Azul

ⴰⵣⵓⵍ

Measuring Differences To Compare Sets Of Models And Improve Diversity In MDE

Adel Ferdjoukh Florian Galinier, Eric Bourreau, Annie Chateau and Clémentine Nebut

ICSEA, Αθήνα, Ελλάδα, october 10th 2017

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

UNIVERSITÉ DE NANTES

LS2N

Synopsis

Context & Introduction

Model Driven Engineering

- Intensive use of models during software development process.
- A model is defined by a modelling language (meta-model).
- Models are manipulated by programs called model transformations.

Context & Introduction

Models play the key role

- Validate concepts (meta-model).
- test model transformations.

Solution to get sets of models

- Automated generation is preferred.

Context & Introduction

Models play the key role

- Validate concepts (meta-model).
- test model transformations.

Solution to get sets of models

- Automated generation is preferred.

Context & Introduction

Many generators exist

- $\mathcal{G}$RIMM.
- emftocsp.
- PRAMANA, etc.

Generated sets of models suffer from

- Close to each other in structure.
- Element naming is poor.
- Solutions' space is not covered.

Context & Introduction

Many generators exist

- $\mathcal{G}$RIMM.
- emftocsp.
- PRAMANA, etc.

Generated sets of models suffer from

- Close to each other in structure.
- Element naming is poor.
- Solutions' space is not covered.

Context & Introduction

Our objectives

1. Measure the quality of a set of models.
2. Improve the quality of a set of models.

Solutions we propose

1. Compare two models.
2. Handle a whole set of models.
3. Increase the diversity of generated sets.

Context & Introduction

Our objectives

1. Measure the quality of a set of models.
2. Improve the quality of a set of models.

Solutions we propose

1. Compare two models.
2. Handle a whole set of models.
3. Increase the diversity of generated sets.

Measuring model differences

Comparing two models with 4 distance measures.

Inspired from well-known distances.

- Mathematics.
- Natural language processing.
- Graph theory.

Adapted to models in MDE.

- structure of models.
- semantics of models.

Hamming distance

Original Hamming Distance

- Introduced in 1952 by Richard Hamming.
- Compares vectors.
- Used for fault detection and code correction.

Hamming distance

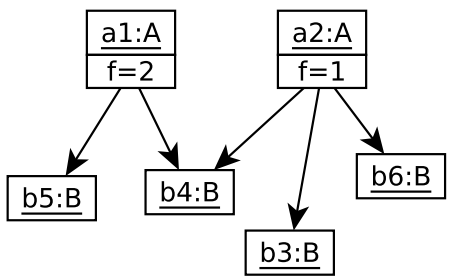
Original Hamming Distance

- Introduced in 1952 by Richard Hamming.
- Compares vectors.
- Used for fault detection and code correction.

Our version for models

- Vectorial representation for models

$$a = (\overbrace{\underbrace{5,4,0,}_{links}\ \underbrace{2,}_{attributes}}^{instance\ a1}\ \overbrace{\underbrace{4,3,6,}_{links}\ \underbrace{1}_{attributes}}^{instance\ a2})$$



model *a*

Hamming distance

Counting differences

$$
\begin{aligned}
\text{a} &= (5, 4, 0, 2, 4, 3, 6, 1) \\
&= \textcircled{1}\ \textcircled{1}\ \textcircled{1}\ \textcircled{1}\ \textcircled{0}\ \textcircled{1}\ \textcircled{1}\ \textcircled{0} \\
\text{b} &= (6, 5, 3, 3, 4, 7, 0, 1) \\
\text{d(a,b)} &= 6/8
\end{aligned}
$$

Optimisations: permutation sensitive.

Levenshtein distance

Original Levenshtein Distance

- Introduced in 1965 by Vladimir Levenshtein.
- Compares string.
- Used for orthographic corrections.

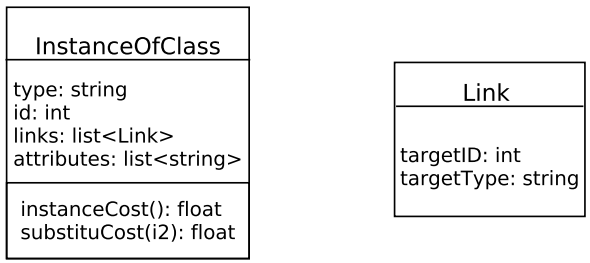
Levenshtein distance

Original Levenshtein Distance

- Introduced in 1965 by Vladimir Levenshtein.
- Compares string.
- Used for orthographic corrections.

Our version for models

- Vectorial representation for models



Model for vectorial representation

- Computing distance
  - Classical Levenshtein algorithm
  - Based on addition, suppression and substitution costs.

Centrality distance

Centrality measure

- In graphs, a function associating a value to each node.
- Many well-known centrality functions: `degree`, `betweenness`, `closeness`, etc.

Centrality distance

Centrality measure

- In graphs, a function associating a value to each node.
- Many well-known centrality functions: `degree`, `betweenness`, `closeness`, etc.

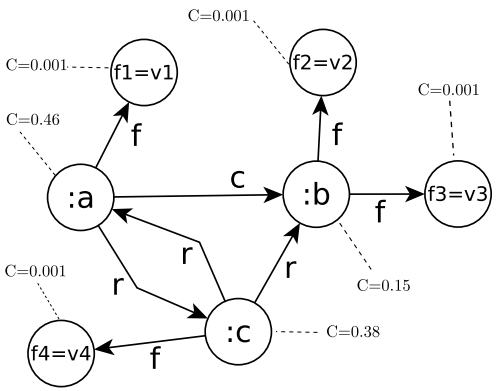
Custom centrality measure

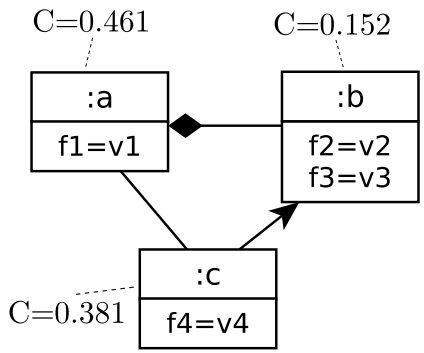
- Based on `eigenvector` centrality (used by `Google` in `Pagerank` algo.)

Centrality distance

Computation

- Transforming models into graphs
- Example of centrality vector





- Comparing two models using (euclidean) norm(s).

Handle sets of models

Objectives

- Compare the models of a set.
- Select the most representative ones.
- Bring a graphical view of the inter-model diversity.

Usefulness

- Reduce the amount of models for testing.
- Achieve a good coverage of meta-models.

Compare the models of a set

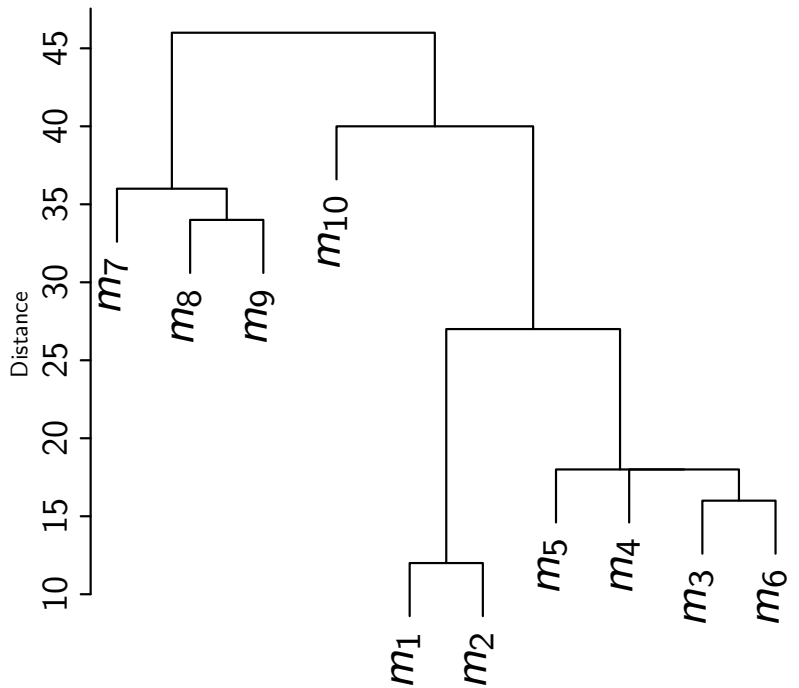
- Use distance metrics.
- Compute distances for each pair of models.
- Produce a distance matrix

	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}
m_1	0	12	27	27	27	26	46	44	45	39
m_2	12	0	27	26	27	27	45	45	43	40
m_3	27	27	0	18	17	16	46	45	46	39
m_4	27	26	18	0	18	18	45	44	45	40
m_5	27	27	17	18	0	18	45	43	44	38
m_6	26	27	16	18	18	0	45	44	46	40
m_7	46	45	46	45	45	45	0	36	36	41
m_8	44	45	45	44	43	44	36	0	34	37
m_9	45	43	46	45	44	46	36	34	0	39
m_{10}	39	40	39	40	38	40	41	37	39	0

Select representative models

Hierarchical clustering of matrix

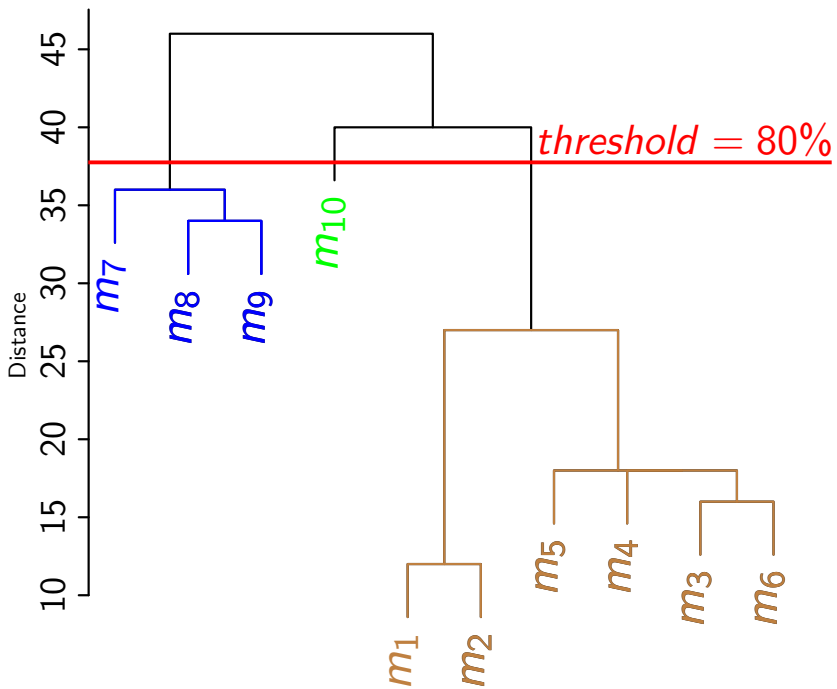
- Construct the clustering tree.
- Derive the clusters using a proximity threshold.
- Pick the representative models.



Select representative models

Hierarchical clustering of matrix

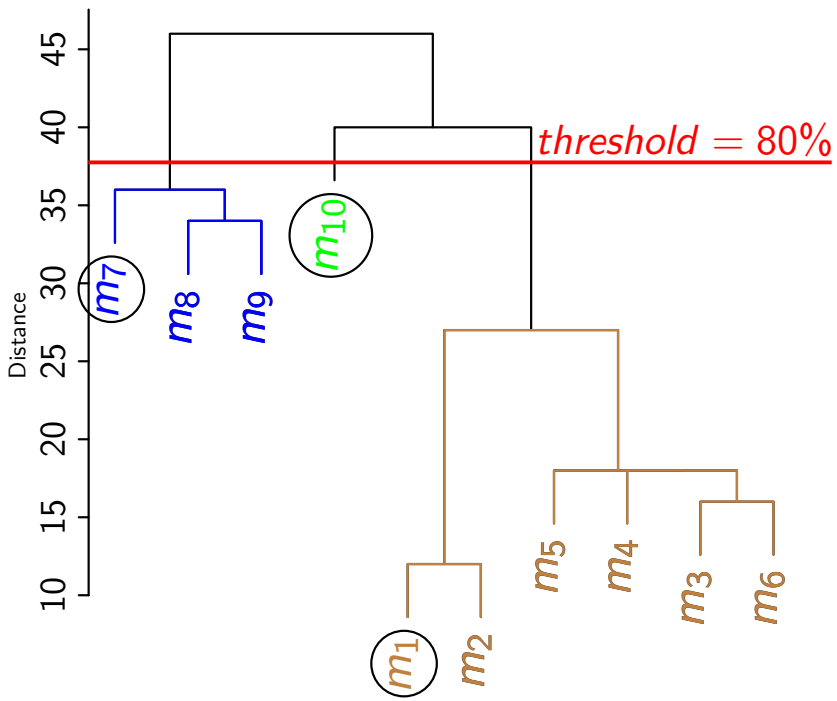
- Construct the clustering tree.
- Derive the clusters using a proximity threshold.
- Pick the representative models.



Select representative models

Hierarchical clustering of matrix

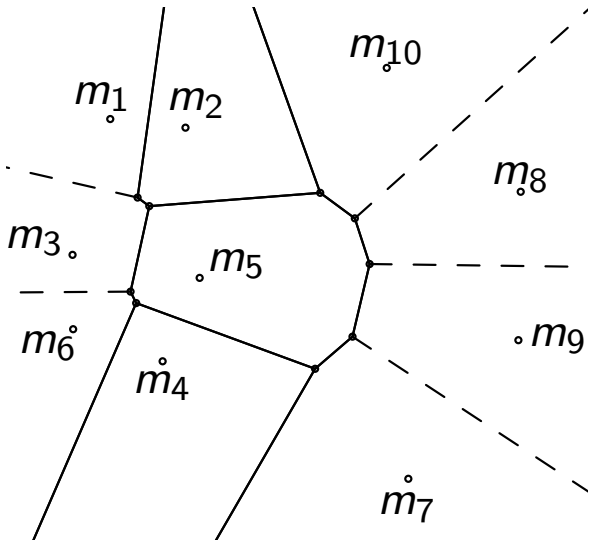
- Construct the clustering tree.
- Derive the clusters using a proximity threshold.
- Pick the representative models.



Graphical view of diversity

Voronoi Diagram

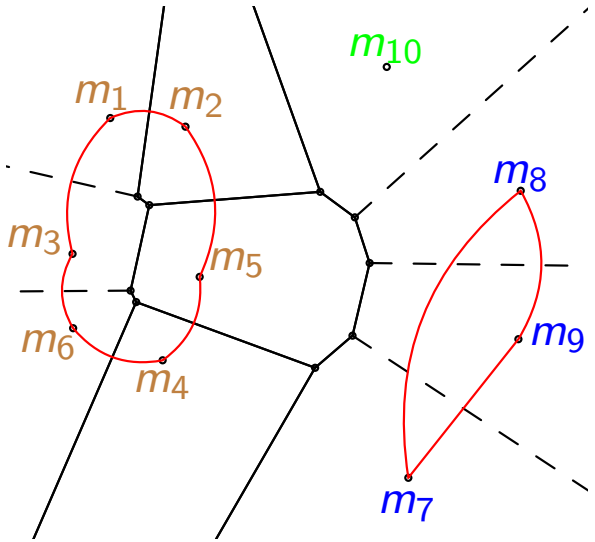
- 2D representation of models and distance between them.
- Manual selection of representative models.
- Manual comparison of model sets.



Graphical view of diversity

Voronoi Diagram

- 2D representation of models and distance between them.
- Manual selection of representative models.
- Manual comparison of model sets.



Application: improve diversity

Case study for application

- Scaffolding process in Bioinformatcis.
- Particular graphs.
- Lack of data.

With our approach

- Improve diversity of a set of generated models.
- Diverse sizes, structures, element naming.

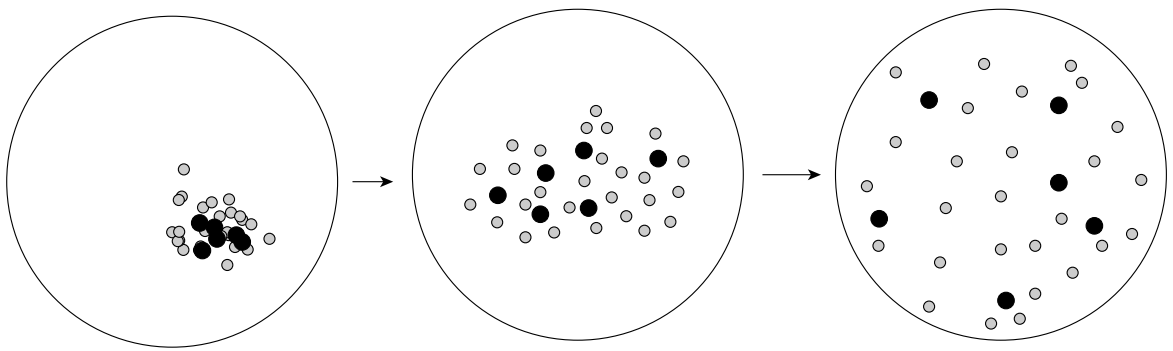
Application: improve diversity

Case study for application

- Scaffolding process in Bioinformatcis.
- Particular graphs.
- Lack of data.

With our approach

- Improve diversity of a set of generated models.
- Diverse sizes, structures, element naming.



Genetic algorithm

Generate a first set of 100 models.

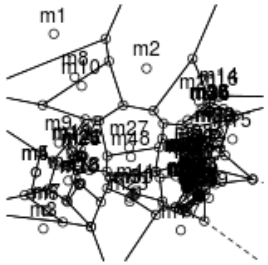
Genetic algorithm (NSGAII)

- Model the problem as a genetic algorithm.

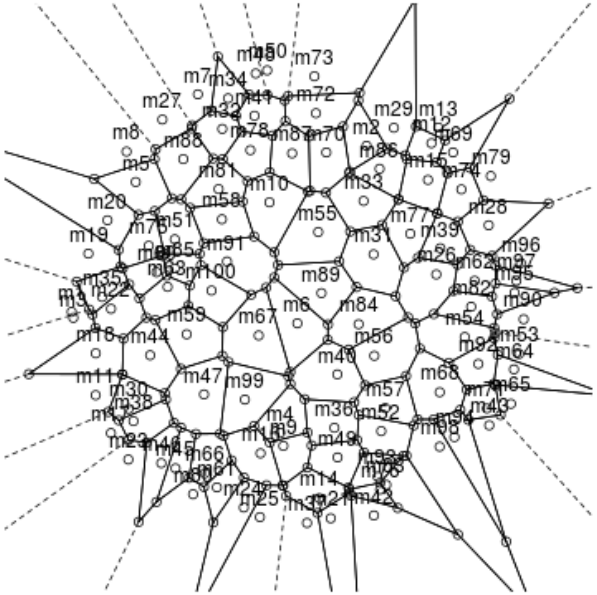
Running the GA (500 times)

- At each round, compute model distances and select representative models (S).
- Use S to produce the next population.

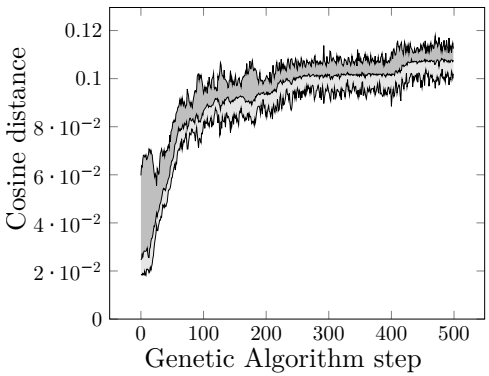
Experimental results

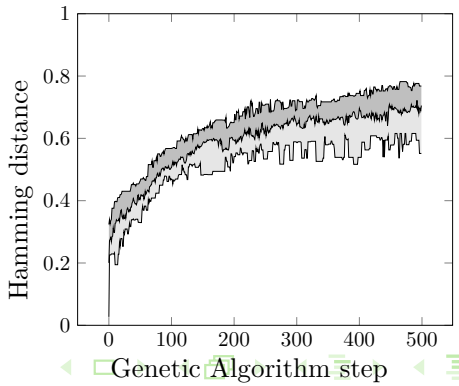


Round 0



Round 500





Conclusion & future work

Generated models suffer from

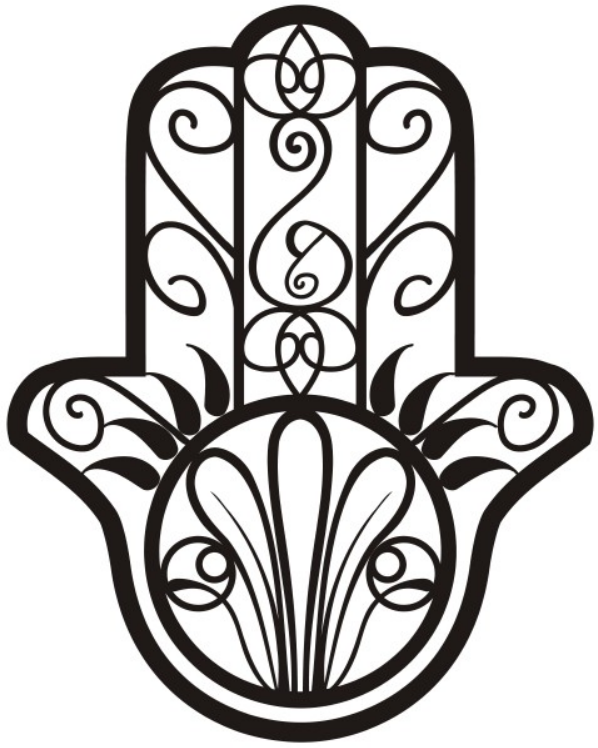
- Close to each other in structure.
- Element naming is poor.
- Solutions' space is not covered.

Contributions

- Four different measures for comparing models
- A method for comparing sets of models
- Select representative models (matrix clustering)
- Graphical viewing of covering (Voronoi diagrams)

Application

- Generate scaffold graphs in bio-informatics
- Improve diversity of generated models



Tanmirt

ⵜⴰⵏⵎⵉⵔⵜ