link alternative spellings and synonyms.

The resulting search strings are described as follows: Software AND (effort/OR cost/) AND (“prediction” OR “estimation” OR “forecasting”) AND (Analogy-based OR “Analogy based” OR “case based reasoning” OR “CBR”) AND (“feature subset selection” OR “feature selection” OR “feature weighting” OR “feature weights” OR “weight” OR “feature significance”).

3.2.2 *Literature sources*

Our search included important software engineering journals as per [20] and conferences which publish literature on software development effort estimation. The search also involved following up the references of included papers ‘backward snowballing’. We also looked at the papers that cited our included papers known as ‘forward snowballing’, which was accomplished by means of Google Scholar.

We did not place any restriction in terms of the start date of inclusion period. Any published paper that met our inclusion criteria was selected because our intention was to have a broad coverage since to our knowledge this was the first SLR on the topic. The feasibility of CBR in software effort estimation was carried out in the early 1990’s by Mukhopadhyay et al. [37] therefore we expected to find papers published starting from the mid 1990’s or early 2000’s to current date 2014.

Our initial search was performed on the online bibliographic library BESTweb[1] maintained by Simula Research Laboratory. BESTweb supplies journal and conference papers on software cost and effort estimation that have been classified according to research topic, estimation approach, research approach, study context and data set. The rest of the search involved searching five electronic databases (IEEE Xplore, ScienceDirect, Google

[1] (http://www.simula.no/BESTweb)

Scholar, ACM Digital Library and Springer). Some other important resources such as CiteSeer or DBLP were not considered, because they are almost covered by the selected five electronic databases.

The search terms discussed and constructed in section 3.2.1 were used to search for papers in the selected databases. The search covered title, abstract, and keywords.

3.2.3 *Search process*

A comprehensive search process of 'all' relevant sources is required for any SLR. Therefore to achieve this objective we divided the search into the following three search phases described in Figure 1.

> ***Search Phase 1***: Using the in-built filter we searched the online bibliographic library BESTweb database and located potential relevant papers.
>
> ***Search Phase 2***: Using search terms we searched the five electronic databases individually and then merged the located relevant papers with those from BESTweb forming a set of potential candidate papers
>
> ***Search Phase 3***: Searched the references of the included papers plus papers that cited the included papers in order to locate further papers.

Zotero[2], Microsoft® Excel and Dropbox[3] were used to manage and store search results. The search process and the number of papers identified at each phase are shown in detail in Figure 1.

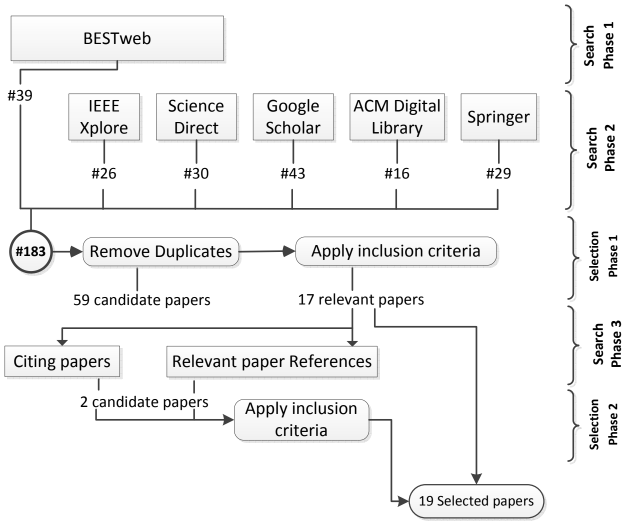

Figure 1: Search and selection process

3.3 Study selection and data extraction

From Search Phases 1 and 2 a total of 183 potential candidate papers for inclusion in the SLR were identified (see Figure 1). Since these papers were from independent sources they were checked to eliminate duplicates. A total of 59 unique papers were obtained after removing duplicates.

The next step involved reading the titles, abstracts, key words or full text of these 59 papers to select relevant papers based on the inclusion criteria defined in Table 1. From this 17 relevant studies were selected. Then the references of every included study were searched to identify other relevant studies that we might have been missed and we also examined citing papers. This effort led to the identification of two further studies, resulting in a total of 19 papers. The quality of the selected studies was linked to our inclusion criteria, where only papers from demonstrably rigorously refereed sources were included.

We created a master search form (see Table 2) to manage and keep record of all the candidate papers. The first four items were essentially for bibliometric and housekeeping purposes. The remaining categories of information are cross-referenced to our four research questions. The search form was also used for data extraction. We exploited the selected papers for data that contributed to addressing the research questions concerned in this review. The initial search and categorising was carried out by the first author and the second author's role was to check reliability via independent checking of the included papers. Any differences in inclusion or categorisation were resolved through discussion.

Table 2: The search form for data extraction

Search form fields	RQ(s) addressed
Article identifier	
Year of publication	
Source	
Article title	
Weighting Technique	*RQ1*
Technique framework	*RQ1*
Limitations and strengths	*RQ2*
Performance metrics	*RQ4*
Statistical testing?	*RQ4*
Datasets used	*RQ4*
Data quality/issues	*RQ4*
Performance	*RQ3*

3.4 Data synthesis

The goal of data synthesis is to identify different FWTs from the selected studies in order to address the research questions. The extracted data obtained consisted of both quantitative data (e.g., values of prediction accuracy or results) and qualitative data (e.g., strengths and weaknesses of FWTs). We used different strategies to synthesise the extracted data relating to different sets of research questions. The strategies are explained in detail as follows.

For the data relating to RQ1, the results were tabulated to represent the distribution of FWTs. The FWTs were categorised based on the dimensions from a review by Wetterschereck et al. [47]. Their dimensions (see Table 3) are as follows:

- *Bias*: refers to whether the weight learning utilises feedback from the performance algorithm [19].
- *Weight space*: is used to distinguish feature weighting from conventional feature subset selection algorithms.

[2] https://www.zotero.org/

[3] http://www.dropbox.com/

- The *Representation*: distinguishes algorithms that transform the given representation (to yield better results) from those that use the 'given' representation.
- *Generality*: is used to distinguish algorithms that assume weights differ among 'local' regions of the instance space form those that do not.
- *Knowledge*: highlights algorithms that use domain specific knowledge to determine feature weights.

In RQ2, the limitations of existing techniques were identified from analysis of the text.

Table 3: Dimensions For Distinguishing Feature Weighting Methods from [47]

Dimension	Possible Values
Bias	{Performance, Preset}
Weight Space	{Continuous, Binary}
Representation	{Given, Transformed}
Generality	{Global, Local}
Knowledge	{Poor, Intensive}

For RQ3 differing response variables and the absence of detailed results (e.g., for prediction performance indicators typically only measures of centre are given rather than variance) make it difficult to conduct a formal meta-analysis. For our purposes we use a simple vote counting procedure although we recognise this can be problematic as it may bias the results to no effect [16]. For this reason we restrict our analysis to the question of is there an effect as opposed to how large is the effect and make no judgement concerning statistical significance when vote counting.

For RQ4 we consider the accuracy indicators utilised such as MMRE and R-squared. the data sets used for validation and the experimental design in terms of benchmarks and so forth. As it turns out all the primary studies were based on experiments although in theory other forms of empirical study were not excluded from our review.

4. RESULTS

In this section we present and discuss the findings from our systematic review. We start by giving an overview of the selected studies. Then, we present a detailed description of the findings of this review for each research question. We also interpret the review results, not only within the context of the research questions, but also in a broader context related to the effort estimation.

We have identified 19 papers describing original primary studies to be included in this SLR. These papers were published during the time period 2000 - 2014. A total of 14 (74%) papers were published in journals and 5 (26%) papers appeared in conference proceedings, however we did not locate any papers from book chapters. The respective numbers, date published and relative sources of the selected studies are shown in Table 4. There is some evidence of the topic gaining momentum since six studies were published in the first half of this time period (2000-6) whereas a further 13 were published in the second half (2007-13), essentially doubling the rate of output.

In terms of the actual composition of the selected studies, we observe that all the studies are experimental in nature. No case studies or other forms of empirical research were located. All studies, but for one, used a minimum of one project data set from industry to validate the feature weighting techniques. Finally we reiterate that since the quality of the selected studies was linked to our inclusion criteria, where only papers from demonstrably refereed sources are included, we believe these represent good experimental and research practice.

RQ1: Range and diversity of FWTs

From the 19 selected studies, 12 distinct techniques for feature weighting were identified that have been applied to estimate software development effort to models using CBR. Other techniques are either variations on combinations on these 12 techniques. They are listed in table 5: It is clear that there is considerable diversity in approach with the majority of studies proposing and evaluating new techniques although the majority of techniques (13 out 19) adopt a general approach to feature weighting as opposed to a binary (included/excluded) view. Search heuristics are non-population searches such as hill climbing, therefore different from GA. We also note that some of the FWTs are combinations, obtained by combining two or more FWTs or by combining FWTs with non-FWTs. Overall we might characterise the area as one that is still in an early stage of development. Since we have a maximum of three observations (genetic algorithms) for any one FWT, our meta-analysis asks a more general question (RQ3). Namely do the techniques collectively offer improvement over not using a feature weighting regime?

Table 5: Range and diversity of FWTs

Technique	Studies	#
Genetic Algorithms (GA)	[S06], [S12],[S17]	3
Exhaustive Search	[S01], [S04],[S05]	3
Weighted Means	[S03], [S04]	2
Particle Swarm Optimisation	[S19],[S18]	2
Mantel's Correlation	[S07],[S10]	2
Principal Component Analysis	[S11],[S13]	2
Kernel Methods	[S15]	1
Heuristic Search (non-population)	[S02]	1
Fuzzy Logic	[S09]	1
Rough Sets Analysis (RSA)	[S08]	1
Grey Relational Analysis (GRA)	[S16]	1
Mutual Information	[S14]	1

RQ2: Strengths and limitations of FWTs

Our findings and discussions on strengths and limitations of FWTs are based on the framework dimensions [47]. The first three dimensions present inconclusive results while the last two dimensions present limitations. In general the greatest strength of FWTs is that they do not assume that features contribute equally to the output, therefore they assign different weights to the features. This potentially results in improved accuracy since theoretically the worst

Table 4: Selected studies

ID	Year	Study Author(s)	Source	Weighting Technique	Weight space	Ref
[S01]	2000	Mair et al.	Journal	Exhaustive Search	Binary	[33]
[S02]	2002	Kirsopp and Shepperd	Conference	Search Heuristics	Binary	[23]
[S03]	2003	Mendes et al.	Journal	Inverse Rank Weighted Mean	Continuous	[34]
[S04]	2004	Auer and Biffl	Conference	Exhaustive Dimension Weighting	Continuous	[5]
[S05]	2006	Auer et al.	Journal	Exhaustive Search	Continuous	[6]
[S06]	2006	Huang and Chiu	Journal	Genetic Algorithm	Continuous	[18]
[S07]	2007	Keung and Kitchenham	Conference	Mantel's Correlation	Continuous	[21]
[S08]	2008	Li and Ruhe	Journal	Rough Set Analysis	Continuous	[30]
[S09]	2008	Azzeh et al.	Conference	Fuzzy Logic	Binary	[7]
[S10]	2008	Keung et al.	Journal	Mantel's Correlation	Binary	[22]
[S11]	2009	Wen et al.	Conference	Principal Components Analysis (PCA)	Continuous	[46]
[S12]	2009	Li et al.	Journal	Genetic Algorithm	Continuous	[32]
[S13]	2009	Tosun et al.	Journal	PCA with Correlation Weighting	Continuous	[44]
[S14]	2009	Li et al.	Journal	Mutual Information	Binary	[31]
[S15]	2011	Kocaguneli et al.	Journal	Kernel Methods	Continuous	[27]
[S16]	2011	Song and Shepperd	Journal	Grey Relational Analysis	Binary	[43]
[S17]	2013	Bardsiri et al.	Journal	Genetic Algorithm	Continuous	[9]
[S18]	2013	Bardsiri et al.	Journal	Particle Swarm Optimisation	Continuous	[8]
[S19]	2013	Wu et al.	Journal	Combinations	Continuous	[48]

case is where all features do indeed merit equal weights which is a situation that should be discoverable by the FWT. The discussions on the first three dimensions are as follows:

- *Bias*: Most algorithms in the selected studies use performance bias methods therefore their search for feature weights is guided by the efficiency of the performance settings.
- *Weight space*: Thirteen of the 19 studies use continuous weight space and reported improved accuracy when compared with studies which used binary weight space. Therefore the use of continuous weight space by FWTs is a strength.
- *Representation*: All the algorithms in the selected studies transform the set of features used to represent the instances. Transforming the given representation before assigning weights assist to overcome insensitivity to correlated or interacting features. This may lead to improved accuracy [36].

While significant efforts have been invested in developing and improving FWTs, some limitations still exist which require research attention. These limitations are about the algorithms used, which in summary, can be described as follows:

- *Generality*: Despite the findings of the survey by Atkenson et al. [4] showing that assuming weights differ among 'local' regions of the instance space may improve results. All FWTs in selected studies use algorithms do not assume weights differ among 'local' regions of the instance space i.e. they use a single set of weights, and employ it globally (i.e., over the entire instance space). Studies such as [2, 3, 13, 15] also reached the same conclusion as Atkenson et al. [4], therefore based on these findings it could be suggested that FWTs may benefit from assuming weights differ among 'local' regions of the instance space.
- *Knowledge*: Several researchers such as [3, 41, 11, 38] and [10] demonstrated the use of domain knowledge to assign weights and that it may lead to improved accuracy. Unfortunately all feature weighting techniques in selected studies do not use domain specific knowledge to assign weights feature and this could be a limiting factor.

In summary, for researchers to efficiently estimate effort in software development using FWTs in CBR, there is need to urgently address these limitations.

RQ3: Estimation accuracy of FWTs

Having considered the range, diversity, strengths and limitations of the feature weighting techniques we now turn to the central and most actionable aspect of our systematic review. Do FWTs perform better than conventional EBA where all features have equal weights? In order to make this comparison we need studies that use conventional EBA as a benchmark. Fortunately 17 out of 19 studies do this (i.e. not S01 and S10) consequently we use these 17 primary studies for our meta-analysis (see Table 6).

Although the 17 studies all use a comparable benchmark each study has differences in their experimental design, choice of accuracy statistic and the level of reporting. As a result we adopt a simple vote-counting procedure. Only counting results that are statistically significant is known to be problematic and indeed the probability of making the correct decision tends to zero as the number of primary study results becomes large (see Hedges and Olkin [16, pp48-52] for a detailed discussion). Mindful of the potential problems of this procedure we follow the procedure recommended by the Cochrane Collaboration [17] ignore statistical significance and simply classify studies as supporting the intervention (using FWT), neutral (i.e., no difference) and negative (favouring constant feature weights).

Based on Table 6 we observe that 16 out of 17 studies report a positive effect. As a formality one could use a one-sample sign test which rejects the null hypothesis of no effect ($p = 0.0003$). Thus despite our reservations about our meta-analysis procedure of vote counting there is clearly a strong result and we can be reasonably confident that FWTs have the effect of reducing prediction error for software effort when using CBR techniques.

The above analysis does not differentiate between differing classes of FWT. Examining our selected studies more closely we see that the two most popular accuracy metrics are MMRE (Mean Magnitude of Relative Error) and Pred(25) (Percentage of predictions that are within 25% of the actual value). We also note that the Desharnais data set is the most widely utilised. Therefore to make comparisons between the differing FWTs we use the subset of eight primary studies that utilise the same accuracy metrics on the same data set (see Table 7). For some basic reference we also give the original results reported in Shepperd and Schofield [40] although we would caution against over-interpretation of the results. First differing procedures are used and the procedures for the exploration of the number of neighbours to use (k) also vary. Finally, as has been extensively discussed elsewhere, we lack confidence in MMRE and Pred(25) as unbiased measures of prediction performance [26]. In addition, using multiple measures can yield contradictory results, so for example, the GAs in study S12 are ranked 3rd for MMRE and worst (9th) for Pred(25). This may be explained by the choice of objective function for the GA.

Table 7: Performance of non-binary weight-space FTWs on Desharnais data set

Study	Technique	Criteria		
		MMRE(%)	Pred(25)%	k
[40]	Benchmark - no FWT	64	36	1-3
[S04]	Exhaustive Dimension Weighting	30	50	1-5
[S07]	Mantel's Correlation	34.5	49.5	1-5
[S12]	Genetic Algorithm	43	32	1-5
[S13]	Principal Components Analysis	46	51	1-5
[S17]	Genetic Algorithm	46	48	1-5
[S05]	Exhaustive Search	48.7	52.6	1-5
[S08]	Rough Set Analysis	59	42	1-4
[S11]	PCA with Correlation Weighting	64	36	1-5

Notwithstanding the above reservations it would seem that the FWTs generally outperform the benchmark and unsurprisingly those based on exhaustive search tend to do best. The latter observation suggests that it will be fruitful to focus on metaheuristic search as a way of finding good approximations of the optima whenever computational considerations militate against exhaustive search.

RQ4: Approaches to experimental evaluation

Table 8 shows evaluation methods used by studies in this review. Jackknifing, sometimes known as Leave-One-Out Cross-Validation, and n-fold Cross-Validation ($n > 1$) are the two most used validation methods in the selected studies. The numbers of the studies that have used these validation methods are as follows: 10 (53%) for Jackknifing, 6 (32%) for n-fold Cross-Validation, and 3 (15%) for other validation methods. MMRE (Mean Magnitude of Relative Error) and Pred(25) (Percentage of predictions that are within 25% of the actual value). We also note that the Desharnais data set is the most widely utilised. MdMRE (Median Magnitude of Relative Error) is also a relatively popular performance metric. The numbers(%) of the studies that used these metrics are as follows: 17 (89%) for MMRE, 17 (89%) for Pred(25), and 7 (37%) for MdMRE.

Unfortunately many researchers still regard the choice of what is the experimental response variable either as a matter of personal preference or adopt a basket approach. As has been shown with respect to RQ3 differing choices can lead to rank or preference reversals and consequently we need to appeal to theory. Absolute residuals have the property of being unbiased and measures relative to a naïve or guessing strategy are most informative (e.g., Relative Accuracy [39]).

It is also interesting to note from Table 8 that statistical testing, that is the use of inferential statistical procedures to determine the significance of the result is by no means universal with less than half (7 out of 19) of the studies following this procedure. Whilst we would argue in favour of some formal statistical evaluation we encourage the adoption of more modern approaches based upon effect sizes [39], though no study in our review used such an approach.

Table 9: Popular datasets used for FWTs construction and validation

Data set	Type	Studies(%)	Features (p)	Size (n)
Desharnais	W	15 (79%)	10	81
Kemerer	W	6 (32%)	7	15
Albrecht	W	6 (32%)	7	24
ISBSG	C	4 (21%)	many	>1000
COCOMO	C	4 (21%)	17	63
Finnish	C	2 (11%)	8	38
Maxwell	C	2 (11%)	27	62
Miyazaki	W	2 (11%)	8	48
NASA	W	2 (11%)	17	93

The abbreviations used are: C = cross-company, W = within-company

Table 9 shows the diversity of data sets that have been employed by at least two primary studies. Interestingly the Desharnais, Kemerer, Albrecht and COCOMO data set dominate although these are amongst the oldest in some cases in excess of 35 years. Given the rapid pace of change in software technology we as a community do need to consider how appropriate this is. The data set size in terms of the number of features also strongly impacts the computational demands on the various FWTs with exhaustive search being infeasible for situations where p is much greater than ten.

Table 6: Performance of FWTs against EBA benchmark

Study	Feature Weighting Technique	Statistical Testing	Benchmarking	Improvement wrt EBA
[S01]	Exhaustive Search	No	No	n.a.
[S02]	Search Heuristics	Yes	Yes	Yes
[S03]	Inverse Rank Weighted Mean	Yes	Yes	Yes
[S04]	Exhaustive Dimension Weighting	No	Yes	Yes
[S05]	Exhaustive Search	No	Yes	Yes
[S06]	Genetic Algorithm	No	Yes	Yes
[S07]	Mantel's Correlation	Yes	Yes	Yes
[S08]	Rough Set Analysis	No	Yes	Yes
[S09]	Fuzzy Logic	No	Yes	Yes
[S10]	Mantel's Correlation	Yes	No	n.a.
[S11]	Principal Components Analysis (PCA)	Yes	Yes	Yes
[S12]	Genetic Algorithm	No	Yes	Yes
[S13]	PCA with Correlation Weighting	No	Yes	Yes
[S14]	Mutual Information	No	Yes	Yes
[S15]	Kernel Methods	Yes	Yes	No
[S16]	Grey Relational Analysis	No	Yes	Yes
[S17]	Genetic Algorithm	No	Yes	Yes
[S18]	Particle Swarm Optimisation	No	Yes	Yes
[S19]	Combinations	Yes	Yes	Yes

5. DISCUSSION AND CONCLUSIONS

In this systematic review we located 19 primary studies published since 2000 that have explored the application of feature weighting techniques to enhance the predictive performance of case-based reasoning for software project effort.

Our main findings are:

- A wide range of techniques are being proposed and there have been relatively few replications.
- There is a lack of case studies, action research or other detailed studies from industry which may imply that the application of FWTs in CBR practice is still immature.
- There is a great diversity in both the conduct and reporting of experimental validations. This hinders our ability to make sense of and compare results through formal meta-analysis. Specifically we were obliged to resort to vote counting and a one-sample sign test.
- The strongest and actionable result is that despite some methodological reservations concerning our meta-analysis it is clear that FWTs are collectively valuable. This is in line with findings from many other areas and problem domains of machine learning [14].
- The approaches are exclusively algorithmic so expert judgement is not used for feature weighting in CBR either as a standalone technique or to augment other techniques.
- There is a tendency to keep advocating new techniques and it may now be useful to consider replication and more benchmarking of existing techniques. This will be facilitated if researchers give more consideration to the details of reporting and possibly as a community we consider specific reporting protocols.
- As a research community we should determine whether it remains fruitful to continue to use the older data sets and whether or not they are now obsolete.

As with any systematic review there are limitations:

- We may not have located all relevant primary studies, although we believe the blend of hand search and automation should be effective.
- The relatively small number of papers included in the review limits our ability to conduct the meta-analysis, in particular to make comparisons between specific FWTs.
- We have had to rely on the researchers' choices of accuracy indicator e.g., MMRE, Pred(25) despite our reservations about bias and exactly what is being captured. In addition, studies have tended to report measures of central tendency rather than variance or spread so we cannot estimate effect sizes or confidence limits for particular results.

Despite these potential limitations, we consider the most striking and actionable finding of our review to be that feature weighting techniques are consistently beneficial. We believe this is useful information for both practitioners and researchers.

6. REFERENCES

[1] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. Mahrin. A systematic literature review of software

Table 8: Evaluation methods used by FWTs

Study	Performance metrics	Cross-validation	Statistical Testing	Benchmarking
[S01]	MMRE	Jackknifing	No	No
[S02]	Mean Absolute Residuals	Jackknifing	Yes	Yes
[S03]	MMRE, MdMRE, Pred(n)	Jackknifing	Yes	Yes
[S04]	MMRE, Pred(n)	Jackknifing	No	Yes
[S05]	MMRE, Pred(n), and Var	Jackknifing	No	Yes
[S06]	MMRE, Pred(n), MdMRE	n-fold	No	Yes
[S07]	MMRE, Pred(n)	Jackknifing	Yes	Yes
[S08]	MMRE, Pred(n)	Jackknifing	No	Yes
[S09]	MMRE, MdMRE, Pred(n)	n-fold	No	Yes
[S10]	MMRE, Pred(n)	Jackknifing	Yes	No
[S11]	MMRE, Pred(n)	Jackknifing	Yes	Yes
[S12]	MMRE, Pred(n), MdMRE	Other	No	Yes
[S13]	MMRE, Pred(n)	Other	No	Yes
[S14]	MMRE, MdMRE, Pred(n)	n-fold	No	Yes
[S15]	MdMRE, MAR, Pred(n)	Other	Yes	Yes
[S16]	MMRE, Pred(n)	Jackknifing	No	Yes
[S17]	MMRE, Pred(n)	n-fold	No	Yes
[S18]	MMRE, Pred(n)	n-fold	No	Yes
[S19]	MMRE, MdMRE, Pred(n)	n-fold	Yes	Yes

requirements prioritization research. *Information and Software Technology*, 56(6):568–585, 2014.

[2] D. W. Aha and R. L. Goldstone. Concept learning and flexible weighting. In *Proceedings of the fourteenth annual conference of the Cognitive Science Society*, pages 534–539. Citeseer, 1992.

[3] K. D. Ashley and E. L. Rissland. Waiting on weighting: A symbolic least commitment approach. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 239–244, 1988.

[4] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning for control. In *Artificial Intelligence Review*, pages 75–113. Springer, 1997.

[5] M. Auer and S. Biffl. Increasing the accuracy and reliability of analogy-based cost estimation with extensive project feature dimension weighting. In *Empirical Software Engineering, 2004. ISESE'04. Proceedings. 2004 International Symposium on*, pages 147–155. IEEE, 2004.

[6] M. Auer, A. Trendowicz, B. Graser, E. Haunschmid, and S. Biffl. Optimal project feature weights in analogy-based cost estimation: Improvement and limitations. *Software Engineering, IEEE Transactions on*, 32(2):83–92, 2006.

[7] M. Azzeh, D. Neagu, and P. Cowling. Improving analogy software effort estimation using fuzzy feature subset selection algorithm. In *Proceedings of the 4th international workshop on Predictor models in software engineering*, pages 71–78. ACM, 2008.

[8] V. K. Bardsiri, D. N. A. Jawawi, S. Z. M. Hashim, and E. Khatibi. A pso-based model to increase the accuracy of software development effort estimation. *Software Quality Journal*, 21(3):501–526, 2013.

[9] V. K. Bardsiri, A. Khatibi, and E. Khatibi. An optimization-based method to increase the accuracy of software development effort estimation. *Journal of Basic and Applied Scientific Research*, 2013.

[10] R. Bareiss. The experimental evaluation of a case-based learning apprentice. In *Proc. of the 2 nd Workshop on Case-Based Reasoning*, pages 162–167, 1989.

[11] T. Cain, M. J. Pazzani, and G. Silverstein. Using domain knowledge to influence similarity judgements. In *Proceedings of the Case-Based Reasoning Workshop*, pages 191–198, 1991.

[12] T. Foss, E. Stensrud, B. Kitchenham, and I. Myrtveit. A simulation study of the model evaluation criterion MMRE. *Software Engineering, IEEE Transactions on*, 29(11):985–995, 2003.

[13] J. H. Friedman. Flexible metric nearest neighbor classification. *Unpublished manuscript available by anonymous FTP from playfair. stanford. edu (see pub/friedman/README)*, 1994.

[14] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[15] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(6):607–616, 1996.

[16] L. Hedges and I. Olkin. *Statistical methods for meta-analysis.* Academic Press, London, 1985.

[17] J. Higgins and S. Green. *Cochrane Handbook for Systematic Reviews of Interventions: Version 5.1.0*

[updated March 2011]. The Cochrane Collaboration, 2011.

[18] S.-J. Huang and N.-H. Chiu. Optimization of analogy weights by genetic algorithm for software effort estimation. *Information and software technology*, 48(11):1034–1045, 2006.

[19] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings 11th International Conference on Machine Learning*, volume 94, pages 121–129, 1994.

[20] M. Jørgensen and M. Shepperd. A systematic review of software development cost estimation studies. *IEEE Transactions on Software Engineering*, 33(1):33–53, 2007.

[21] J. W. Keung and B. Kitchenham. Optimising project feature weights for analogy-based software cost estimation using the mantel correlation. In *Software Engineering Conference, 2007. APSEC 2007. 14th Asia-Pacific*, pages 222–229. IEEE, 2007.

[22] J. W. Keung, B. A. Kitchenham, and D. R. Jeffery. Analogy-x: providing statistical inference to analogy-based software cost estimation. *Software Engineering, IEEE Transactions on*, 34(4):471–484, 2008.

[23] C. Kirsopp, M. Shepperd, and J. Hart. Search heuristics, case-based reasoning and software project effort prediction. In *GECCO 2002: Genetic and Evolutionary Computation Conf.* AAAI, 2002.

[24] B. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering, version 2.3. Report EBSE Technical Report EBSE-2007-01., Keele University, UK, 2007.

[25] B. Kitchenham, T. Dybå, and M. Jørgensen. Evidence-based software engineering. In *27th IEEE Intl. Softw. Eng. Conf. (ICSE 2004)*. IEEE Computer Society, 2004.

[26] B. Kitchenham, S. MacDonell, L. Pickard, and M. Shepperd. What accuracy statistics really measure. *IEE Proceedings - Software Engineering*, 148(3):81–85, 2001.

[27] E. Kocaguneli, T. Menzies, and J. W. Keung. Kernel methods for software effort estimation. *Empirical Software Engineering*, 18(1):1–24, 2013.

[28] R. Kohavi and G. John. Wrappers for feature selection for machine learning. *Artificial Intelligence*, 97:273–324, 1997.

[29] J. Kolodner. *Case-Based Reasoning.* Morgan-Kaufmann, 1993.

[30] J. Li and G. Ruhe. Analysis of attribute weighting heuristics for analogy-based software effort estimation method aqua+. *Empirical Software Engineering*, 13(1):63–96, 2008.

[31] Y. Li, M. Xie, and T. Goh. A study of mutual information based feature selection for case based reasoning in software cost estimation. *Expert Systems with Applications*, 36(3):5921–5931, 2009.

[32] Y.-F. Li, M. Xie, and T. N. Goh. A study of project selection and feature weighting for analogy based software cost estimation. *Journal of Systems and Software*, 82(2):241–252, 2009.

[33] C. Mair, G. Kadoda, M. Lefley, K. Phalp, C. Schofield, M. Shepperd, and S. Webster. An investigation of machine learning based prediction systems. *Journal of Systems and Software*, 53(1):23–29, 2000.

[34] E. Mendes, I. Watson, C. Triggs, N. Mosley, and S. Counsell. A comparative study of cost estimation models for web hypermedia applications. *Empirical Software Engineering*, 8(2):163–196, 2003.

[35] N. Mittas, M. Athanasiades, and L. Angelis. Improving analogy-based software cost estimation by a resampling method. *Information & Software Technology*, 50(3):221–230, 2008.

[36] T. Mohri and H. Tanaka. An optimal weighting criterion of case indexing for both numeric and symbolic attributes. In *AAAI-94 Workshop Program: Case-Based Reasoning, Working Notes*, pages 123–127, 1994.

[37] T. Mukhopadhyay, S. S. Vicinanza, and M. J. Prietula. Examining the feasibility of a case-based reasoning model for software effort estimation. *MIS Quarterly*, pages 155–171, 1992.

[38] B. W. Porter, R. Bareiss, and R. C. Holte. Concept learning and heuristic classification in weak-theory domains. *Artificial Intelligence*, 45(1):229–263, 1990.

[39] M. Shepperd and S. MacDonell. Evaluating prediction systems in software project estimation. *Information and Software Technology*, 54(8):820–827, 2012.

[40] M. Shepperd and C. Schofield. Estimating software project effort using analogies. *IEEE Transactions on Software Engineering*, 23:736–743, 1997.

[41] D. Skalak. Representing cases as knowledge sources that apply local similarity metrics. In *Proc. of the 14th Annual Conference of the Cognitive Science Society*, pages 325–330, 1992.

[42] D. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *11th Intl. Machine Learning Conf. (ICML-94)*, pages 293–301. Morgan Kauffmann, 1994.

[43] Q. Song and M. Shepperd. Predicting software project effort: A grey relational analysis based method. *Expert Systems with Applications*, 38(6):7302–7316, 2011.

[44] A. Tosun, B. Turhan, and A. B. Bener. Feature weighting heuristics for analogy-based effort estimation models. *Expert Systems with Applications*, 36(7):10325–10333, 2009.

[45] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang. Systematic literature review of machine learning based software development effort estimation models. *Information and Software Technology*, 54:41–59, 2012.

[46] J. Wen, S. Li, and L. Tang. Improve analogy-based software effort estimation using principal components analysis and correlation weighting. In *Software Engineering Conference, 2009. APSEC'09. Asia-Pacific*, pages 179–186. IEEE, 2009.

[47] D. Wettschereck, D. W. Aha, and T. Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 11(1-5):273–314, 1997.

[48] D. Wu, J. Li, and Y. Liang. Linear combination of multiple case-based reasoning with optimized weight for software effort estimation. *The Journal of Supercomputing*, 64(3):898–918, 2013.