Exposé zur Bachelorarbeit: Aufwandsschätzungen von Softwareprojekten durch Random Forest

René Burghardt

10. Dezember 2018

Inhalt

1 Einführung

Aufwandsschätzungen sind ein bedeutsamer, oftmals sogar entscheidender Part in der Umsetzung von Softwareprojekten, da Erfolg bzw. Nichterfolg eines Projekts direkt von der Akkuratheit der Schätzung abhängen kann. So führt eine zu niedrige Schätzung zu Engpässen bezüglich des Zeitplans und/oder des Budgets und eine zu hohe Schätzung beeinträchtigt die Effizienz und Wettbewerbsfähigkeit einer Organisation, da durch die zu hoch angesetzten Kosten die Umsetzung abgelehnt oder anders vergeben wird oder Vertrauen in die Schätzungsfähigkeit bzw. die gelieferte Qualität des Unternehmens verloren geht, wenn plötzlich deutlich weniger Arbeit anfällt.

Dementsprechend wird schon seit Jahrzehnten zu dieser Thematik geforscht und laufend neue Ergebnisse vorgestellt [1]. Ähnlich wie in vielen Feldern, kam und kommt dabei verstärkt maschinelles Lernen zum Einsatz. Und tatsächlich hat es sich oftmals als vorteilhaft erwiesen [2]. Dennoch bietet maschinelles Lernen im Bereich der Unterstützung bei Aufwandsschätzungen für Softwareprojekte immernoch ein großes Forschungspotenzial.

2 Ziel

Dementsprechend soll das Ziel dieser Arbeit die Entwicklung und Evaluierung einer neuen Technik zur Erstellung von Aufwandsschätzungen von Softwareprojekten mithilfe von machinellem Lernen sein.

3 Lösungsweg

Der Weg zur Lösung des oben formulierten Zieles lässt sich in folgende Abschnitte unterteilen:

3.1 Pre-processing der Daten

In [3] ist eine zweistellige Anzahl öffentlich verfügbarer Datensätze von Softwareprojekten mit unterschiedlichen Attributen aufgelistet. Interessante und zuletzt in Forschungen benutzte Datensätze sind dabei NASA [4], Maxwell [5], Desharnais [6] und Cosmic [7].

	# Projekte	# Attribute	Jahr
NASA	93	17	1987
Desharnais	81	11	1988
Maxwell	62	27	2002
Cosmic	42	22	2012

Zuallererst müssen diese Datensätze für die Verarbeitung und Auswertung aufbereitet werden.

3.2 Implementierung und Optimierung

In [8] ergab Random Forest verheißungsvolle Ergebnisse. Hier soll angeknüpft werden und eine Erweiterung dieser ML-Technik implementiert und optimiert werden.

Dabei sollen die Attribute aus den in 3.1 genannten Datensätzen zu Nutze gemacht werden. Als Eingabe dient dann eine (Teil-)Menge der jeweils vorliegenden Attribute und die Ausgabe soll in Personenstunden erfolgen.

Da Random Forest normalerweise ein Klassifikationsverfahren ist, die Ausgabe in Personenstunden aber nicht kategorial geschehen soll, sollen Model Trees als Decision Trees des Random Forests zum Einsatz kommen, an deren Blätter sich (lineare) Model befinden.

3.3 Evaluierung und Vergleich

Die Ergebnisse aus 3.2 sollen zum einen mit einer Teilmenge der in 3.1 genannten Datensätze auf Akkuratheit evaluiert werden und zum anderen sollen die Ergebnisse aus einem Datensatz über die jeweils anderen Datensätze evaluiert werden um eine Aussage über die Abhängigkeit von den Daten treffen zu können. Da das Mapping der Attribute nicht ohne Lücken machbar sein wird, soll hier in solchen Fällen stimmige Daten zum Auffüllen generiert werden.

Um das Potenzial der Methode einschätzen zu können, soll außerdem mit anderen schon auf Aufwandsschätzungen von Softwareprojekten angewandte ML-Techniken wie beispielsweise Artifical Neural Networks [9] oder eben die Technik mit Random Forests aus [8] verglichen werden.

4 Verwandte Arbeiten

Wie oben schon beschrieben hat sich die Forschung im Bereich der softwaregestützten Aufwandsschätzung für Softwareprojekte mit Arbeiten wie [10], [11], [9] und [12] in den letzten Jahrzehnten einen starken Auftrieb durch das Feld des Maschinellen Lernens bekommen.

Die veröffentlichten Ergebnisse wurden in [13] zusammengetragen und betrachtet bzw. wurde der aktuellste Stand und Trends in [3] bzw. [1] nochmal zusammengefasst.

Der Einsatz von Decision Trees für Aufwandsschätzungen von Softwareprojekten hat in den Arbeiten [14], [15] und [16] Platz gefunden, wobei gerade [16] hervorzuheben ist, wo die Ergebnisse durch Model Trees sehr vielversprechend sind.

Innerhalb der letzten zurückliegenden Jahre geriet dann Random Forest in den Fokus. zu nennen sind hier [17], [8] und [18]. [19] behandelt einen Vergleich zwischen Decision Trees und Forest Models, welcher zu Gunsten der Forest Models ausgeht.

5 Referenzen

[1] Sehra, Sumeet Kaur and Brar, Yadwinder and Kaur, Navdeep and Sehra, Sukhjit, 2017, Research Patterns and Trends in Software Effort Estimation, Information and Software Technology.

[2] Baskeles, Bilge and Turhan, Burak and Bener, Ayse, 2007, Software effort estimation using machine learning methods, Computer and information sciences, 2007. iscis 2007. 22nd international symposium on.

[3] Gautam, Swarnima Singh and Singh, Vrijendra, 2017, The state-of-the-art in software development effort estimation, Table 3: Overview of publicly available SDEE data sets, Journal of Software: Evolution and Process.

[4] Menzies, Tim, 2006, COCOMO NASA 2 / Software cost estimation, http://promise.site.uottawa.ca/SERepository/datasets-page.html.

[5] KD, Maxwell, 2002, Applied Statistics for Software Managers, https://zenodo.org/record/268461.XApor9tKiUl.

[6] Desharnais, Jean-Marc, 1988, Analyse Statistique de la Productivite des Projets de Developpement en Informatique a Partir de la Technique des Points de Fonction, Desharnais Software Cost Estimation, http://promise.site.uottawa.ca/SERepository/datasets-page.html.

[7] International Software Benchmarking Standards Group, https://zenodo.org/record/268482.XApqBttKiUl.

[8] Zakrani, Abdelali and Hain, Mustapha and Namir, Abdelwahed, 2018, Software Development Effort Estimation Using Random Forests: An Empirical Study and Evaluation, International Journal of Intelligent Engineering and Systems.

[9] Carolyn Mair and Gada F. Kadoda and Martin Lefley and Keith Phalp and Chris Schofield and Martin J. Shepperd and Steve Webster, 2000, An investigation of machine learning based prediction systems, Journal of Systems and Software, Volume 53, Pages 23-29.

[10] Srinivasan, Krishnamoorthy and Fisher, Douglas, 1995, Machine learning approaches to estimating software development effort, IEEE Transactions on Software Engineering, Volume 21, Pages 126-137.

[11] Shepperd, Martin and Schofield, Chris, 1997, Estimating software project effort using analogies, IEEE Transactions on software engineering, Volume 23, Pages 736-743.

[12] Radlinski, Lukasz and Hoffmann, Wladyslaw, 2010, On predicting software development effort using machine learning techniques and local data, International Journal of Software Engineering and Computing, Volume 2, Pages 123-136.

[13] Wen, Jianfeng and Li, Shixian and Lin, Zhiyong and Hu, Yong and Huang, Changqin, Systematic literature review of machine learning based software development effort estimation models, 2012, Information and Software Technology, Volume 54, Pages 41-59.

[14] W. Selby, Richard and Porter, Adam, 1988, Learning from Examples: Generation and Evaluation of Decision Trees for Software Resource Analysis, IEEE Transactions on Software Engineering, Volume 14, Pages 1743-1757.

[15] Andreou, Andreas S and Papatheocharous, Efi, 2008, Software cost estimation using fuzzy decision trees, Proceedings of the 2008 23rd IEEE/ACM international conference on automated software engineering, Pages 371-374.

[16] Azzeh, Mohammad, 2011, Software effort estimation based on optimized model tree, Proceedings of the 7th International Conference on Predictive Models in Software Engineering, Page 6.

[17] Satapathy, Shashank Mouli and Acharya, Barada Prasanna and Rath, Santanu, 2016, Early Stage Software Effort Estimation Using Random Forest Technique Based on Use Case Points, IET Software, Volume 10.

[18] Banimustafa, Ahmed, 2018, Predicting Software Effort Estimation Using Machine Learning Techniques.

[19] A. B. Nassif and M. Azzeh and L. F. Capretz and D. Ho, 2013, A comparison between decision trees and decision tree forest models for software development effort estimation, 2013 Third International Conference on Communications and Information Technology (ICCIT).