DSE = Data-Driven Search-Based SE

Vivek Nair, Amritanshu Agrawal, Jianfeng Chen
Wei Fu, George Mathew, Tim Menzies
Leandro Minku, Markus Wagner, Zhe Yu

MSR’18, Gothenburg, Sweden
Why did these MSR people meet in Japan in Dec’17?

\[ \text{DSE} = \text{Data-Driven Search-based SE} \]
## Search-based SE: highly acceptable at MSR

<table>
<thead>
<tr>
<th>rank</th>
<th>topic</th>
<th>Comparing with the year 2017</th>
<th>submissions</th>
<th>accepted</th>
<th>acceptance rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MSR on Multiple Projects</td>
<td>-</td>
<td>54</td>
<td>17</td>
<td>0.31</td>
</tr>
<tr>
<td>2</td>
<td>Tools and Techniques for MSR</td>
<td>-</td>
<td>46</td>
<td>14</td>
<td>0.30</td>
</tr>
<tr>
<td>3</td>
<td>Empirical studies</td>
<td>-</td>
<td>31</td>
<td>11</td>
<td>0.35</td>
</tr>
<tr>
<td>4</td>
<td>MSR with NLP</td>
<td>↑ 1</td>
<td>35</td>
<td>9</td>
<td>0.26</td>
</tr>
<tr>
<td>4</td>
<td>Change Patterns and Trends</td>
<td>↑ 4</td>
<td>23</td>
<td>9</td>
<td>0.39</td>
</tr>
<tr>
<td>5</td>
<td>Search-driven software development</td>
<td>↑ 3</td>
<td>20</td>
<td>7</td>
<td>0.35</td>
</tr>
<tr>
<td>6</td>
<td>Other</td>
<td>↑ 4</td>
<td>16</td>
<td>7</td>
<td>0.33</td>
</tr>
<tr>
<td>6</td>
<td>Software project evolution</td>
<td>↑ 3</td>
<td>19</td>
<td>6</td>
<td>0.32</td>
</tr>
<tr>
<td>6</td>
<td>Mining mobile app stores</td>
<td>↑ 2</td>
<td>12</td>
<td>6</td>
<td>0.50</td>
</tr>
<tr>
<td>7</td>
<td>Defect Analysis</td>
<td>↓ 3</td>
<td>23</td>
<td>5</td>
<td>0.22</td>
</tr>
<tr>
<td>7</td>
<td>Integrating mined data</td>
<td>↑ 2</td>
<td>12</td>
<td>5</td>
<td>0.42</td>
</tr>
<tr>
<td>8</td>
<td>Prediction with MSR</td>
<td>↓ 1</td>
<td>17</td>
<td>4</td>
<td>0.24</td>
</tr>
<tr>
<td>8</td>
<td>Social and development processes</td>
<td>↓ 1</td>
<td>14</td>
<td>4</td>
<td>0.29</td>
</tr>
<tr>
<td>9</td>
<td>PL features with MSR</td>
<td>↓ 2</td>
<td>11</td>
<td>4</td>
<td>0.36</td>
</tr>
<tr>
<td>9</td>
<td>Mining interaction data repositories</td>
<td>↓ 1</td>
<td>9</td>
<td>2</td>
<td>0.22</td>
</tr>
<tr>
<td>9</td>
<td>Bias in mining and guidelines</td>
<td>-</td>
<td>7</td>
<td>2</td>
<td>0.29</td>
</tr>
<tr>
<td>9</td>
<td>Sharing Data</td>
<td>-</td>
<td>6</td>
<td>2</td>
<td>0.33</td>
</tr>
<tr>
<td>9</td>
<td>Visualization</td>
<td>↑ 1</td>
<td>6</td>
<td>2</td>
<td>0.33</td>
</tr>
<tr>
<td>10</td>
<td>Mining code review repositories</td>
<td>↓ 3</td>
<td>9</td>
<td>1</td>
<td>0.11</td>
</tr>
<tr>
<td>10</td>
<td>Extracting New Forms of Data</td>
<td>↓ 4</td>
<td>9</td>
<td>1</td>
<td>0.11</td>
</tr>
<tr>
<td>10</td>
<td>Reliability and defect occurrences</td>
<td>↓ 1</td>
<td>6</td>
<td>1</td>
<td>0.17</td>
</tr>
<tr>
<td>10</td>
<td>Software licensing and copyrights</td>
<td>NEW</td>
<td>4</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>10</td>
<td>Energy aware mining</td>
<td>NEW</td>
<td>2</td>
<td>1</td>
<td>0.50</td>
</tr>
<tr>
<td>11</td>
<td>Mining execution traces and logs</td>
<td>↓ 4</td>
<td>4</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>11</td>
<td>Privacy and ethics</td>
<td>NEW</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
</tr>
</tbody>
</table>
What is SBSE?
(Search-based Software Engineering)

- Many SE activities are like optimization problems [Harman’02]
- Due to computational complexity, exact optimization methods are impractical
- Alternative: find good-enough solutions using meta-heuristic search as our optimizers
  - e.g. genetic algorithms
  - e.g. simulated annealing
  - e.g. tabu search
  - e.g. NSGA-II, SPEA2, MOEA/D, Differential Evolution, Bayesian parameter optimization, etc etc
DSE = Data-Driven Search-based SE

- Conceptually, common higher level goal
  - supporting and giving insights to software engineers
Data-Driven Search-based SE (DSE)

• To solve an SE problem:
  – Insert a data miner into an optimizer;
  – Or use an optimizer to improve a data miner.

• A new era for MSR (better MSR)
• A new era for SBSE (better SBSE)
A new era for SBSE: Supercharging MSR

- Black art: hyperparameter optimization
- E.G. learning how many trees in a random forest
- E.G. learning how many “k” in kth-nearest neighbors
- Thanks to SBSE: massive improvements in, say, defect prediction
  - e.g. Agrawal & Menzies, ICSE 2018
  - performance details (after - before) tuning
A new era for SBSE: Let MSR help you run faster

- Landscape analysis
  - Find the lay of the land (shape of data)
  - Jump faster to better conclusions
  - e.g.. GALE, TSE 2015
- Note that this “optimizer” is really a “data miner”
  - clustering, PCA

mutate all orange points this way

Red ignored
Q: Why explore MSR+SBSE?
A: So many application areas

1. Requirements Menzies, Feather, Bagnall, Mansouri, Zhang
2. Transformation Cooper, Ryan, Schielke, Subramanian, Fatiregun, Williams
3. Effort prediction Aguilar-Ruiz, Burgess, Dolado, Lefley, Shepperd
4. Management Alba, Antoniol, Chicano, Di Pentam Greer, Ruhe
5. Heap allocation Cohen, Kooi, Srissa-an
6. Regression test Li, Yoo, Elbaum, Rothermel, Walcott, Soffa, Kampfhamer
7. SOA Canfora, Di Penta, Esposito, Villani
8. Refactoring Antoniol, Briand, Cinneide, O’Keeffe, Merlo, Seng, Tratt
9. Test Generation Alba, Binkley, Bottaci, Briand, Chicano, Clark, Cohen, Gutjahr, Schenfeld, Holcombe, Jones, Korel, Pargass, Reformat, Roper, McMinn, Marre, Singer, Trace, Tonella, Xanthakis, Xiao, Wegener, Wilkins
10. Maintenance Antoniol, Lutz, Di Penta, Madhavi, Mitchell, Swift
11. Model checking Alba, Chicano, Godefroid
12. Probing Cohen, Elbaum
13. Comprehension Gold, Li, Mahdavi
14. Protocols Alba, Clark, Jacob, Troya
15. Component sel Baker, Skaliotis, Steinhofel, Yoo
16. Agent Oriented Haas, Peysakhov, Sinclair, Shami, Mancoridis

so many novel contributions to so many areas
Q: Why explore MSR+SBSE?
A2: cause you got to

- How to get a paper rejected (in 2020):
  - Publish data mining results **without** hyper-parameter optimization

- Coming to the end of “merely mining”
  - See debates on “unsupervised learning”
    - Too easy to just chase precision, recall etc

- Complex problems need complex inference
  - e.g. minimizing #false alarms before first defect [Huang et al.ICSME’17]
  - Needed to reply to (e.g.) [Parnin, Orso, Issta’11]
**http://tiny.cc/data-SE:** A new resource for MSR researchers

89 DSE artifacts, in 13 groups
(e.g. RE, software product lines, software processes)

**Existing results; useful for testing new methods**

<table>
<thead>
<tr>
<th>Domain</th>
<th>Problem</th>
<th>Decision Space</th>
<th>C/D</th>
<th>Projects</th>
<th>Description</th>
<th>Links</th>
<th>Related Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSR</td>
<td>Defect Prediction</td>
<td>Numeric</td>
<td>D</td>
<td>10</td>
<td>CK Metric</td>
<td>raise_data_defect</td>
<td>[29]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 Citemap</td>
<td>raise_data_pits</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6 Pits</td>
<td>raise_data_pits</td>
<td>[1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 StackOverflow</td>
<td>SOProcess</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Text Classification</td>
<td>Text</td>
<td>-</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Performance Optimization</td>
<td>Mixed</td>
<td>D</td>
<td>22</td>
<td>Performance Configuration</td>
<td>raise_data_perf</td>
<td>[66, 67, 69]</td>
</tr>
</tbody>
</table>
So now we know why all these MSR people are so interested in SBSE

- Thanks to organizers the Dec’17 NII Shonan Meeting
  - Data-Driven Search-based SE, Dec 11-14, 2017
  - Markus Wagner, Leandro Minku, Ahmed Hassan, John Clark
DSE = Data-Driven Search-based SE

To solve an SE problem:
- Insert a data miner into an optimizer;
- Or use an optimizer to improve a data miner.

• A new era for MSR (better MSR)
• A new era for SBSE (better SBSE)
... if engineering, then NC State ...
Back-up slides
A new era for MSR: Data farming (MSR + SBSE)

- Big data and massive Monte Carlo analysis
  - find important interactions
- domain intuitions $\Rightarrow$
- model $\Rightarrow$
  - generation $+= 1$
  - simulation $i$
  - data
  - mining
  - insight
  - repeat
Q: Why explore farming data from models?
A: Cause models are everywhere

1. Silicon valley developers, new features are experiments, to be tested


3. Engineers test designs via models: radiation therapy, remote sensing, chip design, [http://goo.gl/qBMylZ](http://goo.gl/qBMylZ)

4. Web analysts use models to analyze clickstreams to improve marketing: [http://goo.gl/b26CfY](http://goo.gl/b26CfY)

5. Stock traders use models to simulate trading strategies [http://www.quantopian.com](http://www.quantopian.com)


7. Journalists use models to analyze economic data [http://fivethirtyeight.com](http://fivethirtyeight.com)

8. In London or New York, ambulances wait at locations determined by a model [http://goo.gl/8SMd1p](http://goo.gl/8SMd1p)

9. Etc etc etc
Why explore SBSE + MSR? (the carrot)

1. Requirements Menzies, Feather, Bagnall, Mansouri, Zhang
2. Transformation Cooper, Ryan, Schielke, Subramanian, Fatiregun, Williams
3. Effort prediction Aguilar-Ruiz, Burgess, Dolado, Lefley, Shepperd
4. Management Alba, Antoniol, Chicano, Di Pentam Greer, Ruhe
5. Heap allocation Cohen, Kooi, Srisa-an
6. Regression test Li, Yoo, Elbaum, Rothermel, Walcott, Soffa, Kampfhamer
7. SOA Canfora, Di Penta, Esposito, Villani
8. Refactoring Antoniol, Briand, Cinneide, O’Keeffe, Merlo, Seng, Tratt
9. Test Generation Alba, Binkley, Bottaci, Briand, Chicano, Clark, Cohen, Gutjahr, Schirlo, Holcombe, Jones, Korel, Pargass, Reformat, Roper, McMinn, Michael, Sthamer, Tracy, Tonella, Xanthakis, Xiao, Wegener, Wilkins
10. Maintenance Antoniol, Lutz, Di Penta, Madhavi, Mancoridis
11. Model checking Alba, Chicano, Godefroid
12. Probing Cohen, Elbaum
13. Comprehension Gold, Li, Mahdavi
14. Protocols Alba, Clark, Jacob, Troya
15. Component sel Baker, Skaliotis, Steinhofel, Yoo
16. Agent Oriented Haas, Peysakhov, Sinclair, Shami, Mancoridis

so many novel contributions to so many areas
## Some technical differences

<table>
<thead>
<tr>
<th></th>
<th>MSR</th>
<th>SBSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inference</td>
<td>induction, visualize</td>
<td>optimization</td>
</tr>
<tr>
<td>Speed</td>
<td>Faster, often more scalable</td>
<td>Becoming faster</td>
</tr>
<tr>
<td>Data</td>
<td>Collected before inference</td>
<td>Sampling controlled by inference</td>
</tr>
<tr>
<td>Tools</td>
<td>R, SciKitLearn, WEKA</td>
<td>jMetal, AutoWeka, AutoSklearn, Opt4j, DEAP</td>
</tr>
<tr>
<td>Example problems</td>
<td>● e.g. defect prediction;</td>
<td>● minimize a test suite</td>
</tr>
<tr>
<td></td>
<td>● StackOverflow mining</td>
<td>● configure software</td>
</tr>
<tr>
<td>Goals</td>
<td>e.g. just a few: recall, precision, MRE</td>
<td>● domain-specific goals.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● meta-criteria (hypervolume, spread, IGD)</td>
</tr>
</tbody>
</table>
Optimization = surfing the landscape

*murmuration* of starlings
(learn safe “shapes” to avoid predators)

Particle Swarm Optimization:
\[
\text{new} = \text{old} + \phi_1 \text{rand}( \text{ourBest} - \text{now} ) + \phi_2 \text{rand}( \text{myBest} - \text{now} )
\]

;; social cognition

;; private cognition

use data miners to learn the landscape, guide our optimizers?
Something is changing. Things are .... different

Strange new words:
- “hyper-parameter optimization”
- “evolutionary algorithms”
- “differential evolution”
- “model-based reasoning”

What is going on?
MSR has much to gain from SBSE

- See paper, Fig4, long list of domain-specific goals
  - e.g. minimizing initial false before first defect [Huang et al., ICSME’17] [Parnin, Orso, Issta’11]
  - e.g. favor the shortest, most readable, model with least error

- Goals are domain-dependent
  - Need tools that adjust to different goals
Q: But why bother?
A: Cause much of SE is about choice

BTW: Linux kernel:
- 7000 terms
- 350,000 constraints
How does SBSE connect to MSR?

- Theoretically:
  - All learners build models that trade off competing goals
  - e.g. maximize recall, minimize false alarms

- Empirically:
  - better algorithms adjust themselves to the curves
What is SBSE? *(Search-based Software Engineering)*

- Many SE activities are like optimization problems [Harman’02]

- Due to computational complexity, exact optimization methods are impractical

- Alternative: find good-enough solutions using meta-heuristic search as our optimizers
  - e.g. genetic algorithms
  - e.g simulated annealing
  - e.g. tabu search
  - e.g. NSGA-II, SPEA2, MOEA/D, Differential Evolution, Bayesian parameter optimization, etc etc
A new era for MSR:
Surfing cost-benefit decisions

- Exploring cost-benefit trade-offs in software engineering
- e.g. learn tests that run fastest, most likely to fail
- e.g. as done manually by Elbaum et al, FSE’14
- e.g. as could be done automatically via SBSE

**Graph:**
- after: 50% tests fail in first hour
- before: weeks before tests fail

**Graph Data:**
- % Test Suites
- % Execution Time
- % Failing Test Suites
- % Failing Test Suites with Random Selection