(How) does Machine Learning impact OR?

Marco Lübbecke
Lehrstuhl für Operations Research
RWTH Aachen University

 @mluebbecke

NGB/LNMB Seminar · Lunteren, NL · January 16, 2019

Operations Research | RWTH AACHEN UNIVERSITY

THE AGE OF OPTIMIZATION: SOLVING LARGE-SCALE REAL-WORLD PROBLEMS

GEORGE L. NEMHAUSER

Georgia Institute of Technology, Atlanta, Georgia

(Received September 1993; accepted October 1993)

In the last decade, new advances in algorithms have been as important as the impressive advances in computer technology. Using the new interior-point algorithms and advanced implementations of simplex methods, we can now solve linear programs with more than one million variables and thousands of constraints. Preprocessing and polyhedral theory have yielded at least an order of magnitude improvement in branch-and-bound algorithms for solving mixed integer programs. Moreover, these algorithmic advances have been incorporated in commercially inexpensive software that is readily available, easily portable, and supported by a variety of systems that make it possible for unsophisticated users to input and check their models and obtain understandable outputs. This paper, based on the Morse Lecture given in May 1993 at the TIMS/ORSA meeting in Chicago, begins with some of the modern history of optimization, then surveys some recent developments (illustrating them with an application in the airline industry), and closes with some remarks about the future.

1994

Great strides have been made in the use of optimization. Models with thousands of variables and constraints are being solved regularly and the results are being applied routinely, but only by a relatively small number of organizations. Therefore, we need to do a better job in making optimization easier to use and in making solutions more meaningful to all types of customers. This is our challenge. It involves education, research, and marketing. A rosy future with dramatically increased use of optimization awaits our meeting the challenge successfully.

2019

Google Scholar

Jacques Desrosiers

FOLGEN AKTIV

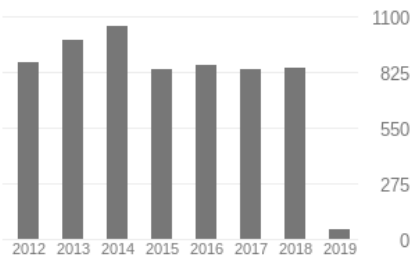
Professor, HEC Montréal
Bestätigte E-Mail-Adresse bei hec.ca - Startseite

Combinatorial optimization Column generation Mathematical programming Vehicle routing Crew scheduling

TITEL	ZITIERT VON	JAHR
A new optimization algorithm for the vehicle routing problem with time windows M Desrochers, J Desrosiers, M Solomon Operations research 40 (2), 342-354	1276	1992
Time constrained routing and scheduling J Desrosiers, Y Dumas, MM Solomon, F Soumis Handbooks in operations research and management science 8, 35-139	1211	1995
Selected topics in column generation ME Lübbecke, J Desrosiers Operations Research 53 (6), 1007-1023	1006	2005

Zitiert von ALLE ANZEIGEN

	Alle	Seit 2014
Zitate	14877	4512
h-index	54	34
i10-index	86	65



Koautoren ALLE ANZEIGEN

Operations Research | RWTH AACHEN UNIVERSITY

2019

Google Scholar

Yoshua Bengio

FOLGEN

Professor, U. Montreal (Computer Sc. & Op. Res.), Mila, CIFAR, CRM, IVADO, REPARTI, GRSNC

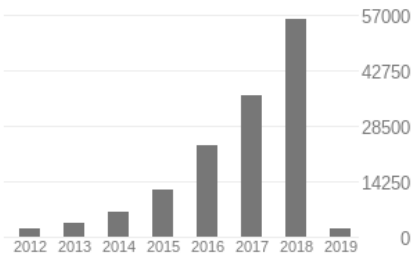
Bestätigte E-Mail-Adresse bei mila.quebec - Startseite

Machine learning deep learning artificial intelligence

TITEL	ZITIERT VON	JAHR
Gradient-based learning applied to document recognition Y LeCun, L Bottou, Y Bengio, P Haffner Proceedings of the IEEE 86 (11), 2278-2324	16148	1998
Deep learning Y LeCun, Y Bengio, G Hinton nature 521 (7553), 436	12253	2015
Generative adversarial nets I Goodfellow, J Pouget-Abadie, M Mirza, B Xu, D Warde-Farley, S Ozair, ... Advances in neural information processing systems, 2672-2680	6756	2014

Zitiert von ALLE ANZEIGEN

	Alle	Seit 2014
Zitate	156106	138169
h-index	133	119
i10-index	413	352

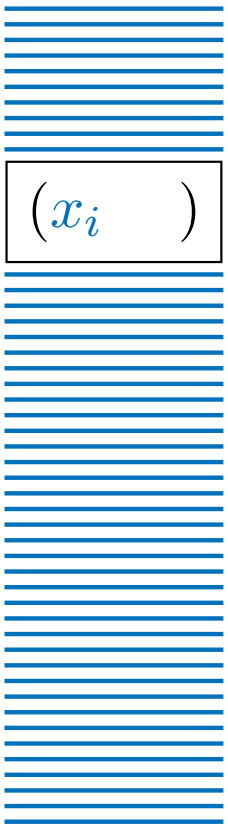


Koautoren ALLE ANZEIGEN

Operations Research | RWTH AACHEN UNIVERSITY

Example: Supervised Learning: Classification

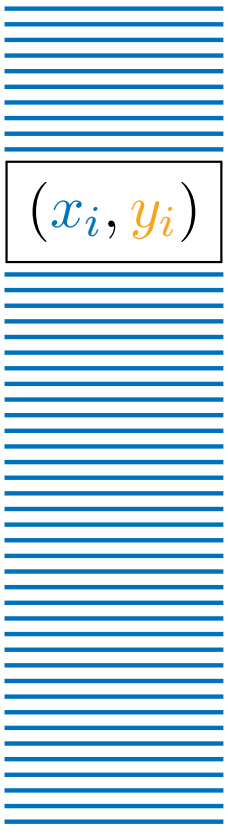
- data $\mathcal{X}$



Example: Supervised Learning: Classification

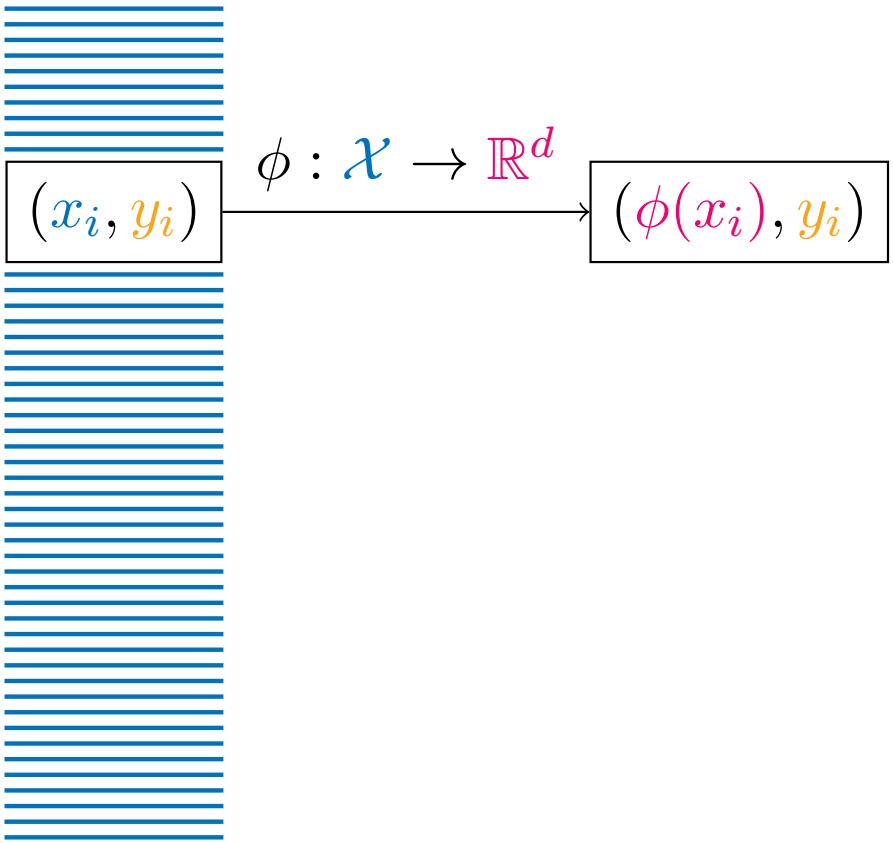
- data $\mathcal{X}$

labels $\mathcal{Y}$



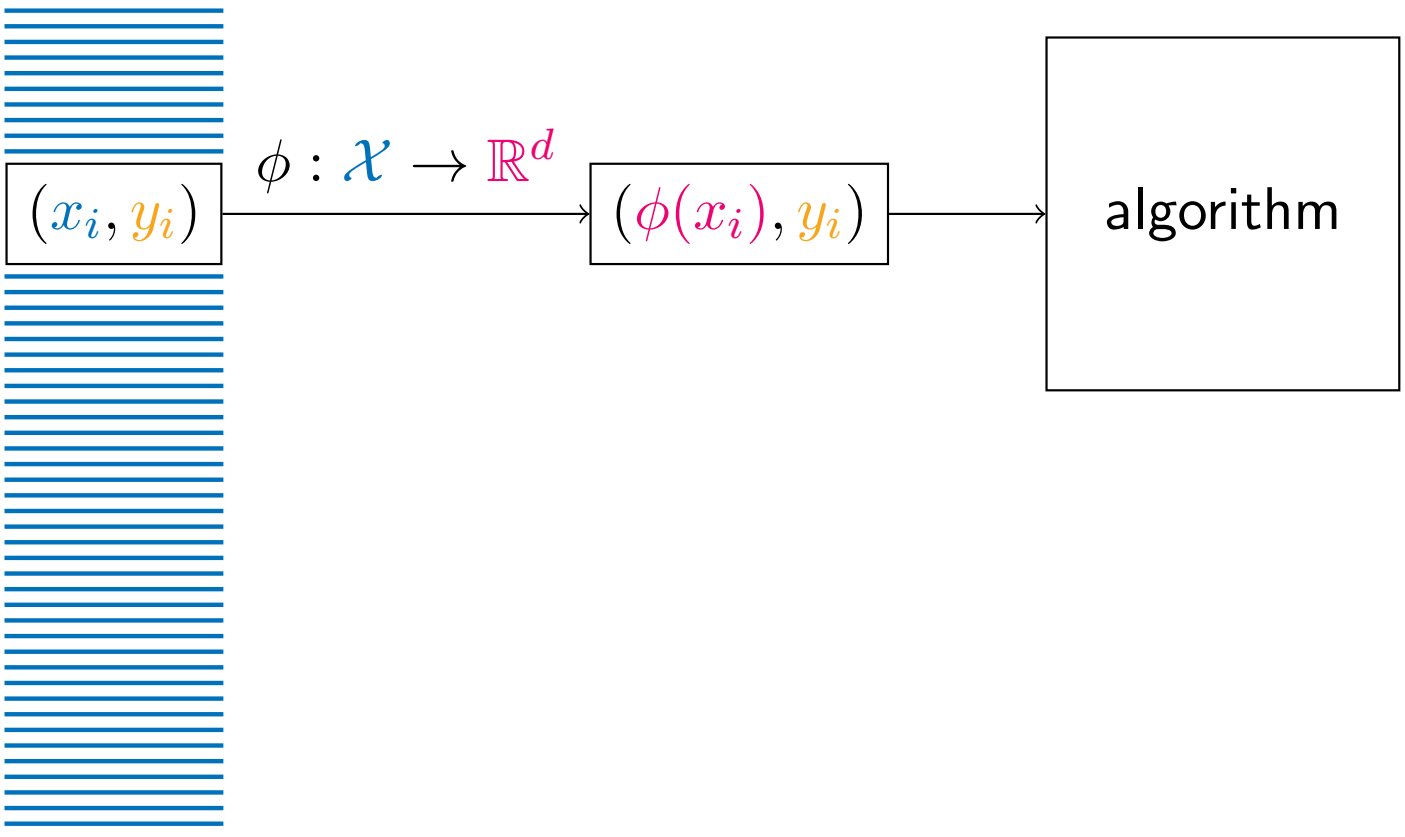
Example: Supervised Learning: Classification

- data $\mathcal{X}$, d features, labels $\mathcal{Y}$



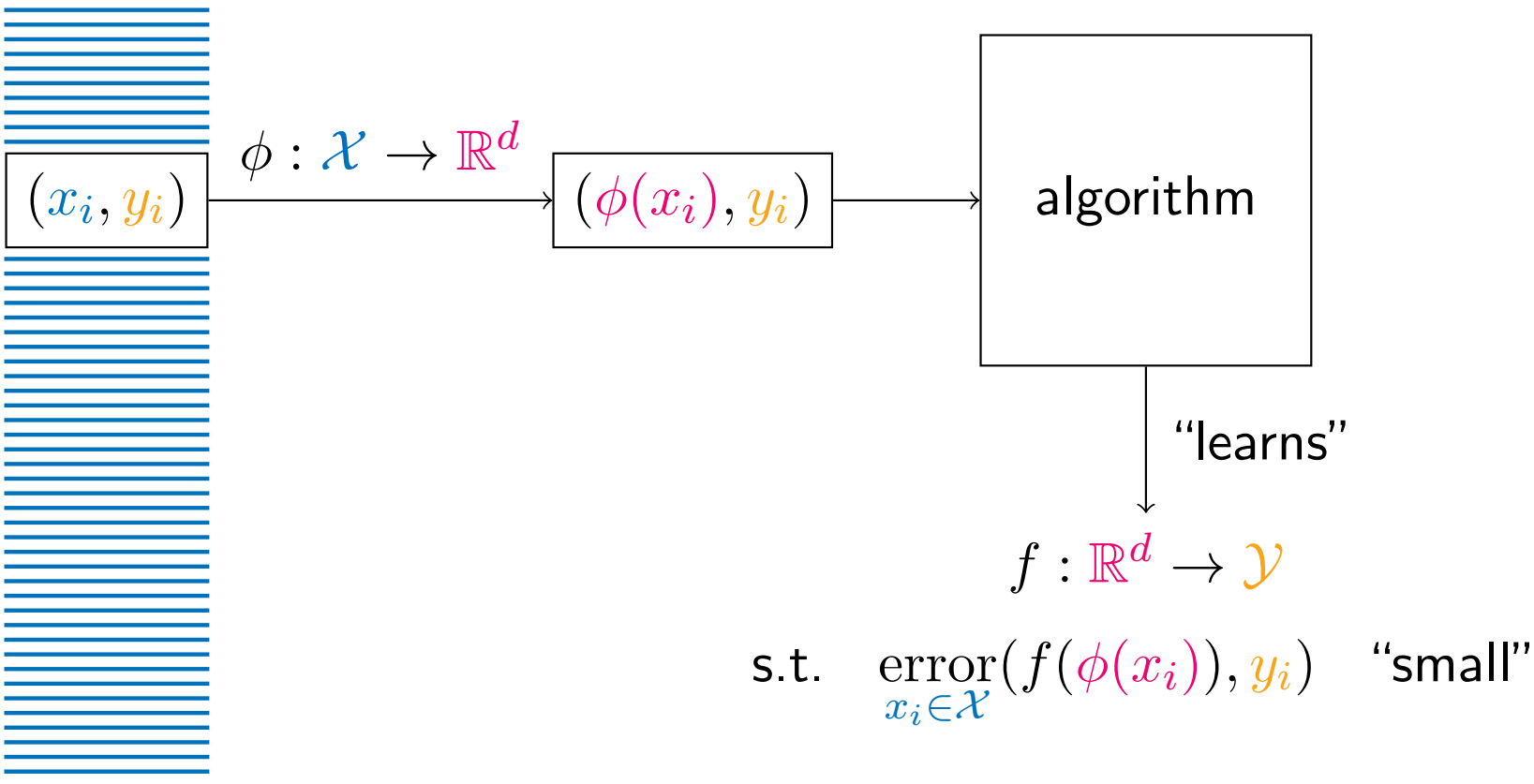
Example: Supervised Learning: Classification

- data $\mathcal{X}$, d features, labels $\mathcal{Y}$



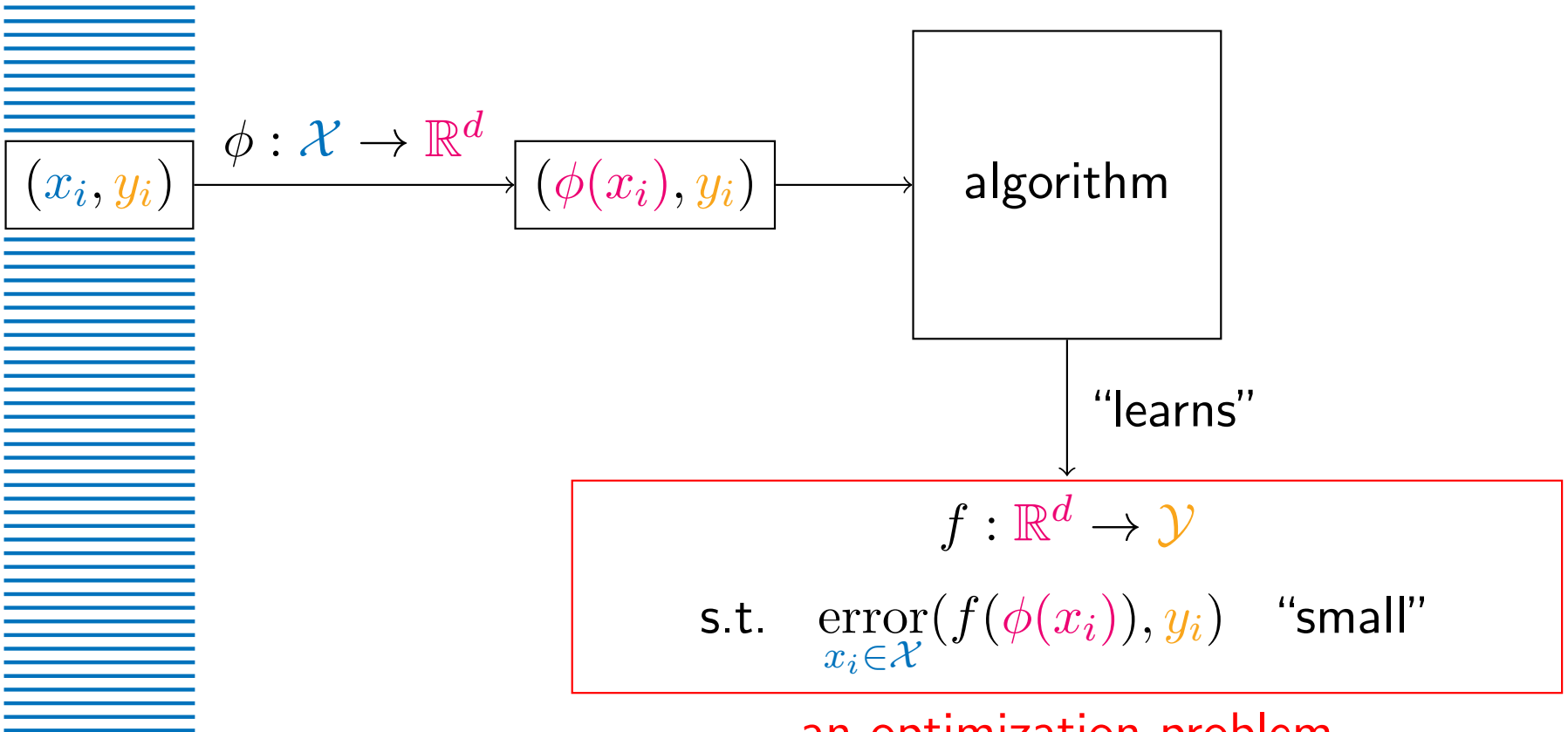
Example: Supervised Learning: Classification

- data $\mathcal{X}$, d features, labels $\mathcal{Y}$



Example: Supervised Learning: Classification

- data $\mathcal{X}$, d features, labels $\mathcal{Y}$

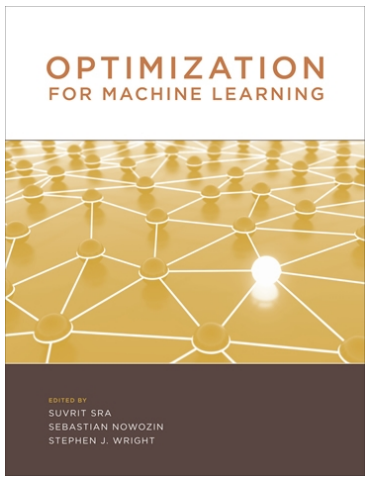


an optimization problem

Optimization naturally appears in ML

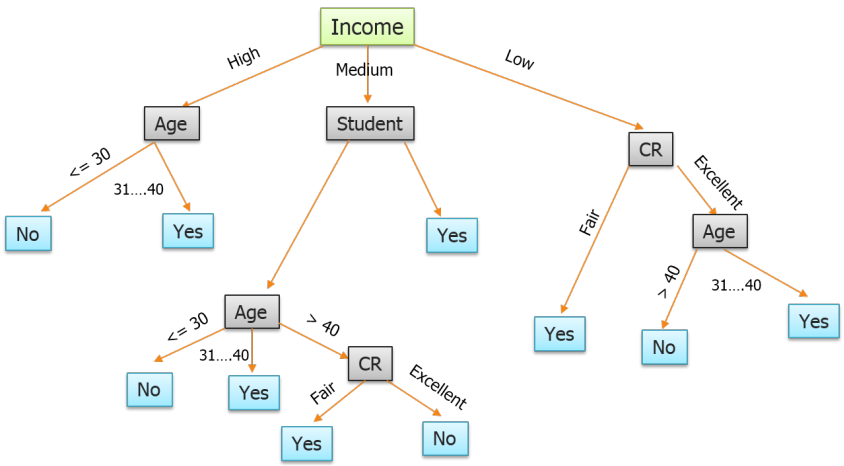
> Optimization lies at the heart of ML. Most ML problems reduce to optimization problems.
>
> — Bennett, Parrado-Hernández (2006)

- minimize e.g., prediction error
- continuous, convex optimization
- discrete, integer optimization



Example: Optimal Classification Trees

- several mixed-integer programs e.g., Bertsimas & Dunn (2017)
- can impose additional structural constraints! Bastert (2018)
- "path-based" integer program Fırat et al. (2018)



source: www.edureka.co/blog/decision-trees

- goals? use few nodes, shallow depth, ...
- improves accuracy over classical CART method

Operations Research | RWTH AACHEN UNIVERSITY

Example: Robustifying against adverserial Inputs

Strong convex relaxations and mixed-integer programming formulations for trained neural networks

Ross Anderson · Joey Huchette · Christian Tjandraatmadja · Juan Pablo Vielma

Abstract We present strong convex relaxations for high-dimensional piecewise linear functions that correspond to trained neural networks. These convex relaxations can be used for a number of important tasks, such as verifying that an image classification network is robust to adversarial inputs, or providing optimality guarantees for decision problems with machine learning models embedded inside (i.e. the "predict, then optimize" paradigm). Our convex relaxations arise from mixed-integer programming (MIP) formulations, and so they can be paired with existing MIP technology to produce provably optimal primal solutions, or to further strengthen the relaxations via cutting planes. We provide convex relaxations for networks with many of the most popular nonlinear operations (e.g. ReLU and max pooling) that are strictly stronger than other approaches from the literature. We corroborate this computationally on image classification verification tasks on the MNIST digit data set, where we show that our relaxations are able to match the bound improvement provided by state-of-the-art MIP solvers, in orders of magnitude less time.

Operations Research | RWTH AACHEN UNIVERSITY

Many Opportunities for Discrete Optimization in ML

- capture combinatorial explosion
- feature selection
- outlier detection
- parameter tuning
- ...

→ check out what the Montréal people are up to

How about the converse Direction?

- "emulating the expert" Bärmann, Martin, Pokutta & Schneider (2017)
- observe a decision maker $\max c_{\text{true}}^T x : x \in X(p)$
- learn their objective function given $(p_t, x_t^*)_{t=1,\dots,T}$
- *online learning*

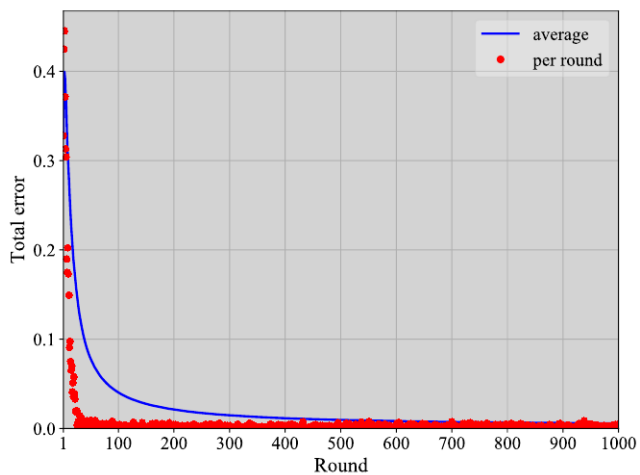


Figure 1. Linear Knapsack problem with $n = 100$ items over

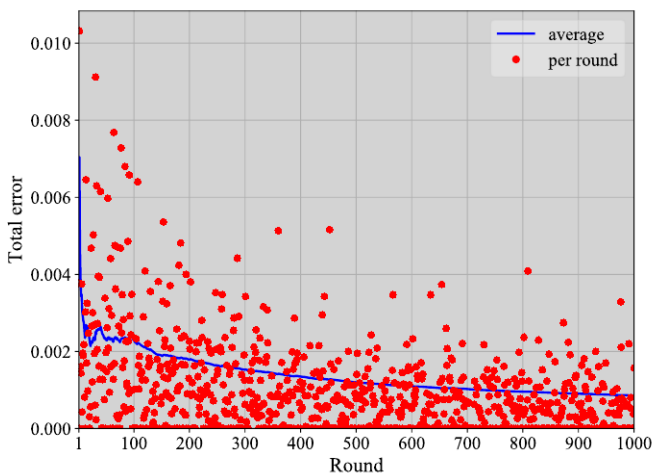


Figure 2. Resource-constrained shortest path problem on a grid

Operations Research | RWTH AACHEN UNIVERSITY

ML may help improving Optimization Algorithms

- e.g., branching in B&B
- *full strong branching* gives locally perfect information

- predict the strong branching score of a variable
- features describe state of a variable
- supervised learning: regression

→ promising proof-of-concept

Marcos Alvarez, Louveaux & Wehenkel (2017)

- survey on ML in branching/searching Lodi & Zarpellon (2017)

→ a lot more research in 2018

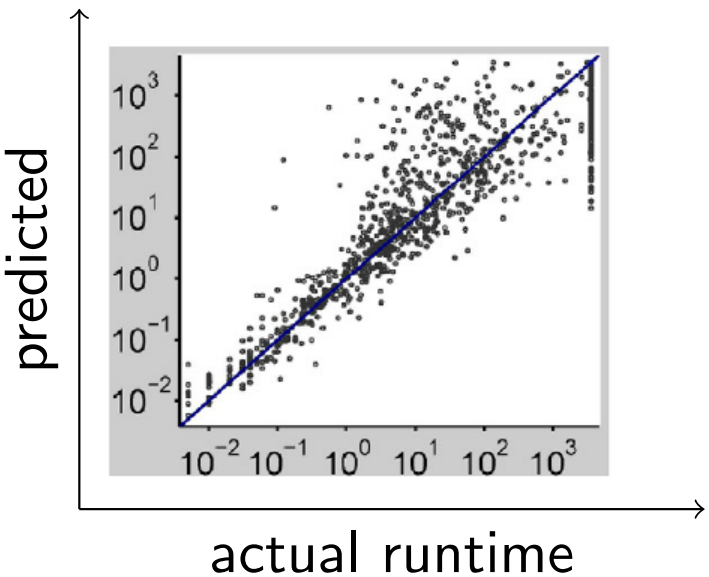
More ML for algorithmic Ingegredients of B&B

- search strategy / node selection He et al. (2014)
- which primal heuristic to run when? Khalil et al. (2017)
- challenge: cut selection

A Progress Bar for Branch-and-Bound?

- predict the runtime of branch-and-bound algorithms

  Hutter, Xu, Hoos & Leyton-Brown (2014)
- CPLEX 12.1 on 1510 publicly available MIPs



Learning the Optimum (for specific Problems)

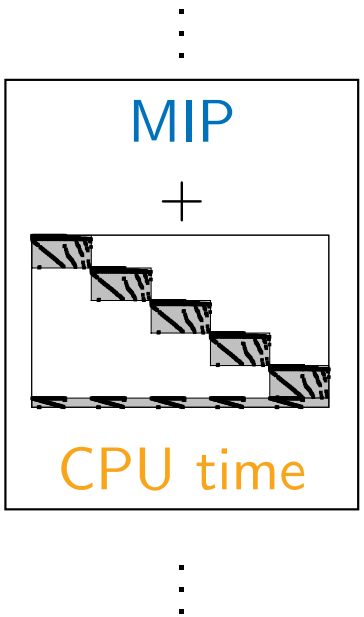
Using OR + AI to predict the optimal production of offshore wind parks: a preliminary study

Martina Fischetti and Marco Fraccaro

Abstract In this paper we propose a new use of Machine Learning together with Mathematical Optimization. We investigate the question of whether a machine, trained on a large number of optimized solutions, can accurately estimate the value of the optimized solution for new instances. We focus on instances of a specific

Learning when to solve a MIP by Branch-and-Price

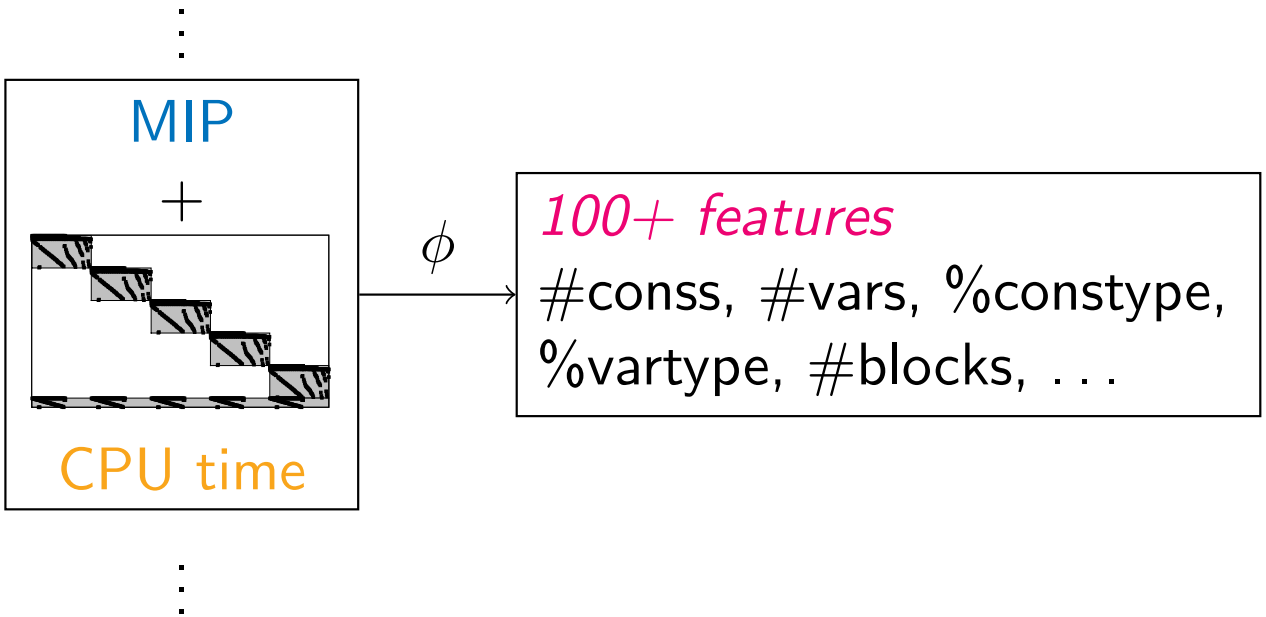
- our MIP solver `GCG` detects many potential DW reformulations



Kruber, L, Parmentier (2017)

Learning when to solve a MIP by Branch-and-Price

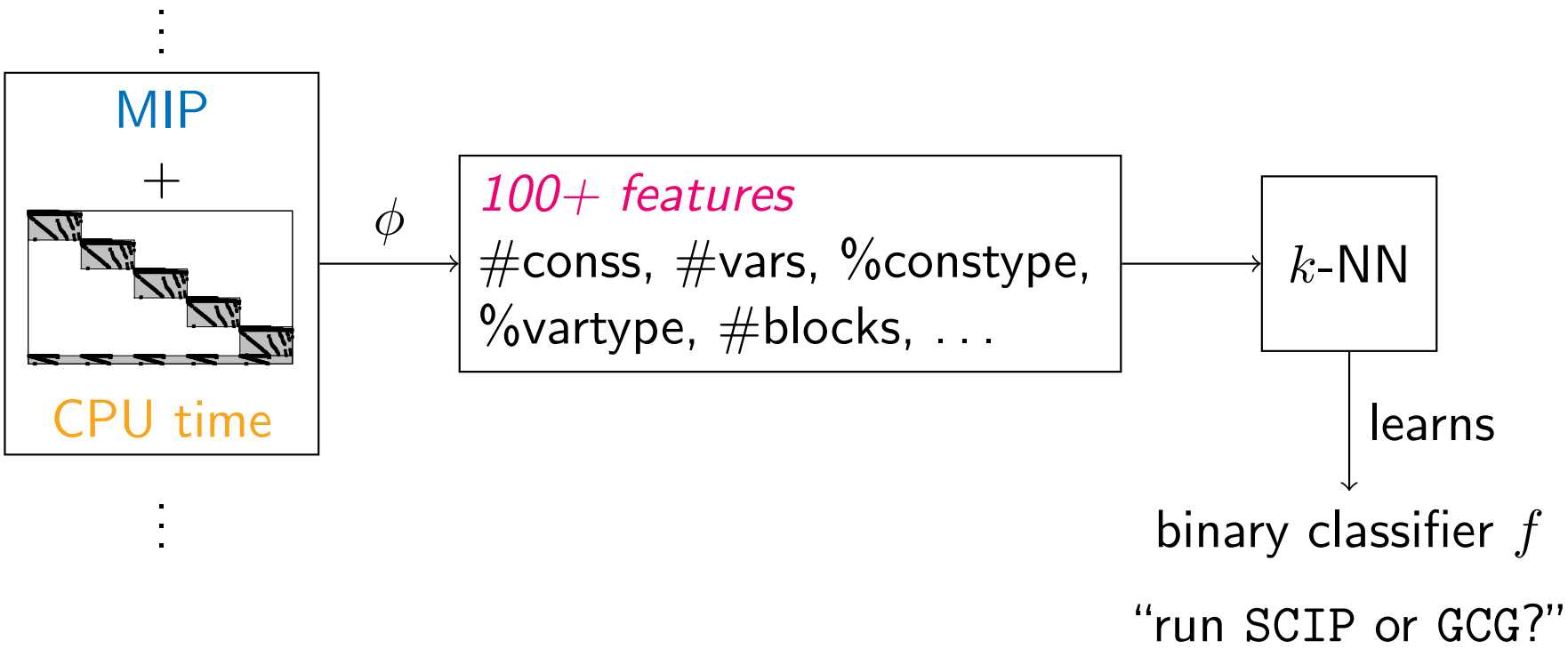
- our MIP solver GCG detects many potential DW reformulations



Kruber, L, Parmentier (2017)

Learning when to solve a MIP by Branch-and-Price

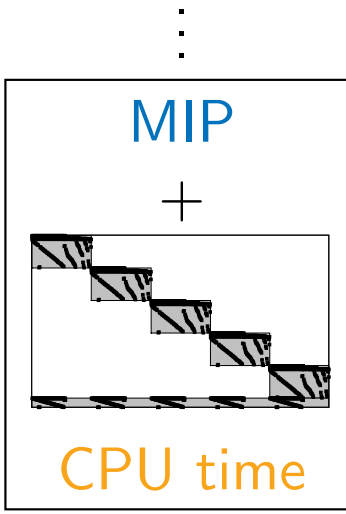
- our MIP solver GCG detects many potential DW reformulations



Kruber, L, Parmentier (2017)

Learning when to solve a MIP by Branch-and-Price

- our MIP solver GCG detects many potential DW reformulations



ϕ

100+ features
#conss, #vars, %constype, %vartype, #blocks, ...

k-NN

learns

binary classifier f

"run SCIP or GCG?"

faster

f	SCIP	GCG
SCIP	69.5%	9.9%
GCG	6.9%	13.7%

Kruber, L, Parmentier (2017)

Learning how to attack MIQPs

- linearize quadratic part or not Bonami et al. (2018)
- "ML to parametrize optimization algorithm"

\+ try to derive structural explanations/insights

What ML Answers can we (Optimizers) expect?

- we get statistical answers $\rightarrow$ not what we are used to see

- we have domain/expert knowledge: e.g., pseudo-costs
- ML may give a better predictor, but no *explanation*

$\rightarrow$ more precisely: no *structural* explanation

- some info can be extracted from most influential features

$\rightarrow$ interpretability is a huge theoretical and practical topic

Decision Making: Machine Learning

> Machine Learning and Artificial Intelligence delivers the most value when you need to make lots of similar decisions quickly.
>
> — Ingo Mierswa, Rapidminer

- simple decisions: e.g., auto correct current word
- solution: often a single score $\rightarrow$ greedy
- keep/learn habits: extrapolate from the past (!)

Typical Example: Predictive Maintenance

source: blog.capterra.com/should-you-invest-in-a-predictive-maintenance-strategy/

Exploit all Options: Prescriptive Maintenance

source: www.siemens.com/press/pool/de/pressebilder/photonews/pn200826/300dpi/pn200826-12_300dpi.jpg

Decision Making: Optimization

> How often can the result of an optimization model be captured in a single variable?
>
> — Ed Rothberg, Gurobi

- solution: not only the objective value!
- complex decisions/plans: e.g., timetables, crew schedules, ...
- global scope: models all (reasonable) interdependencies

Perfect Partners

"Current Standard:" Predictive then Prescriptive Analytics

ML harnesses the bigness of data (the past and present);
Optimization captures the bigness of options (the future).

- e.g., predict arc travel times Gmira et al. (2017)
- e.g., predict availability of charging stations L., (ongoing)

Learning (about) ~~optimal~~ good Solutions

- learn how good (partial) solutions look like
- this may help finding good solutions faster

- reinforcement learning to predict trajectories Le, Liu, Lau (2016)
- predict partial optimal job-shop schedules Shylo, Shams (2018)

→ guide a tabu search

- predict (aggr.) solutions to stochastic ILPs Larsen et al. (2018)

→ high-speed, high-accuracy for particular problems

- deep-learning / container pre-marshalling Tierney et al. (2018)

→ guide a tree search

⋮

MLers think ML can do Optimization

MLers think ML can do Optimization

17 Oct 2018

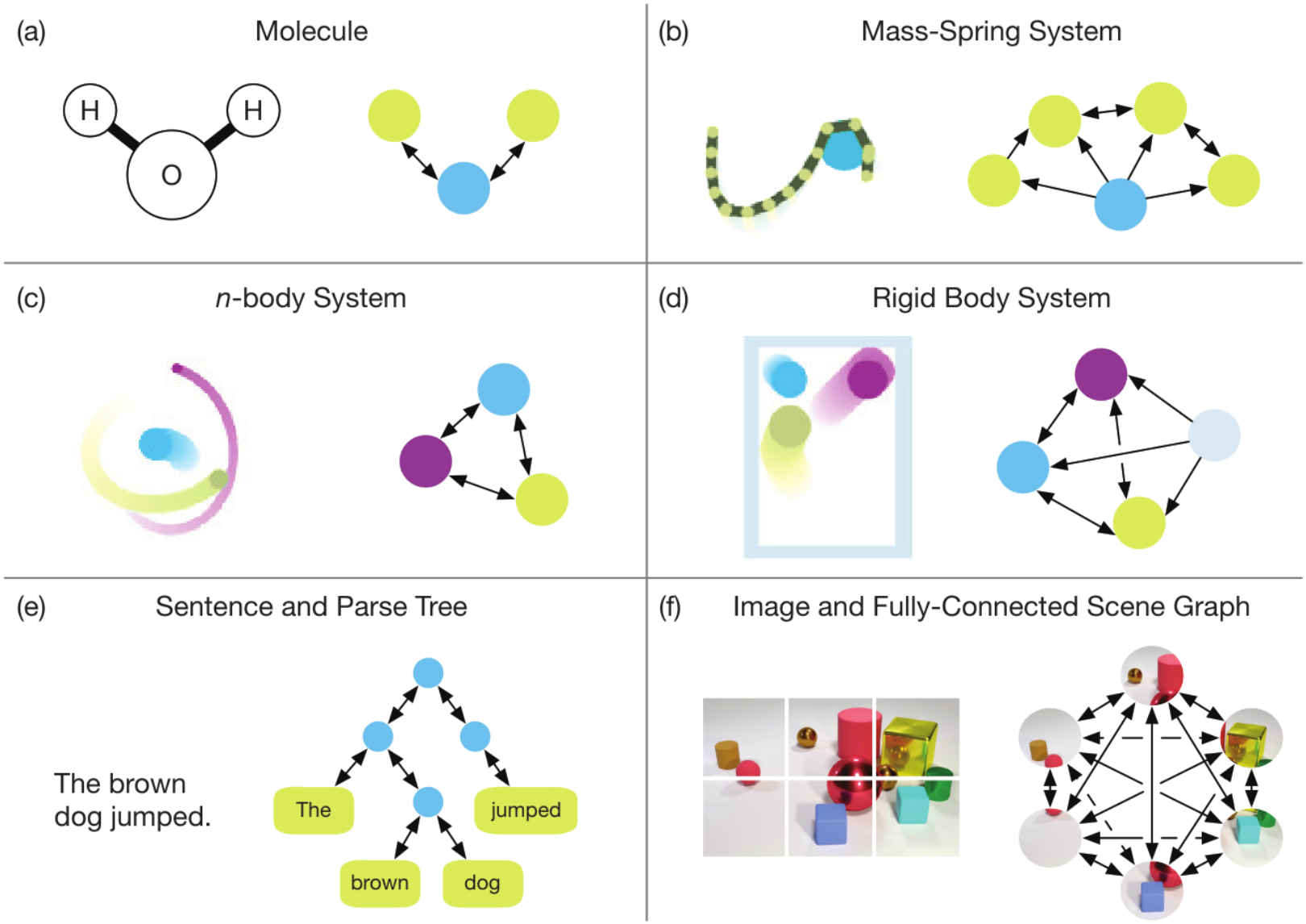
Relational inductive biases, deep learning, and graph networks

Peter W. Battaglia[1]*, **Jessica B. Hamrick**[1], **Victor Bapst**[1], **Alvaro Sanchez-Gonzalez**[1], **Vinicius Zambaldi**[1], **Mateusz Malinowski**[1], **Andrea Tacchetti**[1], **David Raposo**[1], **Adam Santoro**[1], **Ryan Faulkner**[1], **Caglar Gulcehre**[1], **Francis Song**[1], **Andrew Ballard**[1], **Justin Gilmer**[2], **George Dahl**[2], **Ashish Vaswani**[2], **Kelsey Allen**[3], **Charles Nash**[4], **Victoria Langston**[1], **Chris Dyer**[1], **Nicolas Heess**[1], **Daan Wierstra**[1], **Pushmeet Kohli**[1], **Matt Botvinick**[1], **Oriol Vinyals**[1], **Yujia Li**[1], **Razvan Pascanu**[1]

[1]DeepMind; [2]Google Brain; [3]MIT; [4]University of Edinburgh

- open source library `github.com/deepmind/graph_nets`

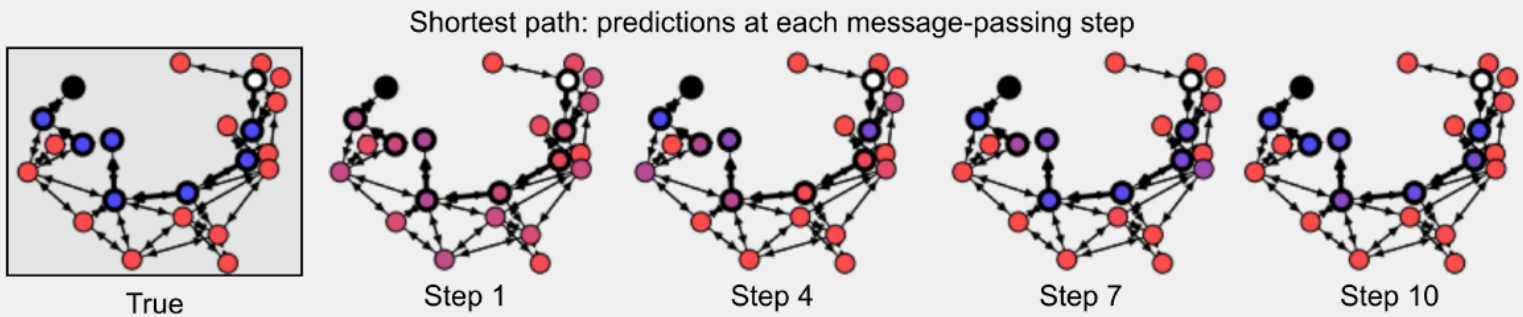
MLers think ML can do Optimization



MLers think ML can do Optimization

Shortest path demo: tinyurl.com/gn-shortest-path-demo

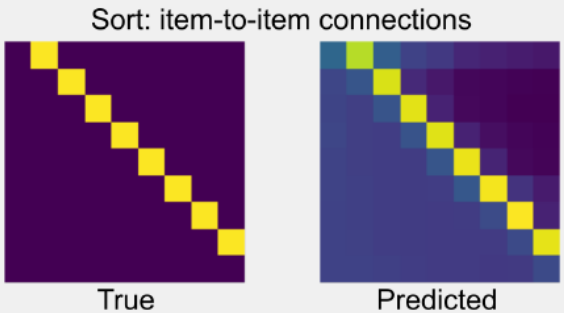
This demo creates random graphs, and trains a GN to label the nodes and edges on the shortest path between any two nodes. Over a sequence of message-passing steps (as depicted by each step's plot), the model refines its prediction of the shortest path.



MLers think ML can do Optimization

Sort demo: tinyurl.com/gn-sort-demo

This demo creates lists of random numbers, and trains a GN to sort the list. After a sequence of message-passing steps, the model makes an accurate prediction of which elements (columns in the figure) come next after each other (rows).



Maybe they are right?

NEURAL COMBINATORIAL OPTIMIZATION WITH REINFORCEMENT LEARNING

Irwan Bello*, **Hieu Pham***, **Quoc V. Le, Mohammad Norouzi, Samy Bengio**
Google Brain
{ibello,hyhieu,qvl,mnorouzi,bengio}@google.com

ABSTRACT

This paper presents a framework to tackle combinatorial optimization problems using neural networks and reinforcement learning. We focus on the traveling salesman problem (TSP) and train a recurrent neural network that, given a set of city coordinates, predicts a distribution over different city permutations. Using negative tour length as the reward signal, we optimize the parameters of the recurrent neural network using a policy gradient method. We compare learning the network parameters on a set of training graphs against learning them on individual test graphs. Despite the computational expense, without much engineering and heuristic designing, Neural Combinatorial Optimization achieves close to optimal results on 2D Euclidean graphs with up to 100 nodes. Applied to the KnapSack, another NP-hard problem, the same method obtains optimal solutions for instances with up to 200 items.

Maybe they are right?

REVISED NOTE ON LEARNING QUADRATIC ASSIGNMENT WITH GRAPH NEURAL NETWORKS

Alex Nowak[†,§], *Soledad Villar*[‡,∗,§], *Afonso S Bandeira*[‡,∗] *and Joan Bruna*[‡,∗]

† Sierra Team, INRIA and Ecole Normale Superieure, Paris
‡ Courant Institute, New York University, New York
∗ Center for Data Science, New York University, New York

30 Aug 2018

Operations Research | RWTH AACHEN UNIVERSITY

Maybe they are right?

Learning Combinatorial Optimization Algorithms over Graphs

Hanjun Dai*, **Elias B. Khalil***, **Yuyu Zhang**, **Bistra Dilkina**, **Le Song**
College of Computing, Georgia Institute of Technology
{hanjun.dai, elias.khalil, yuyu.zhang, bdilkina, lsong}@cc.gatech.edu

Abstract

The design of good heuristics or approximation algorithms for NP-hard combinatorial optimization problems often requires significant specialized knowledge and trial-and-error. Can we automate this challenging, tedious process, and learn the algorithms instead? In many real-world applications, it is typically the case that the same optimization problem is solved again and again on a regular basis, maintaining the same problem structure but differing in the data. This provides an opportunity for learning heuristic algorithms that exploit the structure of such recurring problems. In this paper, we propose a unique combination of reinforcement learning and graph embedding to address this challenge. The learned greedy policy behaves like a meta-algorithm that incrementally constructs a solution, and the action is determined by the output of a graph embedding network capturing the current state of the solution. We show our framework can be applied to a diverse range of optimization problems over graphs, and learns effective algorithms for the Minimum Vertex Cover, Maximum Cut and Traveling Salesman problems.

We need to (and do) talk to each other

Machine Learning for Combinatorial Optimization: a Methodological Tour d'Horizon*

Yoshua Bengio[2,3], Andrea Lodi[1,3], and Antoine Prouvost[1,3]

yoshua.bengio@mila.quebec
{andrea.lodi, antoine.prouvost}@polymtl.ca

[1]Canada Excellence Research Chair in Data Science for Decision Making, École Polytechnique de Montréal
[2]Department of Computer Science and Operations Research, Université de Montréal
[3]Mila, Quebec Artificial Intelligence Institute

- classification ML4Opt: end-to-end, offline, online

We need to (and do) talk to each other

This example does not do justice to the rich TSP literature that has developed far more advanced algorithms performing orders of magnitude better than ML ones. Nevertheless, the point we are trying to highlight here is that given a fixed context, and a decision to be made, ML can be used to discover new, potentially better performing policies. Even on state-of-the-art TSP algorithms (*i.e.* when exact solving is taken to its limits), many decisions are made in heuristic ways, *e.g.* cutting plane selection, thus leaving room for ML to assist in making these decisions.

What's next?

Robust Optimization

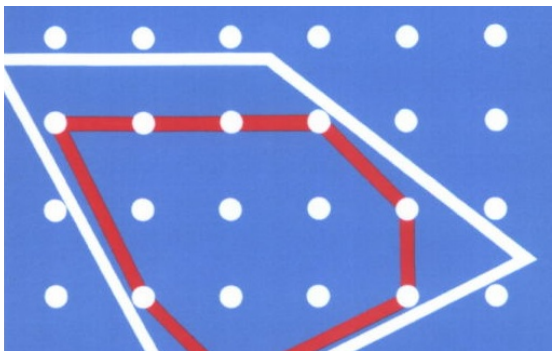
- data-driven design of uncertainty sets Bertsimas et al. (2018)
- a *practical* notion of robustness?

Learning (about) Optimization Models?

- a business school student asked me why we need models

- ML can make sense of data
- optimization models are also "data"

$\rightarrow$ (how) can ML help us make sense of optimization models?



- can ML learn (+ help automate) good modeling?

Learning (about) Optimization Problems?

potential first step:

- can ML learn the *semantics* of a MIP model ("the problem")?

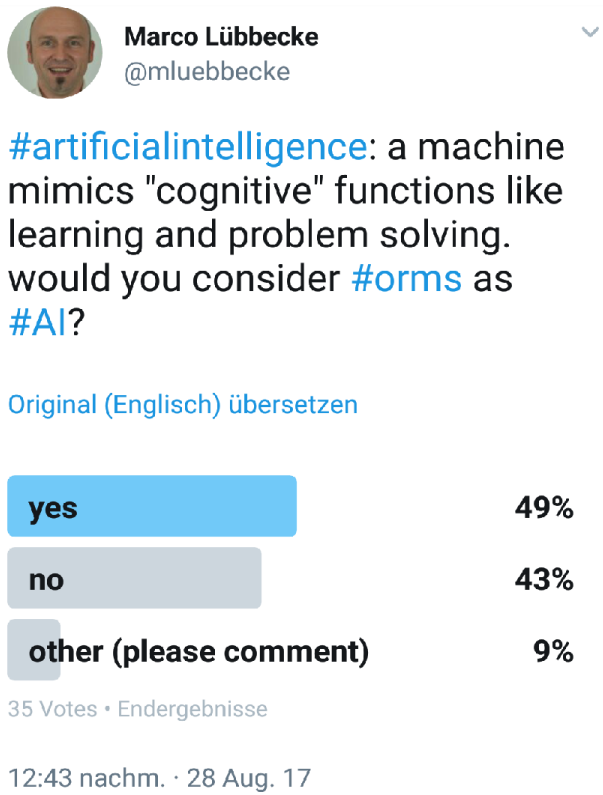
$$\min \sum_{j=1}^{n} y_j$$
$$\text{s.t.} \sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, \ldots, n$$
$$\sum_{i=1}^{n} a_i x_{ij} \leq b \qquad j = 1, \ldots, n$$
$$x_{ij} \leq y_j \qquad i, j = 1, \ldots, n$$
$$x_{ij},\, y_j \in \{0, 1\} \qquad i, j = 1, \ldots, n$$

$$\min \sum_{p \in P} \nu_p$$
$$\text{s.t.} \sum_{p \in P: i \in p} \nu_p = 1 \qquad i = 1, \ldots, n$$
$$\nu_p \in \{0, 1\} \qquad p \in P$$

$\Rightarrow$ e.g., help the modeler find a better formulation

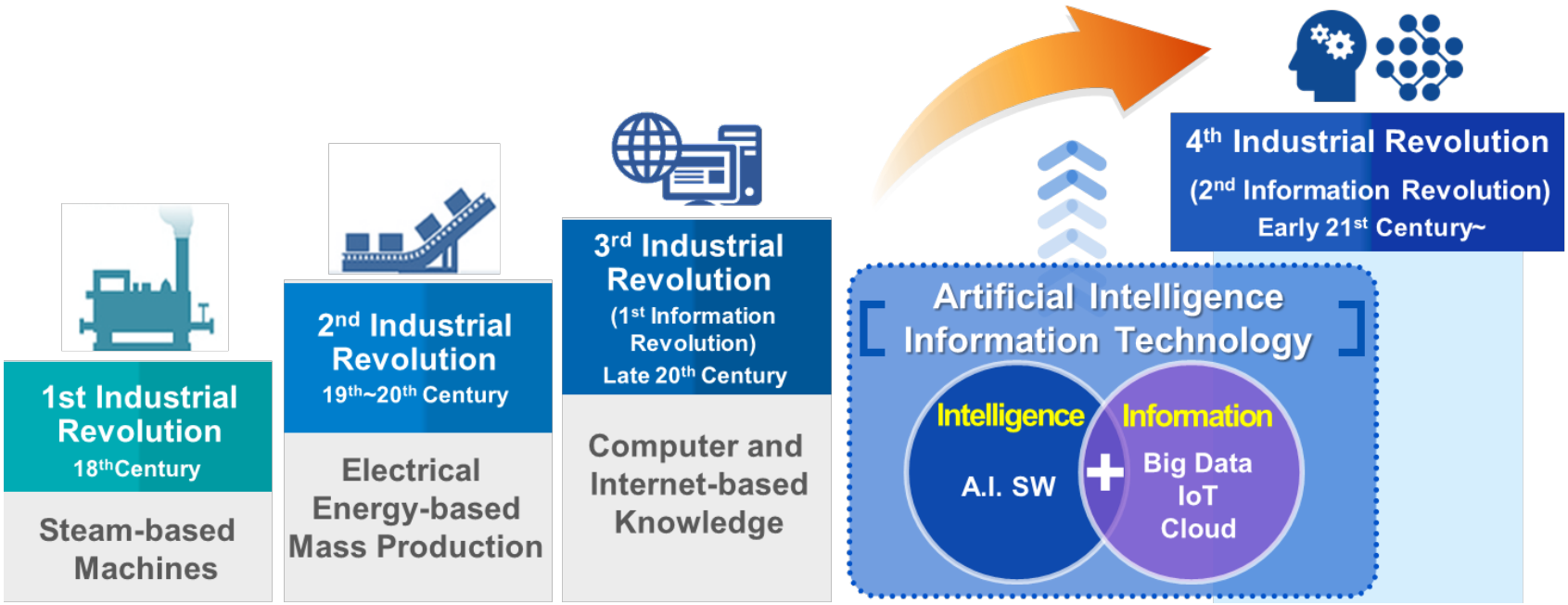
The AI Umbrella?

- OR vs. analytics discussion ✓
- OR vs. AI discussion ?



Operations Research | RWTH AACHEN UNIVERSITY

Why is this relevant?



source: blogs.worldbank.org/category/tags/artificial-intelligence

- *if* the fourth industrial revolution is about AI, OR should be part of it

Why is this relevant?

- besides (perceived) relevance
- and besides the enormous public reception:

- job openings in industry
- position titles in academia
- calls by funding bodies
- journal mission statements
- recognition by peers

$\rightarrow$ all AI/ML "framed"...

(How) does Machine Learning impact OR?

Marco Lübbecke
Lehrstuhl für Operations Research
RWTH Aachen University

 @mluebbecke

NGB/LNMB Seminar · Lunteren, NL · January 16, 2019