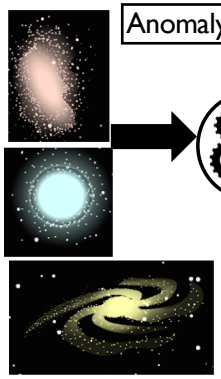
Learning On-the-Job to Re-rank Anomalies from Top-1 Feedback


Hemank Lamba
Carnegie Mellon University
School of Computer Science


Leman Akoglu
Carnegie Mellon University
HeinzCollege



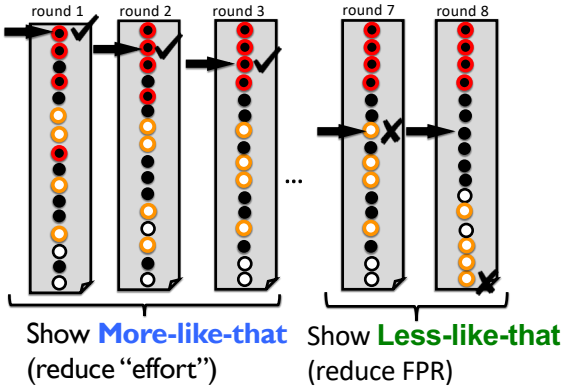
Applications

As the human investigates, they produce labels
How can we learn **on-the-job**?
i.e. Re-rank anomalies from Top-1 Feedback

Goals (two-fold)

True Positive
show more-like-that
(reduce FPR)

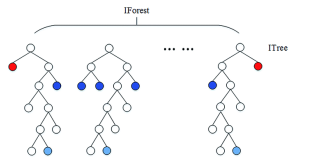
False Positive
show less-like-that
(reduce effort)



Problem Sketch

Given representation over diff. components

$\mathbf{S}_u$



$\mathbf{s} = S.1$

$\mathbf{s} = S.\mathbf{w}$

Learning-To-Rank

Training Examples $\langle (u, v), p_{uv} \rangle$

Optimization Problem

$$\min_{\mathbf{w}} f = \sum_{(u,v)\in T} -p_{uv} \log \hat{p}_{uv} - (1 - p_{uv}) \log(1 - \hat{p}_{uv})$$

Estimated probabilities based on current estimates

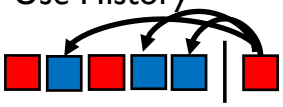
$$\hat{p}_{uv} = \frac{e^{(s_u - s_v)}}{1 + e^{(s_u - s_v)}}$$

Update Equation

$$\Delta \mathbf{w} = \eta \sum_{(u,v)\in P} (\hat{p}_{uv} - p_{uv})(\boldsymbol{S}_u - \boldsymbol{S}_v)$$

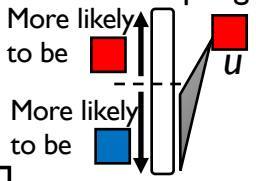
Generating Pairs

Use History

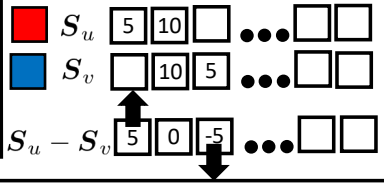


$p_{uv} = p_{ab}$

Use Sampling



$p_{uv} = (1 \pm \delta)\hat{p}_{uv}$



Method Highlights

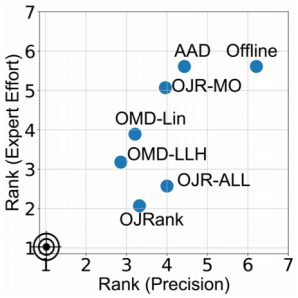
OJRank

- Works alongside expert
- Maximizes *precision*
- Minimizes *effort*
- Instantaneous updates

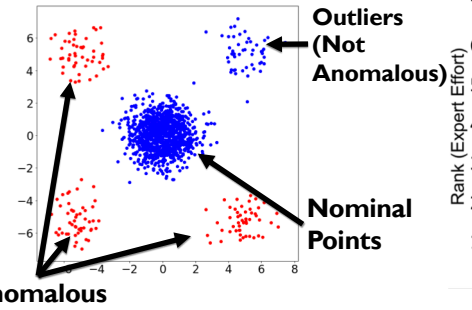
#correct instances shown

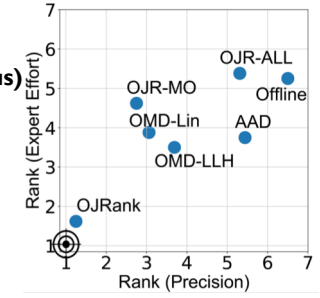
$$\sum_{r=1}^{b-1} 1 - sim(\mathbf{s}_{\pi(r)}, \mathbf{s}_{\pi(r+1)})$$

Benchmark Datasets



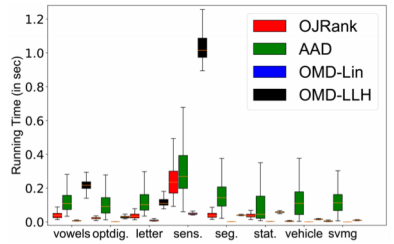
Clustered Datasets





Metric	Baselines / Datasets	AAD [1]	OMD Lin[2]	OMD LLH[2]	OJR MO	OJR ALL
prec. @ b	Benchmark	0.015	0.5	0.5	0.005	0.008
	Clustered	0.003	0.007	0.027	0.003	0.003
expert effort	Benchmark	0.001	0.010	0.024	$1e-4$	0.014
	Clustered	0.027	0.007	0.012	0.004	0.004

OJRank performs well on both the metrics.



Updates are within 1/10th of a second

Carnegie Mellon University

Code & data available at: ojrank.github.io