LEARNING TO RANK FOR SYNTHESIZING PLANNING HEURISTICS

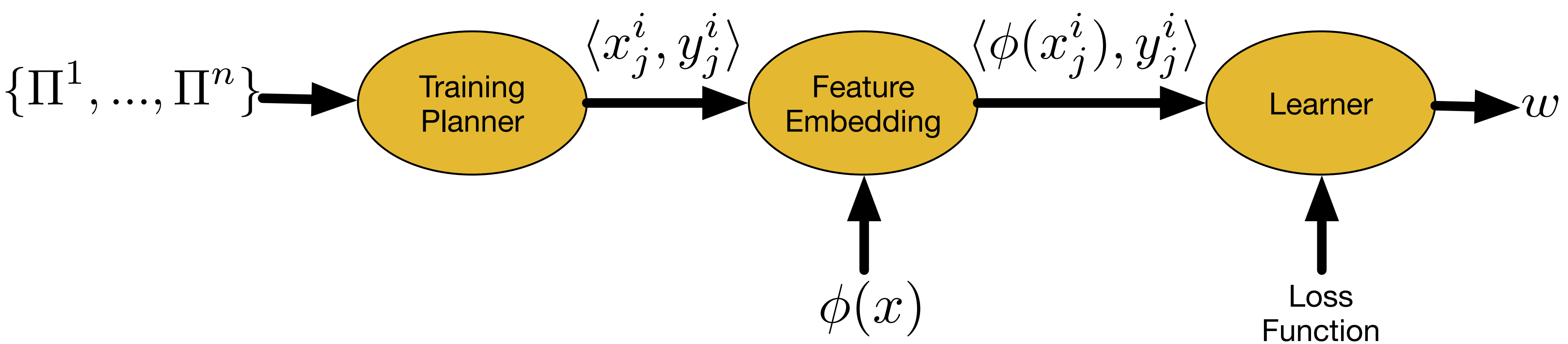
Caelan Reed Garrett, Leslie Pack Kaelbling, Tomás Lozano-Pérez
MIT CSAIL - {caelan, lpk, tlp}@csail.mit.edu

Learning a Heuristic Function

- Deterministic satisficing classical planning
  - Distribution of problems $\{\Pi^1, ..., \Pi^n\}$
- Focus on improving **coverage** and runtime
  - Greedy best-first search
- Learn domain-specific heuristic function
  - But learn it in a domain-independent way
- Use statistical supervised learning

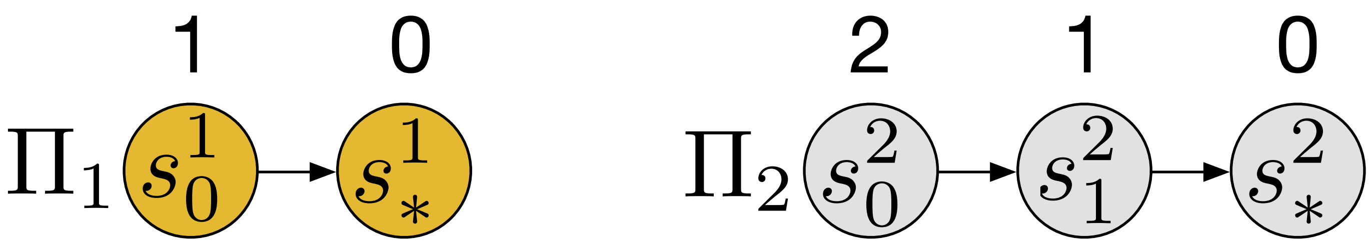
Supervised Heuristic Learning

- Assume linear function $h(x) = \phi(x)^T w$
- Gather training examples $\langle x_j^i, y_j^i \rangle$
- Specify feature representation $\phi(x)$
- Choice of loss function
- Optimize w using the loss function



Gathering Training Examples

- Generate training problems (~10 problems)
- Solve and use states on plan as examples $\langle x_j^i, y_j^i \rangle$
  - Inputs are state and problem $x_j^i = \langle s_j^i, \Pi^i \rangle$
  - Outputs are costs-to-go y_j^i



$$\{\langle x_0^1, 1\rangle, \langle x_*^1, 0\rangle, \langle x_0^2, 2\rangle, \langle x_1^2, 1\rangle, \langle x_*^2, 0\rangle\}$$

- Use large-timeout portfolio to produce plans

Feature Representation

- Obtain generic features from existing heuristics
  - Derive from approximate partially-ordered plans
  - FastForward (FF), Context-Enhanced Add (CEA), …
- New method for features
  - Counting **single** instance per action type
  - Counting **pairwise** instances per action types and interactions

Heuristic in a Greedy Search

- A heuristic is typically an estimate of the cost-to-go
- A heuristic in greedy search sorts the explored state space by its estimate of cost-to-go
- **Only the ordering is important, not the numerical heuristic value**

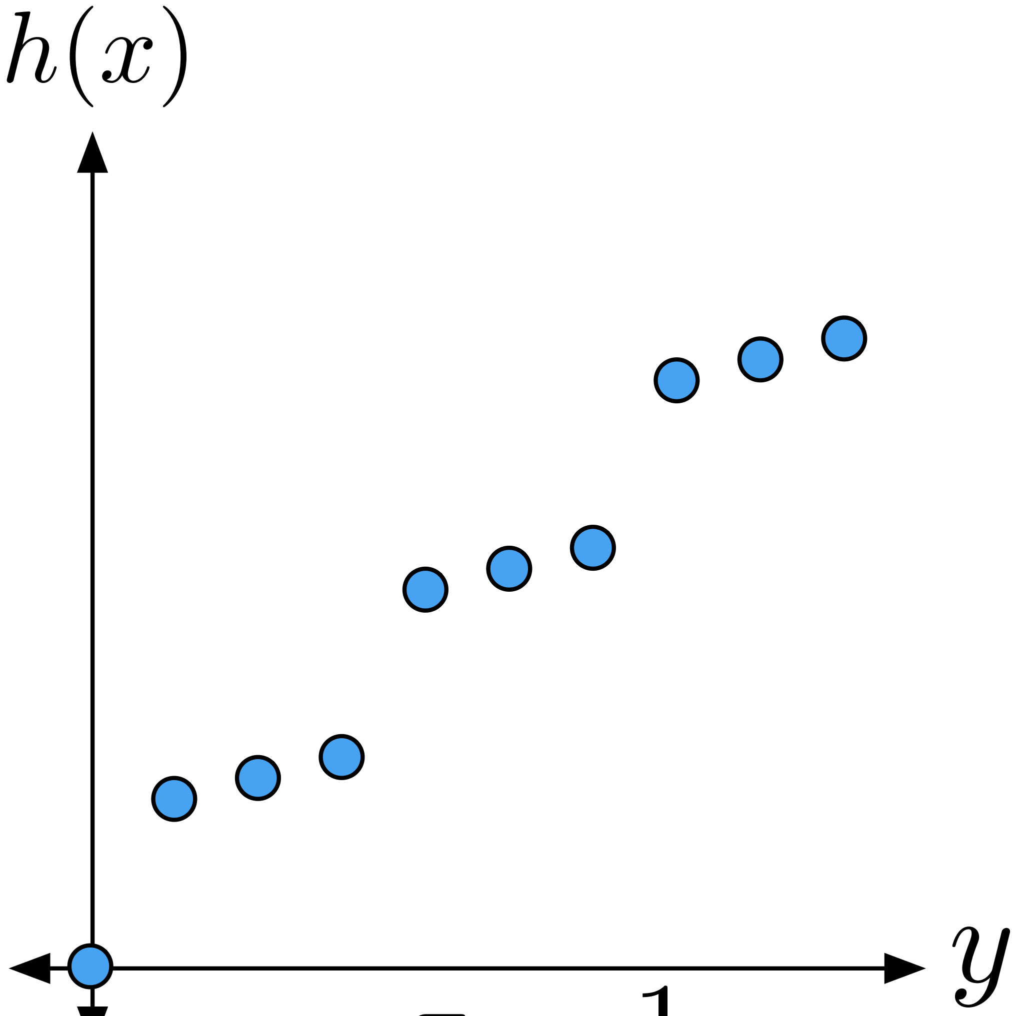
- Ideal heuristic is any monotonically increasing function of true cost-to-go

$$y_j^i > y_k^i \iff h(x_j^i) > h(x_k^i) \quad \forall i, j, k$$

Measuring Ordering

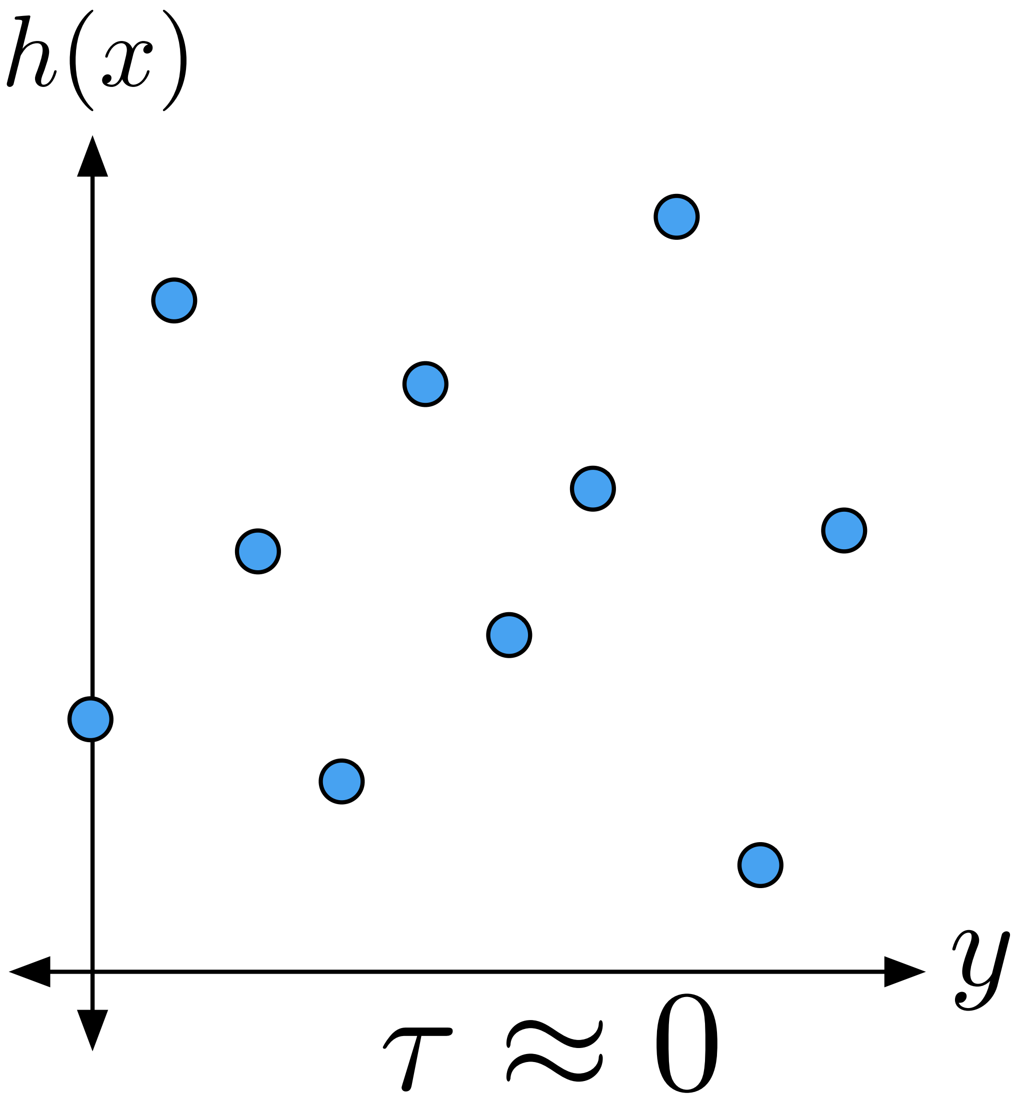
- Want a statistic that counts the number of incorrect orderings
- Kendal Rank Correlation Coefficient (τ)
  - Measure of monotonic correlation $\tau \in [-1, 1]$
  - Normalized number of correctly ranked pairs
- **Only rank examples from the same problem instance**
- Larger τ is generally better in our application

Positive Kendall Rank Correlation

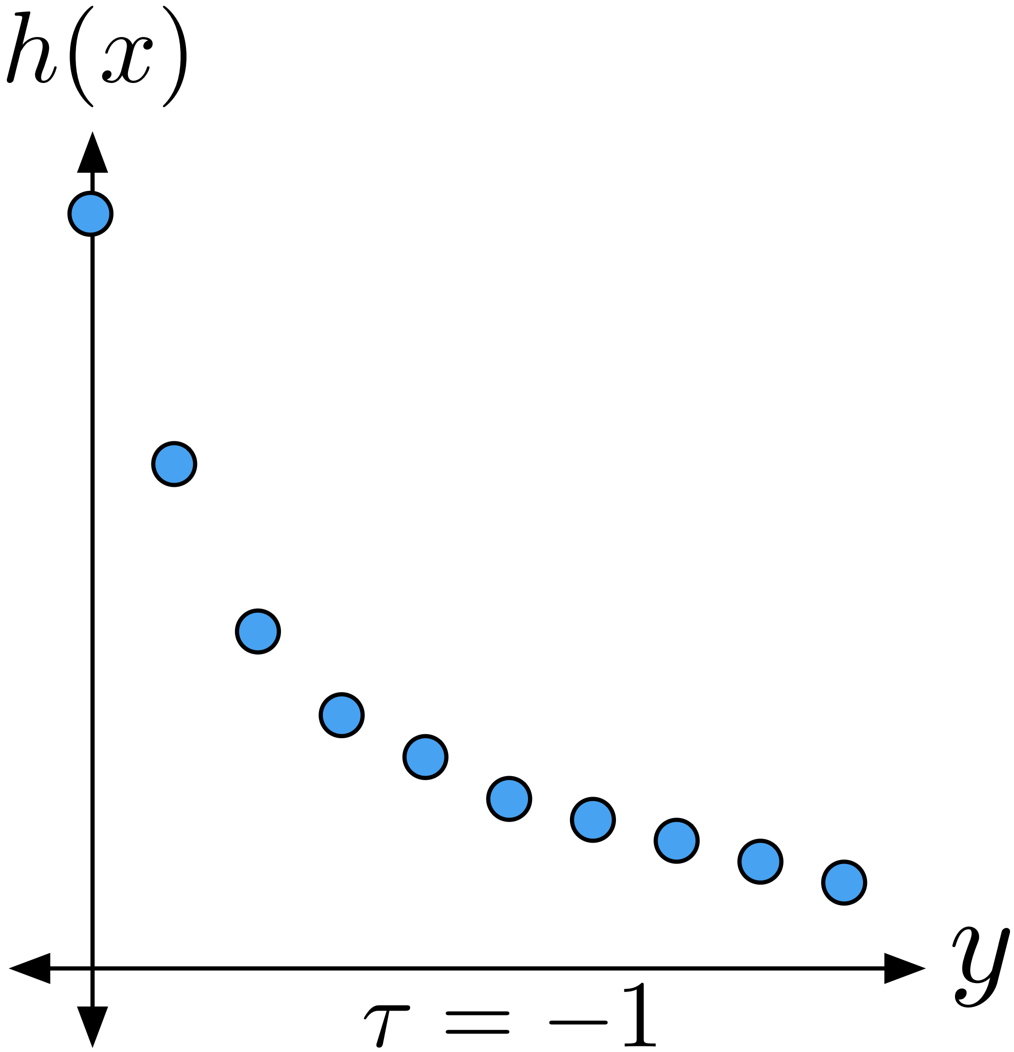


$\tau = 1$

Zero Kendall Rank Correlation



Negative Kendall Rank Correlation



Optimizing the Ranking Loss Function

- Rank Support Vector Machine (RSVM)
  - Optimize hinge loss relaxation of τ
  - Equivalent to SVM classifying ranking pairs

$$\min_{w} \quad ||w||^2 + C \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=j+1}^{m_i} \xi_{ijk}$$

$$\text{s.t.} \quad (\phi(x_j^i) - \phi(x_k^i))^T w \geq 1 - \xi_{ijk}, \; \forall y_j^i \geq y_k^i, \; \forall i$$

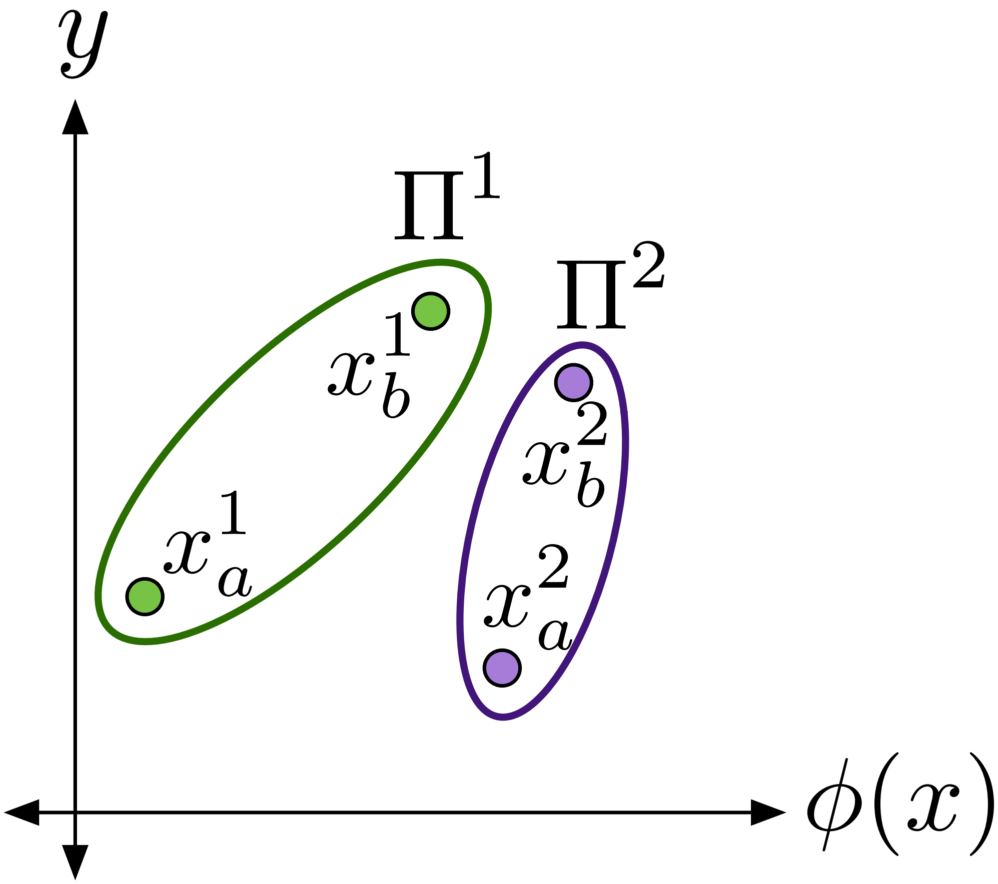
$$\xi_{ijk} \geq 0, \; \forall i, j, k$$

Why Not Use Regression?

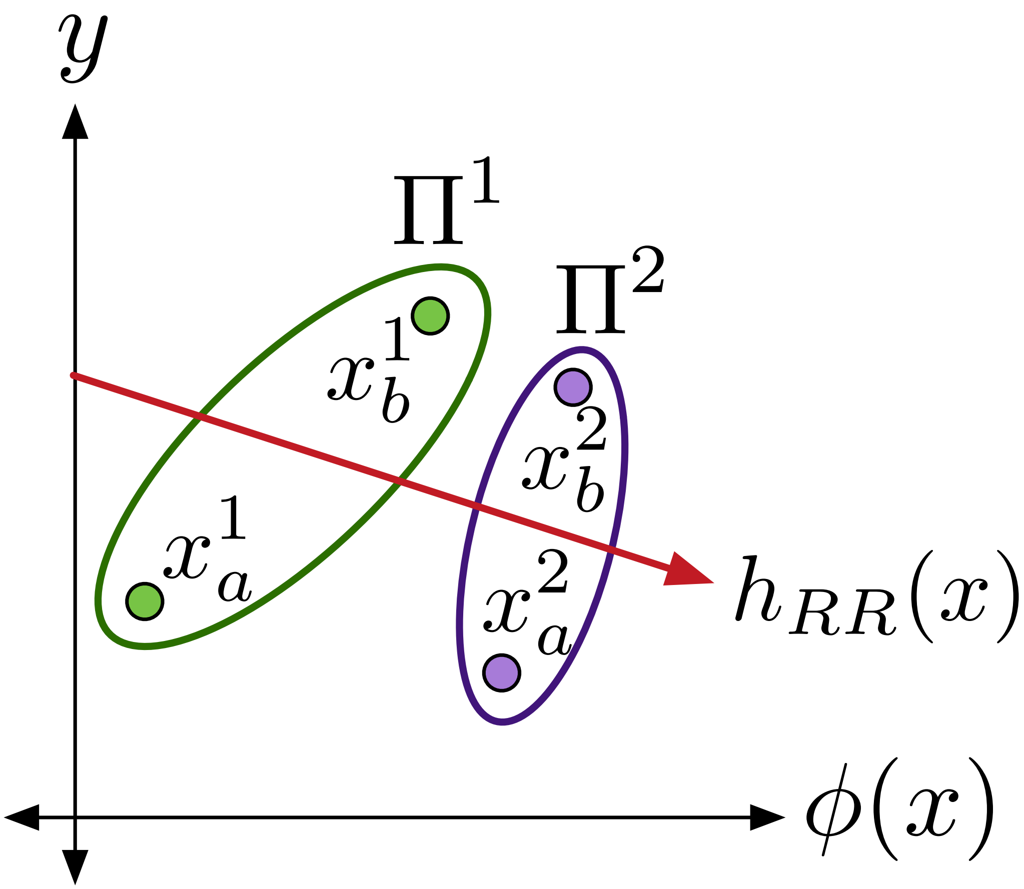
- Prior works optimize Root Mean Squared Error (RMSE) using Ordinary Least Squares Regression

- Can compromise ordering to obtain better average numerical error
- May sometimes create/widen local minima
- Particularly problematic with training noise and imperfect feature representation

1D - 2 problems with 2 states each

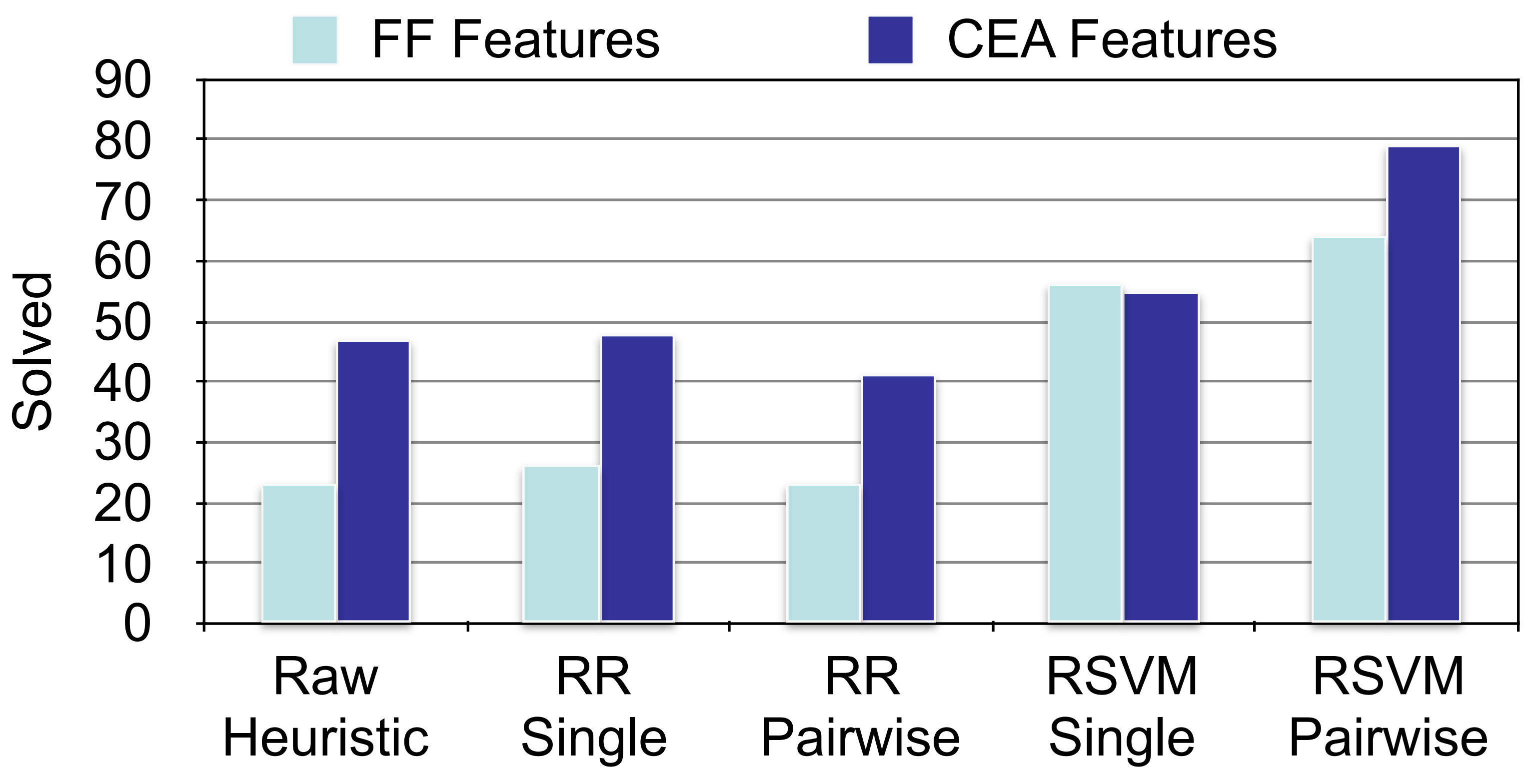


Ridge Regression Solution



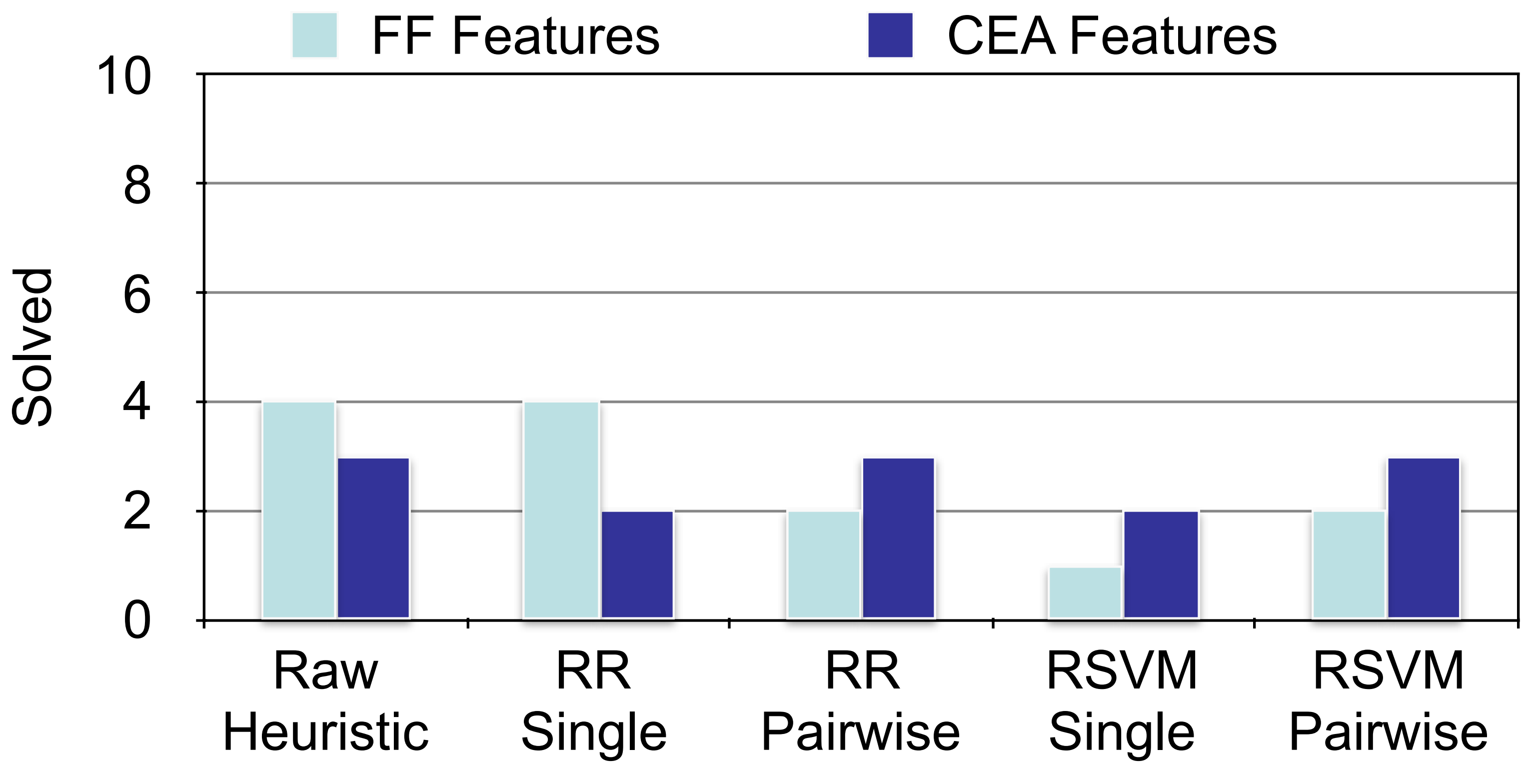
Testing Problems Coverage

- IPC 2014 Learning Track generators - *elevators, transport, parking, no-mystery*
- Harder than instances used in competition



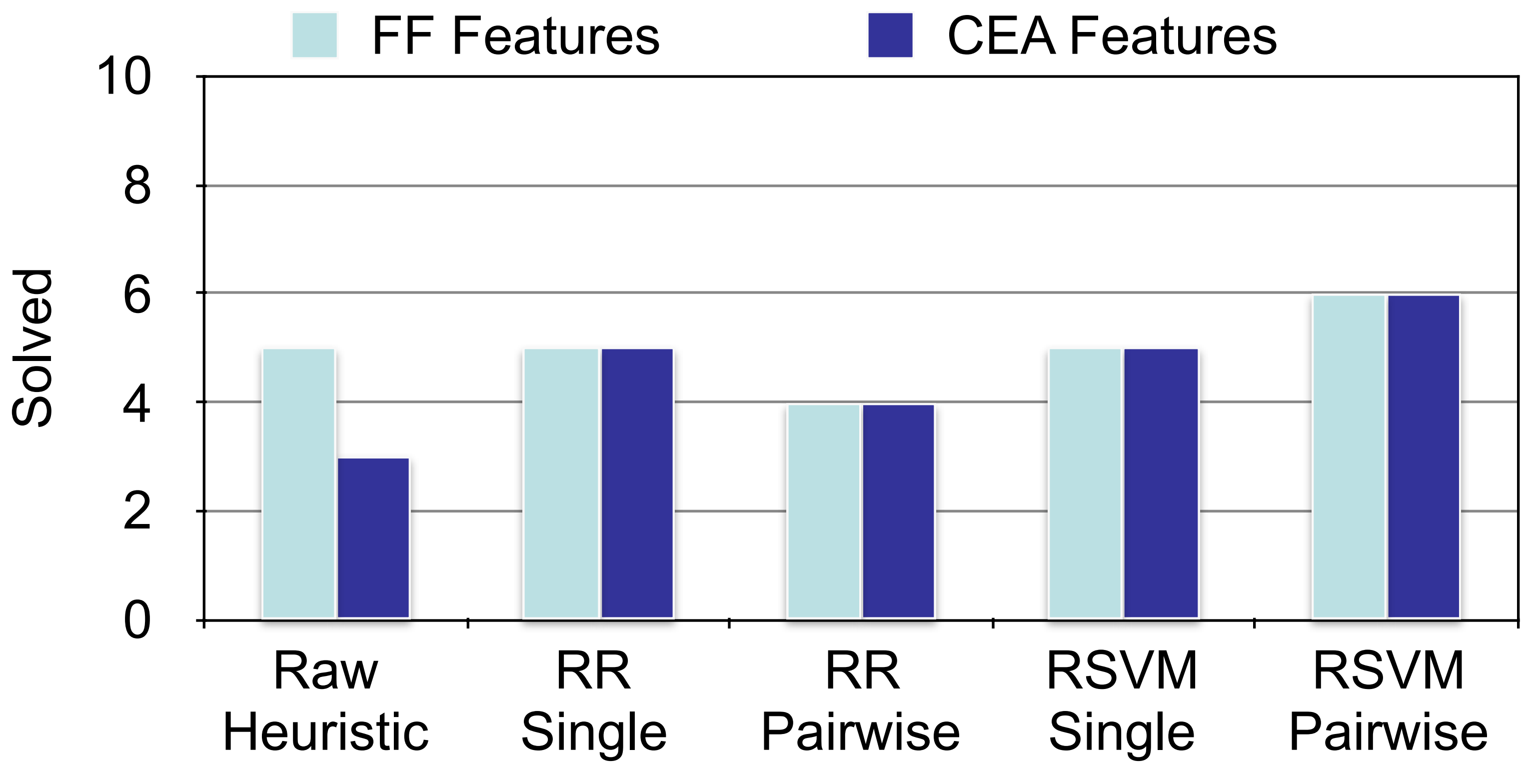
No-Mystery Lazy Best-First

- Heuristic learning alone not effective on domains with harmful dead-ends



No-Mystery Eager Best-First

- Eager best-first search does better but the improvement is not significant



Comparison on IPC Instances

- RSVM lazy best-first search
  - *elevators* (5/5), *transport* (5/5), *parking* (5/5), *no-mystery* (1/5)
- RSVM eager best-first search
  - *no-mystery* (5/5)
- Combined coverage (20/20)
- Competition winner using portfolio (17/20)

Takeaways

- **Only ordering matters** for greedy search heuristics
- **Learning with a ranking loss function** improves performance over raw heuristic or regression heuristic
- **Pairwise features** able to encode information about approximate plan without feature extraction
- Learning most effective on large domains without harmful dead-ends

Selected References

Jorg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. Journal Artificial Intelligence Research (JAIR), 14:253–302, 2001.

Malte Helmert and Hector Geffner. Unifying the causal graph and additive heuristics. In ICAPS , pages 140–147, 2008.

Sungwook Yoon, Alan Fern, and Robert Givan. Learning control knowledge for forward search planning. The Journal of Machine Learning Research, 9:683–718, 2008.

Christopher Makoto Wilt and Wheeler Ruml. Building a heuristic for greedy search. In Eighth Annual Symposium on Combinatorial Search, 2015.

Thorsten Joachims. Optimizing search engines using clickthrough data. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 133–142. ACM, 2002.