UFMG UNIVERSIDADE FEDERAL DE MINAS GERAIS

Information Retrieval

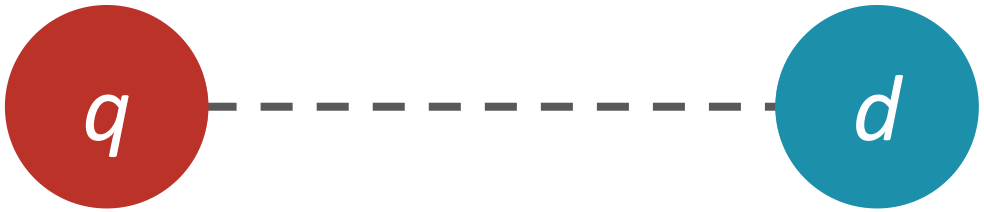
Online Learning to Rank

Rodrygo L. T. Santos
rodrygo@dcc.ufmg.br

The ranking problem

$$f(q,d)$$

Learning to rank



$$f(\mathbf{x})$$

Learning to rank

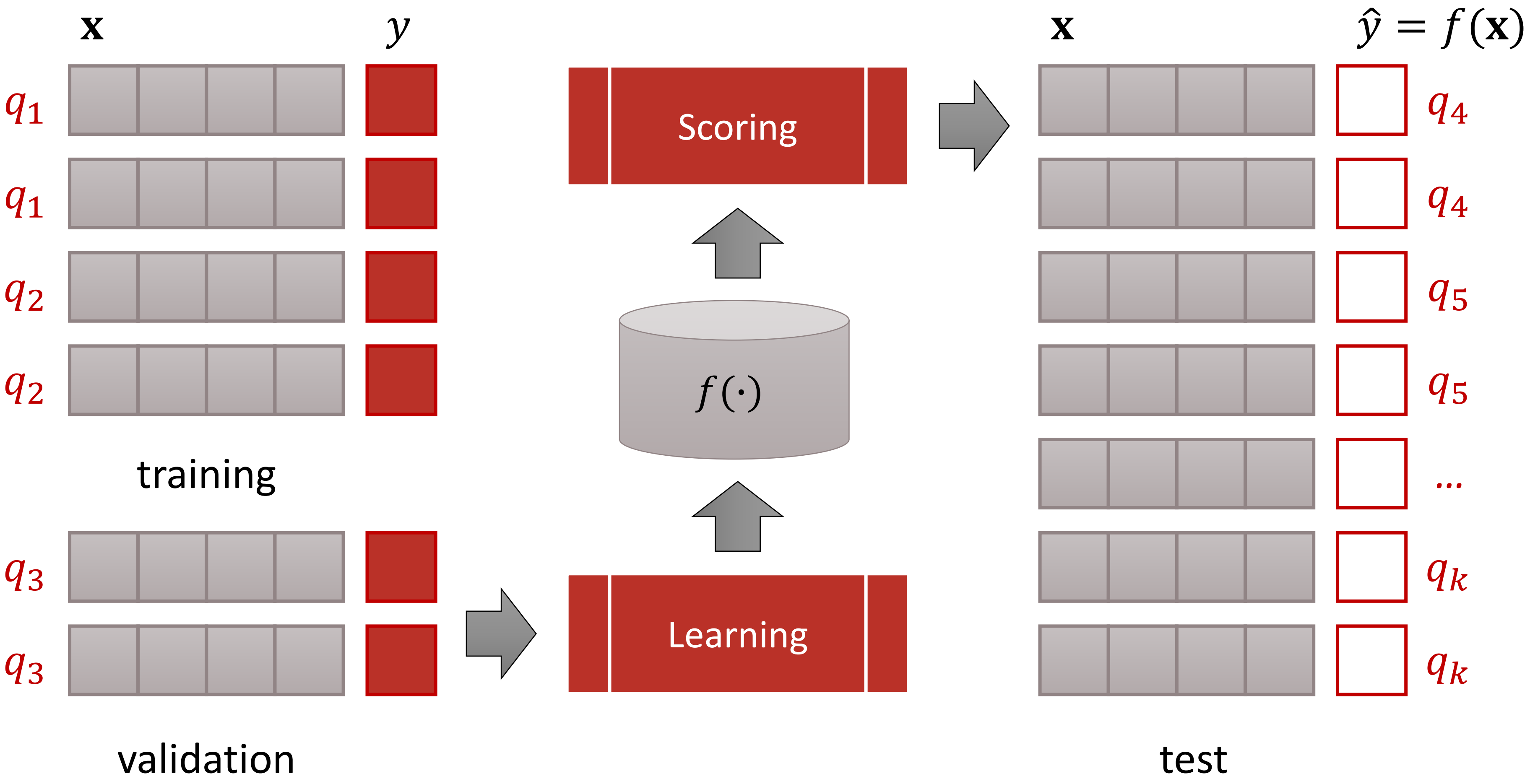
Feature-based representation

- Individual models as ranking ”features”

Discriminative learning

- Effective models learned from data
- Aka machine-learned ranking

Discriminative learning framework



Drawbacks

Scalability

- Relevance labels are costly
- More so if expert labels are needed

Realism

- Hired judges aren’t real users
- Real users’ preferences are contextualized

Credit: NIST

Contextualized preferences

Context has a significant influence on search behavior

- Addressing all possible settings individually, through supervised learning, is not feasible

Can’t sacrifice users while learning ranking models

- Need to look for scalable methods that can learn effective rankings without expensive tuning

Online learning to rank

Learn directly from natural interactions with users

- Typically implicit feedback (e.g., clicks)

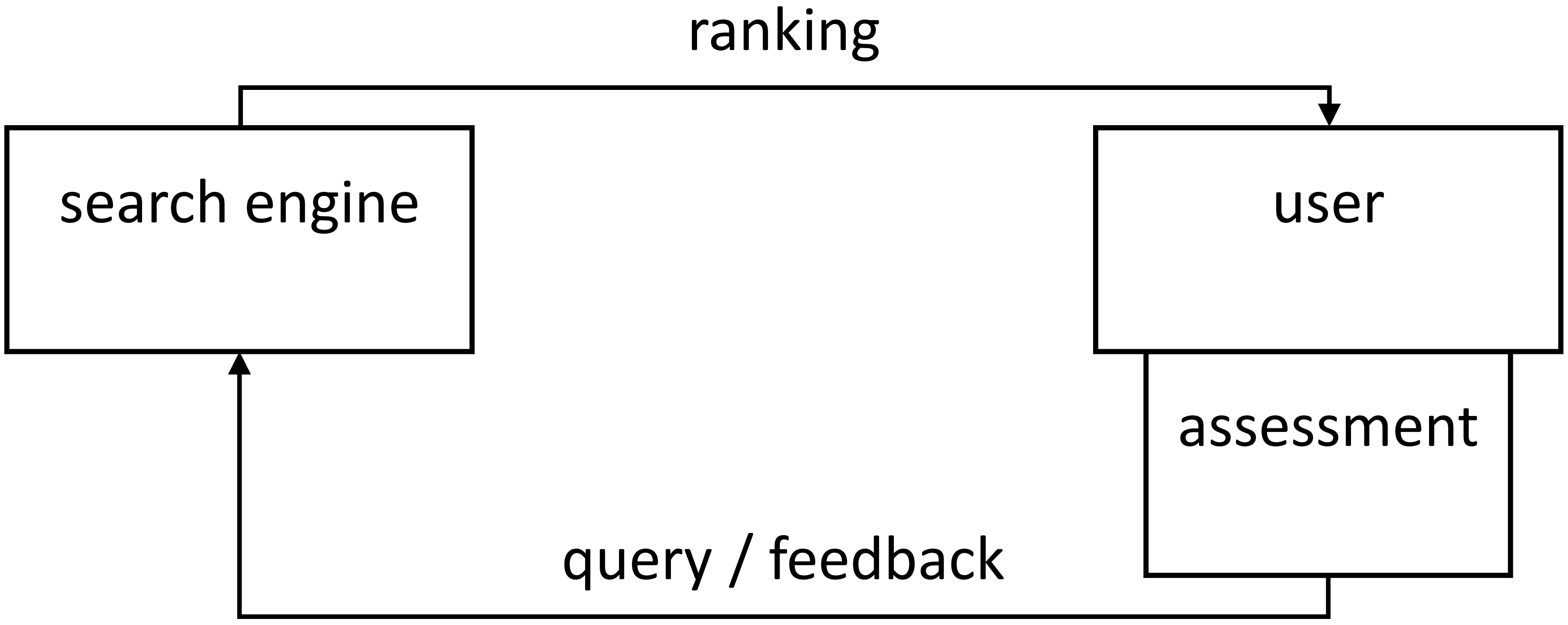
Update learned models incrementally

- Sequential as opposed to batch learning

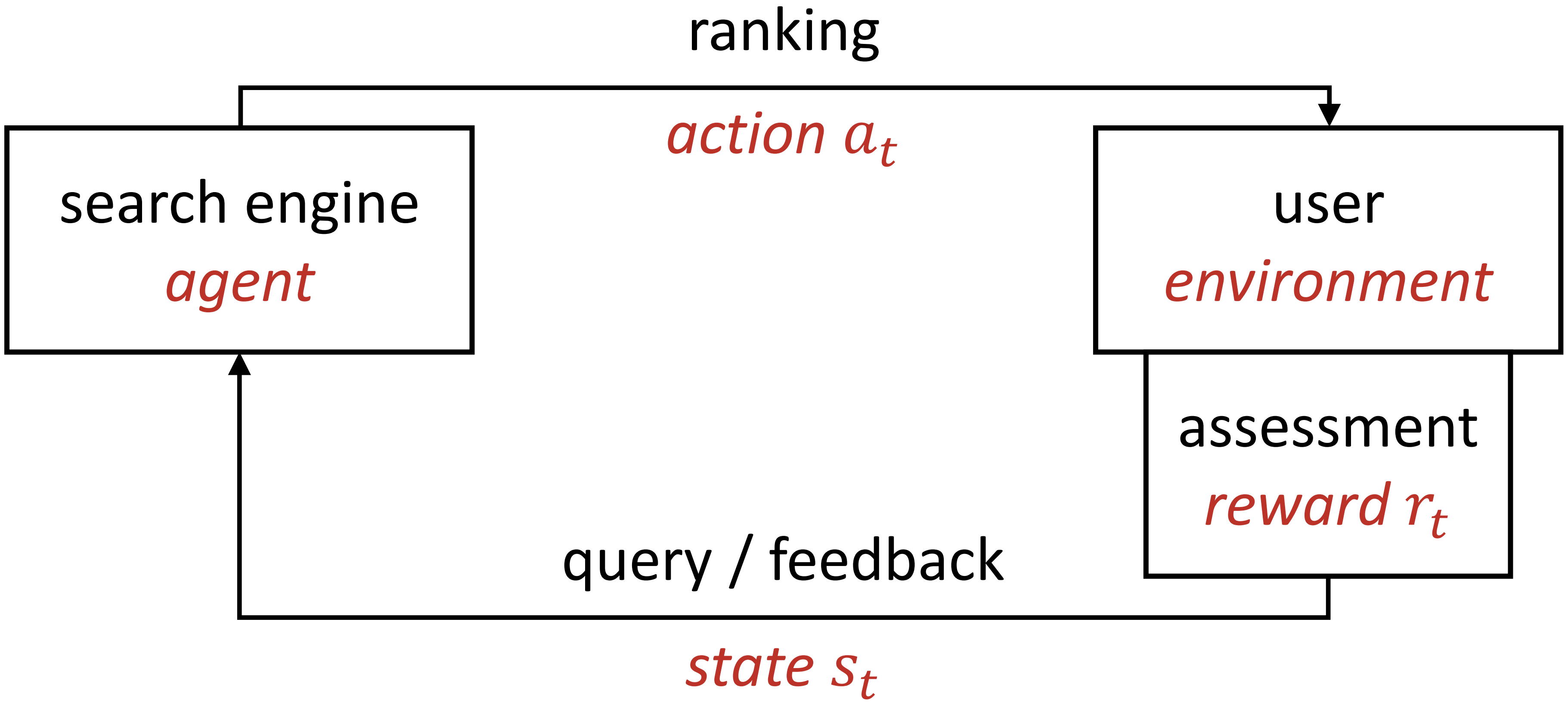
Continuously adapt as interactions progress

- From ”random” to fine-tuned rankings

Online learning to rank



Reinforcement learning



Reinforcement learning

At each discrete time step t

- Agent observes state s_t
- Agent selects action a_t
- Environment provides reward r_t
- Environment moves to a new state s_{t+1}

Goal is to maximize cumulated reward as $t \to \infty$

Credit: Google DeepMind

Exploration-exploitation trade-off

Exploration

- Reward is only provided for the selected action
- We want to uncover other rewarding actions

Exploitation

- Pure exploration risks selecting unrewarding actions
- We want to maximize total reward in the long run

Reinforcement learning for ranking

Online learning to rank as a bandit problem

- State (s_t) does not depend on past actions

Multi-armed bandits (MAB)

- State is not provided

Credit: Wikipedia

Reinforcement learning for ranking

Online learning to rank as a bandit problem

- State (s_t) does not depend on past actions

Multi-armed bandits (MAB)

- State is not provided

Contextual bandits

- State (context vector) is provided

Multi-armed bandits

No state (e.g., query or user profile) is provided

- Fits well with non-personalized recommendation

Example: homepage news recommendation

- New user arrives, recommender returns a news story (a_t) to show, observes whether user clicks (r_t)
- Goal: maximize clicks in a given period

Example: ϵ-greedy

Simple strategy

- Explore with probability $\epsilon \in [0,1]$
- Exploit with probability $1 - \epsilon$

Often works well in practice

- With hyperparameter ϵ suitably tuned

Example: UCB

True reward distribution of an action is unknown

- We can only observe samples r of it
- Each observed sample increases our confidence

Idea: explore actions with large confidence bounds

- Try the action a_x that maximizes $\bar{r}_x + \sqrt{(2\ln n)/n_x}$ for an average reward $\bar{r}_x$ after n_x observations

Contextual bandits

State (context) does not depend on past actions

- Fits well with independent user searches

Example: (stateless) web search

- User submits a query (s_t), search engine displays a ranking (a_t), observes if/where the user clicks (r_t)
- Goal: maximize ranking quality in a given period

Contextual bandits

k-armed contextual bandits

- Find the best among k ranking models
  (e.g., $k - 1$ candidate models vs. current best model)

Continuous-armed contextual bandits

- Find the best among infinite ranking models
  (i.e., feature-based ranking models)

How to infer ranking quality implicitly?

Problem

Rewards aren’t directly measurable

Absolute metrics

Document-level

- Click rate, click models

Ranking-level

- Reciprocal rank, CTR@k, time-to-click, abandonment

Session-level

- Queries per session, session length, time to first click

Relative metrics

Absolute document-level metrics are biased

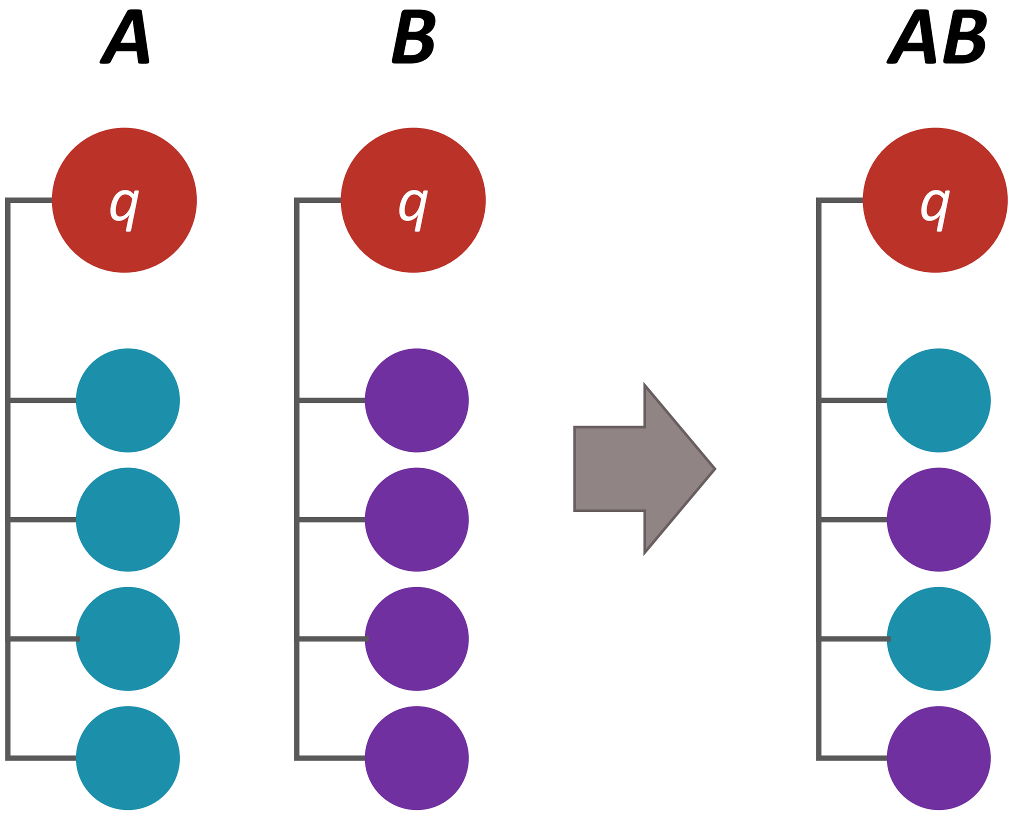
- Position bias: top ranked document favored
- Presentation bias: highlighted documents favored

Relative document-level metrics are less affected

- Click-skip, fair pairs

Better: relative ranking-level metrics!

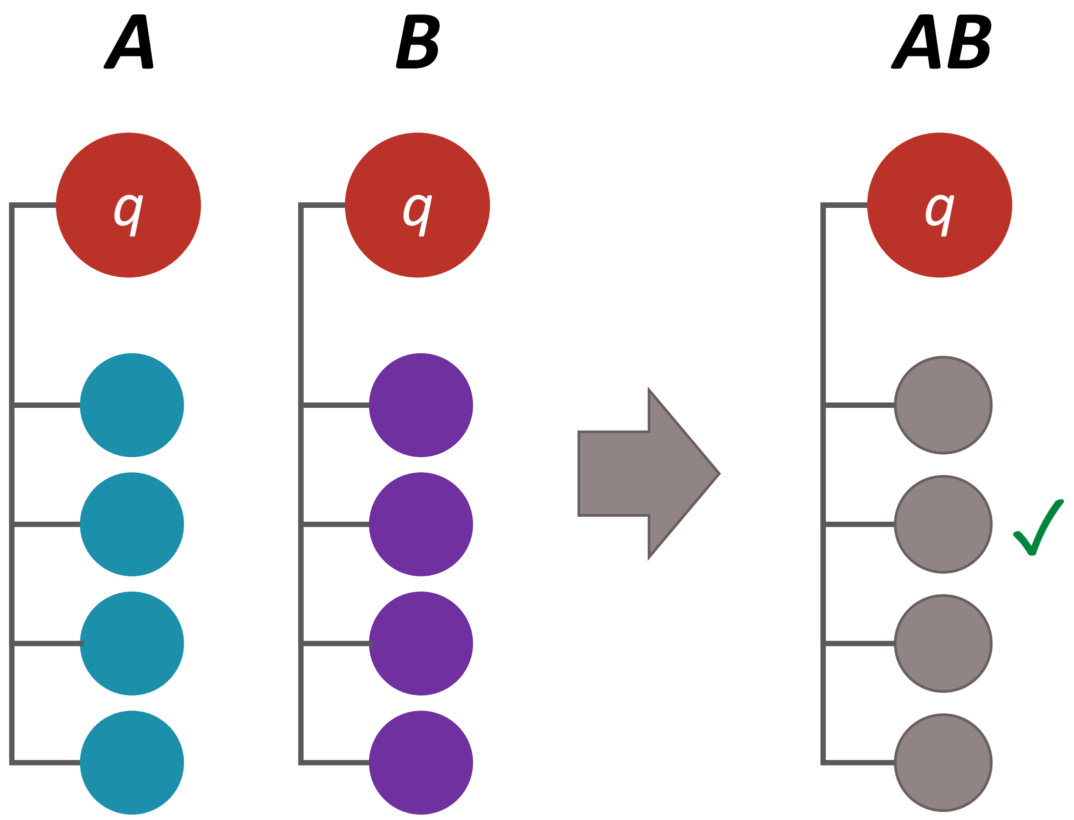
Interleaved comparisons
[Joachims, KDD 2002]



Blend results from both conditions into a single ranking

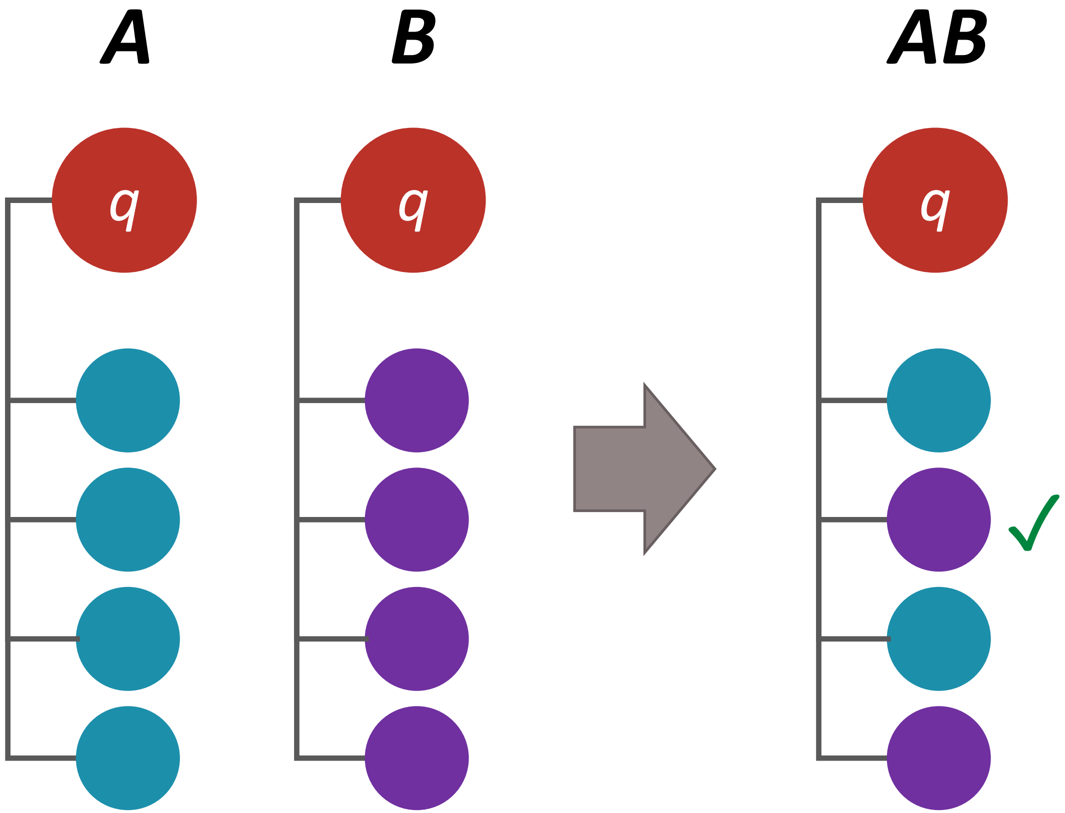
Interleaved comparisons

[Joachims, KDD 2002]



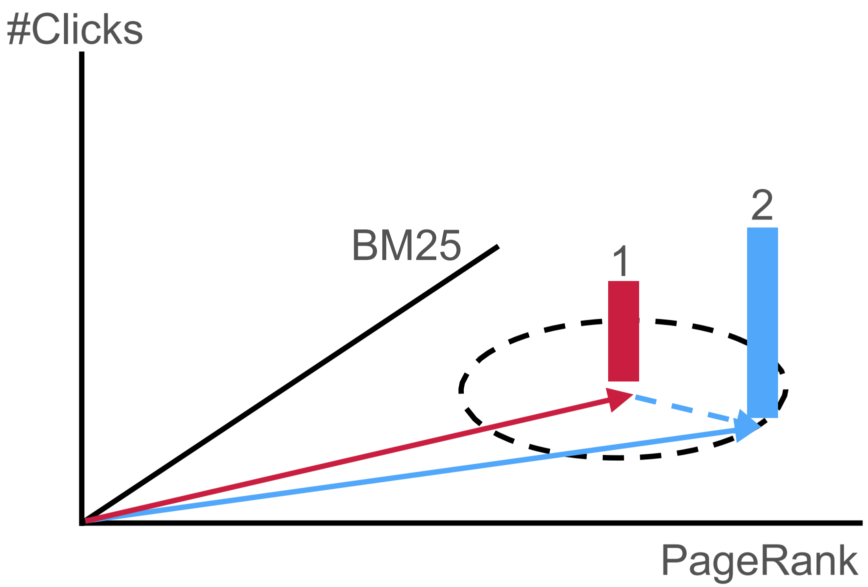
Hide provenance from the user and collect feedback

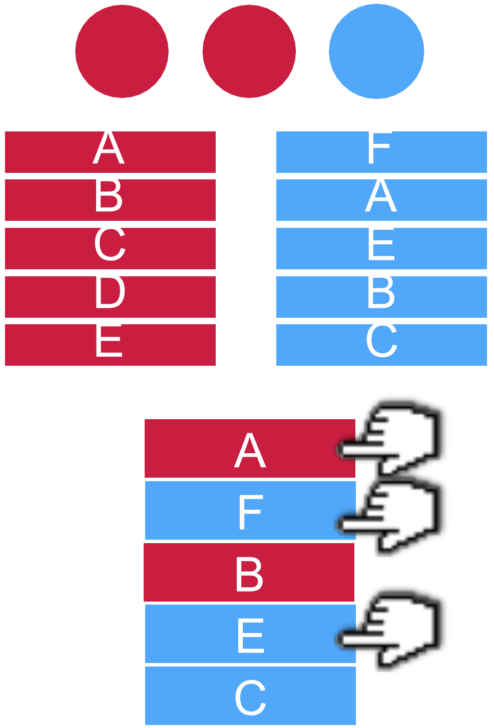
Interleaved comparisons
[Joachims, KDD 2002]



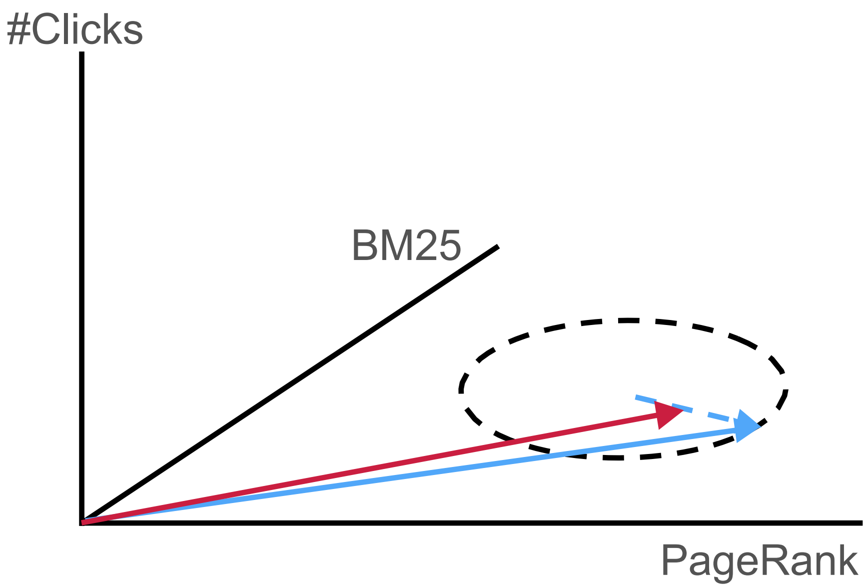
Assign credit:
e.g., B wins over A

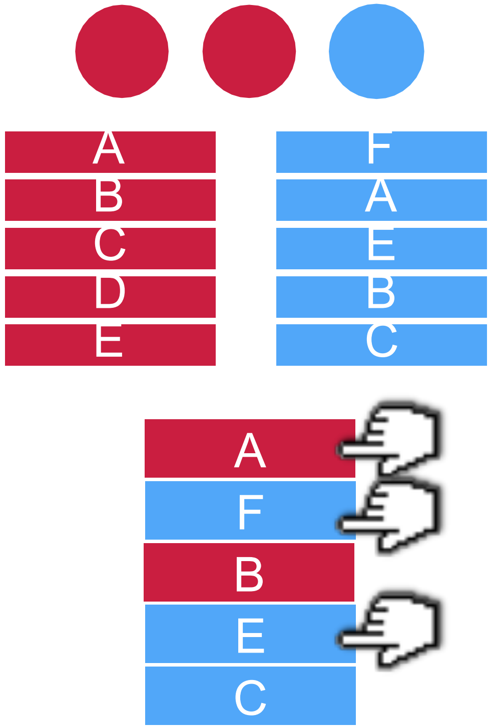
Dueling bandit gradient descent (DBGD)
[Yue et al., ICML 2009]





Dueling bandit gradient descent (DBGD)
[Yue et al., ICML 2009]





Dueling bandit gradient descent (DBGD)

[Yue et al., ICML 2009]

Algorithm 2 Dueling Bandit Gradient Descent (DBGD).

Require: $\alpha, \delta, \mathbf{w}_0^0$

for $t \leftarrow 1..\infty$ **do**

- $q_t \leftarrow receive\_query(t)$ — *// obtain a query from a user*
- $\mathbf{l}_0 \leftarrow generate\_list(\mathbf{w}_t^0, q_t)$ — *// ranking of current best*
- $\mathbf{u}_t^1 \leftarrow sample\_unit\_vector()$
- $\mathbf{w}_t^1 \leftarrow \mathbf{w}_t^0 + \delta \mathbf{u}_t^1$ — *// create a candidate ranker*
- $\mathbf{l}_1 \leftarrow generate\_list(\mathbf{w}_\mathbf{t}^\mathbf{1}, q_t)$ — *// exploratory ranking*
- $\mathbf{m}_t, \mathbf{t}_t \leftarrow TDI\_interleave(\mathbf{l})$ — *// interleaving and teams*
- $\mathbf{c}_t \leftarrow receive\_clicks(\mathbf{m}_t)$ — *// show interleaving to the user*
- $\mathbf{b}_t \leftarrow TDI\_infer(\mathbf{t}_t, \mathbf{c}_t)$ — *// set of winning candidates*
- **if** $\mathbf{w}_t^0 \in \mathbf{b}_t$ **then**
  - $\mathbf{w}_{t+1}^0 \leftarrow \mathbf{w}_t^0$ — *// if current best wins or ties, no update*
- **else**
  - $\mathbf{w}_{t+1}^0 \leftarrow \mathbf{w}_t^0 + \alpha \mathbf{u}_t^1$ — *// update α step towards candidate*

Dueling bandit gradient descent (DBGD)
[Yue et al., ICML 2009]

DBGD explores one direction at a time

- Exploring multiple directions could improve efficiency

Interleaving requires pairwise comparisons

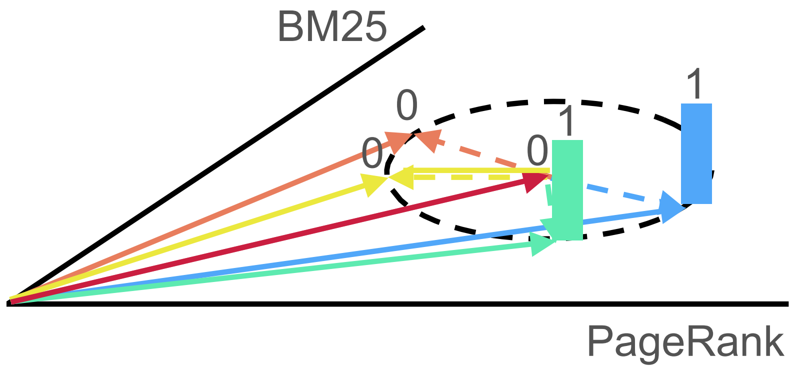
- Quadratic on the number of candidate models

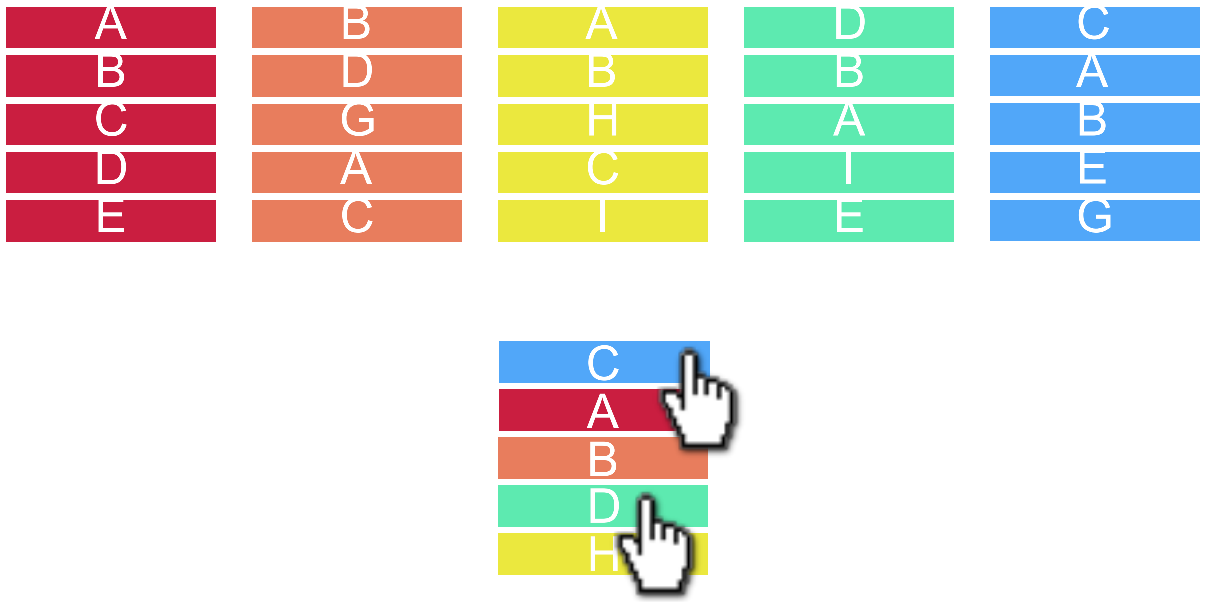
Multileaving to the rescue

- Credit clicks independently, declare winner(s)

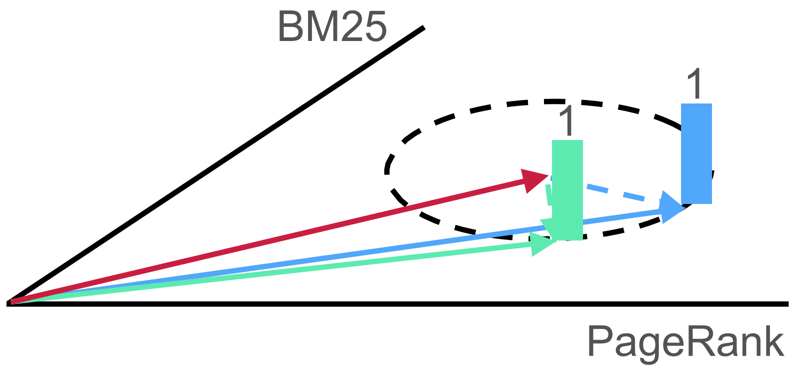
Multileave gradient descent (MGD)

[Schuth et al., WSDM 2016]





Multileave gradient descent (MGD)
[Schuth et al., WSDM 2016]



Winner takes all (MGD-W)

Randomly choose one of the winning models

Mean winner (MGD-M)

Use the average of the winning models

Multileave gradient descent (MGD)

[Schuth et al., WSDM 2016]

Algorithm 10 Multileave Gradient Descent (MGD).

Require: $n, \alpha, \delta, \mathbf{w}_0^0, update(\mathbf{w}, \alpha, \{\mathbf{b}\}, \{\mathbf{u}\})$

for $t \leftarrow 1..\infty$ **do**
  $q_t \leftarrow receive\_query(t)$ — *// obtain a query from a user*
  $\mathbf{l}_0 \leftarrow generate\_list(\mathbf{w}_t^0, q_t)$ — *// ranking of current best*
  for $i \leftarrow 1...n$ **do**
    $\mathbf{u}_t^i \leftarrow sample\_unit\_vector()$
    $\mathbf{w}_t^i \leftarrow \mathbf{w}_t^0 + \delta \mathbf{u}_t^i$ — *// create a candidate ranker*
    $\mathbf{l}_t^i \leftarrow generate\_list(\mathbf{w}_t^i, q_t)$ — *// exploratory ranking*
  $\mathbf{m}_t, \mathbf{t}_t \leftarrow TDM\_multileave(\mathbf{l}_t)$ — *// multileaving and teams*
  $\mathbf{c}_t \leftarrow receive\_clicks(\mathbf{m}_t)$ — *// show multileaving to the user*
  $\mathbf{b}_t \leftarrow TDM\_infer(\mathbf{t}_t, \mathbf{c}_t)$ — *// set of winning candidates*
  if $\mathbf{w}_t^0 \in \mathbf{b}_t$ **then**
    $\mathbf{w}_{t+1}^0 \leftarrow \mathbf{w}_t^0$ — *// if current best among winners, no update*
  else
    $\mathbf{w}_{t+1}^0 \leftarrow update(\mathbf{w}_t^0, \alpha, \mathbf{b}_t, \mathbf{u}_t)$ — *// Algorithm 11 or 12*

Summary

Online learning helps improve implicit metrics

- Explicit metrics still important – don’t fire assessors!

Research on MAB mostly focused on absolute feedback

- Relative feedback methods improve sensitivity

Inter- and multileaving further improves

- Orders of magnitude faster convergence

Open directions

Choosing among infinitely many models

- Probabilistic multileave gradient descent

Online learning of non-linear models

- e.g., regression trees, neural networks

Online learning from offline data

- Counterfactual learning to rank

References

Fast and reliable online learning to rank for information retrieval

Hofmann, PhD thesis 2013

Search engines that learn from their users

Schuth, PhD thesis 2016

References

Online learning to rank for information retrieval
Grotov and de Rijke, SIGIR 2016

Interactively optimizing information retrieval systems as a dueling bandits problem
Yue and Joachims, ICML 2009

UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Coming next...

Seminars

Rodrygo L. T. Santos
rodrygo@dcc.ufmg.br