AI-Augmented Algorithms – How I Learned to Stop Worrying and Love Choice

Lars Kotthoff
University of Wyoming
larsko@uwyo.edu

TU Eindhoven, 09 July 2018
Outline

▷ Big Picture
▷ Motivation
▷ Algorithm Selection and Portfolios
▷ Algorithm Configuration
▷ Outlook
Big Picture

▷ advance the state of the art through meta-algorithmic techniques
▷ rather than inventing new things, use existing things more intelligently – automatically
▷ invent new things through combinations of existing things
Big Picture

▷ advance the state of the art through meta-algorithmic techniques
▷ rather than inventing new things, use existing things more intelligently – automatically
▷ invent new things through combinations of existing things

WE'VE DECIDED TO DROP THE CS DEPARTMENT FROM OUR WEEKLY DINNER PARTY HOSTING ROTATION.
Motivation – What Difference Does It Make?
Performance Differences

Leveraging the Differences

Performance Improvements

Performance models of black-box processes

▷ also called surrogate models
▷ replace expensive underlying process with cheap approximate model
▷ build approximate model based on real evaluations using machine learning techniques
▷ no knowledge of what the underlying process does required (but can be helpful)
▷ allow better understanding of the underlying process through interrogation of the model
Algorithm Selection
Algorithm Selection

Given a problem, choose the best algorithm to solve it.

Algorithm Selection

Portfolio

Algorithm 1  Algorithm 2  Algorithm 3

Training Instances

Instance 1  Instance 2  Instance 3

Feature Extraction

Algorithm Selection

Performance Model

Instance 4: Algorithm 2
Instance 5: Algorithm 3
Instance 6: Algorithm 3

Feature Extraction

Instance 4
Instance 5
Instance 6
Algorithm Portfolios

▷ instead of a single algorithm, use several complementary algorithms
▷ idea from Economics – minimise risk by spreading it out across several securities
▷ same for computational problems – minimise risk of algorithm performing poorly
▷ in practice often constructed from competition winners

“algorithm” used in a very loose sense
  ▶ algorithms
  ▶ heuristics
  ▶ machine learning models
  ▶ ...
Parallel Portfolios

Why not simply run all algorithms in parallel?
▷ not enough resources may be available/waste of resources
▷ algorithms may be parallelized themselves
▷ memory contention
Building an Algorithm Selection System

▷ most approaches rely on machine learning
▷ train with representative data, i.e. performance of all algorithms in portfolio on a number of instances
▷ evaluate performance on separate set of instances
▷ potentially large amount of prep work
Key Components of an Algorithm Selection System

▷ feature extraction
▷ performance model
▷ prediction-based selector/scheduler

optional:
▷ presolver
▷ secondary/hierarchical models and predictors (e.g. for feature extraction time)
Types of Performance Models

Classification Model

Regression Models

Pairwise Classification Models

Pairwise Regression Models

Instance 1: Algorithm 2
Instance 2: Algorithm 1
Instance 3: Algorithm 3...
Benchmark Library – ASlib

▷ currently 29 data sets/scenarios with more in preparation
▷ SAT, CSP, QBF, ASP, MAXSAT, OR, machine learning...
▷ includes data used frequently in the literature that you may want to evaluate your approach on
▷ performance of common approaches that you can compare to
▷ http://aslib.net


http://larskotthoff.github.io/assurvey/
Algorithm Configuration
Algorithm Configuration

Given a (set of) problem(s), find the best parameter configuration.
Parameters?

▷ anything you can change that makes sense to change
▷ e.g. search heuristic, variable ordering, type of global constraint decomposition
▷ not random seed, whether to enable debugging, etc.
▷ some will affect performance, others will have no effect at all
Automated Algorithm Configuration

▷ no background knowledge on parameters or algorithm
▷ as little manual intervention as possible
  ▷ failures are handled appropriately
  ▷ resources are not wasted
  ▷ can run unattended on large-scale compute infrastructure
Algorithm Configuration

Parameter domains & starting values

Configurator

Calls with different parameter settings

Configuration scenario

Target algorithm

Solves

Problem instances

Returns solution cost

General Approach

▷ evaluate algorithm as black box function
▷ observe effect of parameters without knowing the inner workings
▷ decide where to evaluate next
▷ balance diversification/exploration and intensification/exploitation
When are we done?

- most approaches incomplete
- cannot prove optimality, not guaranteed to find optimal solution (with finite time)
- performance highly dependent on configuration space

→ How do we know when to stop?
Time Budget

How much time/how many function evaluations?

▷ too much → wasted resources
▷ too little → suboptimal result
▷ use statistical tests
▷ evaluate on parts of the instance set
▷ for runtime: adaptive capping
Grid and Random Search

▷ evaluate certain points in parameter space

Model-Based Search

- evaluate small number of configurations
- build model of parameter-performance surface based on the results
- use model to predict where to evaluate next
- repeat
- allows targeted exploration of new configurations
- can take instance features into account like algorithm selection

Model-Based Search

performance

actual model

parameter values
Model-Based Search Example

Iter = 1, Gap = 1.9909e−01

Graph showing iterative search process with plotted points and lines for different iterations.
Model-Based Search Example
Model-Based Search Example

Iter = 3, Gap = 1.9909e−01
Model-Based Search Example

Iter = 4, Gap = 1.9992e−01
Model-Based Search Example

Iter = 5, Gap = 1.9992e−01
Model-Based Search Example

Iter = 6, Gap = 1.9996e−01
Model-Based Search Example

Iter = 7, Gap = 2.0000e−01
Model-Based Search Example

Iter = 8, Gap = 2.0000e−01
Model-Based Search Example

Iter = 9, Gap = 2.0000e−01
Model-Based Search Example

Iter = 10, Gap = 2.0000e−01

The graph shows the model-based search process with the following types:
- **y**: Continuous variable
- **yhat**: Predicted variable
- **ei**: Error term

The graph illustrates the search process over iterations, with points indicating the changes in values for each type at different iterations.
Benchmark Library – AClib

- ASP, MIP, planning, machine learning, ...
- 4 algorithm configuration tools from the literature already integrated
- https://bitbucket.org/mlindauer/aclib2

Outlook
Quo Vadis, Software Engineering?
Quo Vadis, Software Engineering?

Meta-Algorithmics in the Physical Realm – AI and Lasers
Tools and Resources

LLAMA  https://bitbucket.org/lkotthoff/llama
SATzilla http://www.cs.ubc.ca/labs/beta/Projects/SATzilla/
iRace http://iridia.ulb.ac.be/irace/
mlrMBO https://github.com/mlr-org/mlrMBO
SMAC http://www.cs.ubc.ca/labs/beta/Projects/SMAC/
Spearmint https://github.com/HIPS/Spearmint
TPE https://jaberg.github.io/hyperopt/

autofolio https://bitbucket.org/mlindauer/autofolio/
Auto-WEKA http://www.cs.ubc.ca/labs/beta/Projects/autoweka/
Auto-sklearn https://github.com/automl/auto-sklearn
Summary

Algorithm Selection choose the best algorithm for solving a problem

Algorithm Configuration choose the best parameter configuration for solving a problem with an algorithm

▷ mature research areas
▷ can combine configuration and selection
▷ effective tools are available
▷ COnfiguration and SElection of ALgorithms group COSEAL
  http://www.coseal.net

Don’t set parameters prematurely, embrace choice!
I’m hiring!

Several funded graduate positions available.