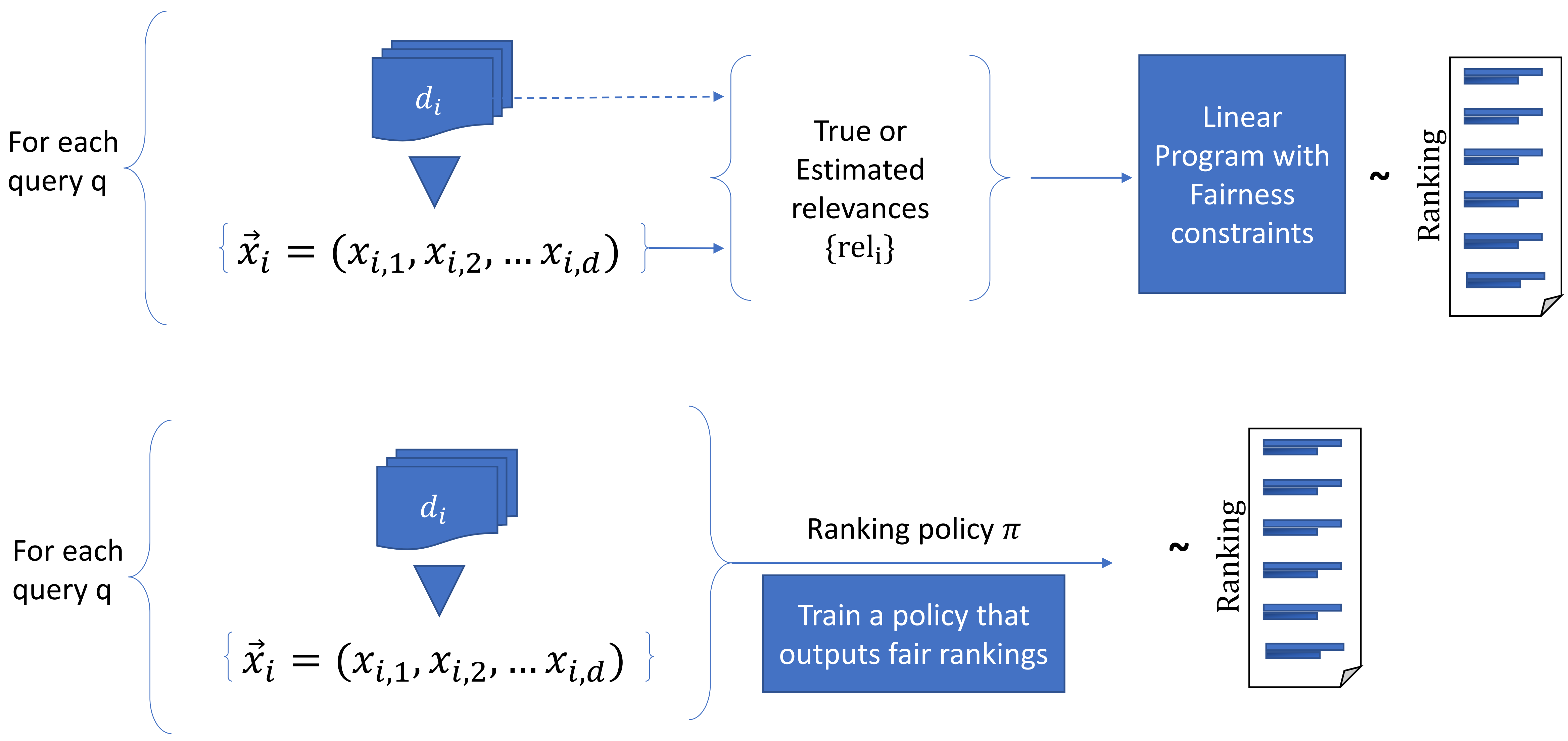
Policy Learning for Fairness in Ranking

Ashudeep Singh, Thorsten Joachims
(Cornell University)

NeurIPS 2019



Learning to Rank with Fairness constraints

Fair-PG-Rank: Learning-to-Rank with Fairness constraints

$$\pi^*_\delta = \operatorname{argmax}_\pi \mathbb{E}_{q\sim\mathcal{Q}}\left[U(\pi|q)\right] \;\; \text{s.t.} \;\; \mathbb{E}_{q\sim\mathcal{Q}}\left[\mathcal{D}(\pi|q)\right] \leq \delta$$

Empirical Risk Minimization:

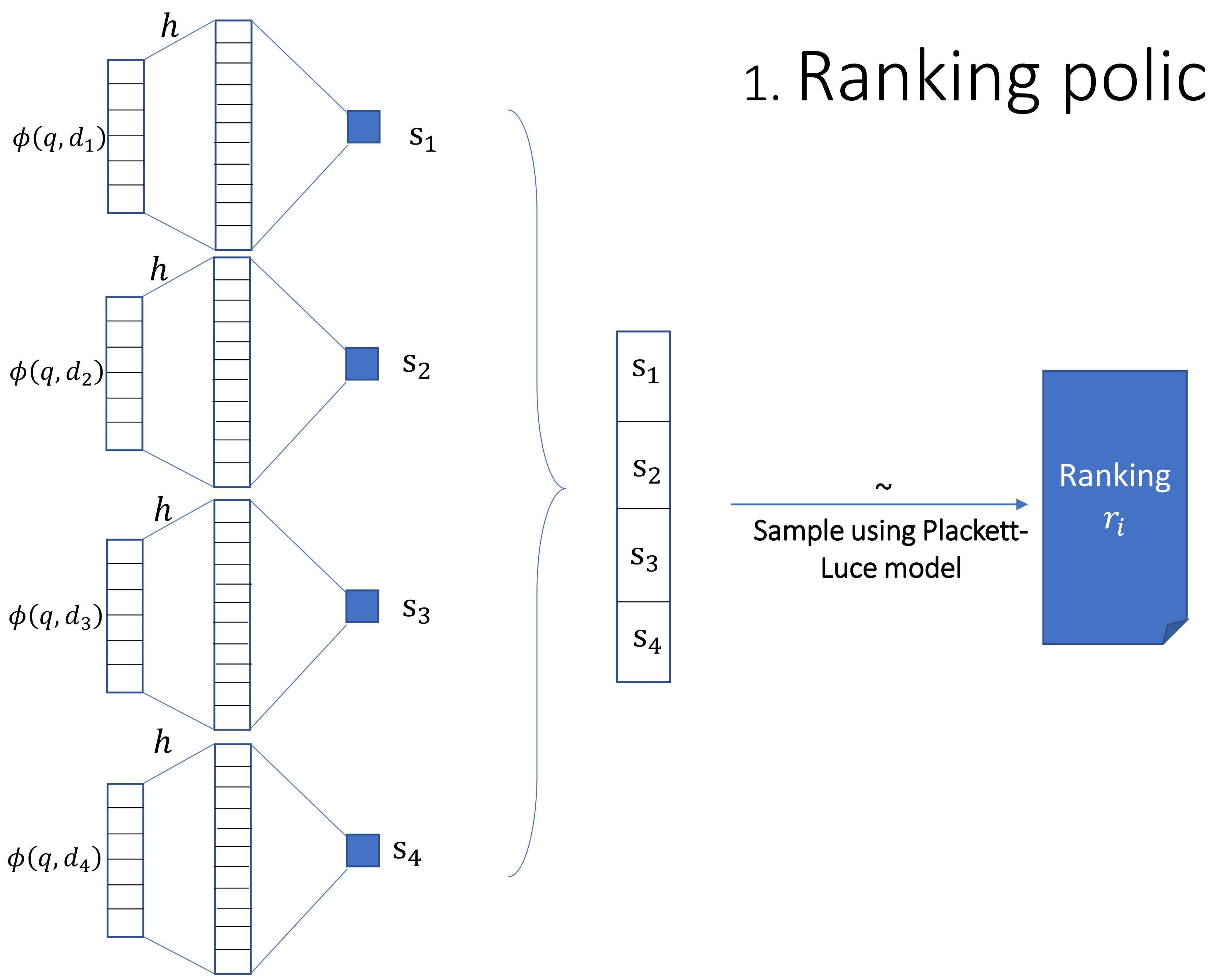
$$\hat{\pi}^*_\lambda = \operatorname{argmax}_\pi \frac{1}{N}\sum_{q=1}^{N} U(\pi|q) - \lambda\frac{1}{N}\sum_{q=1}^{N}\mathcal{D}(\pi|q)$$

Utility

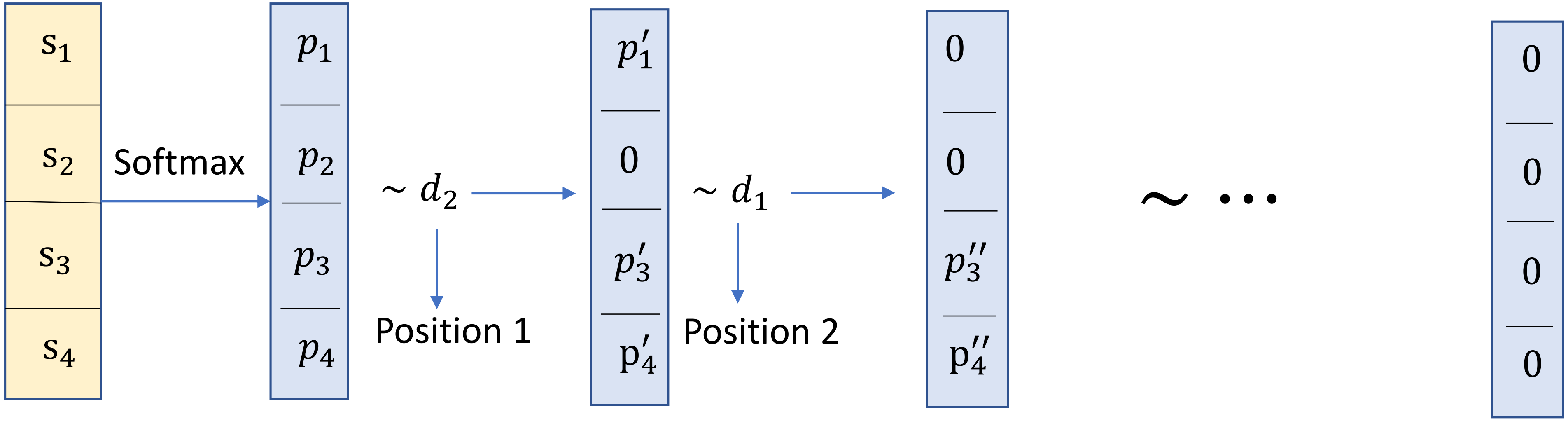
Disparity

1. Ranking policy π
2. Utility of π
3. Disparity measure w.r.t. π

1. Ranking policy π



Sampling from π: Plackett-Luce Model



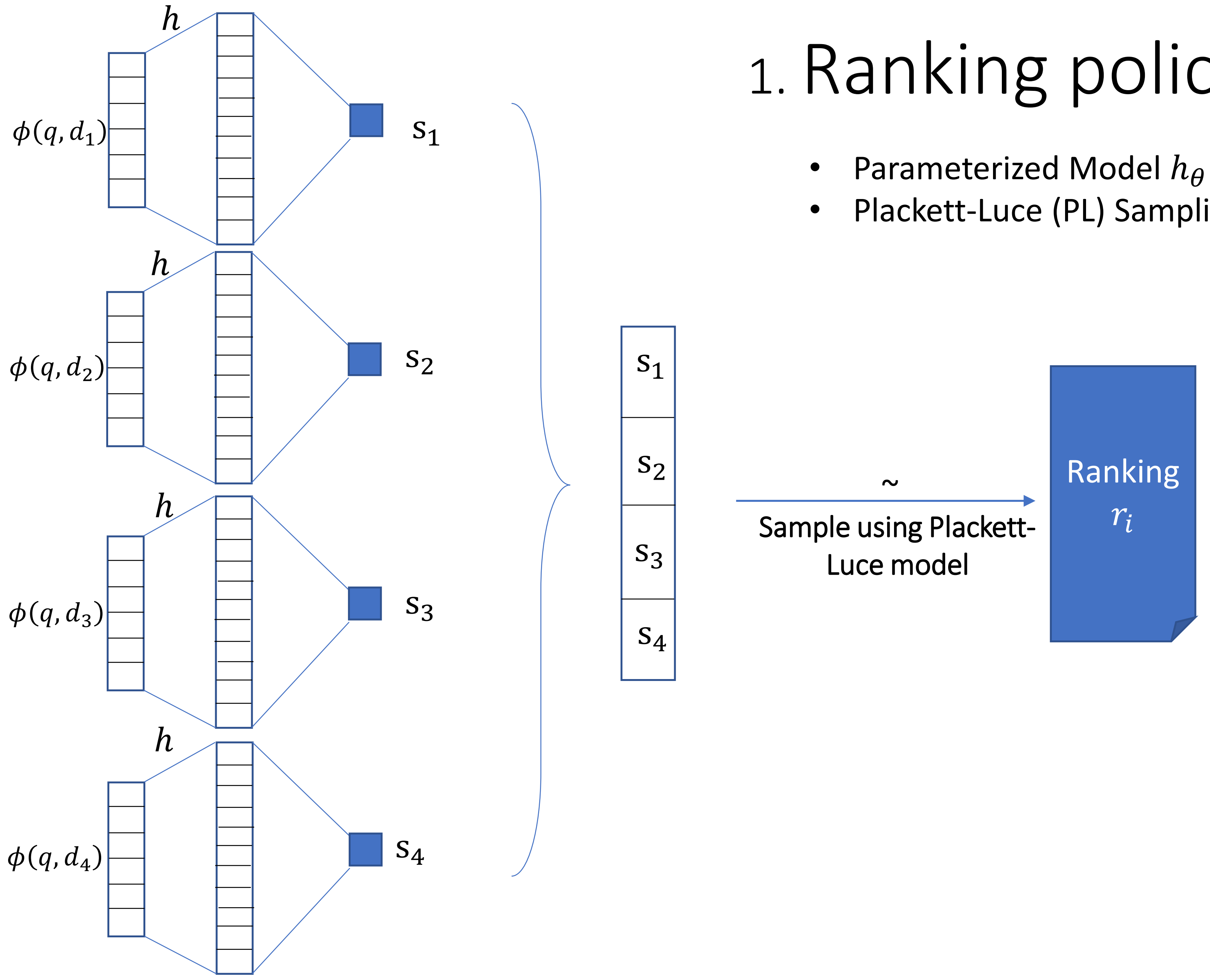
$$p_i = \frac{e^{s_i}}{\sum_j e^{s_j}}$$

$$p'_i = \frac{e^{s_i}}{\sum_{j \neq 2} e^{s_j}}, \mathrm{i} \neq 2$$

$$\boxed{P(ranking = \langle d_2, d_1, d_3, d_4 \rangle) = p_2 \times p'_1 \times p''_3 \times p''_4}$$

1. Ranking policy π

- Parameterized Model h_θ
- Plackett-Luce (PL) Sampling



2. User Utility

$$U(\pi|q) = \mathbb{E}_{r\sim\pi(r|q)}[\Delta(r, \mathrm{rel}_q)]$$

For example:

- $\Delta_{\mathrm{DCG}}(r, rel_q) = \sum_{j=1}^{n_q} \frac{u(r(j)|q)}{\log(1+j)}$
- $\Delta_{N\mathrm{DCG}}(r, rel_q) = \frac{\Delta_{\mathrm{DCG}}(r, rel_q)}{\Delta_{\mathrm{DCG}}(r^*, rel_q)}$

3. Disparity Measures

- $\text{Exposure}(d_i) = v_\pi(d_i) = \mathbb{E}_{r\sim\pi}[\mathbf{v}_{r(d_i)}]$
- $\text{Exposure}(G) = v_\pi(G) = \sum_{d\in G} \text{Exposure}(d_i)$
- Merit of a group $M(G) = \sum_{d\in G} M(d_i)$
- Group fairness disparity:

$$\mathcal{D}_{\text{group}}(\pi|q) = \max\left(0, \frac{v_\pi(G_i)}{M_{G_i}} - \frac{v_\pi(G_j)}{M_{G_j}}\right)$$, where $M_{G_i} \geq M_{G_j}$

- Individual fairness disparity:

$$\mathcal{D}_{\text{individual}}(\pi|q) = \sum_{\substack{i,j \\ s.t.\, M_i \geq M_j}} \max\left(0, \frac{v_\pi(d_i)}{M_i} - \frac{v_\pi(d_j)}{M_j}\right)$$

Learning objective: $\hat{\pi}^*_\lambda = \text{argmax}_\pi \frac{1}{N}\sum_{q=1}^{N} U(\pi|q) - \lambda \frac{1}{N}\sum_{q=1}^{N} \mathcal{D}(\pi|q)$

Utility: $U(\pi|q) = \mathbb{E}_{r\sim\pi(r|q)}[\Delta(r, rel^q)] = \mathbb{E}_{r\sim\pi(r|q)}\left[\sum_{j=1}^{n_q} \frac{u(r(j)|q)}{\log(1+j)}\right]$

Most previous methods optimize a smooth convex proxy of the utility.
(SVMRank, ListNet, RankNet, Softrank etc.)

How do we perform a gradient descent update on utility as well as disparity?

PG-Rank: REINFORCE update for π_θ

$$\nabla_\theta U(\pi_\theta|q) = \nabla_\theta \mathbb{E}_{r\sim\pi_\theta(r|q)} \Delta\big(r, \mathrm{rel}^q\big)$$

Similarly for Disparity $\mathcal{D}_\pi$

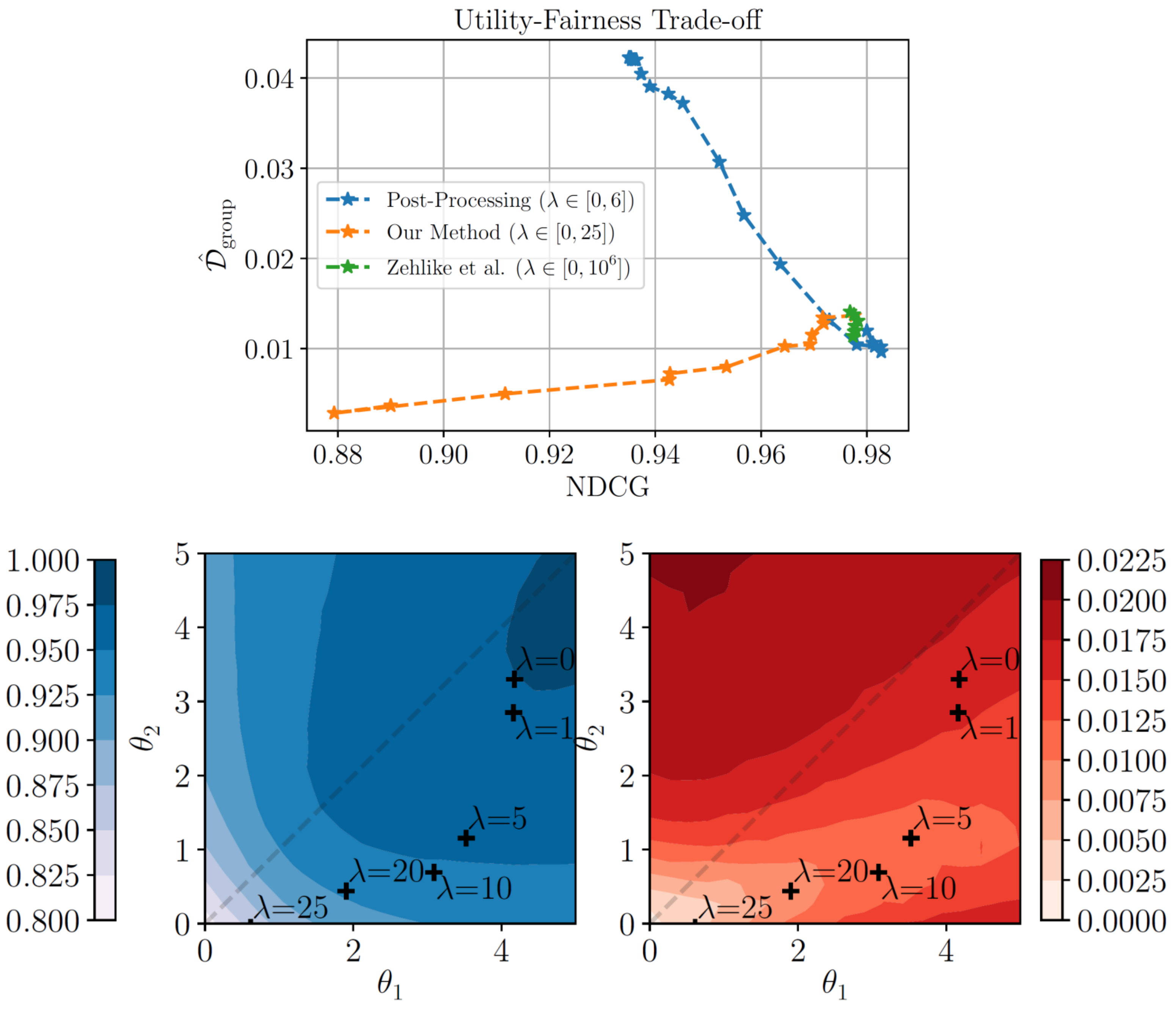
Fair-PG-Rank: Summary

- **Input**: Queries w/ Candidate sets: features (+ relevances available during training)
- **Ranking policy**: A neural network scoring function + PL-sampling
- **Loss function**: Utility - λ . Disparity
- **Train**: Stochastic gradient descent

Experiment 1

- Each document d_i represented by $(x_{i,1}, x_{i,2}) \in [0,3]^2$
- Two groups: G_0, G_1 (Ratio 4:1)
- $\text{rel}_i = x_{i,1} + x_{i,2}$ (True Model)
- $x_{i,2}$ corrupted for G_1 documents
- $h(x_i) = \theta_1 x_{i,1} + \theta_2 x_{i,2}$
- Train the model (θ_1, θ_2) with different values of λ

A fair model only uses the x_1 feature



(a) NDCG (b) Disparity

Experiment 2: Yahoo! Learning-to-Rank Challenge