UNIVERSITY OF CAMBRIDGE

UNIVERSITY OF CAMBRIDGE COMPUTER LABORATORY
RAINBOW RESEARCH GROUP

erc
European Research Council
Established by the European Commission

Active Sampling for Pairwise Comparisons via Approximate Message Passing and Information Gain Maximization

Aliaksei Mikhailiuk, Clifford Wilmot, Maria Perez-Ortiz, Dingcheng Yue, Rafal K. Mantiuk

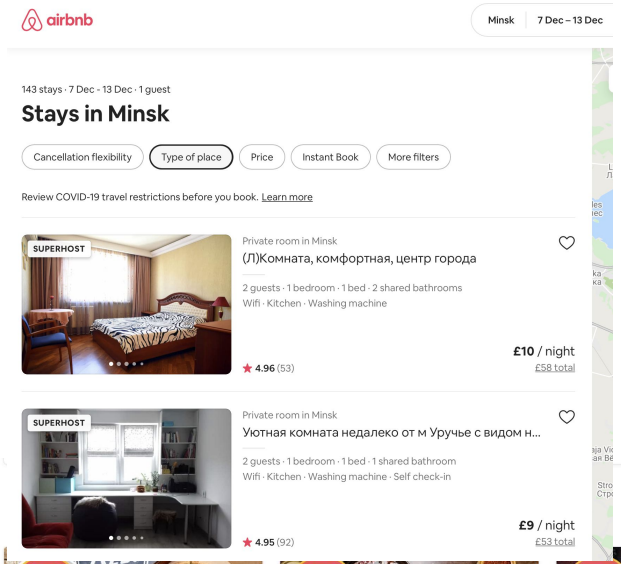
Problem overview - Inferring Scale

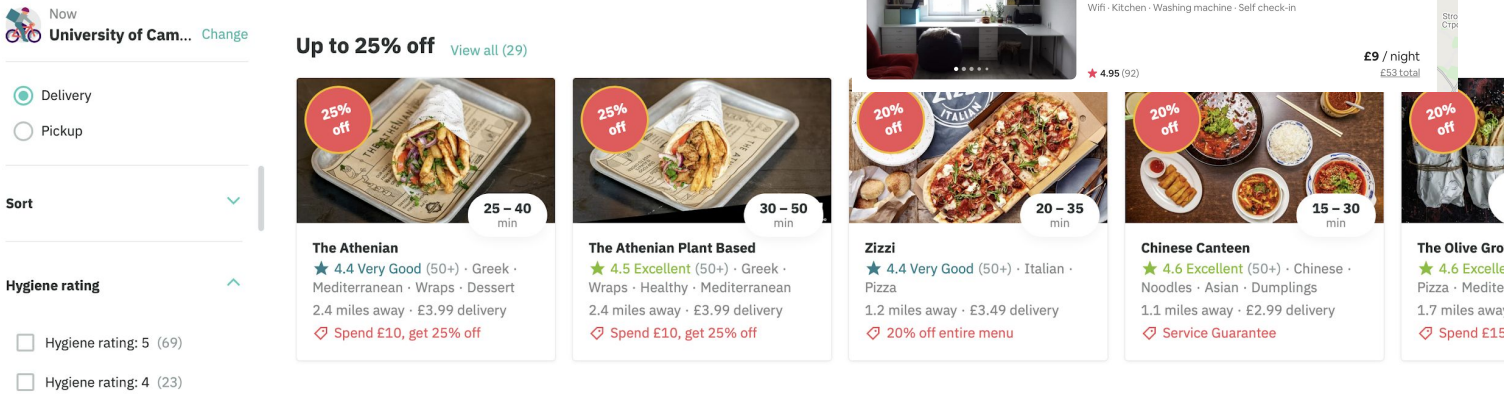
WCSC 2010 Kanazawa 2010

		1	2	3	4	5	6	7	8	9	
1	**Shredder**	*	½	1	½	1	1	1	1	1	7.0 / 8
2	**Rondo**	½	*	½	1	½	1	1	1	1	6.5 / 8
3	**Thinker**	0	½	*	½	1	½	1	1	1	5.5 / 8
4	**Pandix Breakthrough**	½	0	½	*	½		1	1	1	4.5 / 7
5	**Deep Junior**	0	½	0	½	*		1	1	1	4.0 / 7
6	**Jonny**	0	0	½			*		1	1	2.5 / 5
7	**Darmenios**	0	0	0	0	0		*	1	1	2.0 / 7
8	**Fridolin**	0	0	0	0	0	0	0	*	1	1.0 / 8
9	**Hector For Chess**	0	0	0	0	0	0	0	0	*	0.0 / 8

(33 Games)

Problem overview





Motivation

- Pairwise comparisons are often used in crowdsourcing.

Motivation

- Pairwise comparisons are often used in crowdsourcing.
- Drawback - for n conditions there are n(n-1)/2 pairs.

Motivation

- Pairwise comparisons are often used in crowdsourcing.
- Drawback - for n conditions there are n(n-1)/2 pairs.
- How can we minimize the number of comparisons?

Motivation

- Pairwise comparisons are often used in crowdsourcing.
- Drawback - for n conditions there are n(n-1)/2 pairs.
- How can we minimize the number of comparisons?
- Two types of methods:

Motivation

- Pairwise comparisons are often used in crowdsourcing.
- Drawback - for n conditions there are n(n-1)/2 pairs.
- How can we minimize the number of comparisons?
- Two types of methods:
  - Sorting based (Swiss chess tournament system, Adaptive Rectangular Design (ARD))

Motivation

- Pairwise comparisons are often used in crowdsourcing.
- Drawback - for n conditions there are n(n-1)/2 pairs.
- How can we minimize the number of comparisons?
- Two types of methods:
  - Sorting based (Swiss chess tournament system, Adaptive Rectangular Design (ARD))
  - Information gain based - assume data model (Hybrid-MST, Crowd-BT, HR-Active)

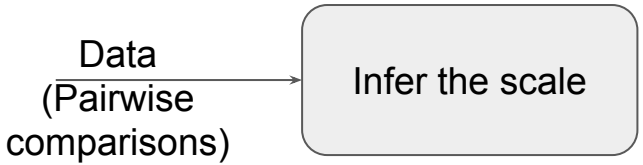
Motivation

- Pairwise comparisons are often used in crowdsourcing.
- Drawback - for n conditions there are n(n-1)/2 pairs.
- How can we minimize the number of comparisons?
- Two types of methods:
  - Sorting based (Swiss chess tournament system, Adaptive Rectangular Design (ARD))
  - Information gain based - assume data model (Hybrid-MST, Crowd-BT, HR-Active, **ASAP**)

Algorithm

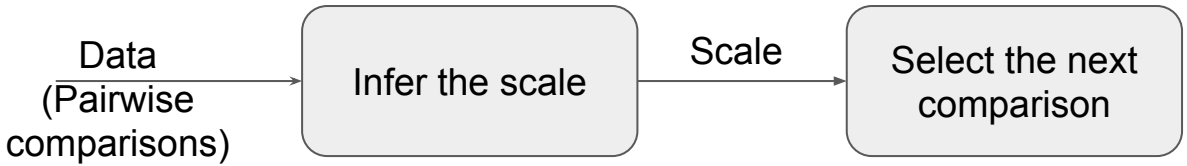
Data
(Pairwise
comparisons)

Algorithm



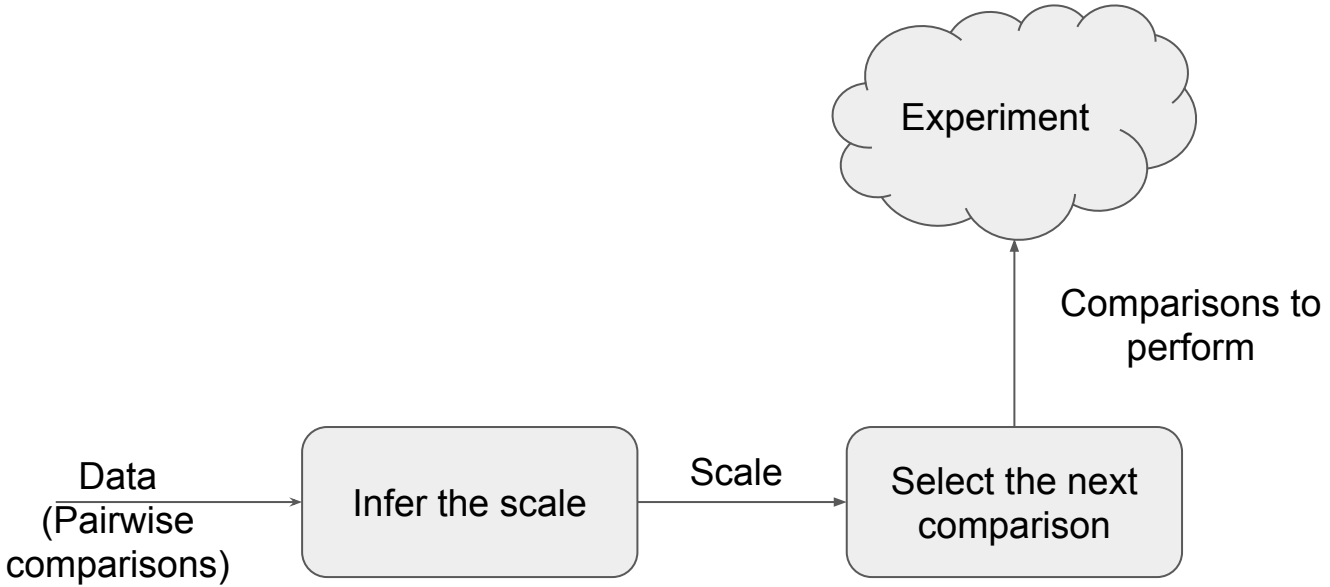
Most methods approximate, we do not

Algorithm

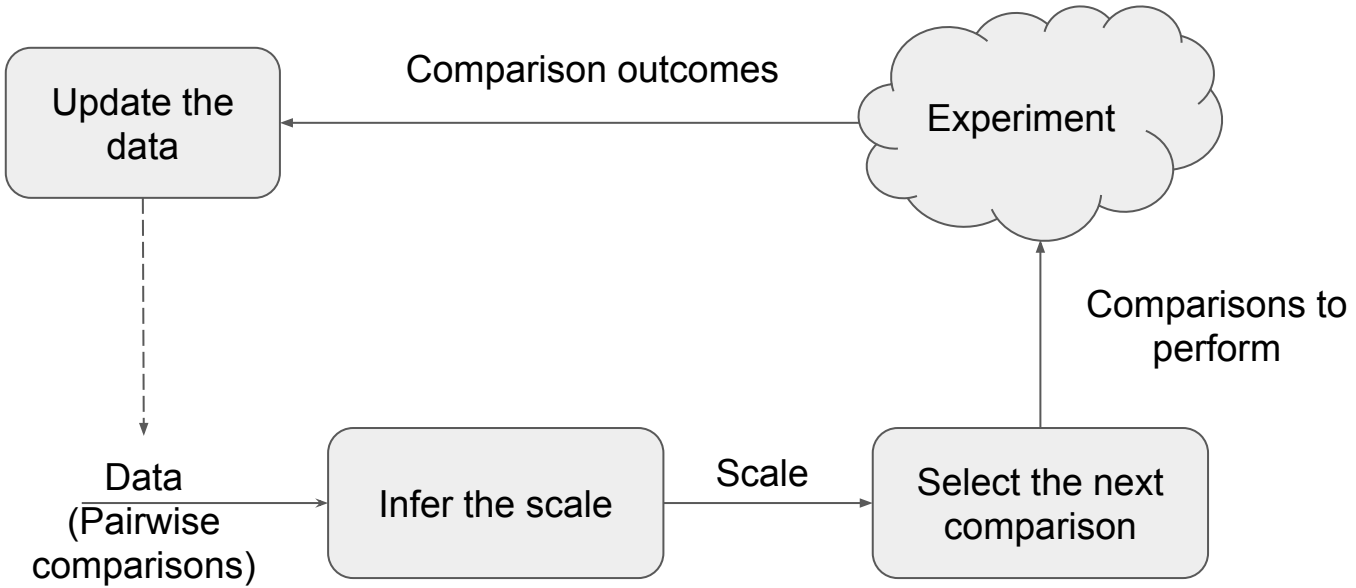


Most methods do not improve speed, we do

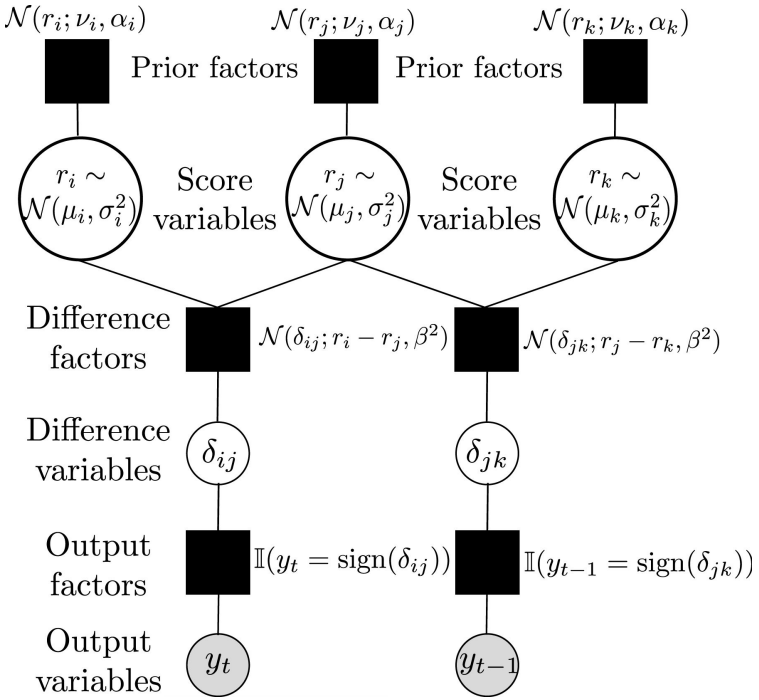
Algorithm


Experiment
Comparisons to perform
Data (Pairwise comparisons)
Infer the scale
Scale
Select the next comparison

Algorithm

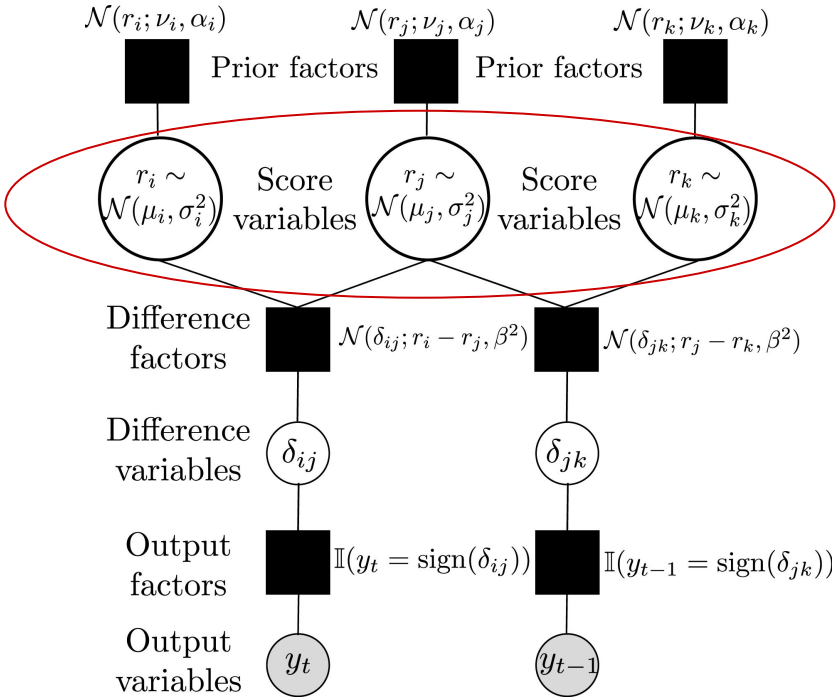

Update the data
Comparison outcomes
Experiment
Comparisons to perform
Data (Pairwise comparisons)
Infer the scale
Scale
Select the next comparison

Bayesian Score Inference (TrueSkill based [1])

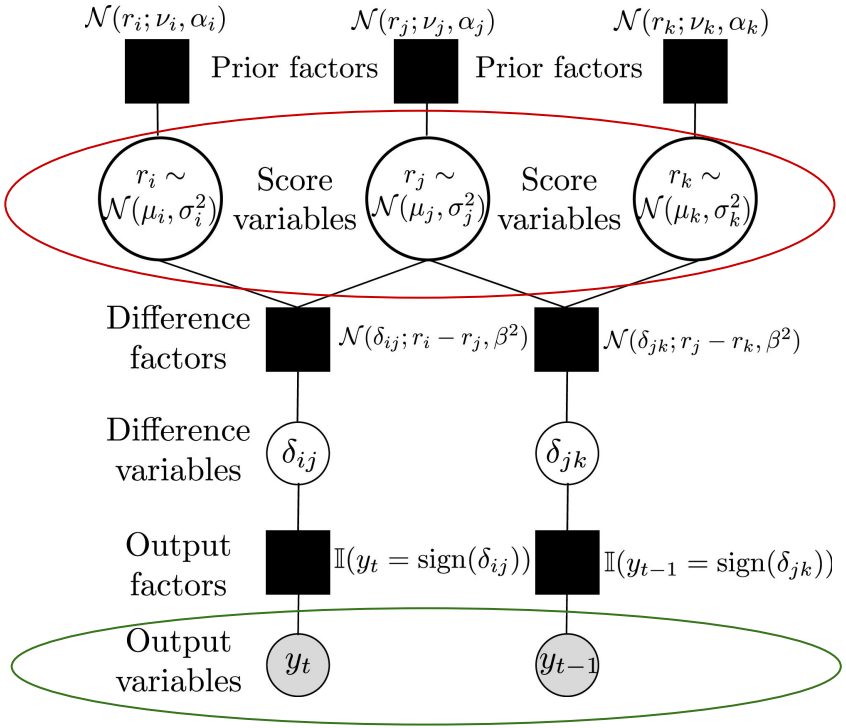


T. Minka, R. Cleven, and Y. Zaykov, “Trueskill 2: An improved bayesian skill rating system,” Microsoft, Tech. Rep. MSR-TR-2018-8, March 2018.

Bayesian Score Inference (TrueSkill based)

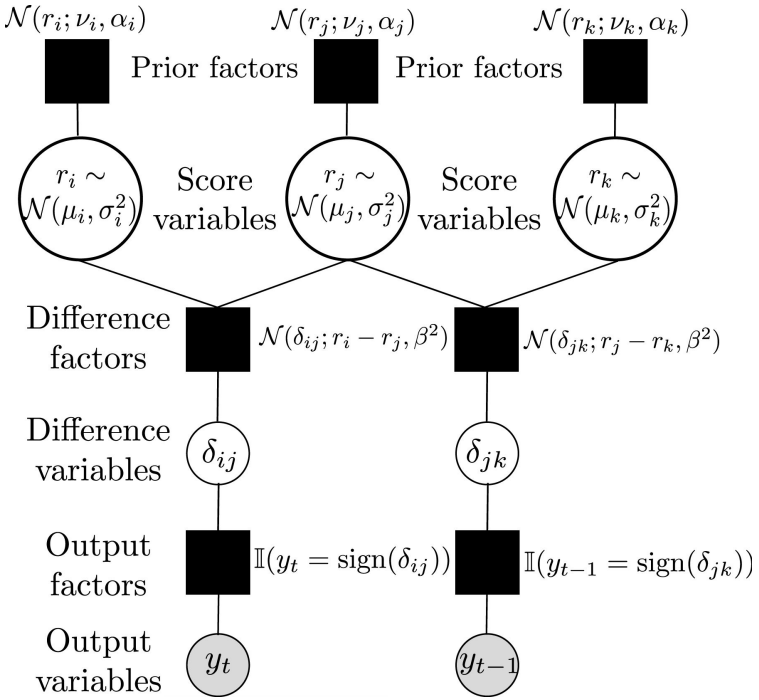


Bayesian Score Inference (TrueSkill based)



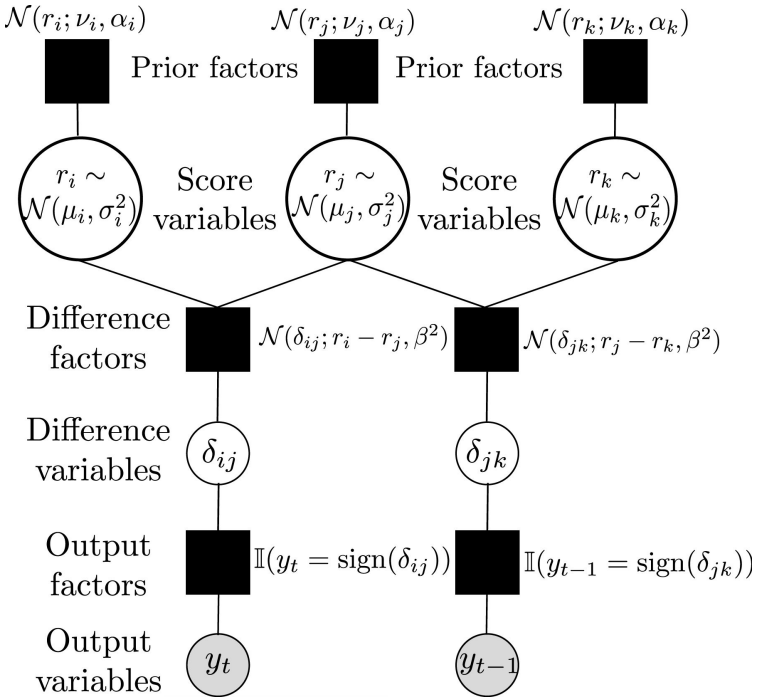
Bayesian Score Inference (TrueSkill based)

- Sum-product algorithm using expectation propagation via moment matching



Bayesian Score Inference (TrueSkill based)

- Sum-product algorithm using expectation propagation via moment matching
- k*O(n+t) complexity of inferring the scores (k - number of iterations)



Expected Information Gain (EIG)

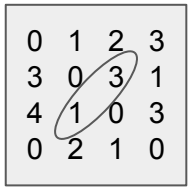
Expected Information Gain (EIG)

Collected data so far

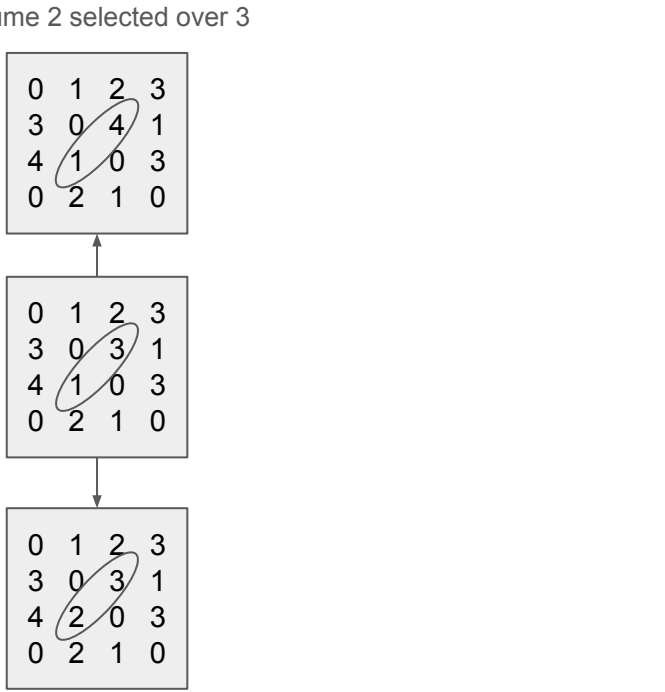
0	1	2	3
3	0	3	1
4	1	0	3
0	2	1	0

Expected Information Gain (EIG)

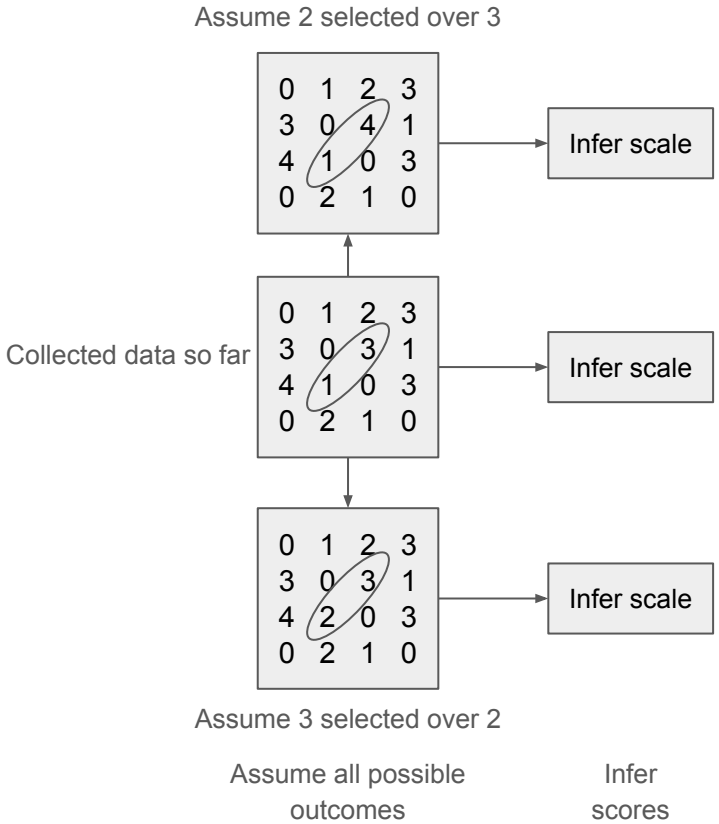
Collected data so far



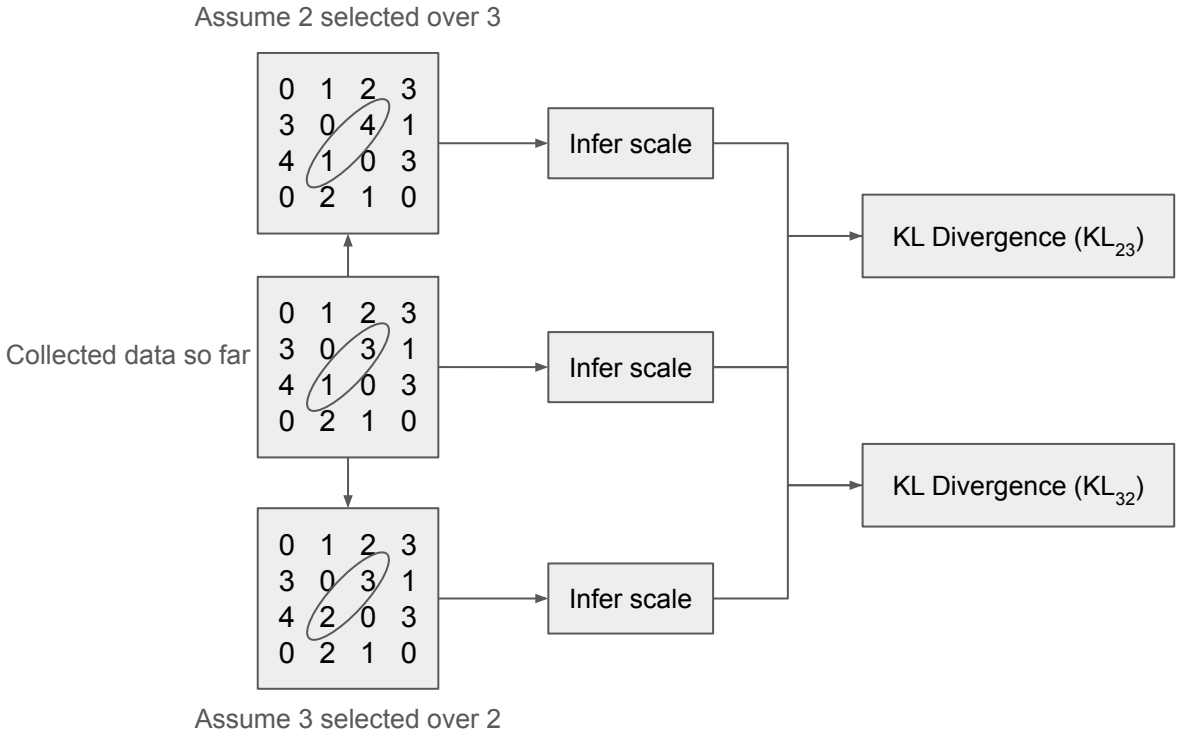
Expected Information Gain (EIG)



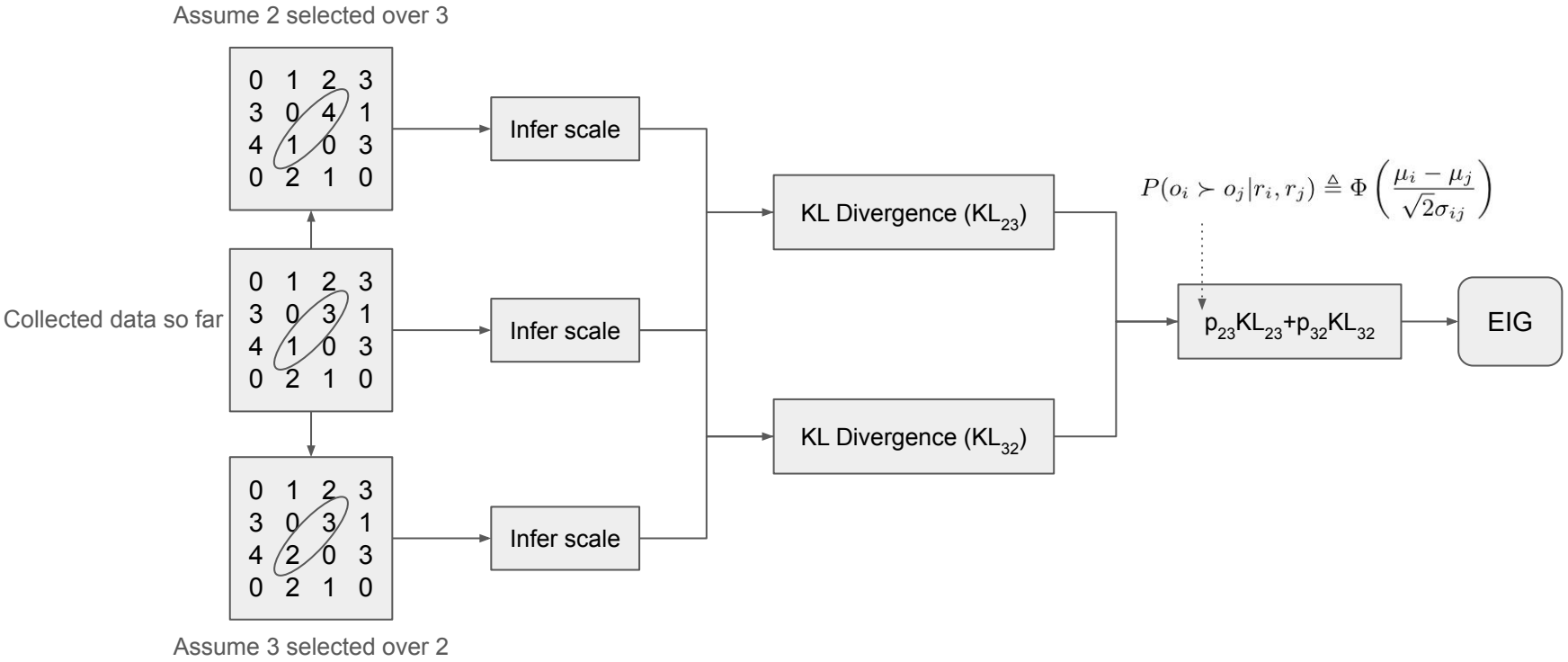
Expected Information Gain (EIG)



Expected Information Gain (EIG)



Expected Information Gain (EIG)



Efficiency considerations

- At every iteration t, there are n(n-1) comparison outcomes to consider.

Efficiency considerations

- At every iteration t, there are n(n-1) comparison outcomes to consider.
- We consider two strategies to improve the speed:

Efficiency considerations

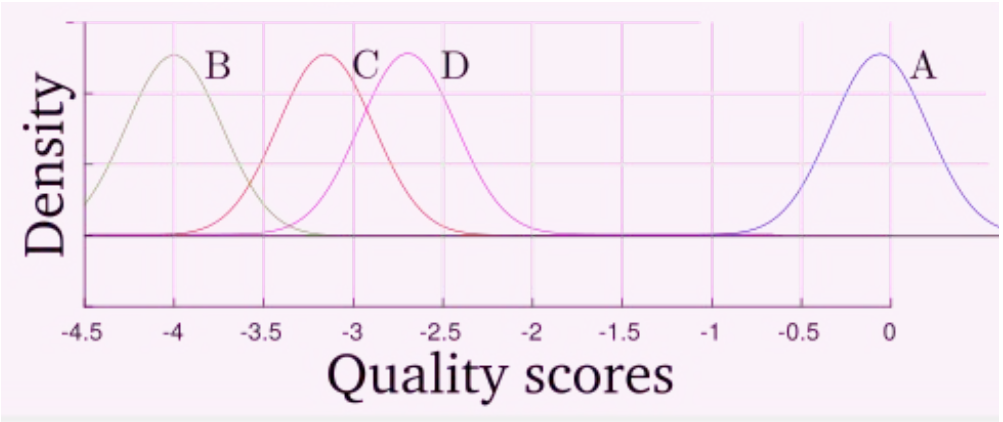
- At every iteration t, there are n(n-1) comparison outcomes to consider.
- We consider two strategies to improve the speed:
  - Selective EIG evaluations

Efficiency considerations

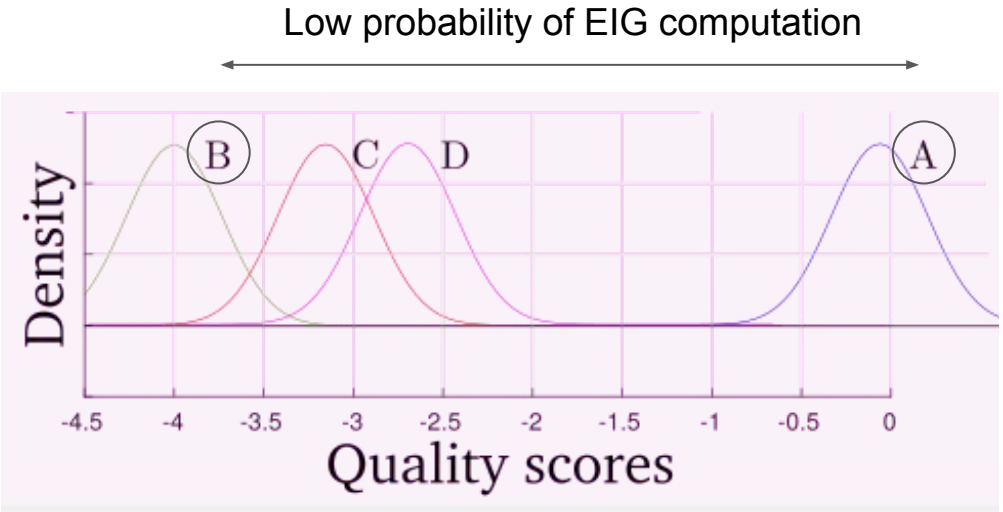
- At every iteration t, there are n(n-1) comparison outcomes to consider.
- We consider two strategies to improve the speed:
  - Selective EIG evaluations
  - Batch mode with Minimum Spanning Tree (MST)

Selective EIG evaluations

Selective EIG evaluations



Selective EIG evaluations



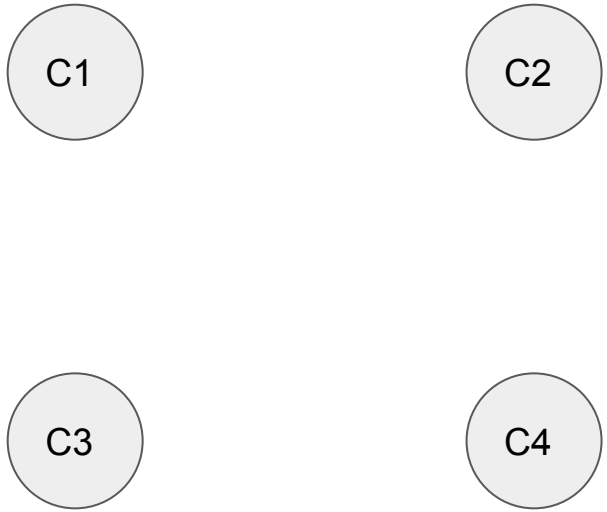
Selective EIG evaluations

High probability of EIG computation

Batch Mode

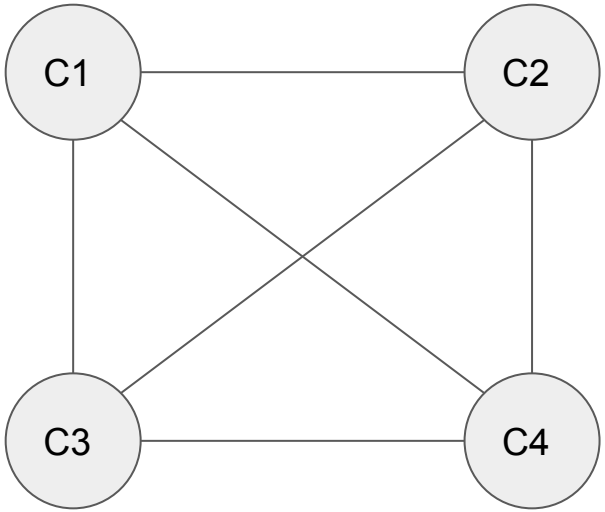
Batch Mode

- Vertices are conditions



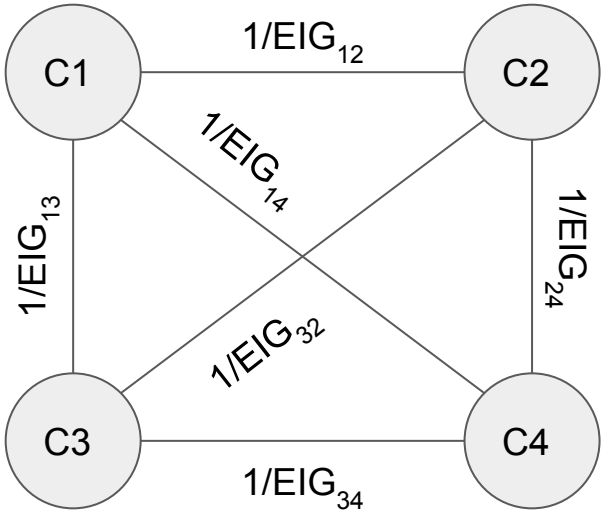
Batch Mode

- Vertices are conditions
- Edges are possible comparisons



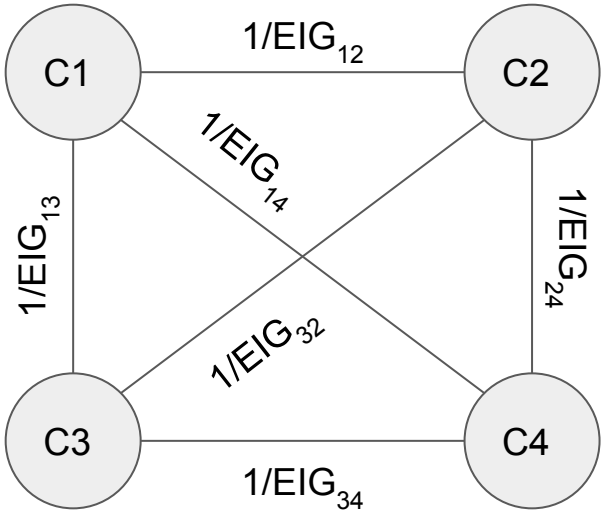
Batch Mode

- Vertices are conditions
- Edges are possible comparisons
- Weights are inverse of the expected information gain for comparison



Batch Mode

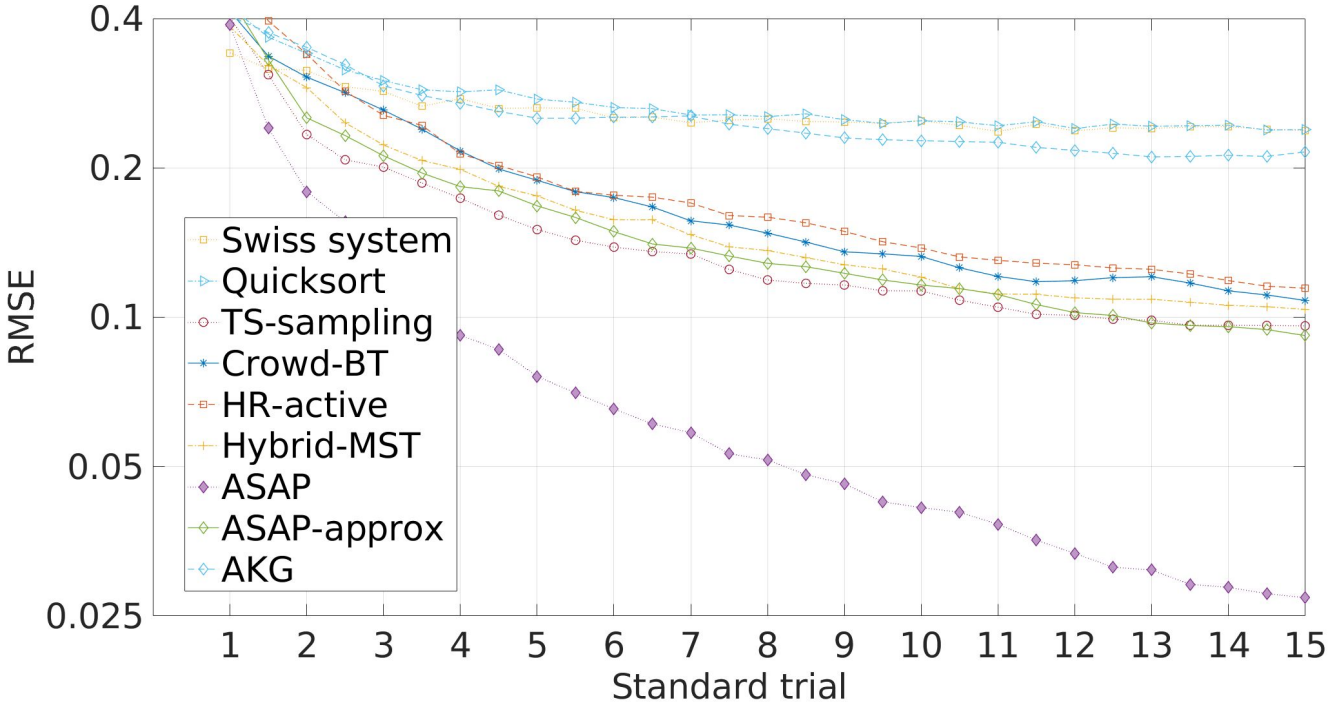
- Vertices are conditions
- Edges are possible comparisons
- Weights are inverse of the expected information gain for comparison
- Batch - comparisons forming edges in the minimum spanning tree (MST) [2]



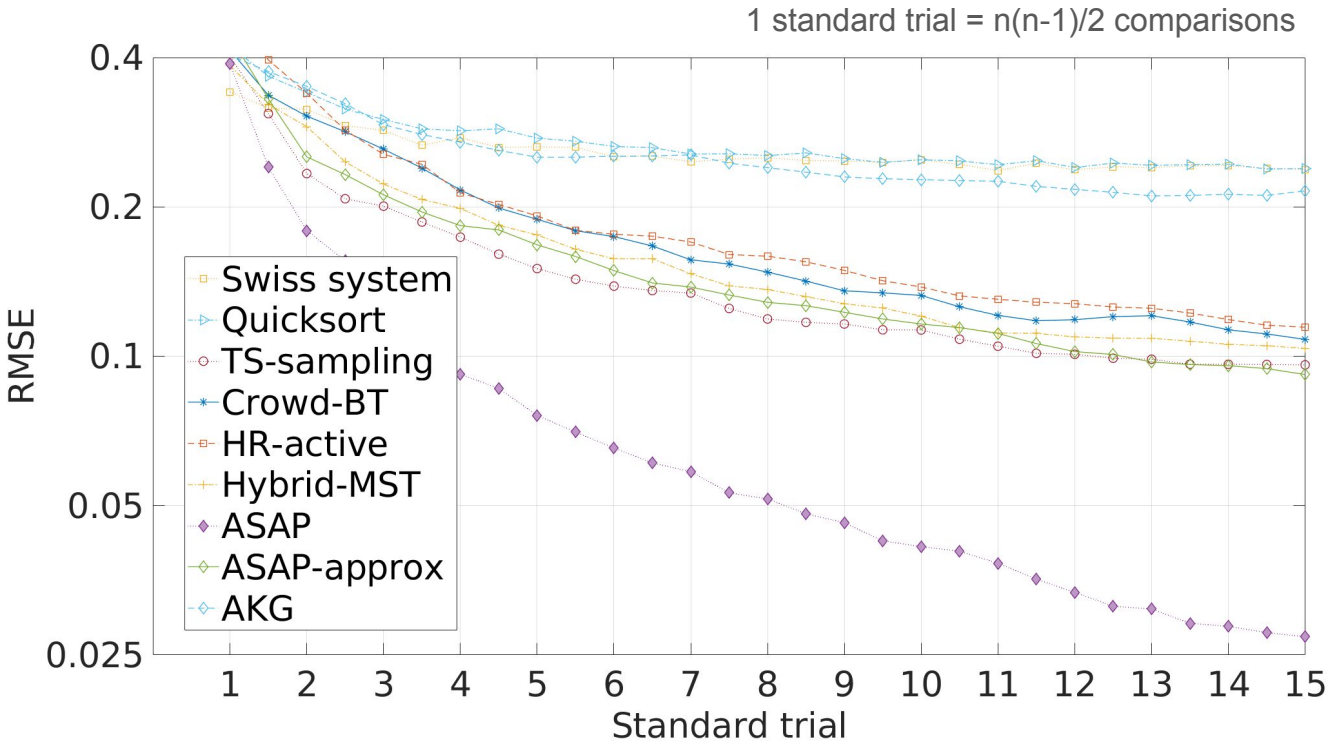
[2] J. Li, R. Mantiuk, J. Wang, S. Ling, and P. Le Callet, “Hybrid-MST: A hybrid active sampling strategy for pairwise preference aggregation,” NIPS, 31st Conference on Neural Information Processing Systems, 2018.

Monte Carlo Simulation

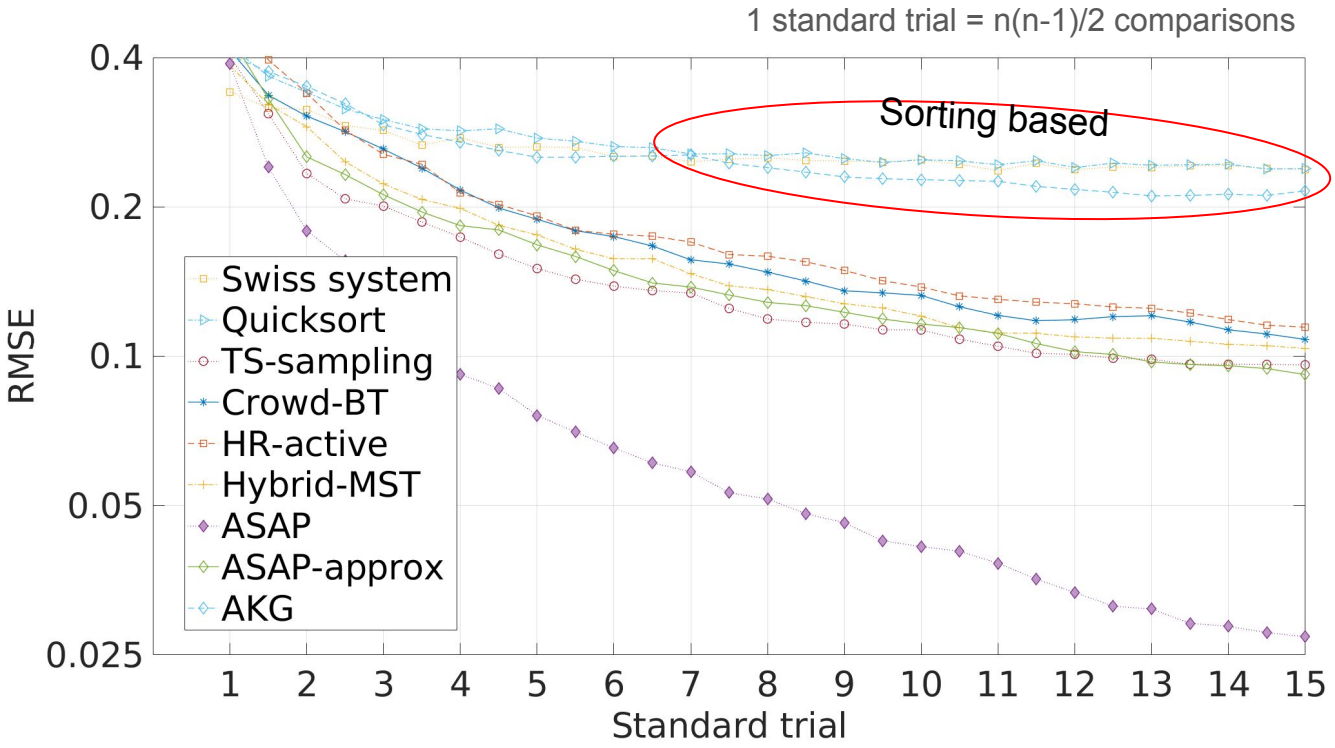
Monte Carlo Simulation



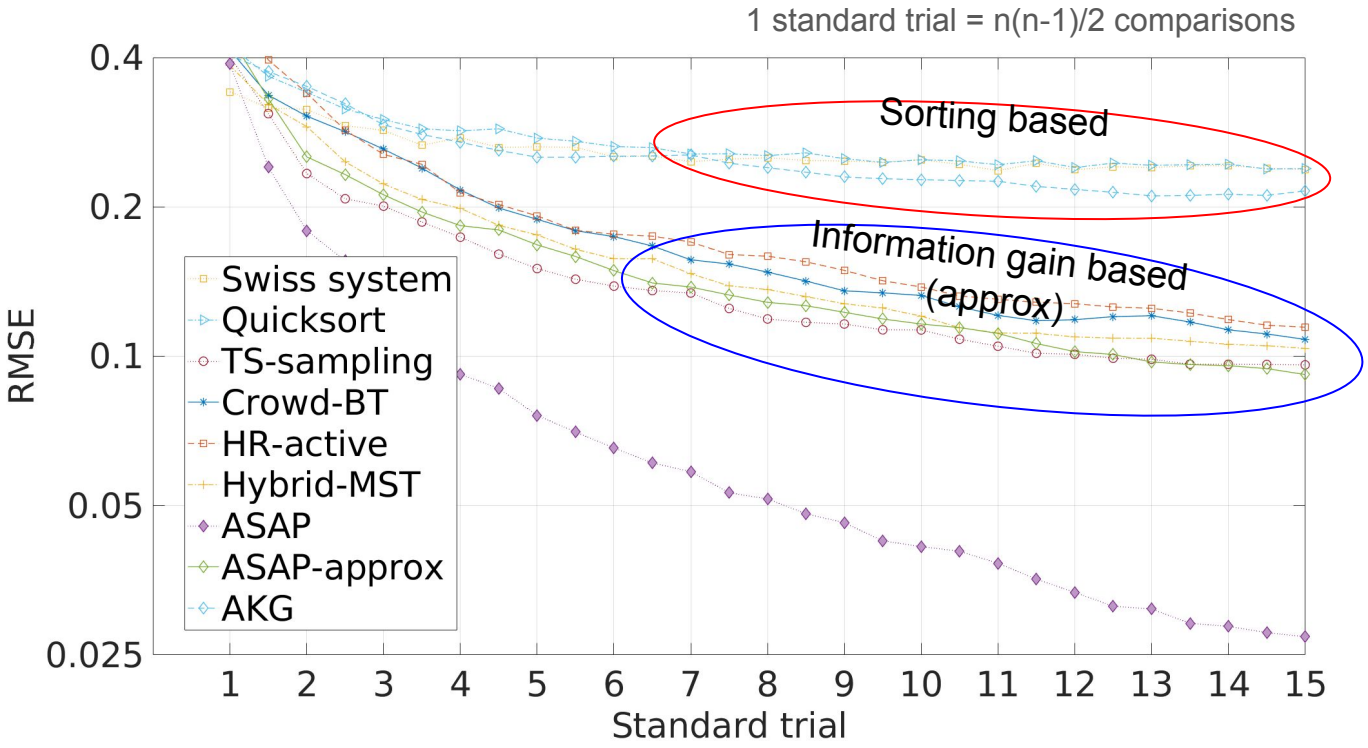
Monte Carlo Simulation



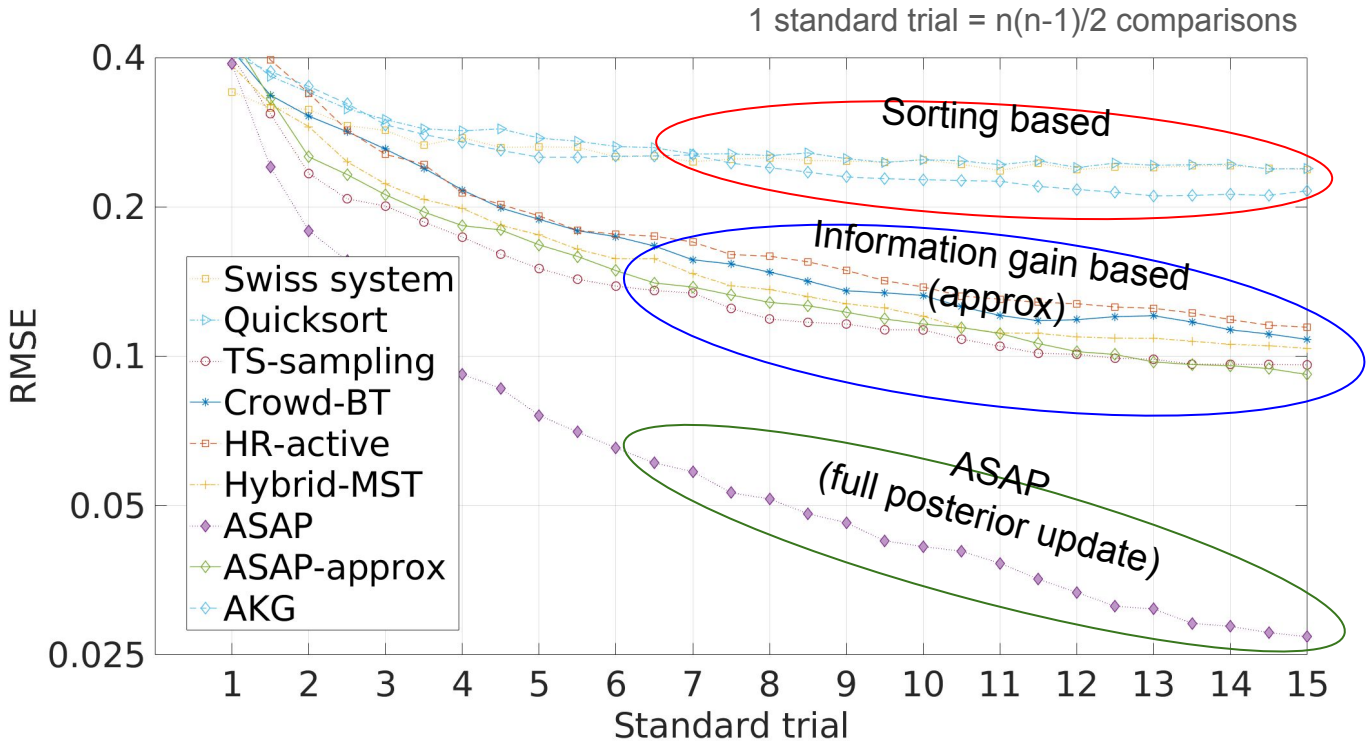
Monte Carlo Simulation



Monte Carlo Simulation

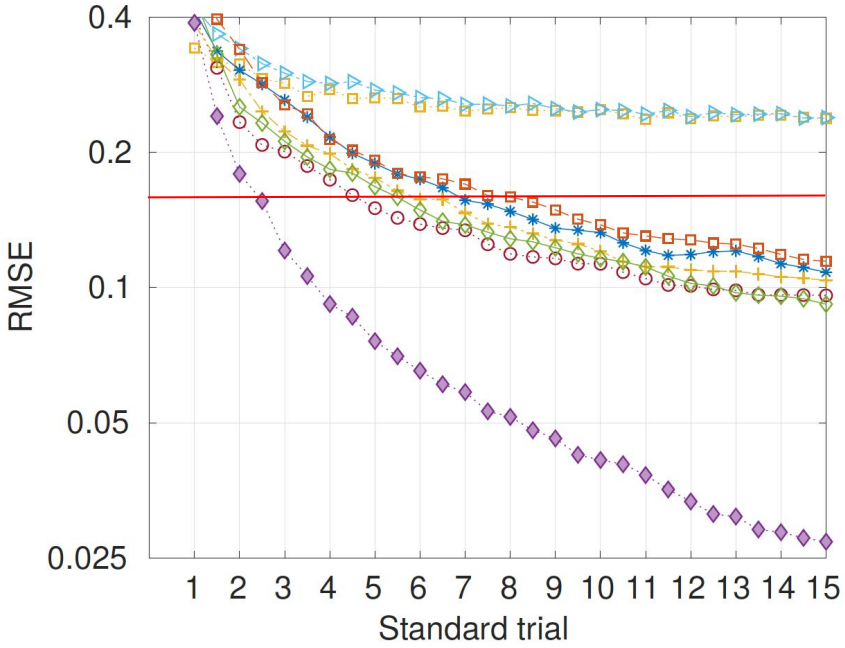


Monte Carlo Simulation



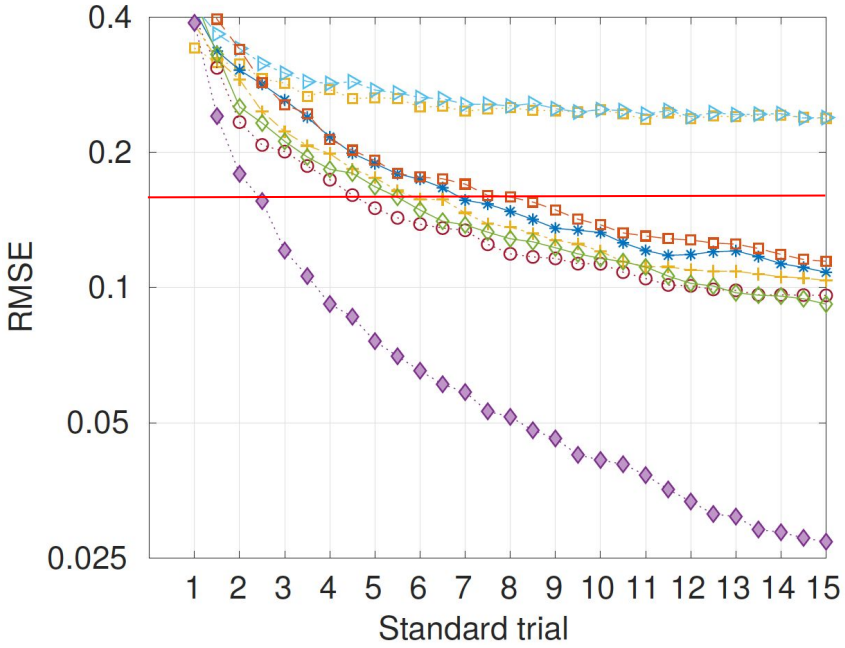
Experimental Effort

- Experimental Effort = time to achieve 0.15 RMSE

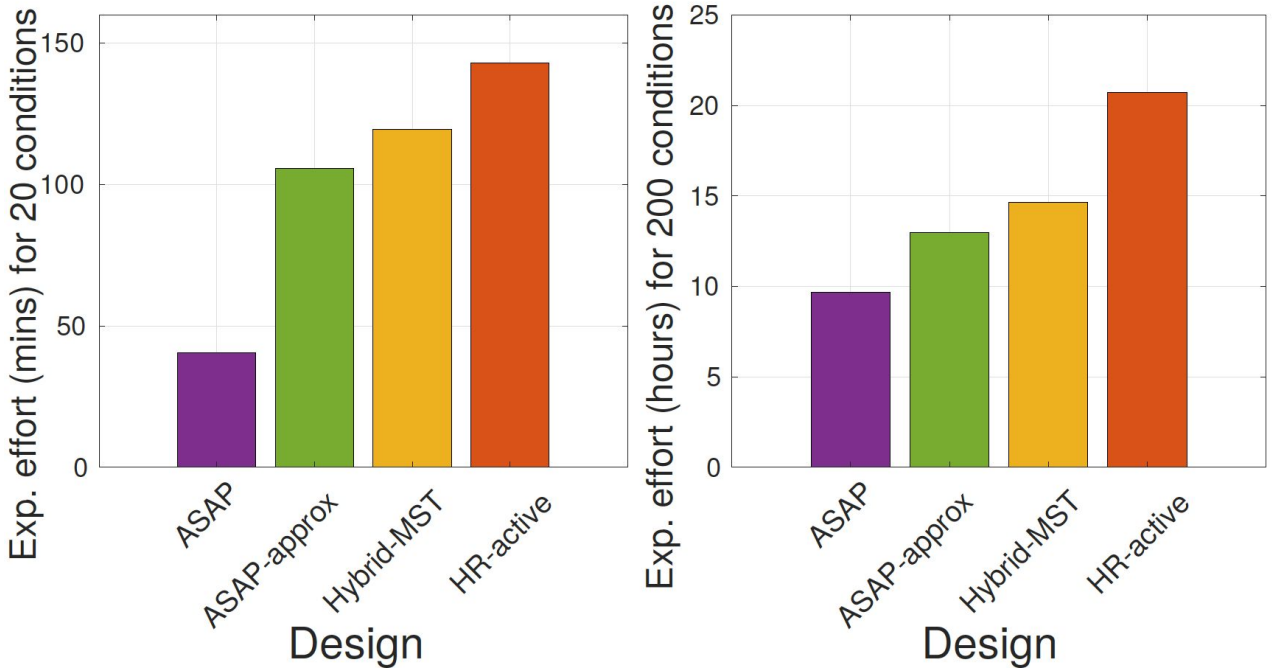


Experimental Effort

- Experimental Effort = time to achieve 0.15 RMSE
- Assume 5s decision time



Experimental Effort



Summary

- **Current methods** are sub-optimal, relying on **partial update** of the posterior distribution;
- **ASAP** computes the **full posterior update**, crucial to achieving **highest accuracy**;
- **ASAP** computes EIG for most informative pairs, **reducing the computational cost** by up to 80%;
- **ASAP** selects **batches** using a minimum spanning tree method**.**

Code: https://github.com/gfxdisp/asap

Contact: am2442@cam.ac.uk