New Conceptual Cohesion Metrics: Assessment for Software Defect Prediction

SYNASC 2021

Miholca Diana-Lucia

December 2021

Outline

Introduction and motivation

Software cohesion. Definition & motivation.

Definition

- software **cohesion** = the extent of relatedness among a software entity's components

Motivation

- low coupling + high cohesion $\Rightarrow \uparrow$ software quality
- proposing new OO cohesion measures is an

$$\left\{\begin{array}{l}\text{emergent [15]}\\ \text{necessary [13]}\\ \text{promising [1, 4, 18, 12]}\end{array}\right. \quad \text{research concern}$$

Software defects prediction. Definition & motivation.

Definition

- Software Defects Prediction (SDP) = identifying defective software components

Motivation

$$\left.\begin{array}{l}\text{measures project evolution}\\ \text{supports process management}\\ \text{streamlines testing}\\ \text{guides code review}\end{array}\right\} \Rightarrow \downarrow \text{cost}$$

Software cohesion in relation to software defects

Motivation

- software defects $\Leftarrow$ poor software quality $\supset$ poor design
- software cohesion $\Leftrightarrow$ software design quality

$\Rightarrow$ software cohesion $\Rightarrow$ design flaws $\Rightarrow$ software defects [8]

Why conceptual cohesion?

- Cohesion is generally computed based on structural information
  $\Rightarrow$ **structural** cohesion

Motivation

- the most desirable form of cohesion is **conceptual** cohesion [5]: the degree to which a class represents an unique and semantically meaningful concept
- there are few conceptual cohesion metrics in the literature

Related work

Lack of Conceptual Cohesion in Methods (LCSM) [10]

Conceptual Cohesion of Classes (C3) [11]

Conceptual Lack of Cohesion on Methods (CLCOM5) [18]

- LSI

- Logical Relatedness of Methods (LORM) [6]
  - knowledge-based system
- Maximal Weighted Entropy (MWE)
  - Latent Dirichlet Allocation (LDA)

The proposed conceptual cohesion metrics

The proposed metrics

Learning conceptual Vectors →

V The source code of each method m_{ij} of a class c_i is transformed into a l-dimensional conceptual vector vector $(m_{ij1}, m_{ij2}, \cdots, m_{ijl})$, by using **Doc2Vec** [9]

- a MLP based prediction model proposed by Le and Mikolov [9]
- shown in the literature to better capture the semantics than statistical, count-based information retrieval methods

The proposed metrics

Learning conceptual Vectors → Conceptual Similarity of methods →

S The conceptual similarity between methods is computed using: • **euclidean** and • **cosine** similarities

D The **Conceptual Similarity between two Methods (COSM)** m_{ij} and m_{ik} is defined as the *similarity* between their conceptual vectors $(m_{ij1}, m_{ij2}, \cdots, m_{ijl})$ and $(m_{ik1}, m_{ik2}, \cdots, m_{ikl})$:

$$COSM^{cos}(m_{ij}, m_{ik}) = \frac{\left| \sum_{p=1}^{l} (m_{ijp} \cdot m_{ikp}) \right|}{\sqrt{\sum_{p=1}^{l} (m_{ijp} \cdot m_{ijp})} \cdot \sqrt{\sum_{p=1}^{l} (m_{ikp} \cdot m_{ikp})}}$$

$$COSM^{euc}(m_{ij}, m_{ik}) = \frac{1}{1 + \sqrt{\sum_{p=1}^{l} (m_{ijp} - m_{ikp})^2}}$$

D The **Average Conceptual Similarity of Methods (ACOSM) in a class** c_i is defined as:

$$ACOSM^{cos/euc}(c_i) = \frac{\sum_{p=1}^{\binom{nm_i}{2}} COSM^{cos/euc}(m_{ij}, m_{ik})}{\binom{nm_i}{2}}$$

The proposed metrics

Learning conceptual Vectors → Conceptual Similarity of methods → Conceptual Cohesion of classes

- The **Conceptual Cohesion of Classes (COCC)** c_i is defined as:

$$COCC^{cos/euc}(c_i) = \begin{cases} ACOSM^{cos/euc}(c_i), ACOSM^{cos/euc}(c_i) > 0 \\ 0, \textit{otherwise} \end{cases}$$

- **Lack of Conceptual Similarity between Methods (LCOSM)** $\cong$ LCSM [10]

Theoretical validation

✓ COCC and LCOSM comply the top three most important [10] mathematical properties of class cohesion metrics, as defined by Briand et al. [2]:

- ✓ *non-negativity*
- ✓ *normalization*
- ✓ *null value*.

Assessment of the proposed cohesion metrics

Experimental case studies

Experimental data

Software system	Number of defective classes	Number of non-defective classes	Percentage of non-defective classes
Ant	166	575	22.4%
Tomcat	77	726	9.6%
JEdit	48	307	13.5%

Case studies

1. First case study
   - to show that COCC & LCOSM capture additional aspects of coupling when compared to existing cohesion metrics
2. Second case study
   - to evaluate COCC & LCOSM vs. existing cohesion metrics for SDP

First case study - Correlation analysis

Preexisting cohesion metrics considered:

Structural metrics:

- LCOM1 [3], LCOM2 [3], LCOM3 [7], LCOM4 [7], LCOM5 [16]
  - have been extensively studied in the literature [1, 11]
- YALCOM [14]
  - the state-of-the-art variant of LCOM

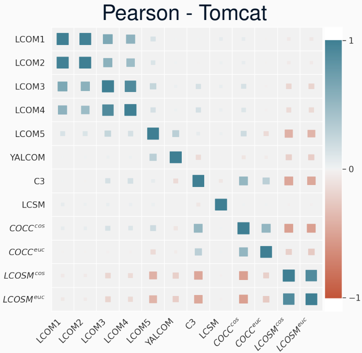
Conceptual metrics:

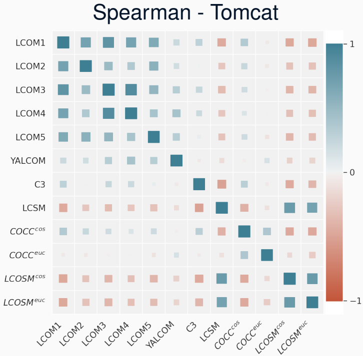
- C3 and LCSM [10]
  - defined using LSI, cosine similarity only
  - LCSM is not normalized

Computed correlation coefficients:

- Pearson
- Spearman

First case study - Correlation analysis - Results





⇒ Predominantly negligible, low or moderate correlations with LCOM1-5, YALCOM, C3 and LCSM

Second case study - Difficulty analysis

The *difficulty* [17] of a SDP data set = the ratio of defective instances for which the nearest neighbor is non-defective.

SDP data sets' difficulty:

Cohesion metrics considered as input features for SDP	Ant	Tomcat	JEdit
{C3}	0.807	0.883	0.896
{$COCC^{cos}$}	0.741	0.804	0.750
{C3, LCSM}	0.801	0.883	0.896
{$COCC^{cos}$, $LCOSM^{cos}$}	0.729	0.804	0.750
{$COCC^{cos}$, $COCC^{euc}$, $LCOSM^{cos}$, $LCOSM^{euc}$}	0.735	0.792	**0.667**
{C3, LCSM, $COCC^{cos}$, $LCOSM^{cos}$}	0.747	0.740	0.708
{C3, LCSM, $COCC^{cos}$, $COCC^{euc}$, $LCOSM^{cos}$, $LCOSM^{euc}$}	**0.663**	**0.701**	0.792

COCC and LCOSM facilitate SDP by reducing the difficulty of distinguishing the defective classes from the others.

Second case study - Supervised SDP analysis

ML models employed:

- k-Nearest Neighbors (kNN)
- Random Forest (RF)

Evaluation methodology:

- leave-one-out (LOO)
- Area under the ROC curve (AUC)

Second case study - Supervised SDP analysis - Results

- AUC values obtained using kNN:

Cohesion metrics considered as input features for SDP	Ant	Tomcat	JEdit
{C3}	0.571	0.624	0.519
{$COCC^{cos}$}	0.644	0.620	0.725
{C3, LCSM}	0.601	0.631	0.517
{$COCC^{cos}$, $LCOSM^{cos}$}	0.656	0.622	0.729
{$COCC^{cos}$, $COCC^{euc}$, $LCOSM^{cos}$, $LCOSM^{euc}$}	**0.758**	0.714	**0.762**
{C3, LCSM, $COCC^{cos}$, $LCOSM^{cos}$}	0.673	0.702	0.740
{C3, LCSM, $COCC^{cos}$, $COCC^{euc}$, $LCOSM^{cos}$, $LCOSM^{euc}$}	0.688	**0.740**	**0.762**

- AUC values obtained using RF:

Cohesion metrics considered as input features for SDP	Ant	Tomcat	JEdit
{C3}	0.514	0.524	0.507
{$COCC^{cos}$}	0.592	0.627	0.639
{C3, LCSM}	0.552	0.523	0.493
{$COCC^{cos}$, $LCOSM^{cos}$}	0.587	0.631	0.591
{$COCC^{cos}$, $COCC^{euc}$, $LCOSM^{cos}$, $LCOSM^{euc}$}	**0.728**	**0.718**	0.705
{C3, LCSM, $COCC^{cos}$, $LCOSM^{cos}$}	0.624	0.686	0.700
{C3, LCSM, $COCC^{cos}$, $COCC^{euc}$, $LCOSM^{cos}$, $LCOSM^{euc}$}	0.659	0.701	**0.711**

Conclusions and future work

Conclusions and future work

- Conclusions
  - a new set of Doc2Vec based metrics for expressing the conceptual cohesion of classes in OO systems
    - ✓ able to capture additional dimensions of cohesion and to be better software defect predictors

Future work directions

- extend the empirical assessment
- define aggregated cohesion metrics
- develope a new extensive metrics suite for SDP
  - aggregated coupling + aggregated cohesion

COHESION

CONCEPTUAL

COCC

THANK YOU!

References i

Al Dallal, J., and Briand, L. C.
A precise method-method interaction-based cohesion metric for object-oriented classes.
ACM Trans. Softw. Eng. Methodol. 21, 2 (Mar. 2012).

Briand, L., Morasca, S., and Basili, V.
Property-based software engineering measurement.
IEEE Transactions on Software Engineering 22, 1 (1996), 68–86.

Chidamber, S. R., and Kemerer, C. F.
Towards a metrics suite for object oriented design.
SIGPLAN Not. 26, 11 (Nov. 1991), 197–211.

References ii

CHOWDHURY, I., AND ZULKERNINE, M.
Using complexity, coupling, and cohesion metrics as early indicators of vulnerabilities.
Journal of Systems Architecture 57, 3 (2011), 294–313.

EDER, J., KAPPEL, G., AND SCHREFL, M.
Coupling and cohesion in object-oriented systems, 1992.

ETZKORN, L., AND DELUGACH, H.
Towards a semantic metrics suite for object-oriented design.

In *Proceedings. 34th International Conference on Technology of Object-Oriented Languages and Systems - TOOLS 34* (2000), pp. 71–80.

HITZ, M., AND MONTAZERI, B.
Measuring coupling and cohesion in object-oriented systems.
In *Proceedings of International Symposium on Applied Corporate Computing* (1995), pp. 25–27.

KARTHA, G. P., ANJALI, C., NAIR, R. V., AND VENKATESWARI, S.
Prediction of defect susceptibility in object oriented software.
In *2017 International Conference on Networks Advances in Computational Technologies (NetACT)* (2017), pp. 467–472.

LE, Q. V., AND MIKOLOV, T.
Distributed representations of sentences and documents.
CoRR abs/1405.4053 (2014).

MARCUS, A., AND POSHYVANYK, D.
The conceptual cohesion of classes.
In *21st IEEE International Conference on Software Maintenance (ICSM'05)* (2005), pp. 133–142.

MARCUS, A., POSHYVANYK, D., AND FERENC, R.
Using the conceptual cohesion of classes for fault prediction in object-oriented systems.
IEEE Transactions on Software Engineering 34, 2 (2008), 287–300.

POSHYVANYK, D., MARCUS, A., FERENC, R., AND GYIMÓTHY, T.
Using information retrieval based coupling measures for impact analysis.
Empirical Softw. Engg. 14, 1 (Feb. 2009), 5–32.

References v

RATHORE, S. S., AND KUMAR, S.
A study on software fault prediction techniques.
Artificial Intelligence Review 51, 2 (2019), 255–327.

SHARMA, T., AND SPINELLIS, D.
Do we need improved code quality metrics?
CoRR abs/2012.12324 (2020).

TIWARI, S., AND RATHORE, S. S.
Coupling and cohesion metrics for object-oriented software: A systematic mapping study.
In *Proceedings of the 11th Innovations in Software Engineering Conference* (New York, USA, 2018), ISEC '18, Association for Computing Machinery.

West, M.
Object-oriented metrics: Measures of complexity, by brian henderson-sellers, prentice hall, 1996 (book review).
Softw. Test. Verification Reliab. 6 (1996), 255–256.

Zhang, D., Tsai, J., and Boetticher, G.
Improving credibility of machine learner models in software engineering.
In *Advances in Machine Learning Applications in Software Engineering*. 2007, pp. 52–72.

Újházi, B., Ferenc, R., Poshyvanyk, D., and Gyimóthy, T.
New conceptual coupling and cohesion metrics for object-oriented systems.
In *2010 10th IEEE Working Conference on Source Code Analysis and Manipulation* (2010), pp. 33–42.