Seminar
Automated Parameter Tuning and Algorithm Configuration

Frank Hutter

Emmy Noether Research Group on
Optimization, Learning, and Automated Algorithm Design

Today's class

- Overview of seminar
- Introduction to seminar topic
- Brief description of available papers
- Brief round of introductions
- Tips for giving a good presentation

Overview of the course

Seminar

- Open to BSc, MSc, and even PhD students
- Worth 4 ECTS credits

Meeting times

- Weekly, Friday 12:00-14:00 c.t.
- 8 slots: April 26, May 3, May 10, May 17, May24,
  May 31, June 7, June 14, June 21, June 28, July 5, July 12, July 19

Mechanics

- We discuss research papers (in English; clarifications in German OK)
- You read each paper that is presented
- You present one paper and lead the discussion for that paper
- If we have ≥ 12 committed participants
  - We'll form teams of 2 (one team of 3 if we have an odd number)
- Grades: combination of all aspects of the course

Your part in the course

For one paper:

- Understand it **in detail**
- Present the paper and lead the discussion;
  receive anonymous feedback from your peers right after class
- End of term: write a report about the paper or a related topic;
  receive anonymous reviews from your peers

For each paper being presented:

- Write a brief summary and formulate some questions
- Attend the presentation
- Participate in a **lively discussion** about the paper
- Give anonymous constructive feedback to the presenter(s) right after class

End of term: write an anonymous review for 3 reports

Warning:

- This course will be more work than a standard block seminar
- But you'll also get more out of it

In detail: preparation for “your” paper

Understand it **in detail**

- Usually requires reading up on some background material
- Often requires downloading the paper’s code and running it

Plan your presentation (it should take 35-40 minutes)

- What you will present (including background from other papers!)
- What you will skip and why
- Outline: **hierarchical** bullet points, with time budget for each point
- Send to paper’s advisor max. 2 weeks before presentation
- Meet with your advisor to discuss the plan & then **adjust it**

Make your slides

- Send to paper’s advisor max. 1 week before presentation
- Meet with your advisor to discuss the slides & then **adjust them**

Practice, practice, practice!

In detail: more about "your paper"

Present the paper and lead the discussion

Open scientific **discussion**

- Strengths & weaknesses of the paper
  - Typically, not everything is perfect
- Relation to other papers we covered
- Interesting future work

Write a **report** about the paper or a related topic

- In LaTeX (because you have to learn it at some point)
- If we have teams, this will be more involved, e.g.
  - run the optimization procedure on some other interesting data
  - compare an optimization procedure against a different one
  - extensive literature review

In detail: preparation for other papers

Send to the paper's advisor max. 2 days before presentation:

- Brief paper summary (one paragraph)
  - Main contributions
  - **In your own words**, non-specialized language
  - Purpose: learn to **concisely & accurately summarize** work that you don't understand in every detail
- Three questions
  - E.g., about
    - something you found unclear
    - how the work relates to something else we covered before
    - any potential problems you noticed
  - Purpose: set up our discussion about the paper

Advisor accepts/rejects summaries & questions

- Max 10% missed or rejected summaries, or you won't pass

What you'll learn in this course

Research skills

- Reading and understanding a specialized research paper
- Exploring the literature for related work & background material
- Assessing strengths & weaknesses of research papers
- Academic writing
- If we have teams: hands-on experience with
  getting someone's research code to run

Soft-skills

- Giving a good oral presentation
- Leading a discussion
- Giving constructive feedback
- Receiving feedback & using it to improve shortcomings
- Communication in English
- If we have teams: team work

The next steps

TODO after this class:

- Browse available papers
- Select a partner with **similar interests**
- Send email to seminar@fhutter.de by Tuesday night, containing
  - The name of your selected partner (in case we form teams)
  - A ranked list of 5 papers you'd be interested in (hopefully overlapping with your partner's list), and reasons why you're interested in them
  - A ranked list of all available time slots (& and hard constraints)
  - Send this email **if and only if you commit** to taking the seminar

We will assign the papers on Wednesday

- If used, 2 early slots (April 26, May 3) get special treatment

Questions about the mechanics?

Today's class

- Overview of seminar
- Introduction to seminar topic
- Brief description of available papers
- Brief round of introductions
- Tips for giving a good presentation

The Big Picture

"Civilization advances by extending the number of important operations which we can perform without thinking of them"
(Alfred North Whitehead)

My group's research agenda:
use machine learning & optimization to automate (parts of) algorithm design

This seminar:
automated methods for tuning the parameters of an algorithm to optimize its performance in practice

Blackbox function optimization

- Optimize a function f over a domain X: $\boxed{\min_{x \in X} f(x)}$

  – **Only mode of interaction: query f(x) at arbitrary $x \in X$**

$$x \rightarrow \blacksquare \rightarrow f(x)$$

- Special characteristics
  – No gradient information
  – Typically, f is not convex
  – Evaluations can be noisy:
    we observe $f(x) + \varepsilon$, with random ε

Generality of the problem definition

- Function can be **implicitly defined**
  - All you need is a way to evaluate your function with different input parameters $x \in X$
  - E.g., run an algorithm with parameters x and measure its performance
  - E.g., run a physical process with control parameters x and measure a quantity to be optimized
- General **performance measures**
  - "Anything that can be measured"
  - E.g., algorithm runtime, approximation error, agreement between output and target output, solution quality, energy consumption, memory consumption, latency, …

Algorithm parameters

Decisions that are **left open during algorithm design**

- E.g., real-valued thresholds
- E.g., which heuristic or which optimizer to use

Parameter types

- Continuous, integer, ordinal
- **Categorical**: finite domain, unordered, e.g. {A,B,C}

Parameter space has **structure**

- E.g. parameters of sub-algorithm A are only active if A is used

Parameters give rise to a **space of algorithms**

- Many “configurations” (e.g. 10^{47})
- Configurations often yield qualitatively different behaviour

$\rightarrow$ Algorithm configuration (as opposed to “parameter tuning”)

The Algorithm Configuration Problem

Definition

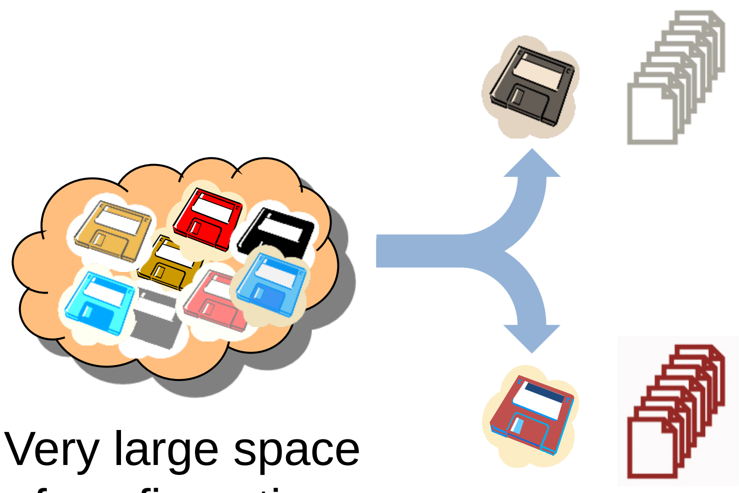
- Given:
  - Runnable algorithm $\mathcal{A}$ with configuration space $\boldsymbol{\Theta} = \Theta_1 \times \cdots \times \Theta_n$
  - Distribution D over problem instances Π
  - Performance metric $m : \boldsymbol{\Theta} \times \Pi \to \mathbb{R}$
- Find:

$$\boxed{\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathbb{E}_{\pi \sim D}[m(\boldsymbol{\theta}, \pi)]}$$

Motivation

Customize versatile algorithms
for different application domains

- Fully automated improvements
- Optimize speed, accuracy, memory, energy consumption, …



Very large space
of configurations

Generalization of performance (1)

- Crucial question in practice:
  which distribution do you want to optimize for?
  - Goal of parameter tuning: solve future problems better
    $\rightarrow$ need **distribution over future problems**
  - Example 1: quickly sort a single list with 1 billion entries
    vs. quickly sort all possible lists with 1 billion entries
  - Example 2: shortest path finding on a compute cluster
    vs. shortest path finding on the iPhone
  - Example 3: learning a regression model that works well on my 20 data points
    vs. learning a model that will generalize to new data points

Generalization of performance (2)

The dark ages

- Student tweaks the parameters manually on 1 problem until it works
- Supervisor may not even know about the tuning
- Results get published without acknowledging the tuning
- Of course, the approach **does not generalize**

A step further

- Optimize parameters on a training set
- Evaluate generalization on a test set

What you should do: also avoid “**peeking**” at the test set

- Put test set into a vault (i.e., never look at it)
- Split training set again into training and validation set
- Use validation set to assess generalization during development
- Only use test set in the very end to generate results for publication

Theory of blackbox optimization

Continuous optimization: $X = R^n$

- **Different assumptions on f** (e.g., smoothness, slope around the optimum, etc) give rise to different algorithms with **different convergence rates**
- Hot topic in theoretical machine learning

Discrete optimization, e.g., $X = \{0,1\}^n$

- Black-box function optimization is NP-hard
- **Under certain assumptions** on f (e.g., submodularity) **efficient approximations are possible**

Much work remains to be done:

- In practice: constants matter, need good solutions quickly, no need to prove optimality
- **TODO: bridge the gap** between theory and practice

Research from several fields is converging

Until recently: **each community used their own methods**

- Evolutionary algorithms to tune evolutionary algorithms
- Gradient-based optimizers to tune gradient-based optimizers
- Machine learning to tune machine learning algorithms
- Local search to tune local search

We advocate: **choose the right optimizer for the task at hand**

- Are the parameters discrete, continuous, or mixed?
- How many parameters are there?
- How much noise is there?
- Etc …

This seminar

Foundations

- Statistics: experimental design, statistical tests
- Machine learning: regression, stochastic processes
- Optimization: global, stochastic, mixed continuous/discrete
- AI: local search, population-based methods
- All of them **"on-the-fly"**, in the context of parameter tuning

Applications

- AI planning
- Formal verification
- Robotics
- Machine Learning
- Graphics
- Parallel Computing
- Algorithm Engineering
- High-Performance Computing
- (Additional applications welcome)

Questions about the seminar topic ?

Today's class

- Overview of seminar
- Introduction to seminar topic
- Brief description of available papers
- Brief round of introduction
- Tips for giving a good presentation

Papers on Foundations

Foundations (papers introducing methods for parameter tuning / algorithm configuration)

- Bayesian optimization for continuous blackbox functions: EGO. *Efficient Global Optimization* of Expensive Black-Box Functions. Journal of Global Optimization, 1998. A classic.
- (Improved) Bayesian optimization for continuous blackbox functions: GPGO (code). Gaussian Processes for Global Optimization. Proceedings of LION 2009: Intl. Conference on Learning and Intelligent Optimization. Various improvements over EGO.
- Evolutionary strategies for continuous blackbox functions: CMA-ES (code). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. Proceedings of CEC 1996: Intl. Conference on Evolutionary Computation. Relatively theoretical, basis for a very successful algorithm.
- Local search for discrete algorithm configuration: ParamILS (code). Automatic Algorithm Configuration based on Local Search. Proceedings of AAAI 2007: Twentysecond Conference on Artificial Intelligence. First approach for algorithm configuration with many discrete parameters.
- Genetic algorithms for algorithm configuration: GGA (code). A Gender-Based Genetic Algorithm for the Automatic Configuration of Algorithms. Proceedings of CP 2009: Principles and Practice of Constraint Programming. Exploits some parallelism.
- Bayesian optimization for algorithm configuration: SMAC (code). Sequential Model-Based Optimization for General Algorithm Configuration. Proceedings of LION 2011: Intl. Conference on Learning and Intelligent Optimization. Improvement & generalization of EGO for algorithm configuration.
- Racing strategies for algorithm configuration: Iterated F-Race (code). F-Race and Iterated F-Race: an overview. Technical report, then book chapter in Experimental methods for the analysis of optimization algorithms. Focusses on selecting the best approach under noise, not so much on the search.

Papers on Applications (1)

Applications (papers applying parameter tuning / algorithm configuration)

- Planning 1: FD-Autotune: Domain-Specific Configuration using Fast Downward, Proceedings of ICAPS-PAL 2011: ICAPS Workshop on Planning and Learning.
- Planning 2: Learning Portfolios of Automatically Tuned Planners, Proceedings of ICAPS 2012: International Conference on Planning and Scheduling.
- Formal verification: Boosting Verification by Automatic Tuning of Decision Procedures, Proceedings of FMCAD 2007: Formal Methods in Computer-Aided Design.
- Robotics 1: Automated Gait Optimization using Gaussian Process Regression, Proceedings of IJCAI 2007: International Joint Conference on Artificial Intelligence.
- Robotics 2: Adapting Control Policies for Expensive Systems to Changing Environments, Proceedings of IROS 2011: International Conference on Intelligent Robots and Systems.
- Machine Learning 1: Practical Bayesian Optimization of Machine Learning Algorithms, Proceedings of NIPS 2012: Neural Information Processing Systems.
- Machine Learning 2: Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms, ArXiv preprint, 2012.
- Computer Vision: Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures, Proceedings of ICML 2013: International Conference on Machine Learning.
- Graphics: A Bayesian Interactive Optimization Approach to Procedural Animation Design, ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2010.

Papers on Applications (2)

- Parallel Computing 1: High-level optimization via automated statistical modeling, Proceedings of PPOPP 1995: Principles and Practice of Parallel Programming.
- Parallel Computing 2: Automated Empirical Optimization of Software and the ATLAS project, Parallel Computing Journal, 2001.
- Algorithm Engineering: Optimizing Sorting with Genetic Algorithms, Proceedings of CGO 1995: International Symposium on Code Generation and Optimization.
- High-Performance Computing: Automatic tuning of inlining heuristics. In Proceedings of SC 2005: ACM/IEEE Conference on Supercomputing.

Today's class

- Overview of seminar
- Introduction to seminar topic
- Brief description of available papers
- Brief round of introduction
- Tips for giving a good presentation

Introductions

- Some information on yourself
  - Your name
  - Your field of study and semester
  - Why you’re interested in this course & what you hope to get out of it
  - Which papers just caught your eye
- Less than 1 minute per person
- Purpose: get to know each other, maybe find a partner

Today's class

- Overview of seminar
- Introduction to seminar topic
- Brief description of available papers
- Brief round of introduction
- Tips for giving a good presentation

How to give a good presentation

This part is heavily based on the excellent slides by Thomas Brox, with permission.

Good scientific behavior

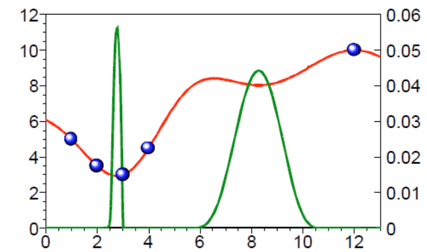
1. Never present other people’s work as your own
   - Never copy-paste (even critical when copying from your own work → self-plagiarism)
   - Clearly mention the material you used for your work (e.g. code, data, papers; if unpublished material, ask before you use it)
   - Say explicitly what is your contribution

2. Never report false scientific results
   - Do not fake data to get the results you want (of course!)
   - Avoid situations that could easily lead to false results
     - Document what you did
     - Make sure comparisons are fair
     - Double check if there is a mistake particularly when results are surprisingly good

- This holds for this seminar, but also for reports, theses, papers, grant proposals, interviews, personal communication

Examples of how to cite others' work

- Quotes from other work should have quotation marks:
  - *X and Y [12] define this problem as follows : "…"*
- Provide references for figures



Source: Jones et al [1998]

- Mention & clarify contributions from others:
  - *The results reported in this section are based on a joint project with X. While he had the main idea and wrote all the code, I was responsible for the experiments.*

    For our implementation, we built upon the source code provided by X [13].

Consequences of bad scientific behavior

- If you cheat in an exam, it will be marked as “failed”
- In severe cases, you can get exmatriculated!
- You can get sued for copyright violations
- You can lose your academic degrees even years after your misbehavior
- You can lose the right to submit grant proposals
- You can lose your job

➔ Never cheat or plagiarize on purpose,
clearly mark your references,
adopt best practices for avoiding mistakes

How to give a good presentation

Communication is hard work.

The work can be done either on the side of the sender or on the side of the receiver.

Importance of good presentation skills

- You'll have to give a lot of presentations in your life (both in academia and industry)
- These presentations can decide whether
  - You get a job
  - Your favourite project gets funded
  - You get the resources you need
  - …
- Presentation skills and communication skills go together
  - Improving one will help with the other

Getting your points across

- What matters is what your audience gets (not which points you "covered")
  - Often, the audience is not as interested in the topic as you
  - You'll have to tell them why they should be care
  - If nobody cares or understands **it's typically your own fault**
- At least the key points must get across to everyone
  - Some details may only be for experts, that's OK

Rule #1: Structure is key

- **High level to low level to high level**
  - Catch your audience's attention
  - Then tell them what you'll tell them and why they should care (priming)
  - Then tell it to them
  - Then tell them what you just told them
- Make transitions clear, don't forget the "**meta-talk**"
  - E.g., "In order to explain X, first I'll need to explain Y"
  - E.g., "Now that we've seen X and Y, we have the ingredients to do Z"
  - Remind the audience where you are in the talk,
    e.g. using a re-occurring outline slide
  - Use meaningful titles
- Don't get lost in details
  - In case of doubt **leave out some details**
  - To scientists, some detail is often important; you can use a "T-structure": combine broad coverage of a topic with depth about one aspect

Rule #2: present in pictures

- Slides full of text are hard to follow
  - The audience will read and not listen to you
  - Reduce text, use more images

Comparing default vs. optimized parameters

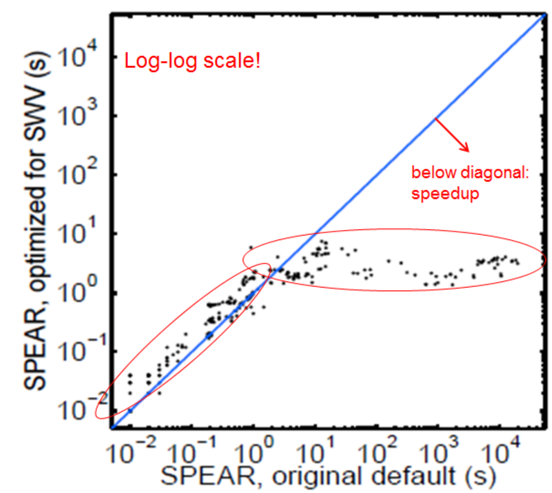
- For easy instances, default parameters perform a little better
- For hard instances, optimized parameters perform much better

- Default runtime ranges from 0.01s to 1day
- Optimized runtime ranges from 0.01s to 10s

- For instance harder than about 5s the optimized parameters perform better

49

Comparing default vs. optimized parameters



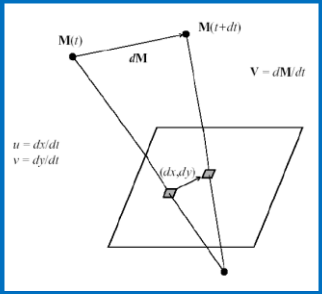
48

Rule #3: Have readable slides

- Can you read this text?

- Also from the back? Remember, the contrast and resolution of your laptop is usually much better than that of the projector

- Sometimes the font size is too tiny



- Sans-serif fonts are easier to read from the back than serif-fonts

Also still quite common is yellow text on white ground

You see this even more often in graphs

Meke sure tere are no typos in yur slides; it’s so unprofessional und unnecessary

Size up figures to use most of the slide. A slide does not need a big frame.

Rule #4: Practice

- Prepare what you want to say, **do not improvise!**
  - Have a **time budget** for each part
  - Write down bullet points of what you want to say in each part
    - Say it out loud a few times & check the timing for the part
    - Then do the part a few times without looking at your notes
  - Write out exactly what you want to say in the first minute and as a closing statement
    - You are most nervous in the beginning
    - You want to end pointedly (also, with a final “Thank you”)
    - Practice first minute and closing statement at least 10 times

- Then put it all together
  - Do the transitions work?
  - Always get stuck at the same point? Change that point!
  - Don’t speak too fast! Speaking too slowly is almost impossible

Rule #5: control you technical equipment

- Prepare and test your equipment before the talk (if possible)
- Checklist:
  - Does your laptop work with the projector?
  - For Mac-Users: do you have the right dongle?
  - Do all videos show properly?
  - Internet connection switched off?
  - Screen saver switched off?
  - Desktop free of too personal items?
  - Enough battery or laptop plugged in?
- Use laser pointer (only)
  for directing attention

Rule #6: Behave naturally

- Keep eye contact with the audience; don't turn your back
  - But do <u>not</u> wonder what they might think of your presentation! (now it's too late)
- Relax
  - Breathing in & out deeply once can help
  - Practice helps building confidence
- Answering questions:
  - First listen to the whole question carefully; don't interrupt
  - Long/multiple questions: take bullet point notes
  - Think about how you can best answer a question before you answer it
  - Give short and precise answers

Rule #7: Adapt your talk to your audience

- The paper you are presenting is written for a specialized research community
- But your audience has a different background
  - Especially for application papers
    - You will need to cover the necessary background
    - We'll be parameter tuning experts – don't bore us with what we know

- For other presentations
  - A talk to the CEO is completely different than one to the tech support group
  - A talk applying method X to problem Y is completely different when you're talking to community studying X or Y

Rule #8: Learn from the mistakes of others

- You cannot follow someone's talk?
- You are totally bored?
- You are irritated by a certain behavior of the presenter?

→ Analyze what the presenter is doing wrong

→ Make sure to give them (friendly & constructive) feedback and do not make the same mistakes

Giving constructive feedback

- Start with something positive
  - In your own reviews you don't want to hear only negative things, either
  - People are more receptive to criticism after hearing something positive
- Make concrete suggestions
  - Bad example: "The lecture was bad"
  - Good example: "I couldn't follow the math because I couldn't read your handwriting on the board – better use a projector or slides"

Today's class

- Overview of seminar
- Introduction to seminar topic
- Brief description of available papers
- Brief round of introduction
- Tips for giving a good presentation

Reminder: the next steps

TODO after this class:

- Browse available papers
- Select a partner with **similar interests**
- Send email to seminar@fhutter.de by Tuesday night, containing
  - The name of your selected partner (in case we form teams)
  - A ranked list of 5 papers you'd be interested in (hopefully overlapping with your partner's list), and reasons why you're interested in them
  - A ranked list of all available time slots (& and hard constraints)
  - Send this email **only if you want to commit** to taking the seminar

We will assign the papers on Wednesday

- If used, 2 early slots (April 26, May 3) get special treatment