Decision-Focused Learning: Through The Lens Of Learning To Rank

Accepted for Presentation at ICML, 2022

Jayanta Mandi, Vrije Universiteit Brussel, Belgium

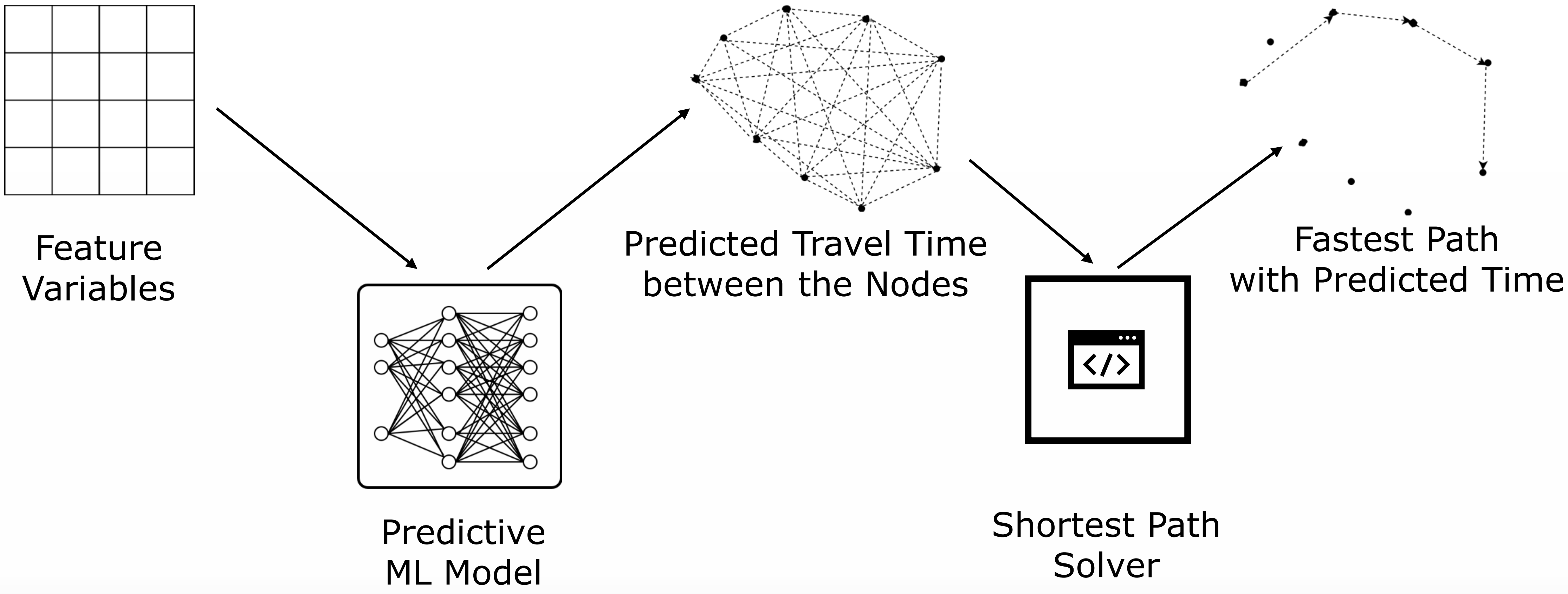
Victor Bucarey Lopez, Universidad de O'Higgins, Chile

Maxime Mulamba, Vrije Universiteit Brussel, Belgium

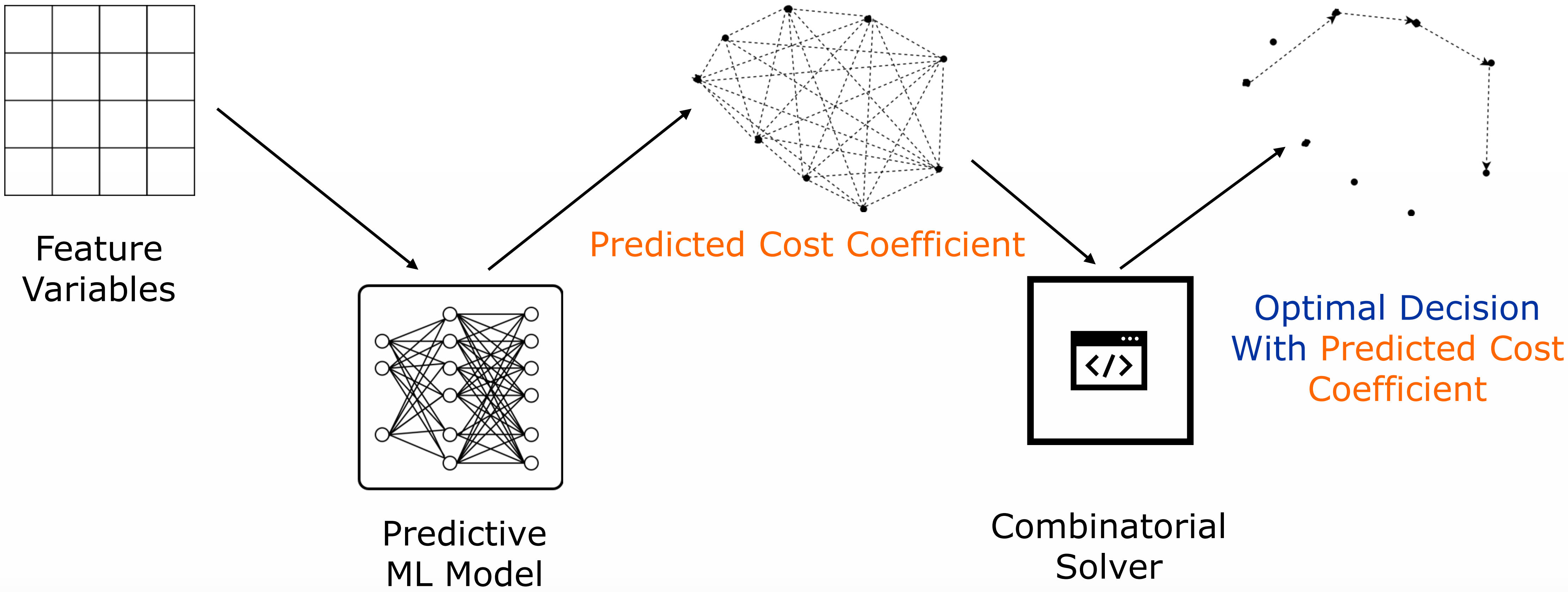
Tias Guns, KU Leuven, Belgium

Problem Description

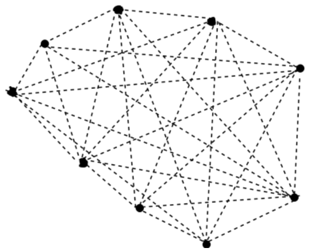
Motivation



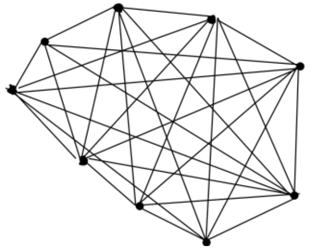
Problem Description



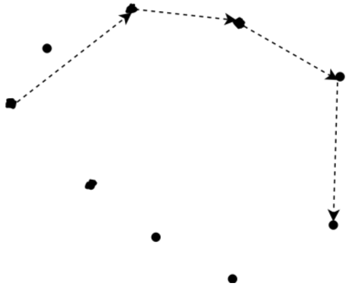
Two-stage Learning vs Decision-Focused Learning (WILDER, B., DILKINA, B., & TAMBE, M.).



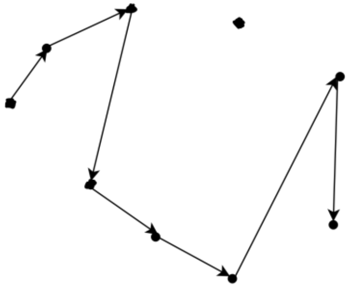
Predicted
Cost Coefficient



Actual
Cost Coefficient



Optimal Decision
With Predicted Cost
Coefficient



True
Optimal Decision

Two-stage Learning

- Train model to minimize the difference between predicted and actual cost coefficient.
- Training does not consider the downstream optimization problem.

Decision-Focused Learning

- Train model to minimize the difference between predicted and actual optimal decision.
- Training to improve the output of the optimization problem.

Decision-Focused Learning

- We study generic combinatorial optimization problem of the following form

$$v^{\star}(c) = \underset{v \in V}{arg\,min}\, f\,(v, c)$$

v: decision variable; V: domain of v; c: cost coefficient

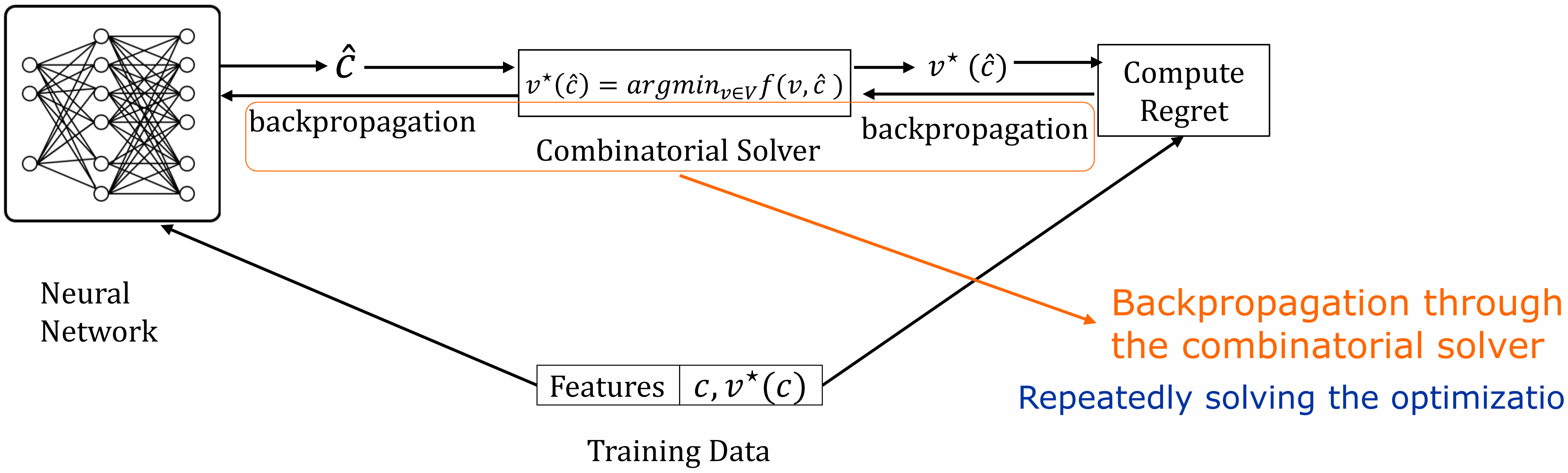
- Our goal is to minimize the regret of the predictions $\hat{c}$. Formally, regret is defined as

$$\text{regret}(\hat{c}, c) = f(v^{\star}(\hat{c}), c) - f(v^{\star}(c), c)$$

Realized objective value with the predicted coefficient

Optimal objective value

Decision-focused Learning Training Loop



Repeatedly solving the optimization problem

Our aim is to facilitate training with fewer call to the solver

Rank Based Loss Functions

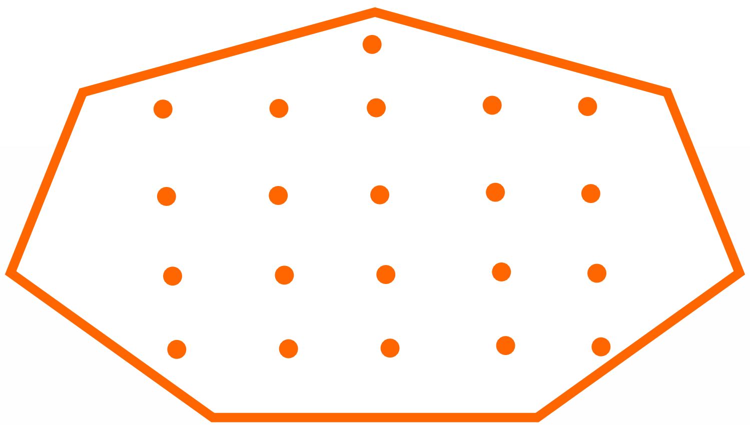
Main Idea

- We want to predict $\hat{c}$, so that $f(v, c)$ and $f(v, \hat{c})$ have same ordering for all $v \in V$

v_i	$f(v_i, c)$	$f(v_i, \hat{c})$
[a,b,c]	1	10
[a,d,c]	2	25
[a,b,e,c]	3	50
[a,d,e,c]	4	80
[a,e,c]	5	100

Implementation By Caching Solution

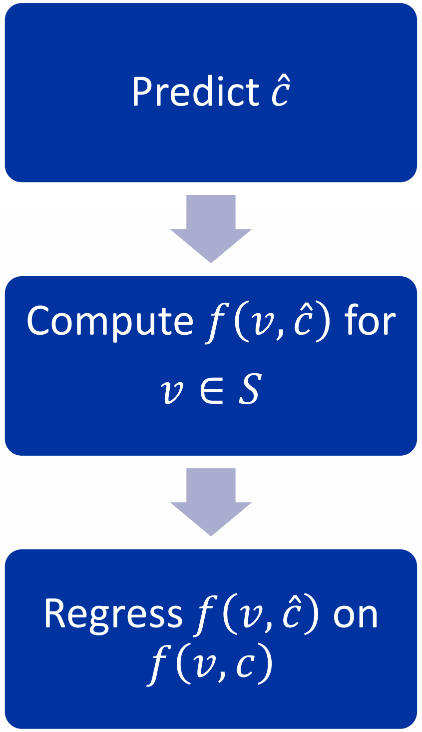
As it is not practically impossible to consider the partial ordering of the whole solution space, in practice, we formulate decision-focused learning as learning the partial ordering of $v \in S \subset V$



Learn to rank $v \in S$

- We implement this by the solution caching scheme proposed by Mulamba, M., et al. (2020).
- The cache is initialized by all optimal solutions present in the training data.
- Then the cache is expanded by adding solutions of predicted $\hat{c}$.
- They experimentally showed it is sufficient to add solutions of 10% of predicted $\hat{c}$.

Pointwise Loss



$f(v_i, c)$	$f(v_i, \hat{c})$	$(f(v_i, c) - f(v_i, \hat{c}))^2$
1	3	2^2
2	1	1^2
3	2	1^2
4	3	1^2
5	4	1^2
Total Loss		8

Pairwise Loss

In pairwise loss formulation, we compare all other v_i with the best v_{best}

If $f(v_i, \hat{c})$ has lower value than $f(v_{best}, \hat{c})$, then their difference is the loss

	v_i	$f(v_i, c)$	$f(v_i, \hat{c})$
v_{best}	[a,b,c]	1	10
	[a,d,c]	2	25
	[a,b,e,c]	3	50
	[a,d,e,c]	4	80
	[a,e,c]	5	100

Pairwise Loss

- We formulate pairwise loss by introducing a margin parameter γ
- The pairwise loss is finally defined as $\mathbf{relu}(\boldsymbol{\gamma} + \mathbf{f}(\mathbf{v}_{\mathbf{best}}, \hat{\mathbf{c}}) - \mathbf{f}(\mathbf{v}_{\mathbf{i}}, \hat{\mathbf{c}}) \ )$
- Extension of NCE loss [Mulamba, M., et al. (2020)]: $(\mathbf{f}(\mathbf{v}_{\mathbf{best}}, \hat{\mathbf{c}}) - \mathbf{f}(\mathbf{v}_{\mathbf{i}}, \hat{\mathbf{c}}) \ )$

Pairwise Difference Loss

$f(v_i, c)$	$f(v_i, \hat{c})$	$(f(v_{best}, c) - f(v_i, c))$	$(f(v_{best}, \hat{c}) - f(v_i, \hat{c}))$	$(col\ 4 - col\ 5)^2$
1	12	0	0	0^2
2	13	1	1	0^2
3	15	2	3	1^2
4	16	3	4	1^2
5	18	4	6	2^2
Total Loss				6

In this loss formulation, we regress predicted difference on actual difference.

Listwise Loss

- In pairwise loss formulation, we compare v_{best} with other v.
- In essence, in pairwise loss formulation, we only consider whether v_{best} is same for c and $\hat{c}$.
- In listwise loss formulation, we consider the partial ordering of all $v \in S \subset V$.

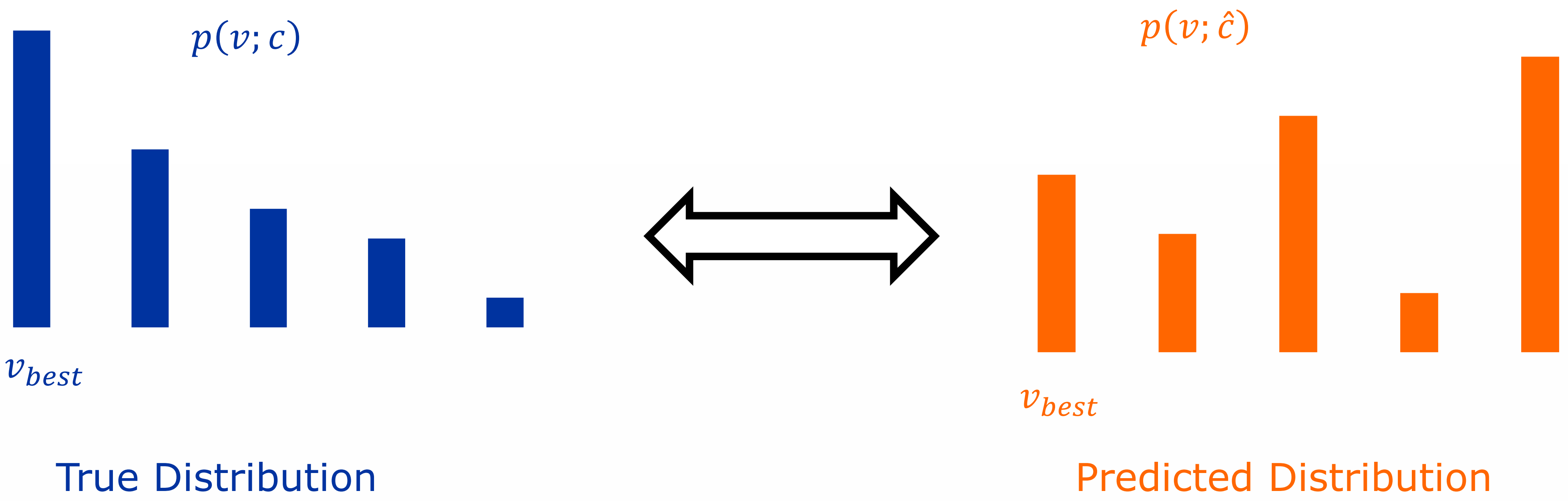
Listwise Loss Formulation

We start by defining the following discrete exponential distribution in the solution space

$$p(v;c) = \begin{cases} \frac{1}{Z} exp(-f(v,c)/\tau) & v \in V \\ 0 & v \notin V \end{cases}$$

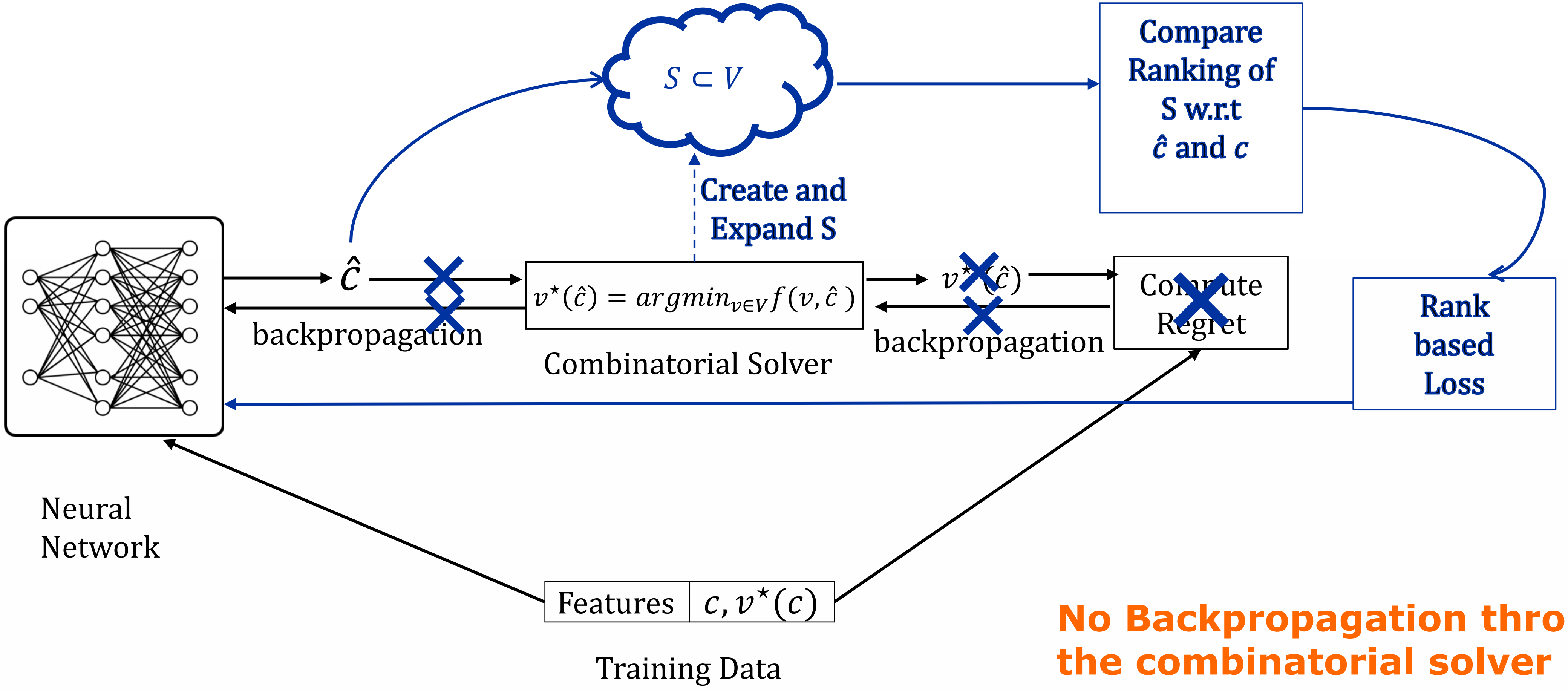
- τ controls the smoothness of the distribution
- $\tau \to 0$, $p(v;c)$ has positive pmf only at v_{best}
- As τ increases, $p(v;c)$ converges to a uniform distribution

Listwise Loss Formulation



Listwise loss is the cross-entropy loss between $p(v; c)$ and $p(v; \hat{c})$

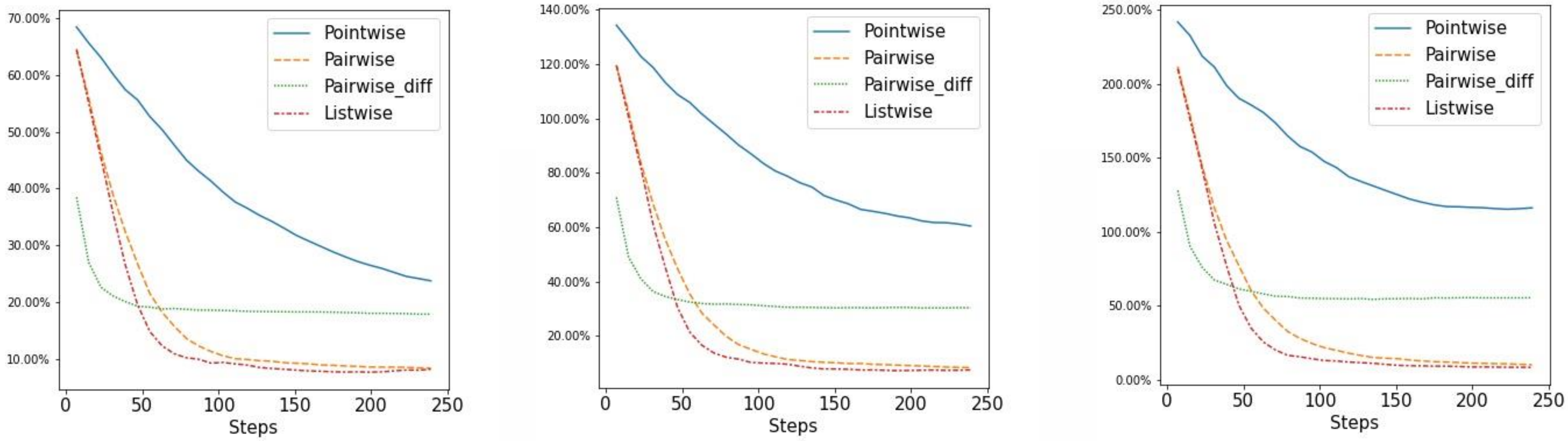
Rank Based Loss With Caching Solution



No Backpropagation through the combinatorial solver

Result

Shortest Path Problem

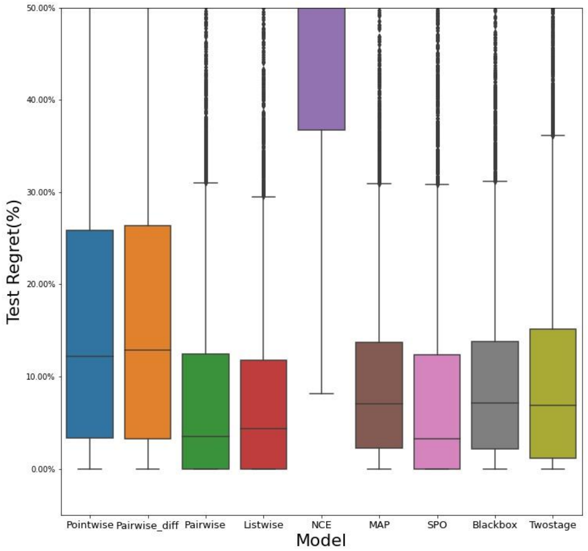


Degree 4 Degree 6 Degree 8

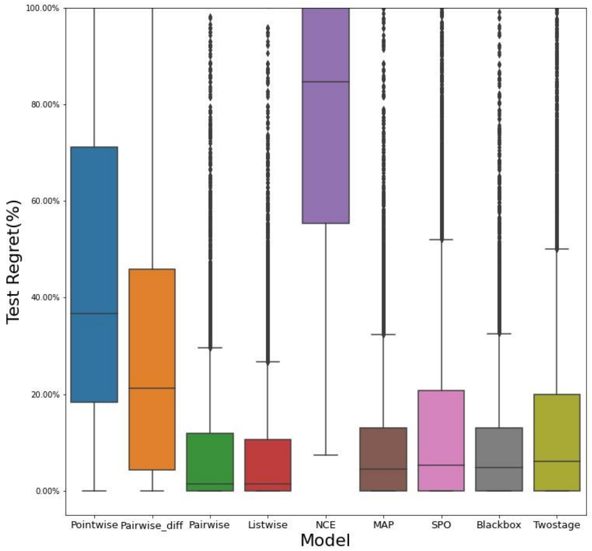
We demonstrate that by minimizing the rank based loss functions we can lower regret.

Result

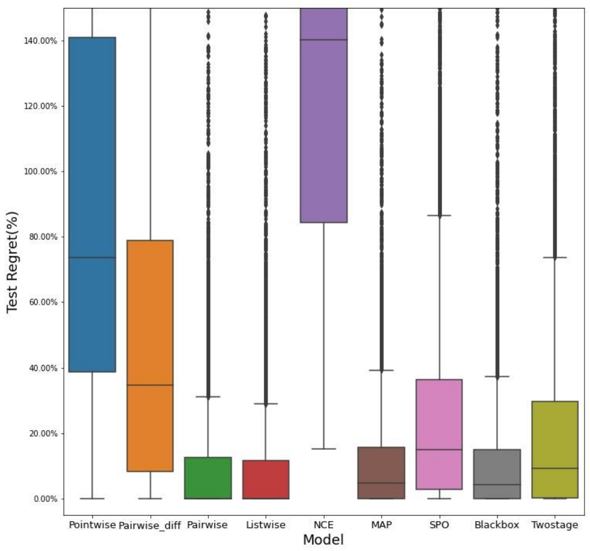
Shortest Path Problem



Degree 4



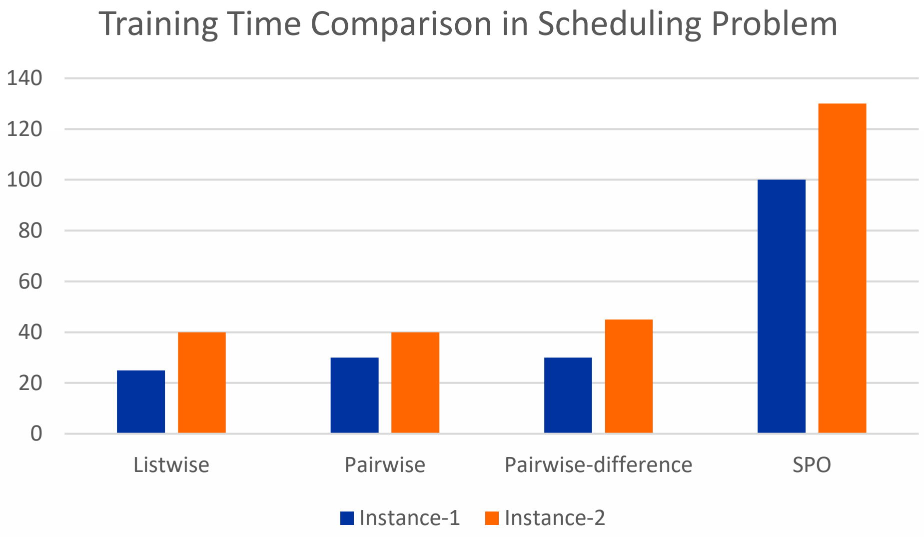
Degree 6



Degree 8

Listwise and Pairwise loss functions generate lowest regret in these problem instances.

Efficiency Gain In Training Time



Contribution

- We formulate decision-focused learning as learning the partial ordering of the solution space with respect to the objective value.
- We propose surrogate learning-to-rank loss functions for decision-focused learning.
- We show the approach of Mulamba, M., et al. (2020) can be viewed as a particular case of the proposed learning-to-rank loss functions.
- We evaluate the performance of the loss functions in three combinatorial optimization problems, where we show that it is possible to lower regret by minimizing the proposed loss functions.
- The performances of the proposed loss functions are comparable to the state of the arts.

References

Mulamba, M., Mandi, J., Diligenti, M., Lombardi, M., Bucarey, V., and Guns, T. Contrastive losses and solution caching for predict-and-optimize. In Zhou, Z.-H. (ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, pp. 2833–2840

Wilder, B., Dilkina, B., & Tambe, M. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In Proceedings of the AAAI Conference on Artificial Intelligence, AAAI 2019 (Vol. 33, No. 01, pp. 1658-1665).

Elmachtoub, A. N., & Grigas, P. (2022). Smart "predict, then optimize". Management Science, 68(1), 9-26.

Mandi, J., Demirovic, E., Stuckey, P. J., & Guns, T. (2020). Smart predict-and-optimize for hard combinatorial optimization problems. In Proceedings of the AAAI Conference on Artificial Intelligence, AAAI 2020 (Vol. 34, No. 02, pp. 1603-1610).

Pogančić, M. V., Paulus, A., Musil, V., Martius, G., & Rolinek, M. Differentiation of blackbox combinatorial solvers. In International Conference on Learning Representations, ICLR 2020

DANK U

Github Repository

VRIJE UNIVERSITEIT BRUSSEL

KU LEUVEN

Universidad de O'Higgins