Audition CRCN CNRS 2022
Efficient Exploration of Colossal Configurable Spaces

Paul TEMPLE

March 2022

Equipes : Spirals (Lille) ; ProGresS (Bordeaux) ; NaoMod (Nantes)

Software variability & system complexity

JHipster: **50** options

Linux Kernel: **15,000** options

Software variability & system complexity

JHipster: **50** options

Linux Kernel: **15,000** options

$$2^{15,000} \approx 10^{3,250} >> 10^{1,000} >> \text{estimated } \# \text{ of particules}$$

Sébastien Mosser @petitroll · 25 févr.

"the number of atoms in the visible universe is 10^80. There are 2^15000 different versions of the Linux kernel. So astrophysicists works with things way simpler than software engineers". @jmjezequel

Software variability & performances

no-mbtree = F
qblur = 0.90
nr = 200

encoding time = 5 min

encoding time = 2 h

encoding time = 10 h

Evaluating performance is complex

	Program Variants				
Inputs				...	
		12	1	...	5
		1	348	...	10
	...				
		50	101	...	260

Do we need to measure?

Assumptions

- Exploring all configurations is **impossible**
- Measuring performances is **costly**

Siegmund *et al.*, Perf. Prediction with feature interaction, ICSE'12
Siegmund *et al.*, Perf.-Influence models for config. systems, FSE'15
Guo *et al.*, Var.-aware perf. prediction, ASE'13

Do we need to measure?

Assumptions

- Exploring all configurations is **impossible**
- Measuring performances is **costly**

Assign a measure without measuring

- Similar configurations produce similar performances
- Performance prediction

Siegmund *et al.*, Perf. Prediction with feature interaction, ICSE'12
Siegmund *et al.*, Perf.-Influence models for config. systems, FSE'15
Guo *et al.*, Var.-aware perf. prediction, ASE'13

Do we need to measure?

Assumptions

- Exploring all configurations is **impossible**
- Measuring performances is **costly**

Assign a measure without measuring

- Similar configurations produce similar performances
- Performance prediction
  - Linear models (+ interactions)
  - Incremental learning

Siegmund *et al.*, Perf. Prediction with feature interaction, ICSE'12
Siegmund *et al.*, Perf.-Influence models for config. systems, FSE'15
Guo *et al.*, Var.-aware perf. prediction, ASE'13

Do we need to measure?

Assumptions

- Exploring all configurations is **impossible**
- Measuring performances is **costly**

Assign a measure without measuring

- Similar configurations produce similar performances
- Performance prediction

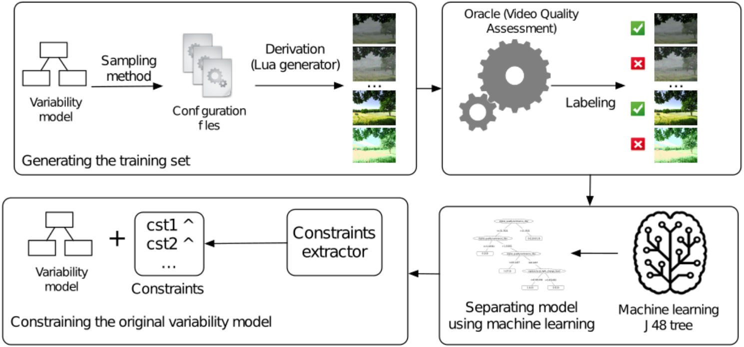
Users know what they want

- Technically & performance-wise
- **Few** configurations are **acceptable**

$\rightarrow$ **Scope** the configuration space

Temple *et al.*, Using machine learning to infer constraints for product lines, SPLC'16

Reducing configuration space with ML



Temple *et al.*, Using machine learning to infer constraints for product lines, SPLC'16

Improving the classification of software configurations

Impacts

Machine Learning is based on statistics $\rightarrow$ errors

- Over-constraining
- Under-constraining

Improving the classification of software configurations

Impacts

Machine Learning is based on statistics → errors

- Over-constraining
- Under-constraining

Over-constraining

- Forbid more configurations than necessary
- Lack of flexibity

Improving the classification of software configurations

Impacts

Machine Learning is based on statistics → errors

- Over-constraining
- Under-constraining

Over-constraining

- Forbid more configurations than necessary
- Lack of flexibity

Under-constraining

- Allow more configurations than necessary
- Waste of resources and can have dramatic outcome

Improving the pipeline

Robustifying the model

- Show new configurations
- Configurations with **high risk** of misclassification

Goodfellow *et al.*, Adversarial examples, ICLR'15
Elsayed *et al.*, Fool both humans and computers, NeurIPS'18
Sharif *et al.*, Accessorize to crime, CCS'16

Improving the pipeline

Robustifying the model

- Show new configurations
- Configurations with **high risk** of misclassification

Improving the pipeline

Robustifying the model

- Show new configurations
- Configurations with **high risk** of misclassification

Goodfellow *et al.*, Adversarial examples, ICLR'15
Elsayed *et al.*, Fool both humans and computers, NeurIPS'18
Sharif *et al.*, Accessorize to crime, CCS'16

Robustifying the model

Configurations with high risk of misclassification

- Adversarial retraining → retrain a model
- Enhanced exploration → what make them misclassified?

Biggio *et al.*, Evasion attacks against SVMs, ECML'13
Temple *et al.*, Adv. Configs for config. systems, EMSE'21
PRALab website

Robustifying the model

Configurations with high risk of misclassification

- Adversarial retraining → retrain a model
- Enhanced exploration → what make them misclassified?

Adversarial Configurations for configurable systems

- 1^{st} application of evasion attacks to configurable systems
- Opportunity to work with PRALab
- SPLC'19 → EMSE'21

Biggio *et al.*, Evasion attacks against SVMs, ECML'13
Temple *et al.*, Adv. Configs for config. systems, EMSE'21
PRALab website

Research Project: Adversarial ML for software testing

Support for constraints

- Constraints on feature values and combinations → forbidding exploring subspaces
- Constraints may be complex → involve several features
- Generation process is iterative → constraint checking strategy

Delobelle *et al.*, Ethical Adversaries, SIGKDD Exploration NewsLetters

Research Project: Adversarial ML for software testing

Support for constraints

- Constraints on feature values and combinations → forbidding exploring subspaces
- Constraints may be complex → involve several features
- Generation process is iterative → constraint checking strategy

Future directions

- Adversarial for improvement → fairness
- Adversarial sampling

Delobelle *et al.*, Ethical Adversaries, SIGKDD Exploration NewsLetters

Research Project: Find an efficient representation for configurations

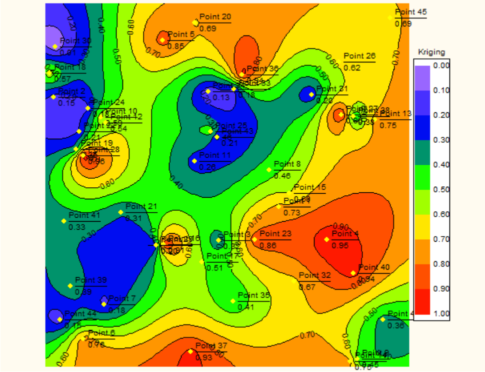
What is wrong?

- Similar configurations → similar performances
- options as a feature vector → interactions?

Research Project: Find an efficient representation for configurations

What is wrong?

- Similar configurations → similar performances
- options as a feature vector → interactions?



Research Project: ML models design with variability management tools

Modern ML models

- 100 epochs ImageNet to train AlexaNet in 24*minutes* for **only 1.2M dollars**

$\Rightarrow$ Impossible if you are not GAFAM

You *et al.*, ImageNet trained in 24 Minutes, ICPP'18

Research Project: ML models design with variability management tools

Modern ML models

- 100 epochs ImageNet to train AlexaNet in 24*minutes* for **only 1.2M dollars**

$\Rightarrow$ Impossible if you are not GAFAM

Goal of variability management

- Reducing costs to make it accessible
- Green computing
- Reduce complexity of models $\rightarrow$ explainability

You *et al.*, ImageNet trained in 24 Minutes, ICPP'18

Integration in Spirals

One of the **most active** French configurable systems team

Variability, prediction performance

- Edouard Guegain
- Clément Quinton
- Romain Rouvoy

Adaptable systems

- Laurence Duchien
- Lionel Seinturier

Machine learning

- Patrick Bas

Integration in ProGresS

Missing a ML dimension to start collaborations

Software variability and evolution

- Thomas Degueule
- Laurent Réveillère

Green computing

- Jean-Rémy Falleri

Machine learning and explainability

- Collaborations with BKB

Integration in NaoMod

Research in relations with companies

Software variability and architecture

- Gerson Sunyé
- Dalila Tamzalit

ML4SE

- Dalila Tamzalit
- Project with GEODES (Montréal, Canada)

Low-code

- Lowcomote EU project
- User in the loop

Efficient Exploration of Colossal Configurable Spaces

- Software variability; Machine learning; Performance
- Testing performances of configurable systems is difficult
- Adversarial configurations
- Research Project:
  - representation problem
  - adversarial for improvement $\rightarrow$ fairness; adversarial sampling
  - var. management for models $\rightarrow$ green computing, reducing complexity
- teams:
  - Spirals, Lille
  - ProGresS, Bordeaux
  - NaoMod, Nantes