School of Computer and Communication Sciences
Laboratory for Communications and Applications 4

Lucas Maystre and Matthias Grossglauser

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

A Simple and Effective Approach to Active Preference Learning

Motivation

Goal: **learn a ranking** from outcomes of **pairwise comparisons**.

GIFGIF (gif.gf) *which better expresses happiness?*

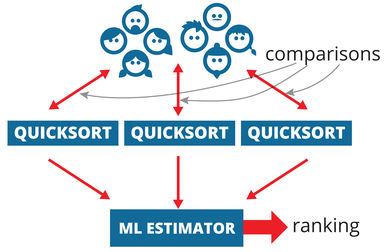
global ranking

Comparison outcomes are **noisy**, i.e., at times **inconsistent with the ranking**.

How to **select which items to compare**?

Key Idea

Just use a **sorting algorithm**!



Computational Gains

Time in seconds to select the (n+1)-st pair.

Strategy	T [s] $n = 10^2$	$n = 10^3$	$n = 10^4$
uncertainty	0.05	0.5	11
entropy	0.3	40	—
KL-divergence	0.9	71	—
Mergesort	ε	ε	ε
Quicksort	ε	ε	ε
random	ε	ε	ε

$\varepsilon < 10^{-5}$

Simple to implement and takes a **fraction of the time** compared to other approaches.

Contributions

1. We study **Quicksort** when comparisons outcomes are **noisy**, under natural assumptions.
2. We show that *repeatedly sorting the items* is an **effective active learning strategy** in practice.

choix — A **Python library** for inference in the Bradley–Terry model. `https://github.com/lucasmaystre/choix`

Quicksort with Noisy Comparisons

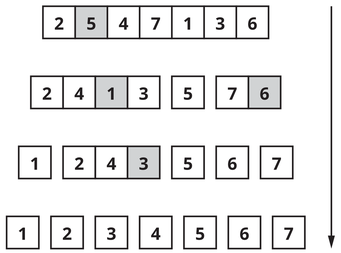
Noise: we assume the **Bradley-Terry model**.

Probability of outcome depends on difference between item's latent scores

$$p(i \prec j) = \frac{1}{1 + e^{-(\theta_i - \theta_j)}}$$

"i is smaller than j" — score difference

Quicksort works by recursively **partitioning items** smaller / larger than a **pivot** item.



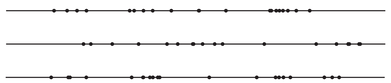
The output of Quicksort depends mostly on the parameters $\{\theta\}$.

We postulate that parameters are drawn from a **Poisson point process** of rate λ.

$$\text{i.i.d.} \quad \theta_{i+1} - \theta_i \sim \mathrm{Exp}(\lambda)$$

controls the amount of **noise**

Examples of parameters (n = 20):



True ranking (no noise):

Examples of Quicksort output for λ = 1.0:

Examples of Quicksort output for λ = 5.0:

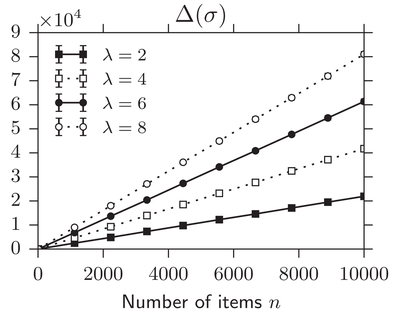
Theoretical Results

To measure the discrepancy between the output of Quicksort and the true tranking, we use **Spearman's footrule distance**.

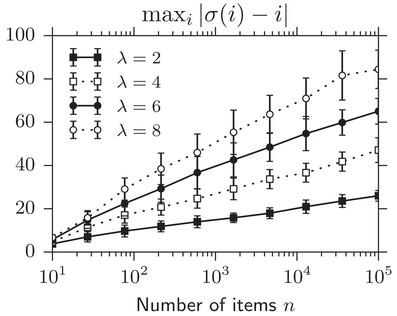
$$\Delta(\sigma) = \sum_{i=1}^{n} |\sigma(i) - i|$$

In short, our results show that **Quicksort's output is optimal** (up to constant factors) asymptotically almost surely.

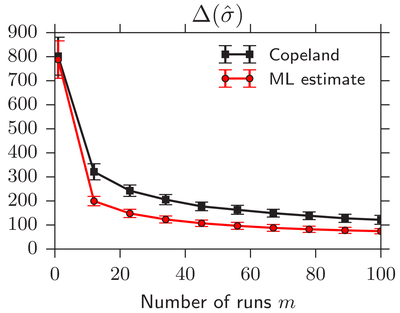
with high probability, $\Delta(\sigma) = O(\lambda^2 n)$



with high probability, $\max_i |\sigma(i) - i| = O(\lambda \log n)$

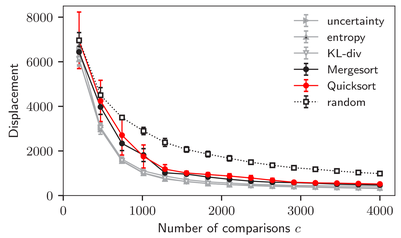


$O(\lambda^2 \log^5 n)$ outputs are enough to recover *exactly* almost all positions

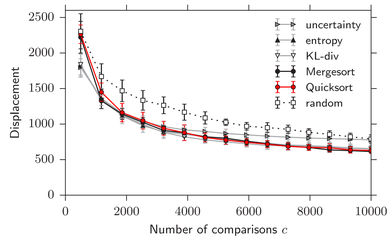


Experimental Results

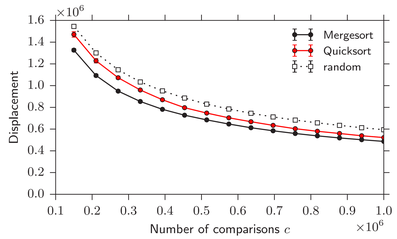
Synthetic dataset, n = 200, λ = 5.0



dataset of sushi preferences

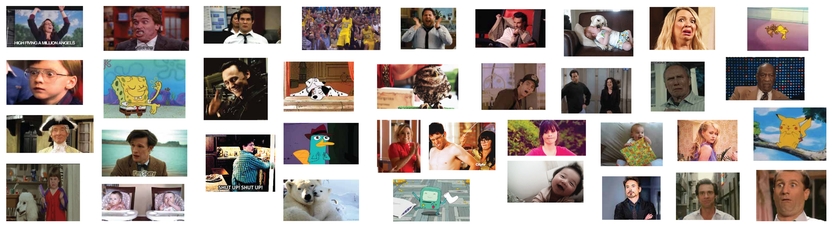


GIFGIF (gif.gf) *which better expresses happiness?*



GIFGIF Dataset

The GIFGIF team kindly allowed us to make their **dataset of pairwise comparison outcomes** public.

2.7+ million pairwise comparisons
6170 images

`http://lucas.maystre.ch/gifgif-data`