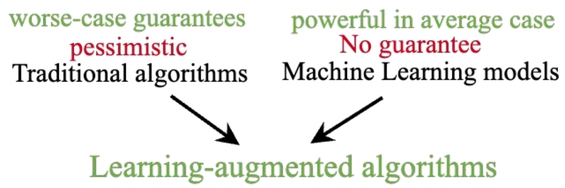
Sorting with Predictions

Xingjian Bai, Christian Coester
xingjian.bai@sjc.ox.ac.uk, christian.coester@cs.ox.ac.uk

Paper Link

Motivation:

Use possibly erroneous predictions from ML models to improve provably correct algorithms in classic algorithmic settings.



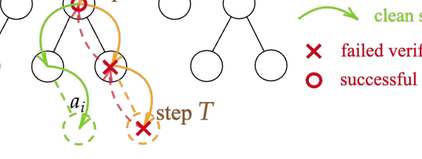
Two models of predictions:

Positional Setting

a_i	510	82	208	813	67	491
p_i	5	2	3	6	1	4
$\widehat{p_i}$	6	1	4	5	3	4

Given predicted ranking of each item.

Dirty Comparisons Setting



Given a faster and dirty comparison between each pair of items.

Error Measure of Predictors

$$\eta_i^{\Delta} := |\hat{p}(i) - p(i)|$$

$$\eta_i = |\{j : (a_i < a_j) \neq (a_i \hat{<} a_j)\}|$$

$$\eta_i^l := |\{j \in [n] : \hat{p}(j) \leq \hat{p}(i) \wedge p(j) > p(i)\}|$$

$$\eta_i^r := |\{j \in [n] : \hat{p}(j) \geq \hat{p}(i) \wedge p(j) < p(i)\}|$$

Comparison Complexity of Proposed Algorithms

Displacement Sort: $O\left(\sum_{i=1}^{n} \log\left(\eta_i^{\Delta} + 2\right)\right)$

Dirty-Clean Sort: $O\left(\sum_{i=1}^{n} \log\left(\eta_i + 2\right)\right)$

Double-Hoover Sort: $O\left(\sum_{i=1}^{n} \log\left(\min\left\{\eta_i^l, \eta_i^r\right\} + 2\right)\right)$

Classic Entropic Bound: O(n log n)

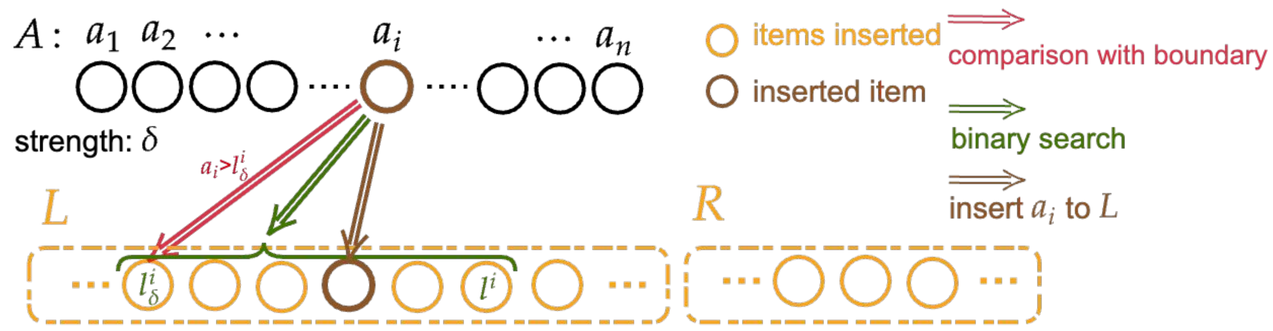
Dirty-Clean Sort:

Insert items into a Binary Search Tree in randomized order.
For each item, insertion contains three phases:

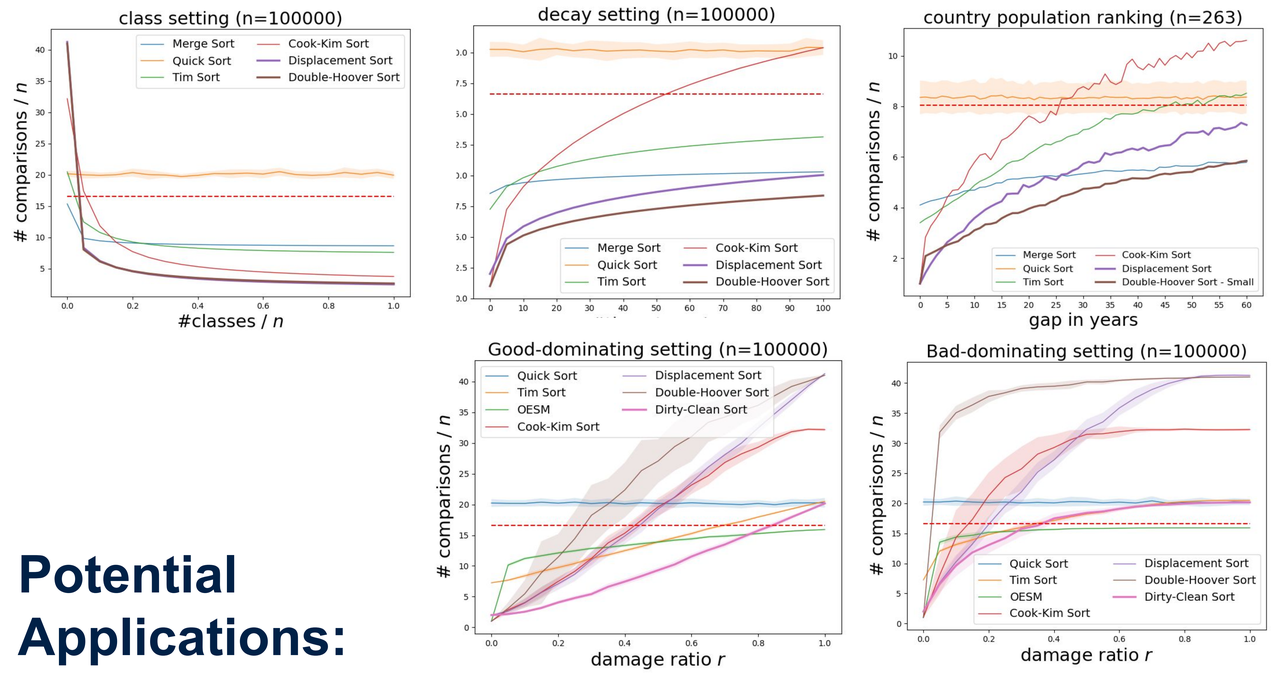
1. **Dirty Search**: trace root-to-leaf path using dirty comparisons.
2. **Verification**: go up by clean comparisons to find the first correct subtree.
3. **Clean Search**: insert the item back to BST following clean comparisons.

Double-Hoover Sort:

1. Two "Hoovers", L and R, have exponentially increasing strengths to "suck in" items: δ=1, 2, 4, and so on.
2. At strength δ, the hoovers suck in items that can be inserted within log(δ) comparisons.
3. To insert an item, a hoover conducts a binary search to locate the inserting position.



Experimental Results:



Potential Applications:

- ❑ Drug efficiency testing
- ❑ Draw conclusions from social experiments
- ❑ (any scenarios with faster rough comparisons available!)