CSCI 699: Topics in Discrete Optimization and Learning

Units: 4
Spring 2020 – Thursday – 2:00-5:20

Thursday 2:00- 5:20pm
Location: TBD

Instructors: Bistra Dilkina
Office: SAL 304
Office Hours: Wednesday 11am-12 or By Appointment

Contact Info: dilkina@usc.edu
D Clearance Form:
https://forms.gle/wihPhqn8TbVaDCv4A
Teaching Assistant: TBD

Course Description

This course will examine recent research that leverages **synergies between machine learning and discrete optimization**, including 1) leveraging machine learning in computational methods for solving NP-hard discrete optimization problems, 2) learning in decision-focused way, and 3) leveraging combinatorial algorithmic ideas for machine learning tasks. The course will introduce students to computational discrete optimization and in particular integer programming, branch and bound, SAT, local search, submodular optimization and empirical evaluation of algorithms, as well as applications of machine learning methods, such as deep learning, graph convolutional networks and reinforcement learning. The course will study in depth recent papers focusing on a data-driven algorithm design for combinatorial problems, where ML techniques such graph convolutional network, reinforcement learning, and deep learning are used to improve existing methods or design new ones altogether. The second part will focus on the combinatorial approaches in the context of machine learning, such as binarized NN, optimal decision trees, interpretability, and adversarial examples. This class is targeted at PhD students. Background in ML is expected. Mathematical maturity, as well as research experience in computer science and/or data science is strongly recommended.

Learning Objectives

1. Learn solution techniques for discrete optimization
2. Learn how to empirically evaluate computational performance of solvers.
3. Learn how to use different ML techniques in the context of discrete optimization
4. Learn how to critically read a technical research paper on this topic.

Prerequisite(s): sufficient mathematical background; some background in AI, machine learning and discrete optimization, with research experience in at least one, is strongly recommended; good programming skills
Recommended Preparation: CS Algorithms class (preferably graduate level) is strongly advisable, ML and optimization course work or background is advisable, not mandatory

Course Notes

Lecture slides/notes will be available online after class.

Required Readings and Supplementary Materials

The course will not have any official textbook but will instead use assigned papers and book chapters reading. This is a preliminary list of the reading list:

Part 1: ML for Discrete Optimization

- Machine Learning for Combinatorial Optimization: a Methodological Tour d'Horizon. Yoshua Bengio, Andrea Lodi, A Prouvost

Graph Optimization problems (and RL)

- Learning combinatorial optimization algorithms over graphs, H. Dai, E. B. Khalil, Y. Zhang, B. Dilkina, L. Song. **NeurIPS** 2017. (code)
- Neural combinatorial optimization with reinforcement learning,I. Bello, H. Pham, Q. V. Le, M. Norouzi, S. Bengio. Arxiv 2017. Lecture by Samy Bengio (Berlin, June 2017.) Background: Pointer networks, O. Vinyals, M. Fortunato, N. Jaitly. 2017.
- Combinatorial optimization with graph convolutional networks and guided tree search. Li, Zhuwen, Qifeng Chen, and Vladlen Koltun. ***NeurIPS*** 2018.
- Reinforcement Learning for Solving the Vehicle Routing Problem, M. Nazari, A. Oroojlooy, L. V. Snyder, M. Takac. **NeurIPS** 2018. Background: Neural machine translation by jointly learning to align and translate, D. Bahdanau, K. Cho, Y. Bengio. 2014. Sequence to sequence learning with neural networks, I. Sutskever, O. Vinyals, Q. V. Le. 2014.
- Combinatorial optimization with graph convolutional networks and guided tree search, Li, Z., Chen, Q., & Koltun, V. **NeurIPS** 2018.
- Learning heuristics for the TSP by policy gradient, Deudon, M., Cournut, P., Lacoste, A., Adulyasak, Y., & Rousseau, L. M. **CPAIOR** 2019
- Neural Large Neighborhood Search for the Capacitated Vehicle Routing Problem. **Arxiv** 2019
- Learning Heuristics over Large Graphs via Deep Reinforcement Learning. Mittal, A., Dhawan, A., Medya, S., Ranu, S., & Singh, A. **ARXIV** (2019).

Attention

- Attention solves your TSP, W. Kool, M. Welling. March 2018. (Source code on GitHub.) Background: Attention is all you need, A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, I. Polosukhin. December 2017.
- Attention, learn to solve routing problems!, W. Kool, M. Welling. ICLR 2019.

Integer Programming

- On learning and branching: a survey, A. Lodi, G. Zarpellon.

MIP Branching:

- Learning to branch in mixed integer programming, E. B. Khalil, P. Le Bodic, L. Song, G. Nemhauser, B. Dilkina, **AAAI** 2016
- Learning to Branch, M-F Balcan, T Dick, Tuomas Sandholm, E Vitercik, **ICML** 2018
- DASH: Dynamic Approach for Switching Heuristics. Liberto, G. Di, S. Kadioglu, K. Leo, Y. Malitsky. European Journal of Operational Research 248(3) 943–953, 2016.
- Exact Combinatorial Optimization with Graph Convolutional Neural Networks. Maxime Gasse, Didier Chételat, Nicola Ferroni, Laurent Charlin, Andrea Lodi. ArXiV 2019. (code)
- Neural Network Branching for Neural Network Verification. J Lu, M. Pawan Kumar. ArXiV 2019

MIP Node Selection:

- Guiding Combinatorial Optimization with UCT, Sabharwal, Ashish, Horst Samulowitz, and Chandra Reddy. **CPAIOR**, 2012.
- Learning to Search in Branch-and-Bound Algorithms, He He, Hal Daume III, Jason Eisner. Advances in neural information processing systems, 3293-3301, 2014.
- Learning to Search via Retrospective Imitation, J Song, R Lanka, A Zhao, Yisong Yue, M Ono. **ArXiv**, 2018

Other MIP topics:

- Learning to run heuristics in tree search. Khalil, E. B., B. Dilkina, G. L. Nemhauser, S. Ahmed, Y. Shao. **IJCAI** 2017.
- Learning when to use a decomposition. Kruber, M., M. E. Lubbecke, A. Parmentier. CPAIOR 2017.
- Nair, Vinod, Krishnamurthy Dvijotham, Iain Dunning, and Oriol Vinyals. "Learning Fast Optimizers for Contextual Stochastic Integer Programs." **UAI** 2018.
- Learning to Solve Large-Scale Security-Constrained Unit Commitment Problems. Alinson S. Xavier, Feng Qiu, Shabbir Ahmed. **ArXiv** 2019
- Accelerating Primal Solution Findings for Mixed Integer Programs Based on Solution Prediction. Jian-Ya Ding, Chao Zhang, Lei Shen, Shengyin Li, Bing Wang, Yinghui Xu, Le Song. **ArXiv**, 2019
- Reinforcement Learning for Integer Programming: Learning to Cut. Yunhao Tang, Shipra Agrawal, Yuri Faenza. **ArXiv** 2019
- Online Mixed-Integer Optimization in Milliseconds. Dimitris Bertsimas, B Stellato. **ArXiv** 2019.
- Learning to Project in Multi-Objective Binary Linear Programming. Alvaro Sierra-Altamiranda, Hadi Charkhgard, Iman Dayarian, Ali Eshragh, Sorna Javadi. **ArXiv** 2019.

SAT and CSP

- Deep Learning for Algorithm Portfolios. Andrea Loreggia, Yuri Malitsky, Horst Samulowitz, Vijay Saraswat. **AAAI** 2016
- Learning Robust Search Strategies Using a Bandit-Based Approach. Wei Xia, Roland H. C. Yap, **AAAI** 2018
- Effective Deep Learning for Constraint Satisfaction Problems. H. Xu, S. Koenig and S. Kumar International Conference on Principles and Practice of Constraint Programming (**CP**), 2018.
- Learning a SAT Solver from Single-Bit Supervision. Daniel Selsam, Matthew Lamm, Benedikt Bunz, Percy Liang, Leonardo de Moura, David L. Dill, **ICLR** 2019

RL for Discrete Optimization tasks

- Algorithm Selection using Reinforcement Learning. Lagoudakis, Michail G., and Michael L. Littman. **ICML** 2000.
- Improving optimization bounds using machine learning: decision diagrams meet deep reinforcement learning, Cappart, Q., Goutierre, E., Bergman, D., & Rousseau, L. M. **AAAI** 2019
- Learning to Perform Local Rewriting for Combinatorial Optimization.Xinyun Chen and Yuandong Tian. **NeurIPS** 2019
- Co-training for Policy Learning.Song, Jialin, Ravi Lanka, Yisong Yue, and Masahiro Ono. **UAI** 2019.
- Causal Discovery with Reinforcement Learning, Zhu S., Ng I., Chen Z., **ICLR** 2020

PART 2: Decision-focused Learning

- Optnet: Differentiable optimization as a layer in neural networks, Amos B, Kolter JZ. **JMLR** 2017
- Task-based end-to-end model learning in stochastic optimization, Donti, P., Amos, B. and Kolter, J.Z. **NeurIPS** 2017
- SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver, Wang, P.W., Donti, P.L., Wilder, B. and Kolter, Z., **ICML** 2019

- Smart "Predict, then Optimize", Elmachtoub, A. N., & Grigas, P. Optimization Online 2018
- Differentiable Submodular Maximization, Tschiatschek, S., Sahin, A., & Krause, A. **IJCAI** 2018
- Melding the Data-Decisions Pipeline: Decision-Focused Learning for Combinatorial Optimization, Wilder, B., Dilkina, B., & Tambe, M. **AAAI** 2019
- End to end learning and optimization on graphs, Wilder B, Ewing E, Dilkina B, Tambe M. **NeurIPS** 2019 (code)
- Differentiation of Blackbox Combinatorial Solvers, Vlastelica, M., Paulus, A., Musil, V., Martius, G., & Rolínek, M. **ICLR** 2020
- MIPaaL: Mixed integer program as a layer. Ferber, A., Wilder, B., Dilina, B., & Tambe, M. **AAAI** 2020 (code)
- Differentiation of Blackbox Combinatorial Solvers. Marin Vlastelica, Anselm Paulus, Vít Musil, Georg Martius, Michal Rolínek. ArXiV 2019.

PART 3: Discrete approaches in ML

- Deep neural networks and mixed integer linear optimization. Fischetti, M. & Jo, J. Constraints (2018). https://doi.org/10.1007/s10601-018-9285-6
- Training Binarized Neural Networks Using MIP and CP. RT Icarte, L Illanes, MP Castro, AA Cire, SA McIlraith, JC Beck. **CP** 2019
- Mapping images to scene graphs with permutation-invariant structured prediction, Herzig, R., Raboh, M., Chechik, G., Berant, J., & Globerson, A. **NeurIPS** 2018 (**Structured Prediction**)
- Optimized Pre-Processing for Discrimination Prevention, Calmon, F., Wei, D., Vinzamuri, B., Ramamurthy, K. N., & Varshney, K. R., **NeurIPS** 2017 (**Fairness)**

Exact Decision Trees

- Optimal classification trees, Bertsimas & Dunn. Journal of Machine Learning, 2017
- Learning Optimal Decision Trees with SAT. Nina Narodytska, Alexey Ignatiev, Filipe Pereira, João Marques-Silva. **IJCAI** 2018
- Optimal Sparse Decision Trees, Hu, X., Rudin, C., & Seltzer, M.. **NeurIPS** 2019
- Learning Optimal and Fair Decision Trees for Non-Discriminative Decision-Making, Aghaei S., Azizi M. J., Vayanos P. **AAAI** 2019

Interpretability

- "Why Should I Trust You?" Explaining the Predictions of Any Classifier, Ribeiro, M. T., Singh, S., & Guestrin, C. **KDD** 2016
- Boolean Decision Rules via Column Generation, S. Dash, O. Gunluk, D. Wei. Winner of the $5000 FICO Explainable Machine Learning Challenge at **NeurIPS** 2018.
- Learning Optimized Risk Scores, Ustun B., Rudin C. **JMLR** 2019
- Compiling Neural Networks into Tractable Boolean Circuits. Arthur Choi and Weijia Shi and Andy Shih and Adnan Darwiche. **AAAI** 2019
- Compiling Bayesian Networks into Decision Graphs. Andy Shih and Arthur Choi and Adnan Darwiche. **AAAI** 2019
- On Validating, Repairing and Refining Heuristic ML Explanations. Alexey Ignatiev, Nina Narodytska, Joao Marques-Silva. ArXiv 2019
- Assessing Heuristic Machine Learning Explanations with Model Counting. Nina Narodytska, Aditya Shrotri, Kuldeep S. Meel, Alexey Ignatiev, Joao Marques Silva. **SAT** 2019

Adversarial examples for NNs via MIP

- Combinatorial Attacks on Binarized Neural Networks. Elias B. Khalil, Amrita Gupta, Bistra Dilkina. **ICLR** 2019
- Evaluating robustness of neural networks with mixed integer programming, V. Tjeng, K. Xiao, R. Tedrake. **ICLR** 2019. (code available).

- Deep neural networks and mixed integer linear optimization, M. Fischetti, J. Jo. Journal Constraints, 2018.
- Maximum resilience of artificial neural networks, C.-H. Cheng, G. Nuhrenberg, H. Ruess, ATVA 2017.
- Output range analysis for deep feedforward neural networks, S. Dutta, S. Jha, S. Sanakaranarayanan, A. Tiwari. NFM 2018.
- Reachability Analysis for Neural Agent-Environment Systems. Michael Akintunde, Alessio Lomuscio, Lalit Maganti, Edoardo Pirovano. KR 2018

Background

- Applied Mathematical Programming, Chapter 9, Integer Programming, S. Bradley, A. Hax, T. Magnanti. Branch and bound is covered in Section 9.5.
- In Pursuit of the Traveling Salesman, Chapter 7. Gentle introduction to branch-and-bound for the TSP and integer programming.
- Integer Programming: The Branch and Bound Method. Detailed example of branch-and-bound applied to a small integer programming problem.
- Constraint Integer Programming, Tobias Achterberg, Ph. D. Thesis, Berlin 2009. Beautiful examples of careful computational testing.
- Experimental analysis of algorithms, Catherine McGeoch. Notices of the AMS, March 2001.
- CH. 5, STOCHASTIC LOCAL SEARCH: FOUNDATIONS AND APPLICATIONS by Holger H. Hoos and Thomas Stützle
- A Practical Guide to Discrete Optimization, Chapter 1, Chapter 7. David Applegate, William Cook, Sanjeeb Dash. Computational studies in discrete optimization.
- Deep Learning. Ian Goodfellow, Yoshua Bengio, Aaron Courville. MIT Press, 2016. Course text for fundamentals of deep learning.

Other Class Pages

William Cook's class "Deep Learning in Discrete Optimization":
http://www.ams.jhu.edu/~wcook12/dl/index.html Spring 2019

Description and Assessment of Assignments

1. **Reviews**: We will explore the course topics through a series of assigned readings in the form of papers (and book chapters). Students will be expected to read the papers before class and submit a one page review for the assigned reading papers as homework. Students will get credit for every review submitted.

2. **Paper Presentations**: Students will present a recent research papers relevant to the course topic and lead discussions with the class.

3. **Project**: A key component of the course will be to get a hands on experience in using ML in the context of discrete optimization. Projects will be alone or in small teams (1-3). Projects will be graded based on their novelty and technical results. Students will be expected to prepare project proposals early in the course, and give presentations of their projects/papers at the end of the course. The final project paper should have the structure of a conference paper with problem statement, lit review, approach, empirical results and discussion. A statement of author contributions (i.e. who did what) must be turned in with the final draft. Rough drafts and partial

drafts will be due at different points throughout the semester so that the instructor may provide students with constructive input along the way.

Grading Breakdown

	Weight
Class participation	**5%**
Paper Reviews (10 @ 1% each)	**10%**
Paper Presentations	**30%**
Final Project	**55%**
Project Proposal (5%)	
Preliminary Paper (10%)	
Final Presentation (10%)	**Last week**
Project/Final Paper (30%)	**Final exam time**

Assignment Submission Policy

Paper reviews and other deliverables should be submitted the day before class at midnight. Assignemnts will be administered through balcboard.

Statement on Academic Conduct and Support Systems

Academic Conduct

Plagiarism – presenting someone else's ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in *SCampus* in Section 11, *Behavior Violating University Standards* https://scampus.usc.edu/1100-behavior-violating-university-standards-and-appropriate-sanctions. Other forms of academic dishonesty are equally unacceptable. See additional information in *SCampus* and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct.

Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the *Office of Equity and Diversity* http://equity.usc.edu or to the *Department of Public Safety* http://capsnet.usc.edu/department/department-public-safety/online-forms/contact-us. This is important for the safety of the whole USC community. Another member of the university community – such as a friend, classmate, advisor, or faculty member – can help initiate the report, or can initiate the report on behalf of another person. *The Center for Women and Men* http://www.usc.edu/student-affairs/cwm/ provides 24/7 confidential support, and the sexual assault resource center webpage http://sarc.usc.edu describes reporting options and other resources.

Note on Collaborative Work

For collaborative projects, students are expected to have equal distribution. If there is any perceived imbalance in the collaborative project, the student should bring this to the attention of the instructor or the teaching assistant.

Support Systems

A number of USC's schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the *American Language Institute* http://dornsife.usc.edu/ali, which sponsors courses and workshops specifically for international graduate students. *The Office of Disability Services and Programs* http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html provides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, *USC Emergency Information* http://emergency.usc.edu will provide safety and other updates, including ways in which instruction will be continued by means of blackboard, teleconferencing, and other technology.