Modeling Class Cohesion as Mixtures of Latent Topics

Yixun Liu, ***Denys Poshyvanyk***, Rudolf Ferenc,
Tibor Gyimóthy, Nikos Chrisochoides

Motivation

- Cohesion is the degree to which the elements in a design unit (class, package) are logically related or "belong together" [Briand 00]
- A cohesive class represents a crisp abstraction from a problem domain
- Class cohesion can significantly affect the design, understandability, maintainability
- Different views of cohesion

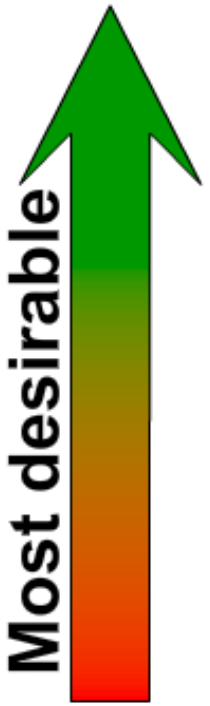
Class Cohesion

- Structural metrics:
  - LCOM1, LCOM2 [Chidamber 94][1]; LCOM3, LCOM4 [Hitz 94]
  - LCOM5 [Henderson 96]
  - Connectivity [Hitz 94]; Coh [Briand 97, 98]
  - ICH[2] [Lee 95]; TCC[3], LCC[4] [Bieman 95, 98]
- Semantic metrics:
  - LORM[5] [Etzkorn 00]; SCF[6] [Maletic 01]; C3[7][Marcus 05]
- Others: information entropy-based metrics [Allen 01]; slice-based metrics [Binkley 04]; etc.

1. *Lack of cohesion in methods*
2. *Information-flow based cohesion*
3. *Tight class cohesion*
4. *Loose class cohesion*
5. *Logical relatedness of methods*
6. *Semantic cohesion of files*
7. *Conceptual cohesion of classes*

Types of Cohesion

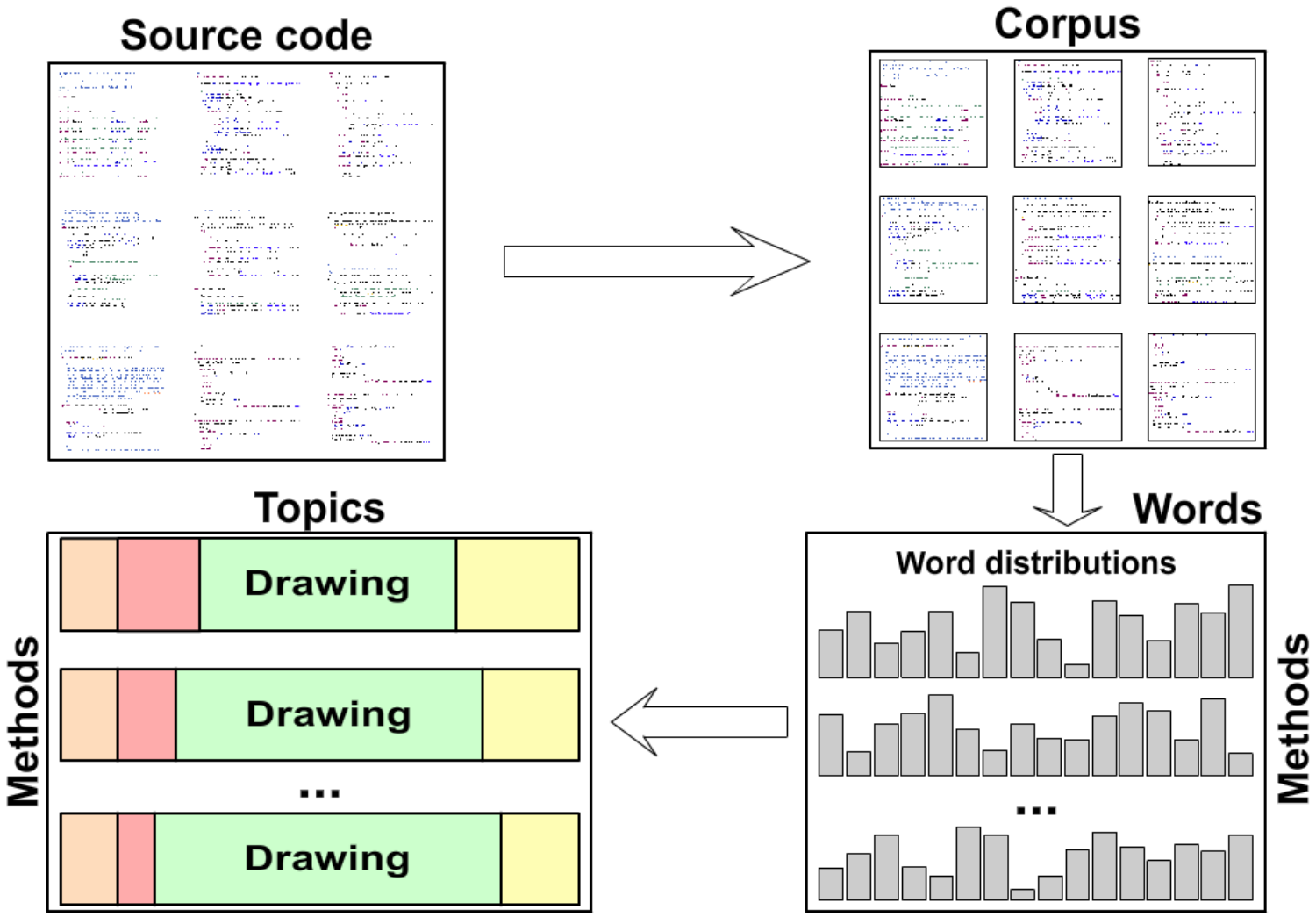
- Functional
- Informational
- Communicational
- Procedural
- Temporal
- Logical
- Coincidental



Using Latent Dirichlet Allocation for Cohesion Measurement

- Using semantic information (i.e., comments, identifiers, etc.) to measure cohesion
  - Prior work: textual similarities among methods in classes using Latent Semantic Indexing [Deerwester 90]
  - Current work: where and how the topics are implemented in the context of the software system using Latent Dirichlet Allocation [Blei 03]
- Latent Dirichlet Allocation
  - Each document is a mixture of different topics
  - Topics consist of words
  - Applied in software engineering before [Baldi'08], [Lukins'08], [Linstead'07]

Using LDA on Source Code



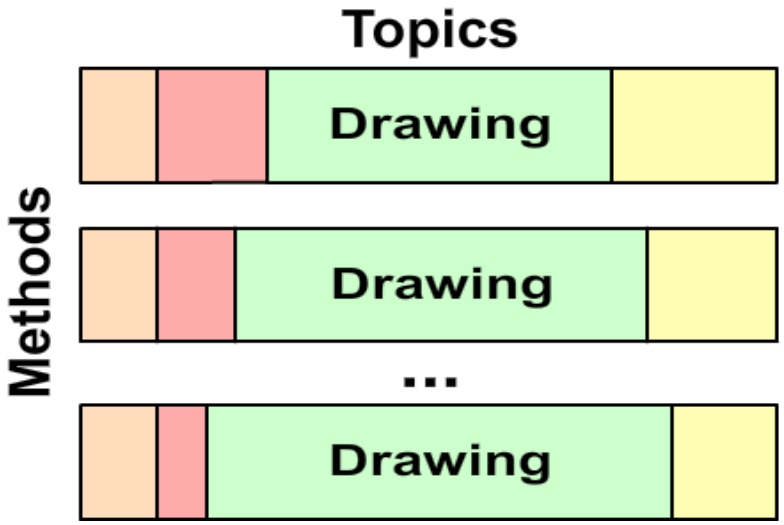
Example: topics in *CircleArea* class

- Topic "**drawing**" in Mozilla = {context, rendering, nscoord, color, device, draw, pixel, get, rect, units}

Methods	Topics (probabilities)			
	Drawing range	Drawing	Drawing context	View
Inside	0.003	**0.420**	0.062	0.012
Draw	0.002	**0.671**	0.002	0.002
GetRectangle	0.002	**0.462**	0.002	0.202

Capturing Class Cohesion using Information Entropy

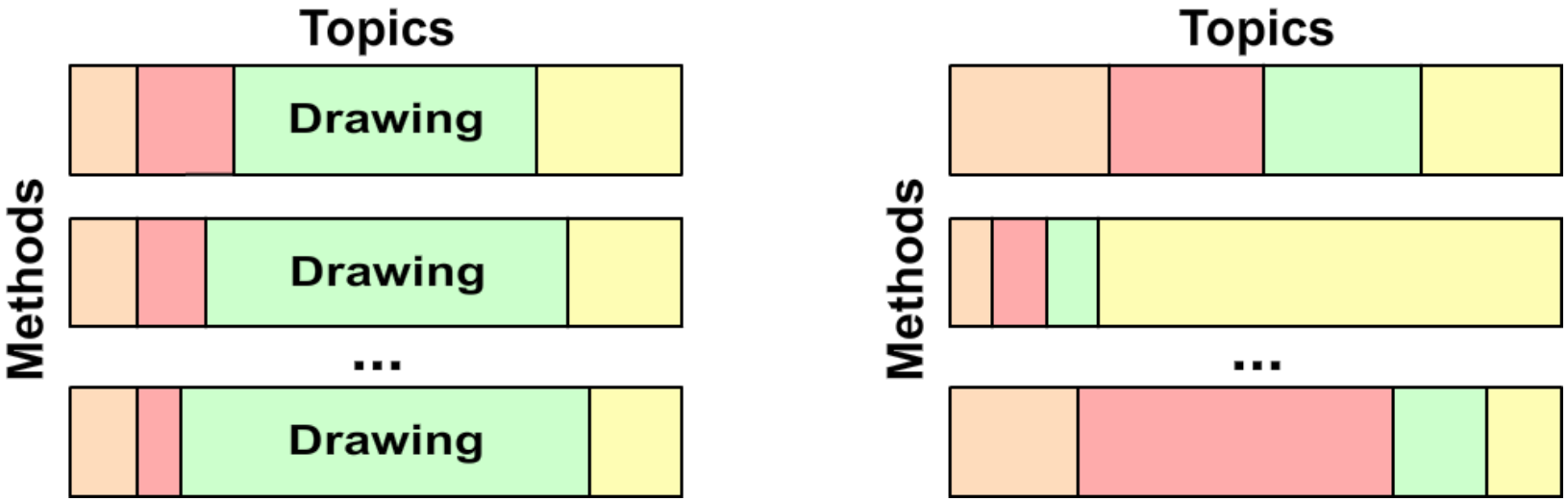
- A class with three methods and four topics



- Observations:
  - Occupancy (weight) of “green” topic
  - Distribution (entropy) of “green” topic

Distribution of Topics and Information Entropy

- Distribution of topic t_i across the methods of a class C_i is captured using Shannon information entropy



- High entropy for a dominating topic implies high cohesion and vice versa

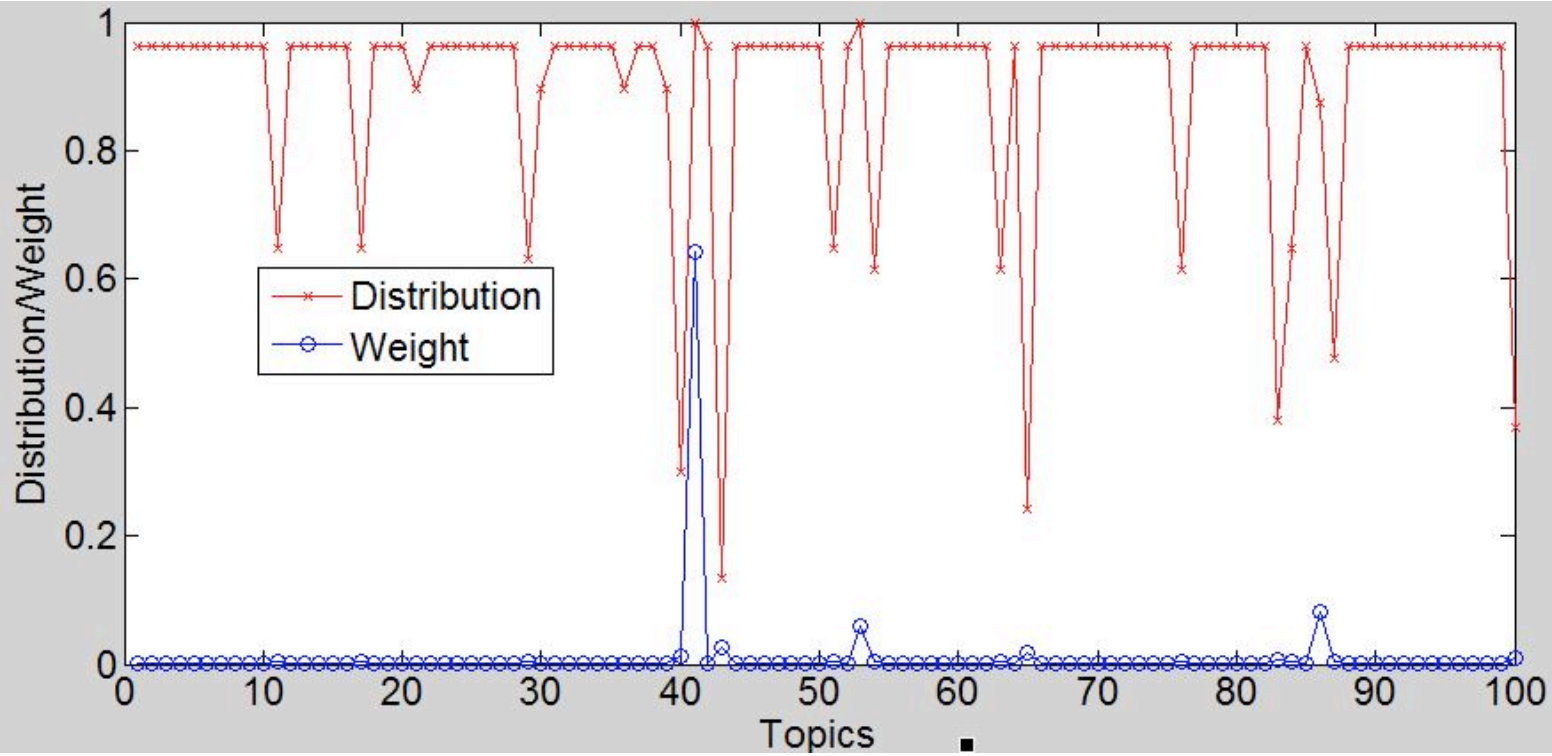
Cohesion Metric – Maximal Weighted Entropy

- MWE – Maximal Weighed Entropy
  - takes into account values of occupancy and distribution for the ***dominating*** topic in the class
  - occupancy of a topic t_i in methods of a class C_i captures the average probability of topic t_i across all methods in a class
  - distribution of a topic t_i across the methods of a class C_i is captured using information entropy (uniform distribution implies high entropy and high cohesion)

$$MWE(C)=\max(Occupancy(t) \times Distribution(t))$$

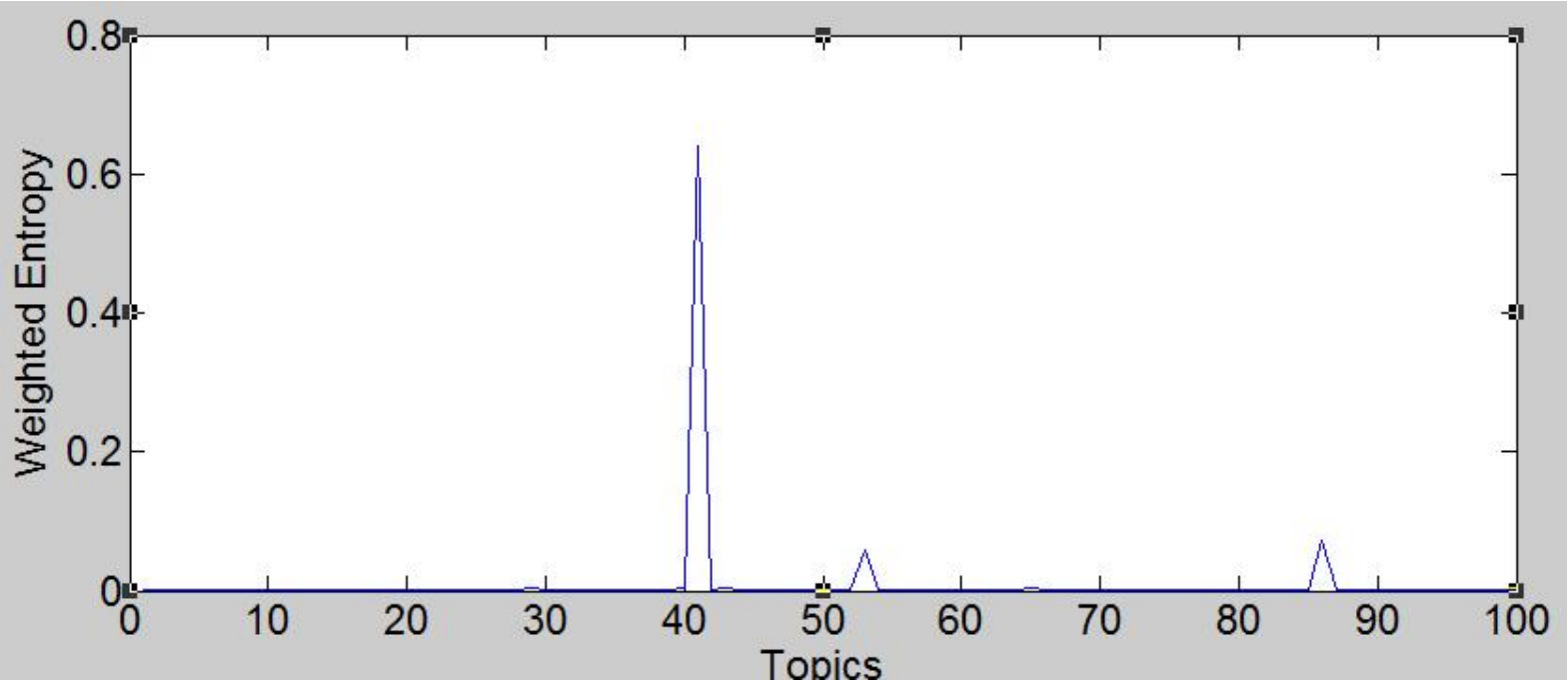
Example – *CircleArea* Class from Mozilla

- Three methods: *IsInside*, *Draw* and *GetRect*
- Topic 41: high weight and high entropy



Example – *CircleArea* Class from Mozilla

- Topic 41: high Maximal Weighted Entropy, MWE = 0.64



Empirical Assessment of Class Cohesion

- Research questions
  - **RQ1**: Does MWE capture aspects of class cohesion that are not captured by other structural and/or conceptual metrics?
  - **RQ2**: Do the combinations of cohesion metrics with MWE provide better results in predicting faults in classes than the combinations of the other metrics?
- Design of the case studies
  - Mozilla 1.6 analyzed with Columbus [Ferenc 04]
  - Metrics: $LCOM_1$, $LCOM_2$, $LCOM_n$, $LCOM_3$, $LCOM_4$, $LCOM_5$, Coh, ICH, TCC, C3 and LCSM and LOC
  - Analyzed 2,068 classes; 35K methods (docs)
  - LDA parameters: 100 topics
  - Case study data: www.cs.wm.edu/~denys/data/icsm09-mwe/

RQ_1 - Principal Component Analysis (PCA)

- Identifying groups of metrics (variables) which measure the same underlying mechanism that defines cohesion(dimension)

- PCA procedure:
  - collect data
  - identify outliers
  - perform PCA

PCA Results: Rotated Components

	PC_1	PC_2	PC_3	PC_4	PC_5	PC_6
Proportion	42.83	19.43	12.05	7.17	4.99	3.61
Cumulative	42.83	62.26	74.31	81.48	86.5	90.1
C3	-0.15	-0.17	0.26	**0.88**	-0.02	0.28
LCSM	-0.50	-0.26	0.01	-0.17	**0.78**	0.08
LOC	0.65	0.61	0.07	-0.21	-0.06	0.10
$LCOM_1$	**0.91**	0.28	-0.03	0.09	0.04	-0.10
$LCOM_2$	**0.96**	0.07	0.01	0.07	0.10	-0.09
$LCOM_n$	**0.96**	0.06	0.01	0.06	0.11	-0.10
$LCOM_3$	**0.94**	-0.11	0.09	0.07	0.07	-0.11
$LCOM_4$	**0.81**	-0.34	0.01	0.17	0.09	-0.30
$LCOM_5$	0.66	0.17	-0.23	0.22	0.27	0.21
ICH	0.58	0.56	0.16	-0.22	-0.01	**0.42**
TCC	-0.38	**0.83**	-0.27	0.16	0.07	-0.12
LCC	-0.26	**0.87**	-0.28	0.13	0.07	-0.07
Coh	-0.65	0.45	-0.31	0.22	0.08	-0.21
MWE	-0.26	0.32	**0.84**	0.04	0.08	-0.13

RQ_2 - Predicting Faults in Mozilla

- Identified bugs between two versions of Mozilla (1.6 and 1.7) [Gyimóthy'05]
- Univariate and multivariate logistic regression
- Quantitative characteristics
  - *Precision*: evaluates how well the model classifies classes as faulty or not;
  - *Correctness*: captures the percentage of the faulty predicted classes that are really faulty;
  - *Completeness*: evaluates the percentage of the total number of faulty classes that can be captured by the model.

Results for Univariate Logistic Regression

Metric	Precision	Precision Rank	Correct	Correct Rank	Complete	Complete Rank	R^2
LOC	64.26	1	71.74	5	65.96	5	0.131
$LCOM_1$	61.99	5	74.55	3	60.46	8	0.109
$LCOM_3$	62.72	2	70.30	6	64.01	7	0.107
$LCOM_2$	62.19	3	76.19	2	58.59	9	0.106
$LCOM_4$	59.86	9	65.99	7	54.72	12	0.079
C3	62.14	4	61.44	8	71.73	4	0.075
ICH	60.88	8	73.25	4	52.99	13	0.069
$LCOM_n$	61.56	7	78.99	1	55.08	11	0.06
Coh	61.79	6	60.17	9	78.18	2	0.034
MWE	57.21	11	55.47	10	65.67	6	0.03
LCSM	56.29	12	53.11	12	91.69	1	0.024
TCC	51.98	13	50.44	13	56.51	10	0.01
$LCOM_5$	57.30	10	55.28	11	75.21	3	0.007
LCC	50.68	14	48.72	14	32.74	14	0.002

Results for Multivariate Logistic Regression

Model	Precision	Precision Rank	Correct	Correct Rank	Complete	Complete Rank	R^2
MWE+LOC	67.31	1	73.02	30	73.97	18	0.166
LOC+$LCOM_4$	66.68	2	74.19	20	72.70	22	0.165
C3+LOC	66.68	3	69.79	50	74.98	12	0.164
C3+$LCOM_3$	66.19	5	68.32	55	75.77	10	0.161
LOC+$LCOM_3$	66.39	4	74.40	16	71.63	25	0.157
C3+$LCOM_1$	65.38	10	68.19	56	73.79	20	0.154
C3+ $LCOM_2$	64.65	16	67.09	59	72.44	23	0.152
LOC+ $LCOM_2$	66.15	6	75.00	10	70.46	28	0.148
LOC+$LCOM_1$	66.10	7	74.55	14	70.59	27	0.145

Combining MWE with other metrics

Metric	Precision	Precision MWE	Correct	Correct MWE	Complete	Complete MWE
C3	62.13	63.05	61.44	61.87	71.72	73.94
LCSM	56.28	59.62	53.10	57.43	91.69	**72.80**
LOC	64.26	67.31	71.73	73.02	65.96	73.97
$LCOM_2$	62.18	63.58	76.19	**71.35**	58.59	65.01
$LCOM_n$	61.55	61.70	78.99	**63.53**	55.08	68.69
$LCOM_1$	61.99	64.65	74.54	**72.24**	60.45	67.81
$LCOM_3$	62.71	63.49	70.29	**67.97**	64.00	68.53
$LCOM_4$	59.86	61.12	65.98	**63.52**	54.72	62.86
$LCOM_5$	57.30	57.64	55.27	55.88	75.21	**67.03**
ICH	60.88	62.57	73.25	**63.83**	52.99	72.01
TCC	51.98	57.15	50.43	55.39	56.51	67.16
LCC	50.67	56.91	48.71	55.17	32.73	66.05
Coh	61.79	**61.60**	60.17	**59.71**	78.17	**74.62**

Threats to Validity

- Only one software system
- Internal validity:
  - cohesion is not the only factor affecting the fault -proneness of classes;
  - LOC - possibly a confounding factor, needs a further investigation
- Number of topics

Related Work

[Baldi, Lopes, Linstead, Bajracharya @ OOPSLA '08]

- Problem statement:
  - Theory of aspects as latent topics;
  - Capturing scattering and tangling of cross-cutting concerns in open-source projects
- Solution:
  - LDA + information entropy
- Validation:
  - Identifying cross-cutting concerns in open-source projects;
  - Comparing with aspect identification approaches.

Current Limitations and Future Work

- MWE does not take into account polymorphism and inheritance
- Method invocations, parameters, attribute references, and types are of interest only at identifier level
- MWE does not make distinction between constructors, accessors, and other method stereotypes. Some of these methods can artificially increase or decrease cohesion

Conclusions

- IR and information theory can be used to capture software cohesion
- MWE is different from other metrics
- MWE is useful indicator of fault-proneness in classes

Thank you. Questions?

SEMERU @ William and Mary

http://www.cs.wm.edu/semeru/

denys@cs.wm.edu