Comparison-based Choices

Johan Ugander
Management Science & Engineering
Stanford University

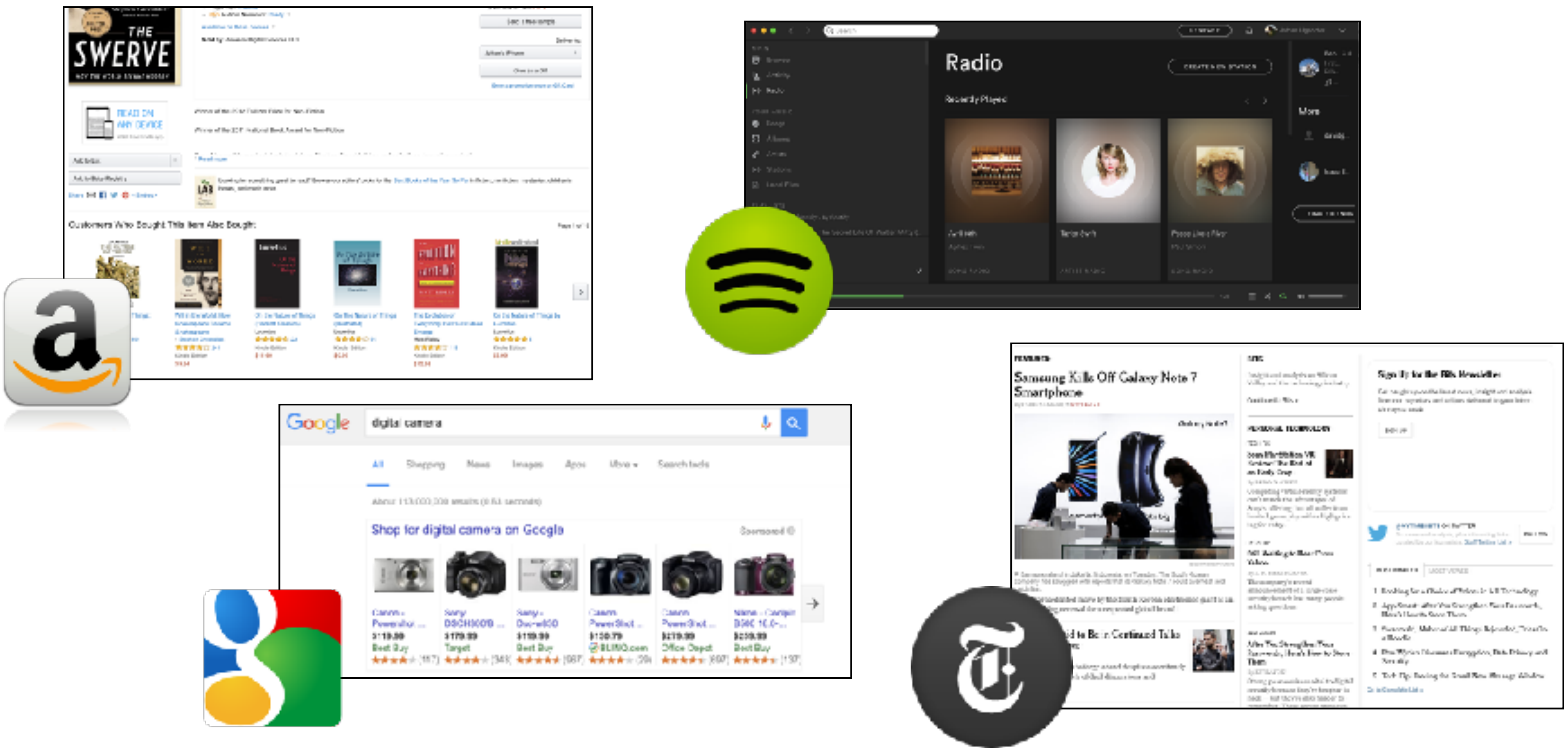
Joint work with:
Jon Kleinberg (Cornell)
Sendhil Mullainathan (Harvard)

EC'17 Boston
June 28, 2017

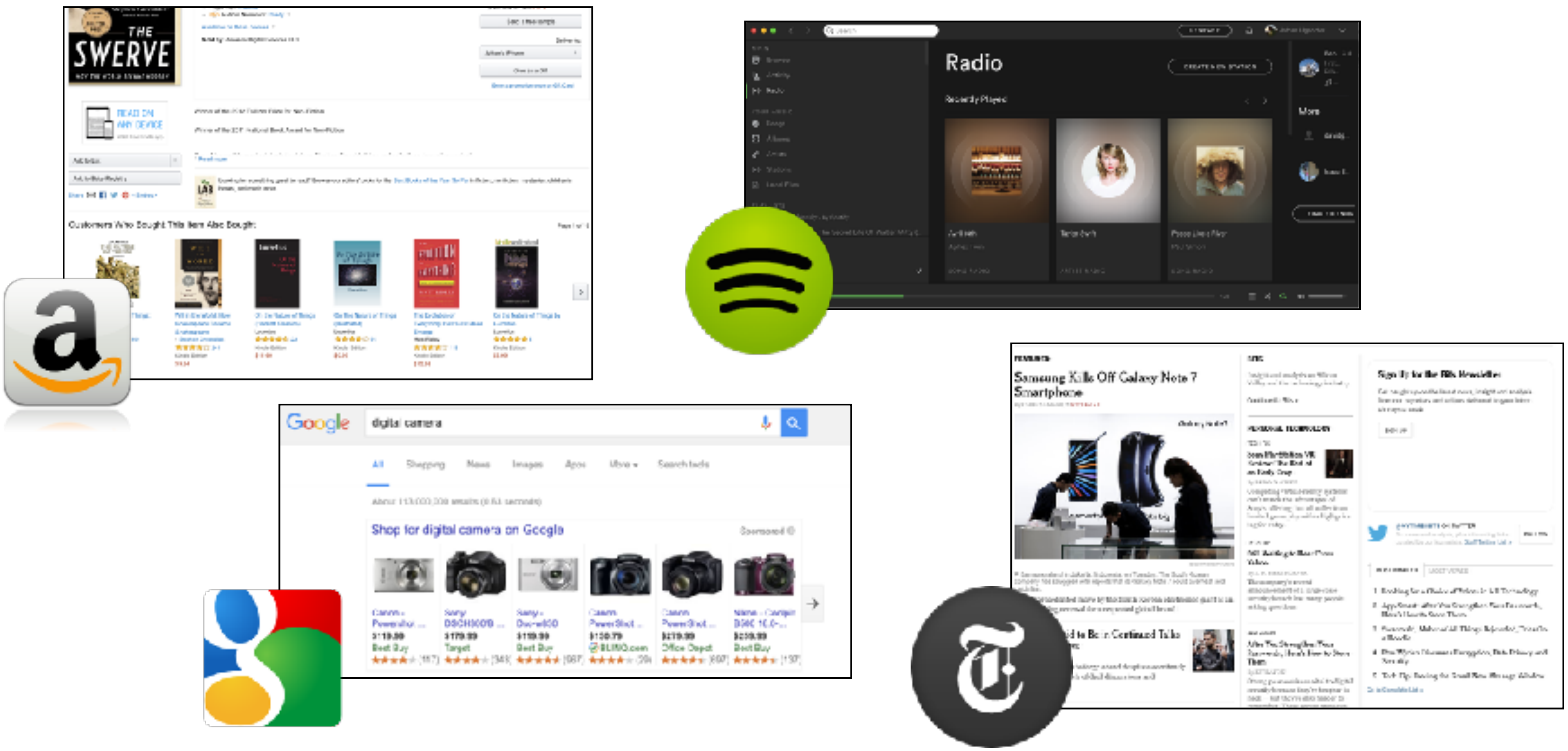
Predicting discrete choices

- Classic problem: consumer preferences [Thurstone ’27, Luce ’59], commuting [McFadden ’78], school choice [Kohn-Manski-Mundel ’76]

Predicting online discrete choices



Predicting online discrete choices



How well can we learn/predict "choice set effects"?

a.k.a. violations of the "independence of irrelevant alternatives" (IIA)

- [Sheffet-Mishra-Ieong ICML 2012, Yin et al. WSDM 2014]

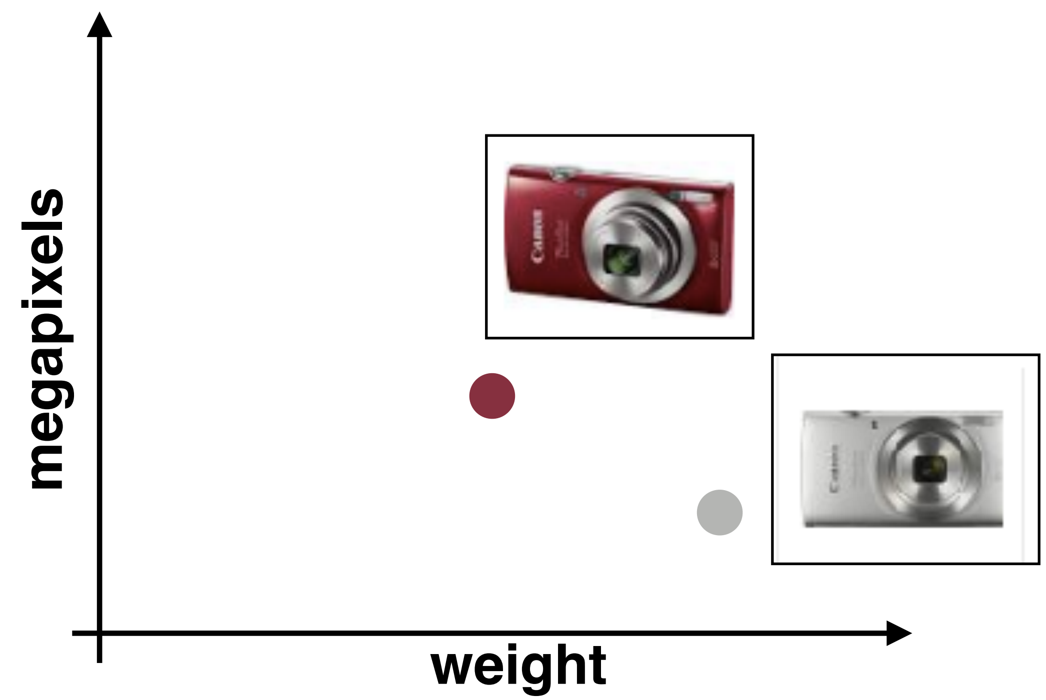
Choice set effects

- Bias towards moderation, compromise effect

- [Simonson 1989, Simonson-Tversky 1992, Kamenica 2008, Trueblood 2013]

Choice set effects

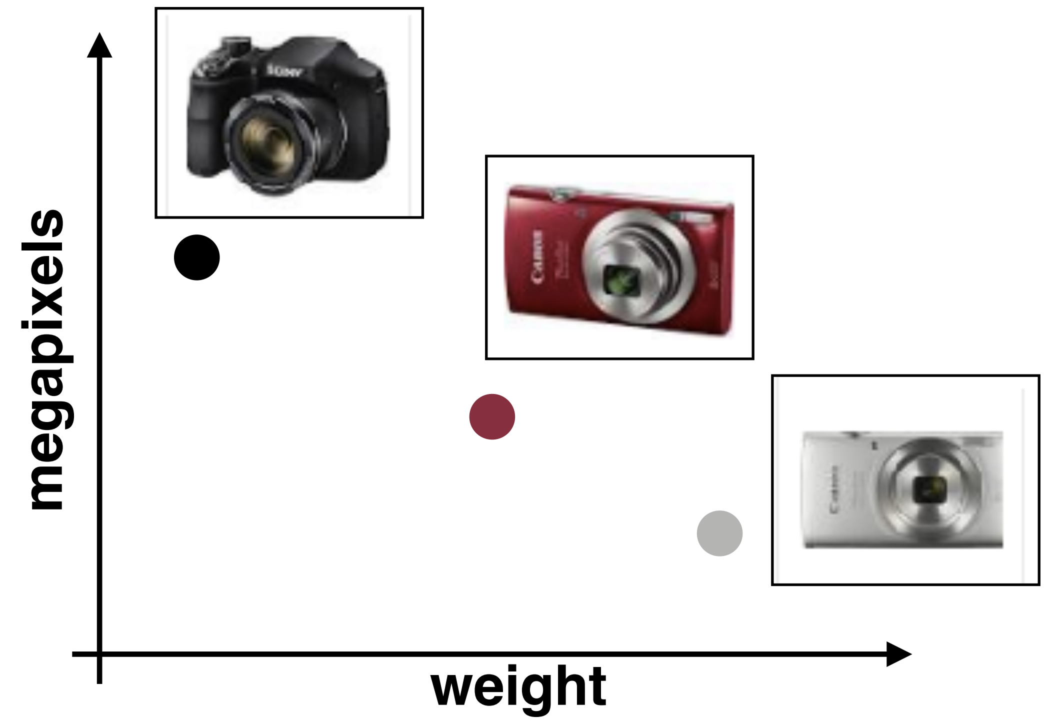
- Bias towards moderation, compromise effect



- [Simonson 1989, Simonson-Tversky 1992, Kamenica 2008, Trueblood 2013]

Choice set effects

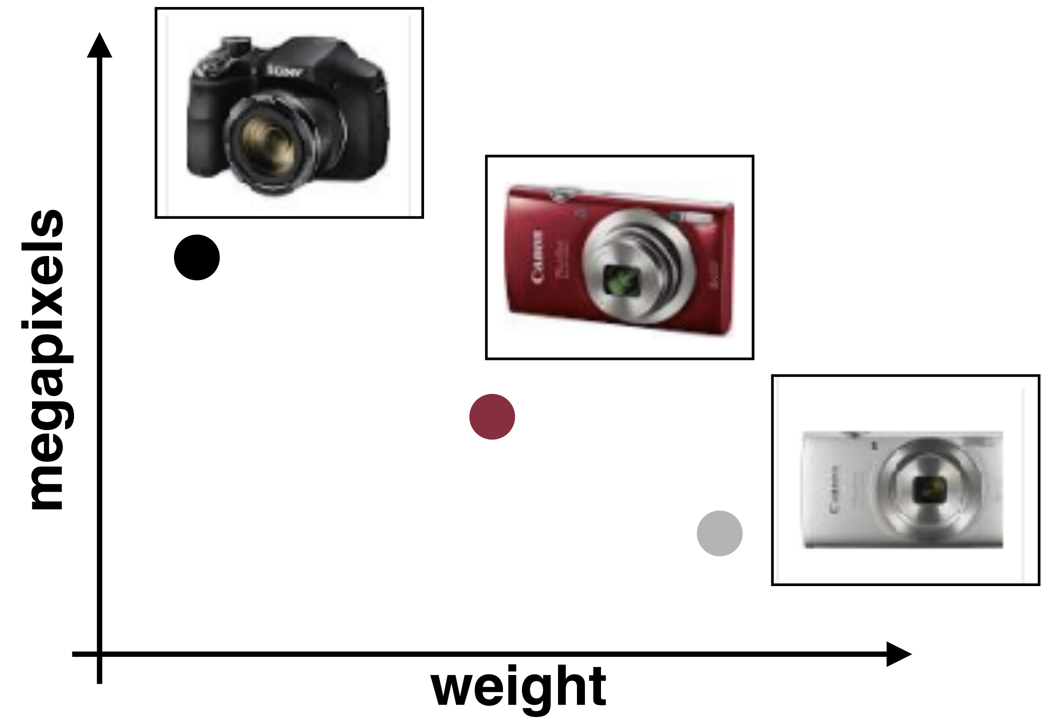
- Bias towards moderation, compromise effect



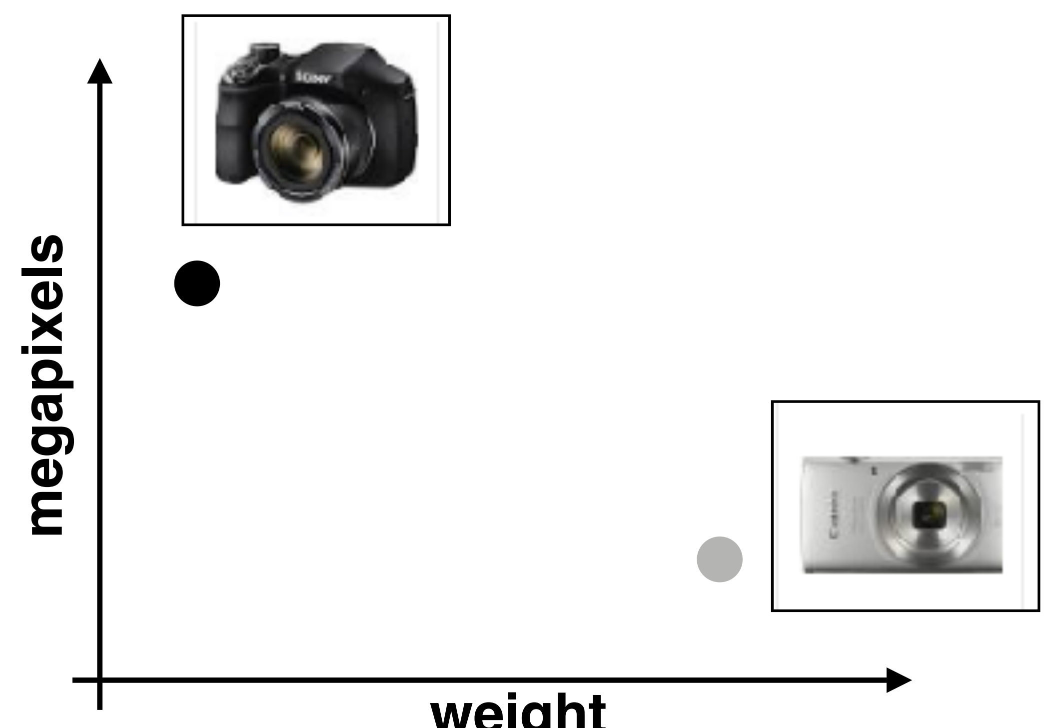
- [Simonson 1989, Simonson-Tversky 1992, Kamenica 2008, Trueblood 2013]

Choice set effects

- Bias towards moderation, compromise effect



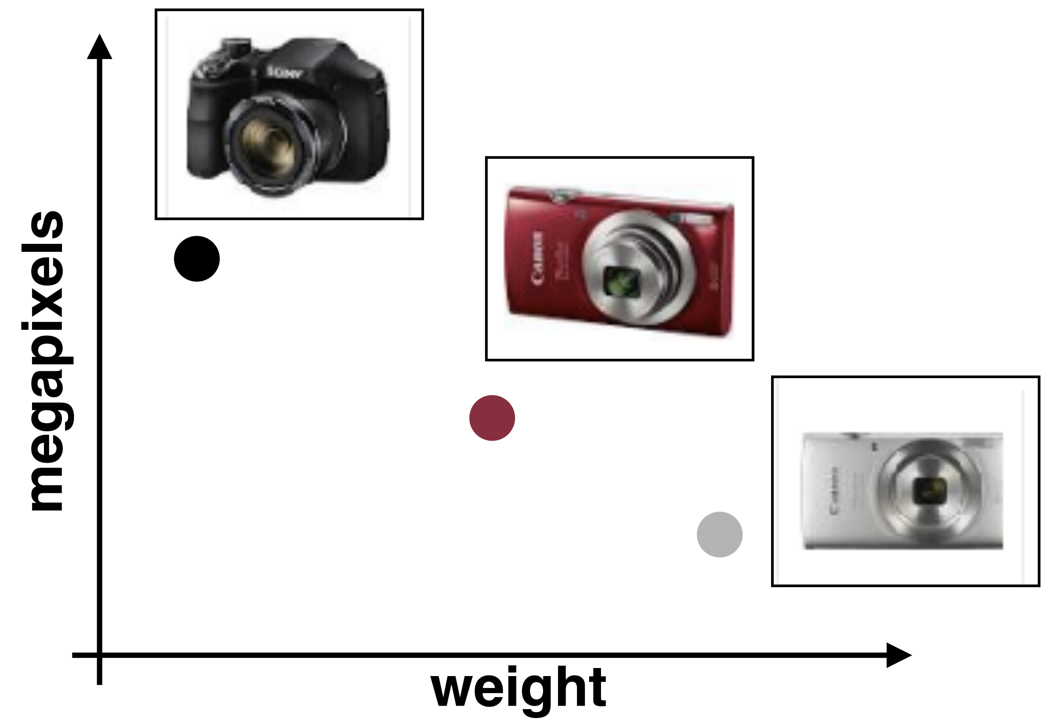
- Similarity aversion



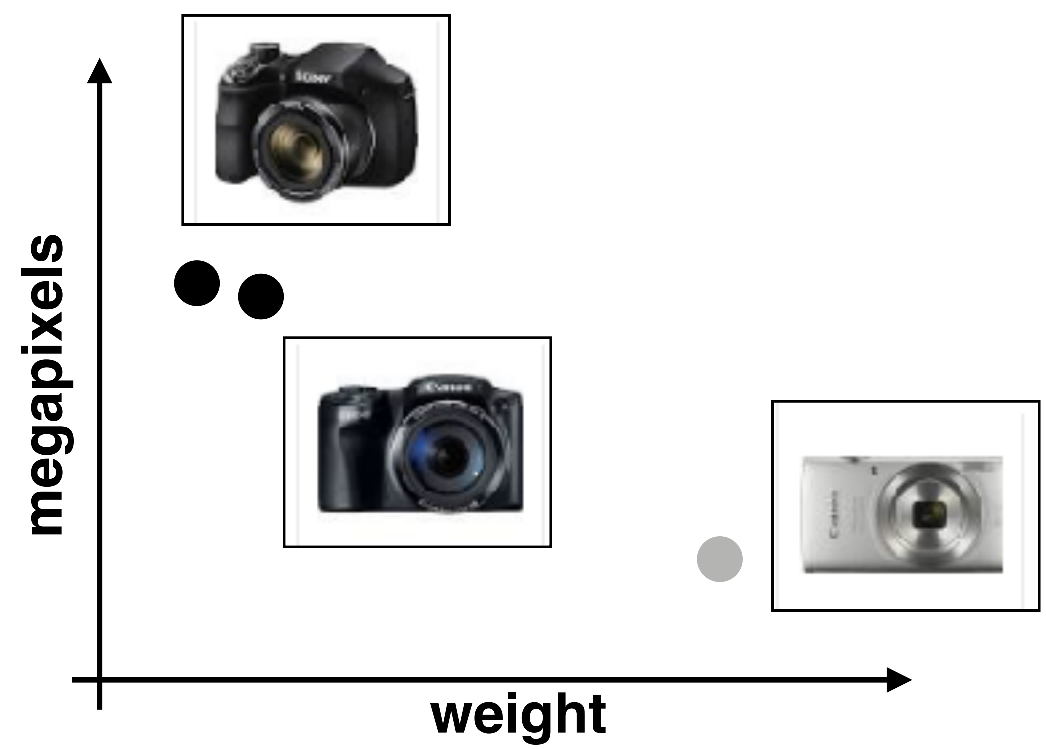
- [Simonson 1989, Simonson-Tversky 1992, Kamenica 2008, Trueblood 2013]

Choice set effects

- Bias towards moderation, compromise effect



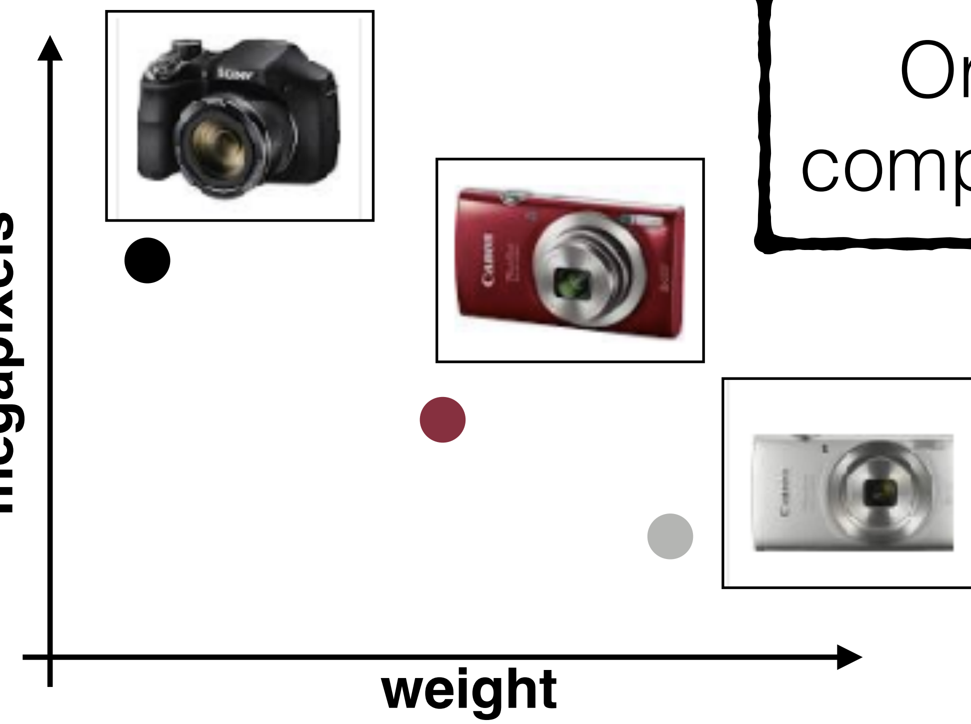
- Similarity aversion



- [Simonson 1989, Simonson-Tversky 1992, Kamenica 2008, Trueblood 2013]

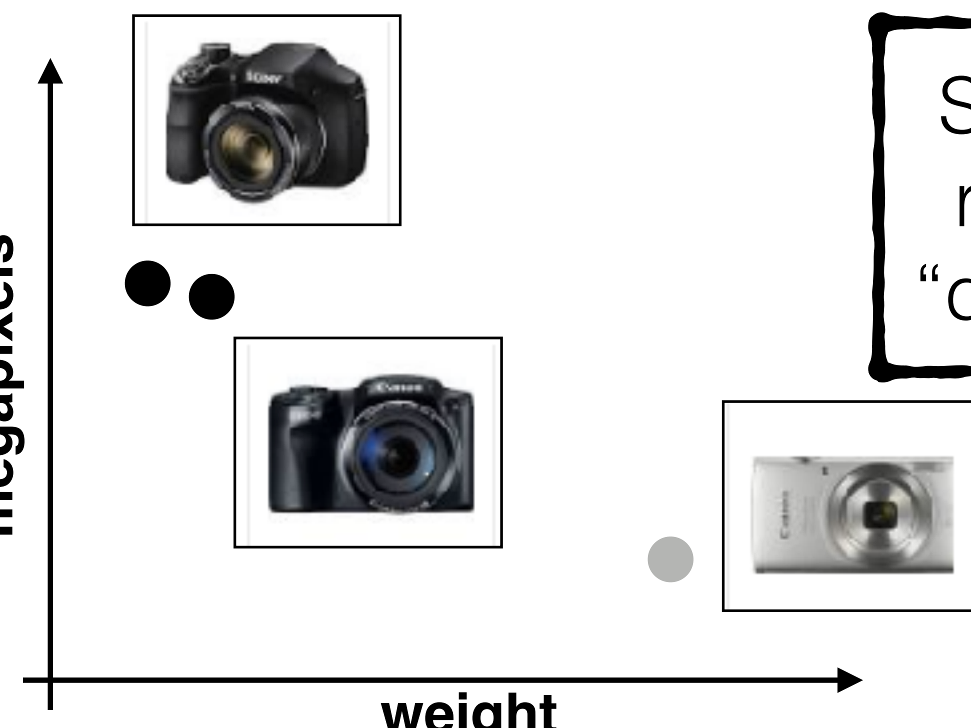
Choice set effects

- Bias towards moderation, compromise effect



Ordinal comparisons

- Similarity aversion



Similarity requires "distance"

- [Simonson 1989, Simonson-Tversky 1992, Kamenica 2008, Trueblood 2013]

The present work

- Focused on **comparison-based functions**.
- Investigate asymptotic **query complexity**: if an agent makes comparison-based choices, how hard to learn their choice function?
- Assume population is not learning, meaning choice set effects are not “transient irrationality”.
- **Several query frameworks:**
  - Active queries vs. passive stream of queries
  - Fixed choice function vs. mixture of choice functions

The present work

- Focused on **comparison-based functions**.

- Investigate asymptotic **query complexity**: if an agent makes comparison-based choices, how hard to learn their choice function?

- Assume population is not learning, meaning choice set effects are not “transient irrationality”.

- **Several query frameworks:**
  - Active queries vs. passive stream of queries
  - Fixed choice function vs. mixture of choice functions

- **Basic takeaway:** comparison-based functions in one dimension (still rich!) are no harder to learn than binary comparisons (sorting).

Comparison-based choice functions

- **Definition:** Given a set of alternatives **U**, a **choice function f** maps every non-empty **S⊆U** to an element **u∈S.**
- **Example:**

U: 

S **u**

f( **) =** 

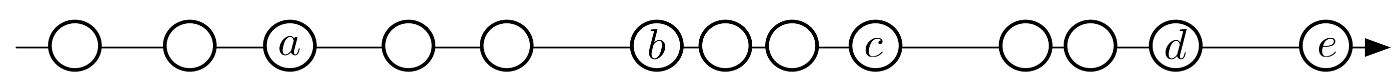
Comparison-based choice functions

- **Definition:** Given a set of alternatives **U**, a **choice function f** maps every non-empty **S⊆U** to an element **u∈S.**
- **Example:**

U: 

S **u**

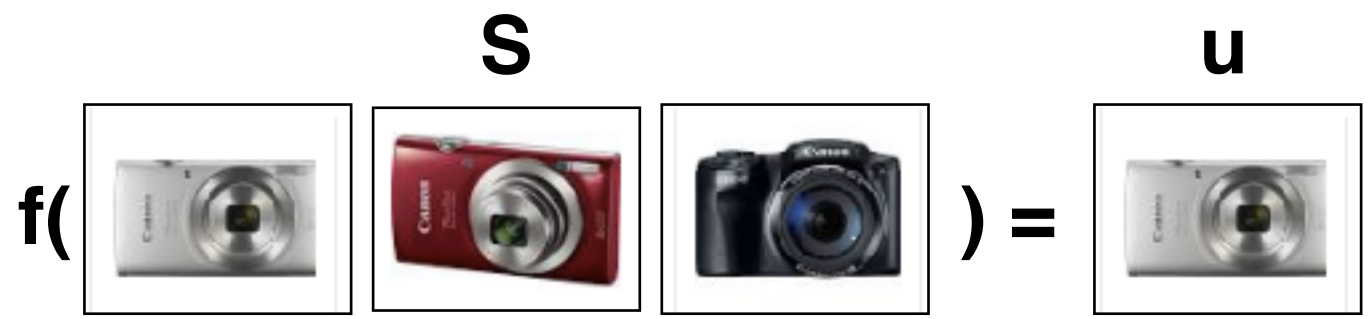
f( **) =** 

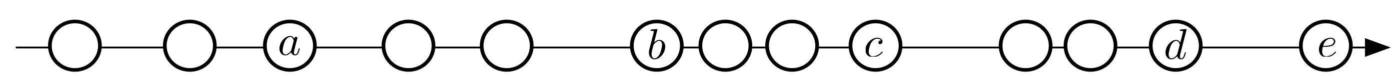
- **Embedding items:**
    - Consider **U** as embedded in attribute space, **h:U->X**
    - For **X =** $\mathbb{R}^1$, **h(u$_i$)** are utilities: 

Comparison-based choice functions

- **Definition:** Given a set of alternatives **U**, a **choice function f** maps every non-empty **S⊆U** to an element **u∈S.**
- **Example:**

U:

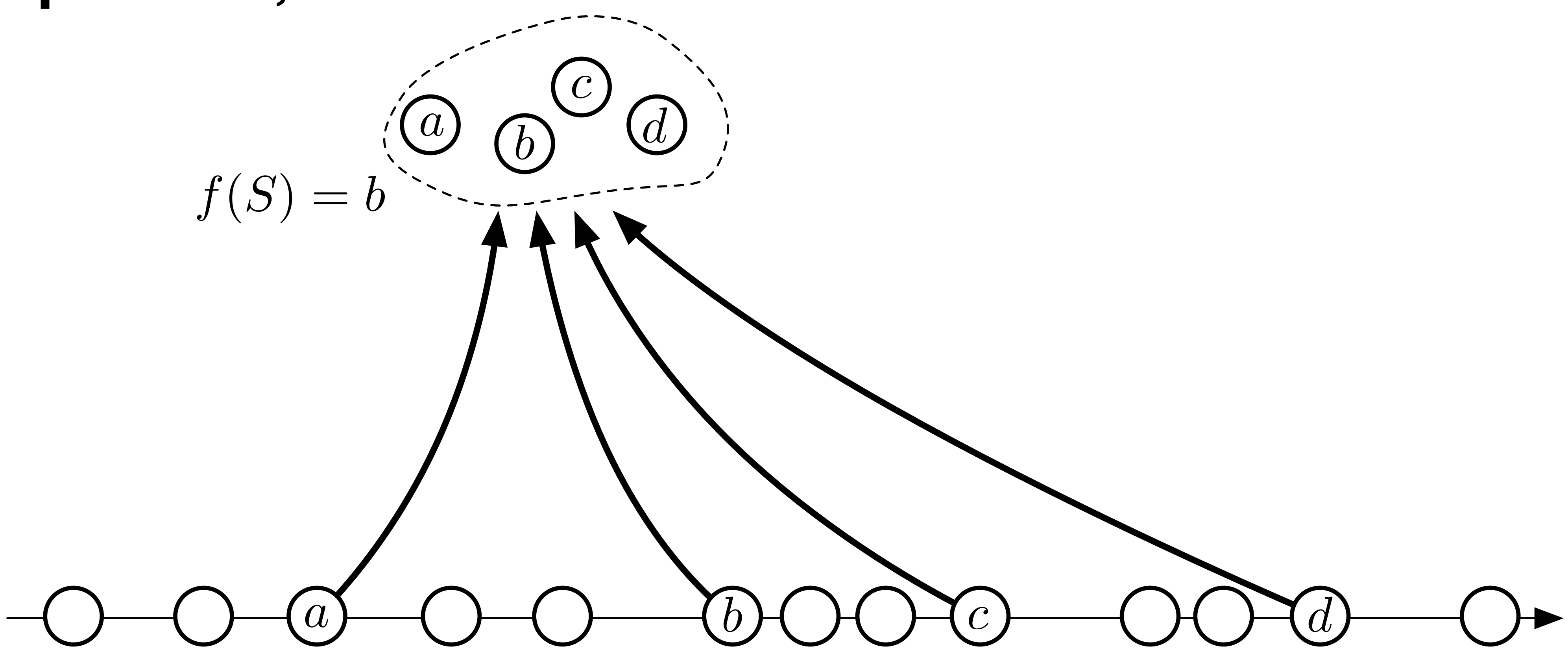


- **Embedding items:**
  - Consider **U** as embedded in attribute space, **h:U->X**
  - For **X =** $\mathbb{R}^1$, **h(u$_i$)** are utilities: 

- **Comparison-based functions:**
  - **Definition:** Choice functions that can be written as comparisons (<,>,=) over **{h(u$_i$): u$_i$∈S}**.

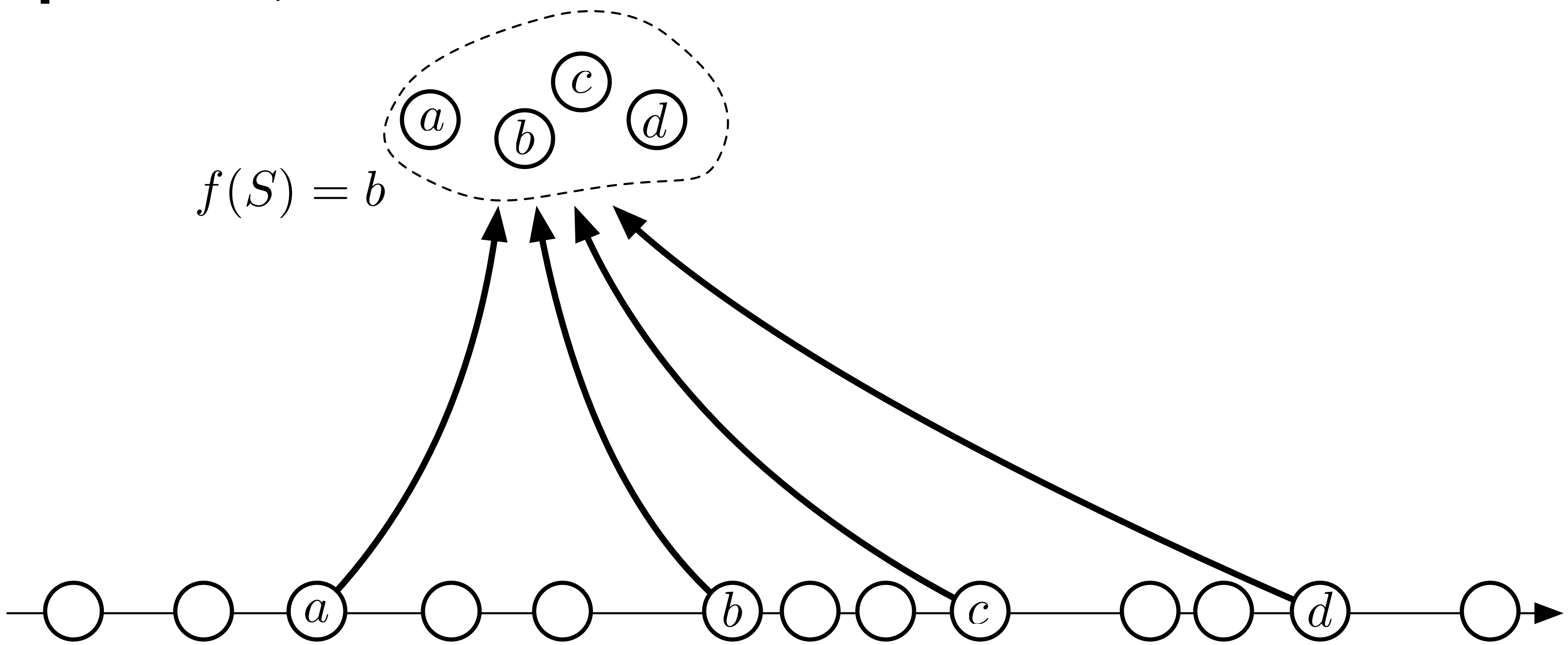
Comparison-based choice functions

- In one dimension, comparison-based functions are all **position-selection functions**: **select ℓ-of-k.**
- **Example: k=4, ℓ=2**



Comparison-based choice functions

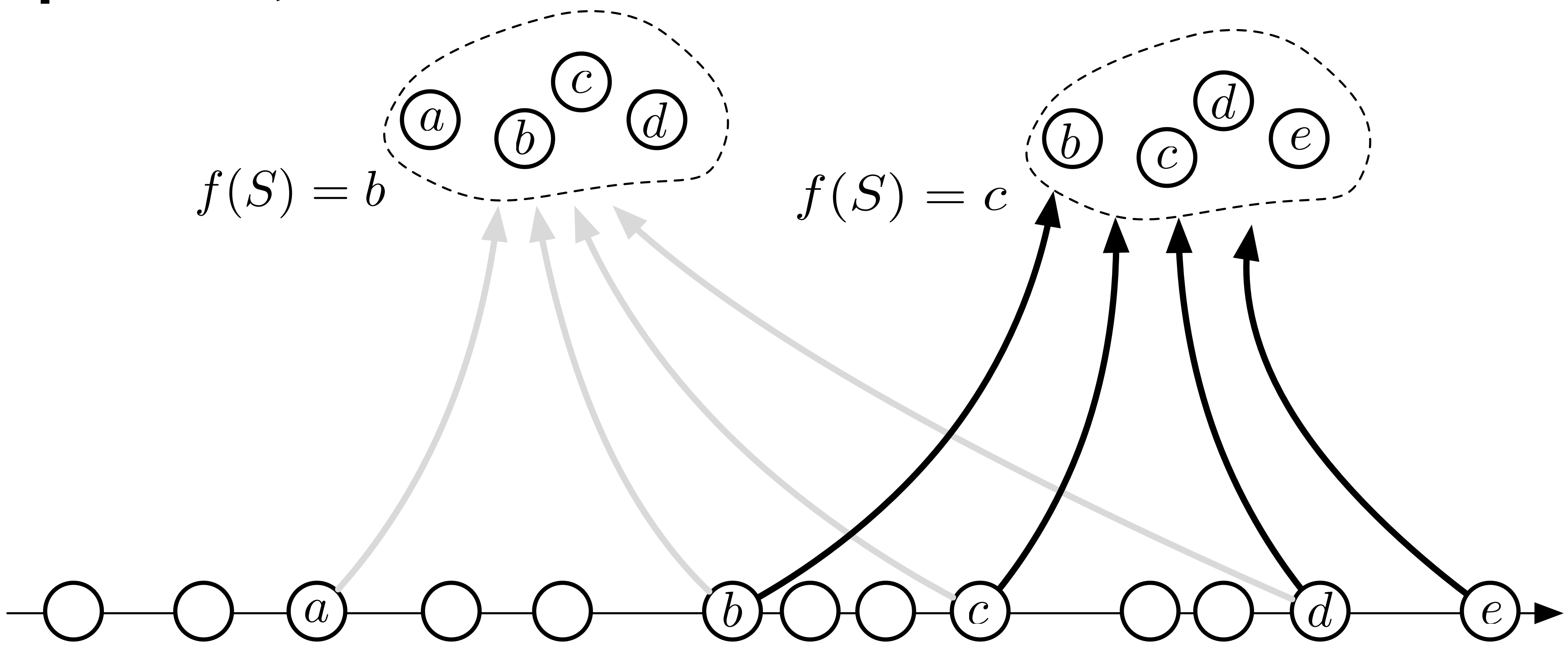
- In one dimension, comparison-based functions are all **position-selection functions**: **select ℓ-of-k.**
- **Example: k=4, ℓ=2**



- Selecting **1-of-2** is **sorting.**
- Focus on **k**-sets **S** with fixed **k**.

Comparison-based choice functions

- In one dimension, comparison-based functions are all **position-selection functions**: **select ℓ-of-k.**
- **Example: k=4, ℓ=2**



- Selecting **1-of-2** is **sorting.**
- Focus on **k**-sets **S** with fixed **k**.
- **Position-selection functions exhibit choice set effects.**

Query complexity

- Observe sequence of (choice set, choice) pairs **(S, f(S)).**
- How many do we need to observe to report **f(S)** for (almost) all **S**?

Query complexity

- Observe sequence of (choice set, choice) pairs **(S, f(S)).**
- How many do we need to observe to report **f(S)** for (almost) all **S**?

- Active vs. passive queries
  - **Active:** can choose what k-set **S** to query next, sequentially.
  - **Passive:** Stream of random k-sets **S**.

- Fixed vs. mixed choice functions
  - **Fixed:** all queries of same ℓ**-of-k** function.
  - **Mixed:** mixture $(\pi_1, ..., \pi_k)$ of different positions selected.

Query complexity, binary choices

- How does sorting (**1-of-2**) fit in this query complexity framework?
- Mixed binary choice functions map to (**p,1-p**) noisy sorting.

	Fixed	**Mixed**
Active	Sorting from comparisons O(n log n)	Sorting with noisy comparisons (Feige et al. 1994) O(n log n)
Passive	Sorting in one round (Alon-Azar 1988) O(n log n loglog n)	?

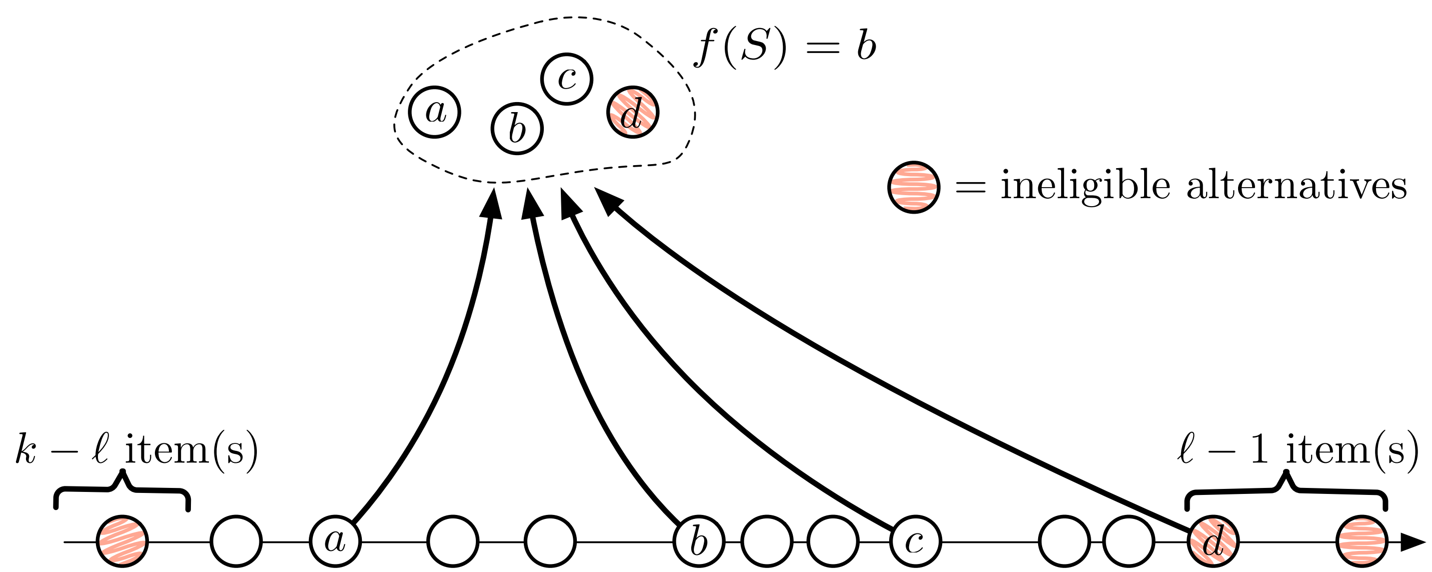
Query complexity, k-set choices

- Sorting results translated to position-selection functions:

	Fixed	Mixed
Active	Two-phase algorithm O(n log n)	Adaptation of two-phase algorithm O(n log n)
Passive	Streaming model $O(n^{k-1} \log n \log\log n)$	?

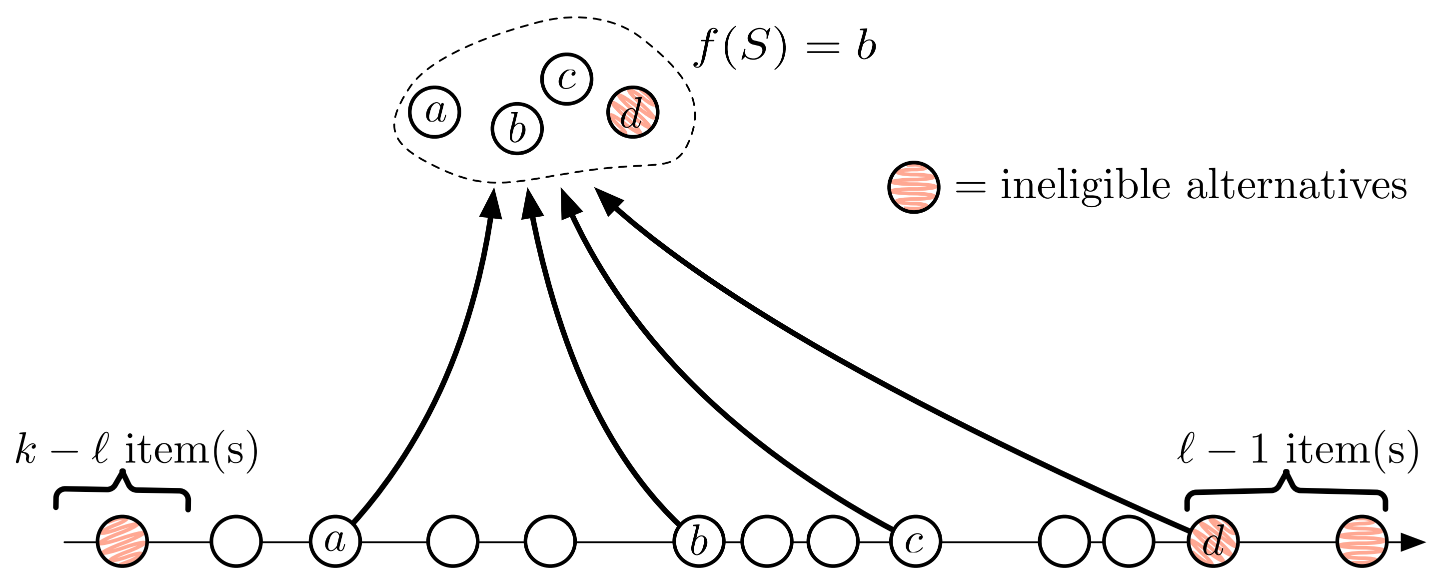
Query complexity: active, fixed

- **Phase 1:** find "ineligible alternatives" via a discard algorithm



Query complexity: active, fixed

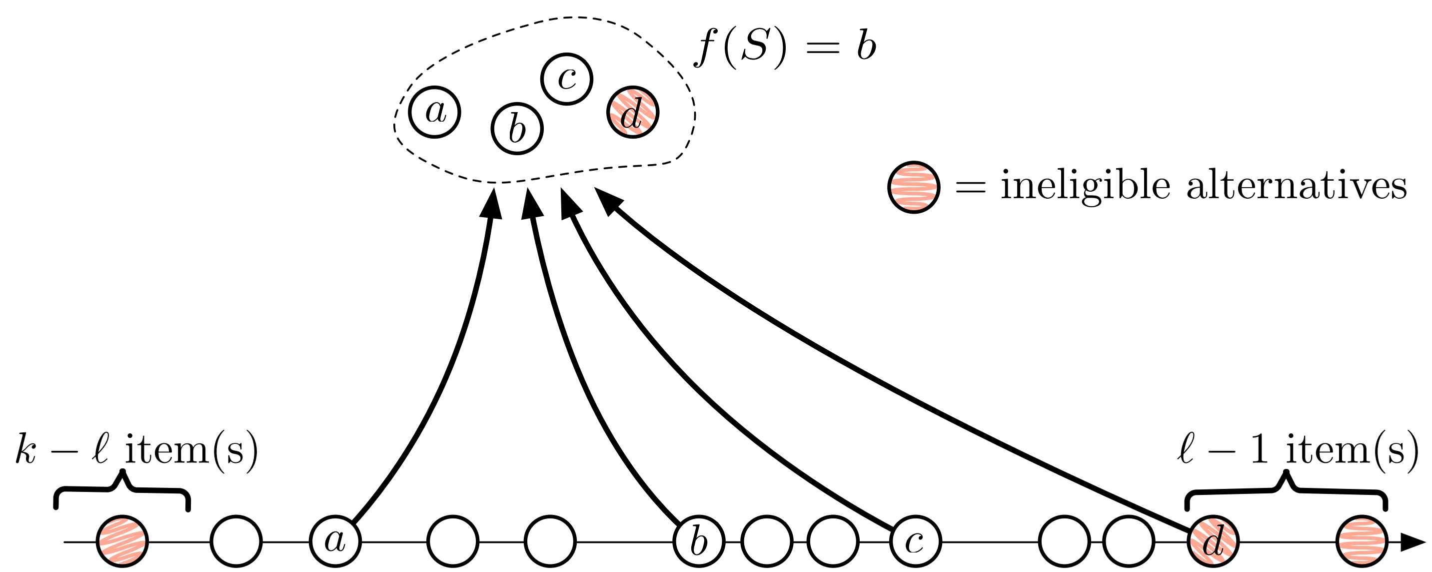
- **Phase 1:** find "ineligible alternatives" via a discard algorithm



- **Phase 2:** Pad a choice set with ineligible alternatives, do binary sort.

Query complexity: active, fixed

- **Phase 1:** find "ineligible alternatives" via a discard algorithm

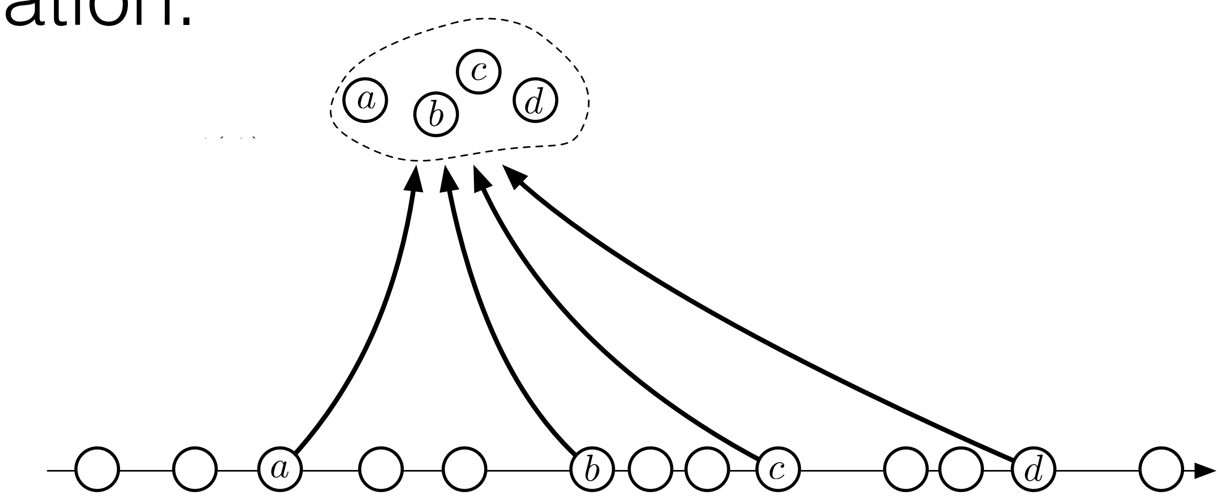


- **Phase 2:** Pad a choice set with ineligible alternatives, do binary sort.

- O(n) queries in discard algorithm, O(n log n) queries to sort.
- Only recovers order, not orientation: don't know if "padded sort" is a "max" or a "min", but not needed to recover **f(S)** for ever **S**.
- Algorithm doesn't depend on what position is being selected for.

Query complexity: active, mixed

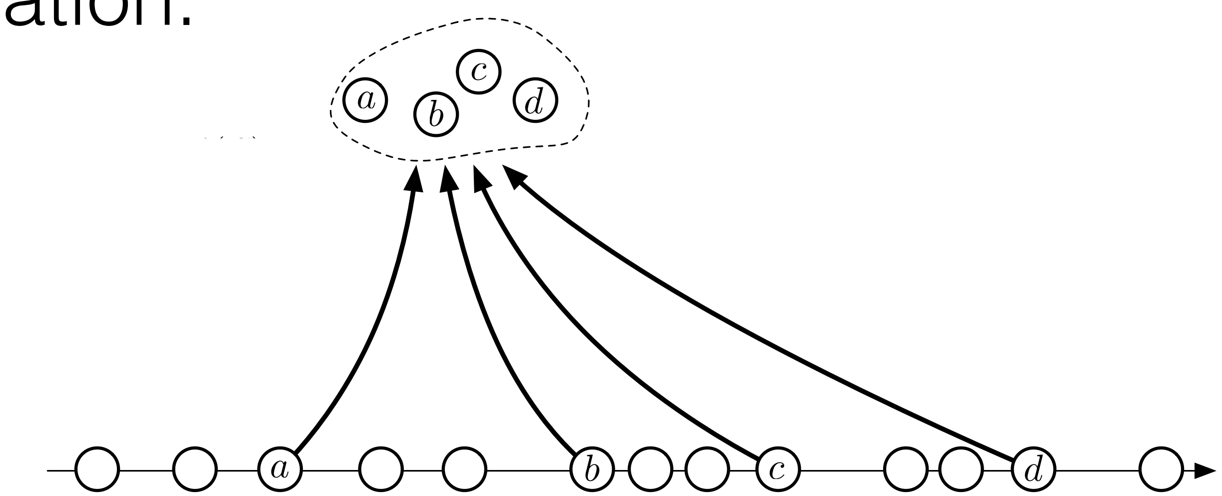
- Instead of ℓ**-of-k,** mixture of positions with probabilities $(\pi_1, ..., \pi_k)$, constant separation.



- **0:** Estimate probabilities of each position by studying a **k+1**-set closely.
- **1:** Run **discard phase** O(log n) times, find "max-ineligible alternatives"
- **2:** Can then **pad choice set** and run a "noisy max" with (max, min, fail) outcomes instead of (max, min) outcomes as in (Feige et al. 1994).

Query complexity: active, mixed

- Instead of ℓ**-of-k,** mixture of positions with probabilities $(\pi_1, ..., \pi_k)$, constant separation.



- **0:** Estimate probabilities of each position by studying a **k+1**-set closely.
- **1:** Run **discard phase** O(log n) times, find “max-ineligible alternatives”
- **2:** Can then **pad choice set** and run a “noisy max” with (max, min, fail) outcomes instead of (max, min) outcomes as in (Feige et al. 1994).

- O(1) queries estimate probabilities, O(n log n) queries in discard algorithm, O(n log n) queries to sort.
- Need to book-keep many failure probabilities, but straight forward.

Query complexity: passive, fixed

- **Passive query model:** Poisson process where each **k**-set enters the stream with equal rate **α.**
- See a given **k**-set in interval [0,T] with probability $\mathbf{p_T}$.
- How long an interval [0,T] do we need to observe stream?

- **Phase 1:** use queries in $[0,T_1]$, with T_1 large enough so that all items except ineligible alternatives are chosen.
- **Phase 2:** Simulate pairwise comparisons using queries where **k-2** of the elements are ineligible.

Query complexity: passive, fixed

- **Passive query model:** Poisson process where each **k**-set enters the stream with equal rate **α.**
- See a given **k**-set in interval [0,T] with probability $\mathbf{p_T}$.
- How long an interval [0,T] do we need to observe stream?

- **Phase 1:** use queries in $[0,T_1]$, with T_1 large enough so that all items except ineligible alternatives are chosen.
- **Phase 2:** Simulate pairwise comparisons using queries where **k-2** of the elements are ineligible.

- For Phase 2 to work, need $\mathbf{p_T}$ to be O(log n loglog n / n). End up seeing ~**log(n)/n** fraction of all **(n choose k)** choice sets.
- For **k≥3,** proof only works for positions $1<\ell<k$, **not ℓ=1 or ℓ=k,** which breaks our analysis ($\mathbf{p_T} \nrightarrow \mathbf{0}$).

Query complexity, k-set choices

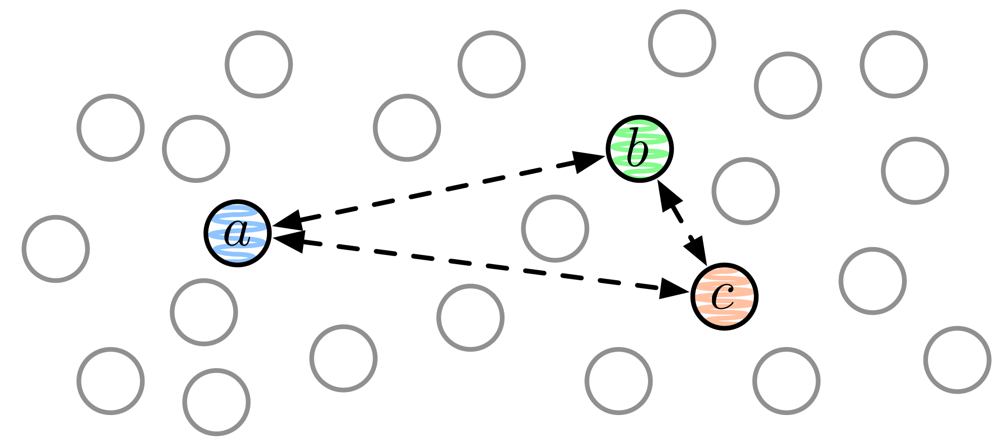
- Sorting results translated to position-selection functions:

	Fixed	**Mixed**
Active	Two-phase algorithm $O(n \log n)$	No new difficulties $O(n \log n)$
Passive	Streaming model $O(n^{k-1} \log n \log\log n)$	?

- **Immediate questions:**
  - Better algo for passive stream; “sorting in one noisy round”; higher-dim comparison functions; distance-comparison.

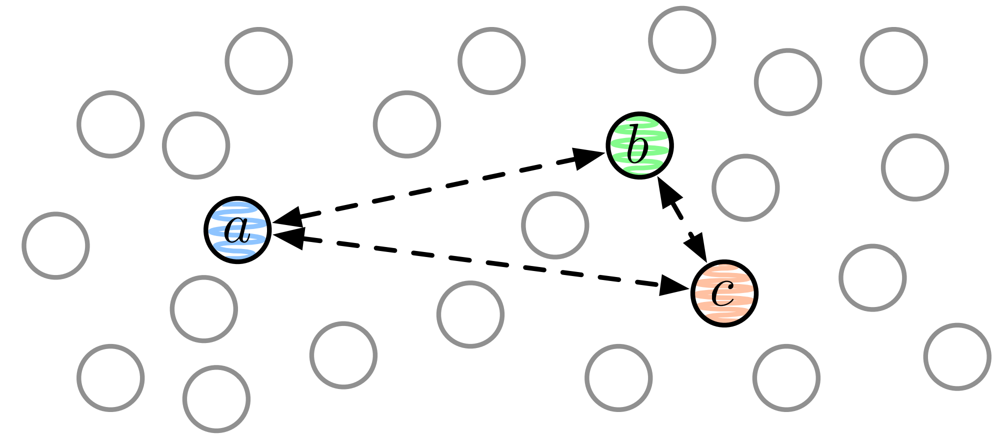
Distance-comparison-based choice

- **Distance-comparison-based functions** are comparison functions on the set of pairwise distances.

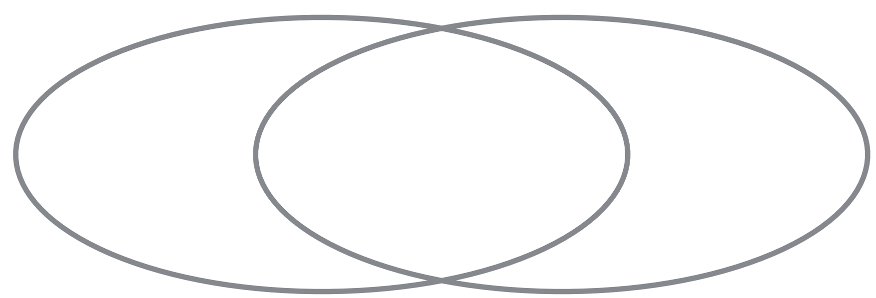


Distance-comparison-based choice

- **Distance-comparison-based functions** are comparison functions on the set of pairwise distances.

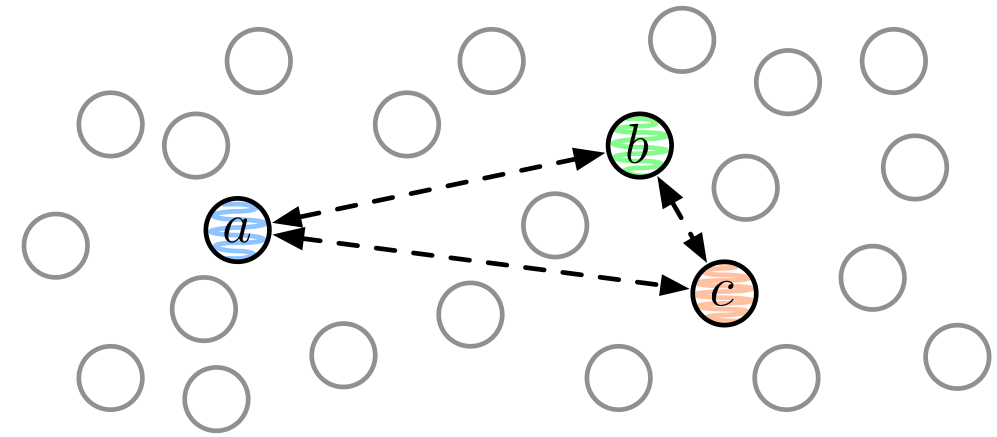


- **Distance-comparison** vs. **comparison** functions are quite different.

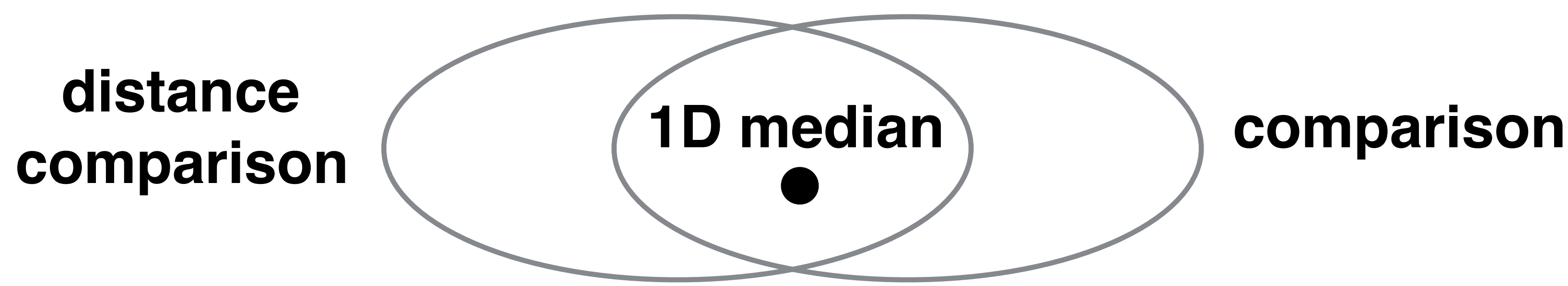


Distance-comparison-based choice

- **Distance-comparison-based functions** are comparison functions on the set of pairwise distances.



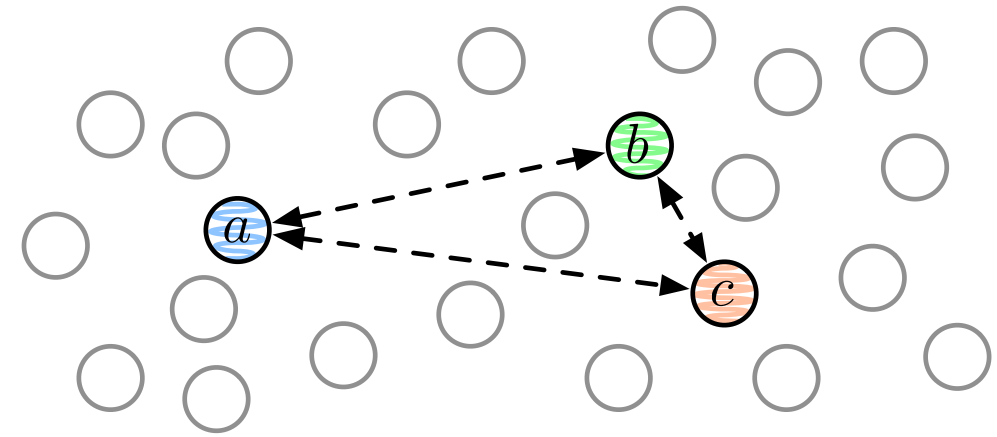
- **Distance-comparison** vs. **comparison** functions are quite different.



- **Comparison functions:**
  - Can not express similarity (only order)
- **Distance-comparison functions:**
  - Can not maximize or minimize (distances are all internal to set)

Distance-comparison-based choice

- **Distance-comparison-based functions** are comparison functions on the set of pairwise distances.



- Paper poses **many questions** about distance-comparison, **few answers.**

- Related to open learning questions for:
  - Crowd median algorithm [Heikinheimo-Ukkonen 2013]
  - Stochastic triplet embedding [Van Der Maaten-Weinberger 2012]
  - Crowdsourced clustering [Vinayak-Hassibi 2016]
  - Metric embedding [Schultz-Joachims 2004].

Summary

- Inference for comparison-based functions generally not more difficult than sorting.
- Active vs. passive, fixed vs. mixed query complexity frameworks.

- **Open questions**:
  - Results for high-dim (EBA?), distance-comparison, RUMs.
  - Learning/non-static agents?

- **Other recent work:**
  - [Benson et al. WWW'16] "On the relevance of irrelevant alternatives"
  - [Ugander-Ragain, NIPS'16] Markov chain model generalizing BTL/MNL, can violate IIA.
  - [Maystre-Grossglauser ICML'17] For BTL with ~uniform quality, $\log^5(n)$ independent Quicksorts recover exact rank for almost all items.
  - [Peysakhovich-Ugander NetEcon'17] Machine learning adaptation of the Simonson-Tversky model for contextual utility.