ifis
Institut für Informationssysteme
Technische Universität Braunschweig

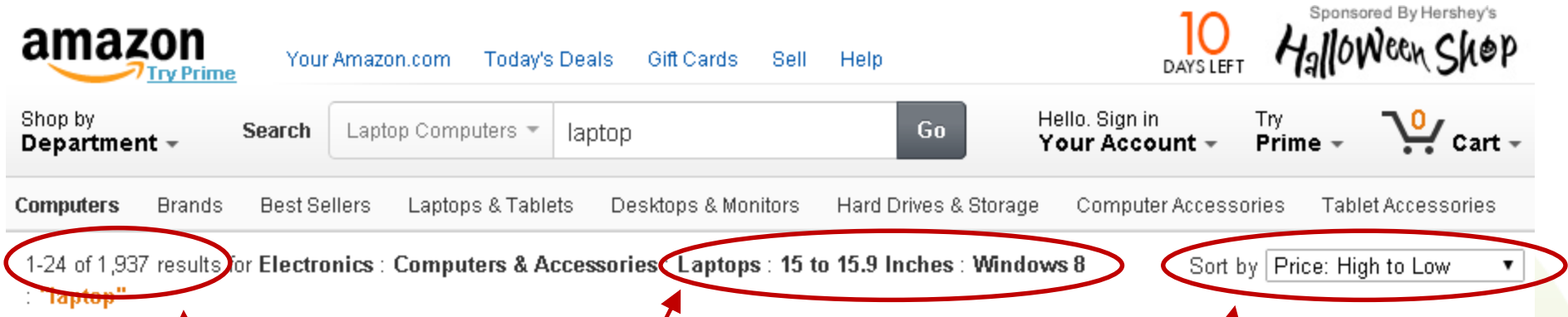
# TopCrowd – Efficient Crowd-enabled Top-k Retrieval on Incomplete Data

**Christian Nieke, Ulrich Güntzer, Wolf-Tilo Balke**

Technische Universität Braunschweig,
Universität Tübingen

# Revisiting Top-k Retrieval

- Traditional SQL-style Queries:




Too many, or too few results

Independent restriction of attributes

Optional sort by single attributes, but not combinations

# Revisiting Top-k Retrieval

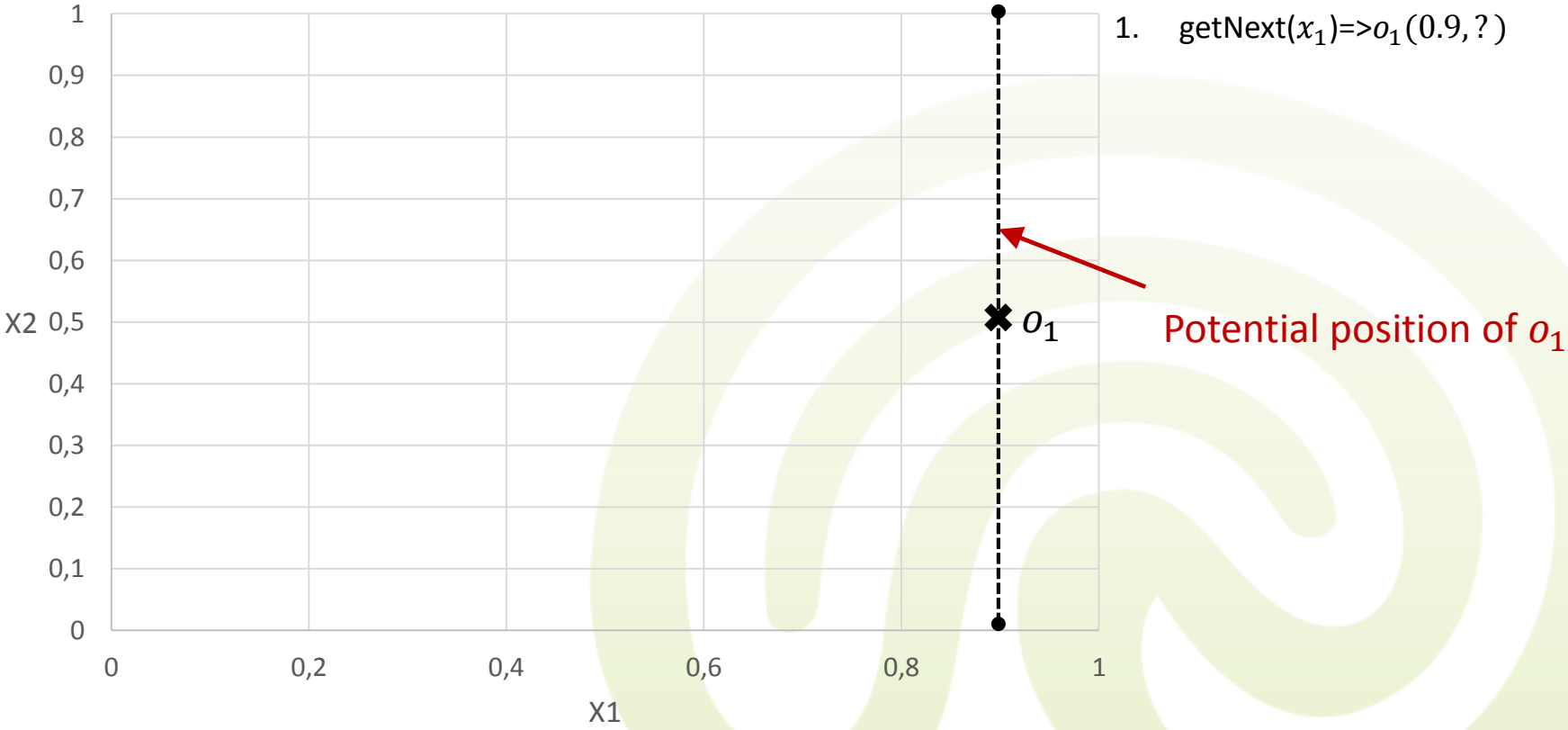
- Top-k Queries:
  - Show only the *k* best results (20,100…)
  - Ordered by *relevance* for the user
    - User defined *scoring function*
    - Allows to balance conflictive goals
  - Example: Laptop
    - Low weight, big screen, long battery duration,…

# Revisiting Top-k Retrieval

- Classical Top-k Algorithms:
  - Mix *random access* and *sorted access* efficiently
    - Random access: retrieves all attributes for one object
      - $get(o_i) \Rightarrow o_i = (0.5\,, 0.2\,, 0.7)$
    - Sorted access: retrieves object with highest value for one attribute
      - $getNext(attribute2) \Rightarrow o_8 = (?\,, 0.95\,, ?\,)$
    - Infer bounds on attribute values from sorted access

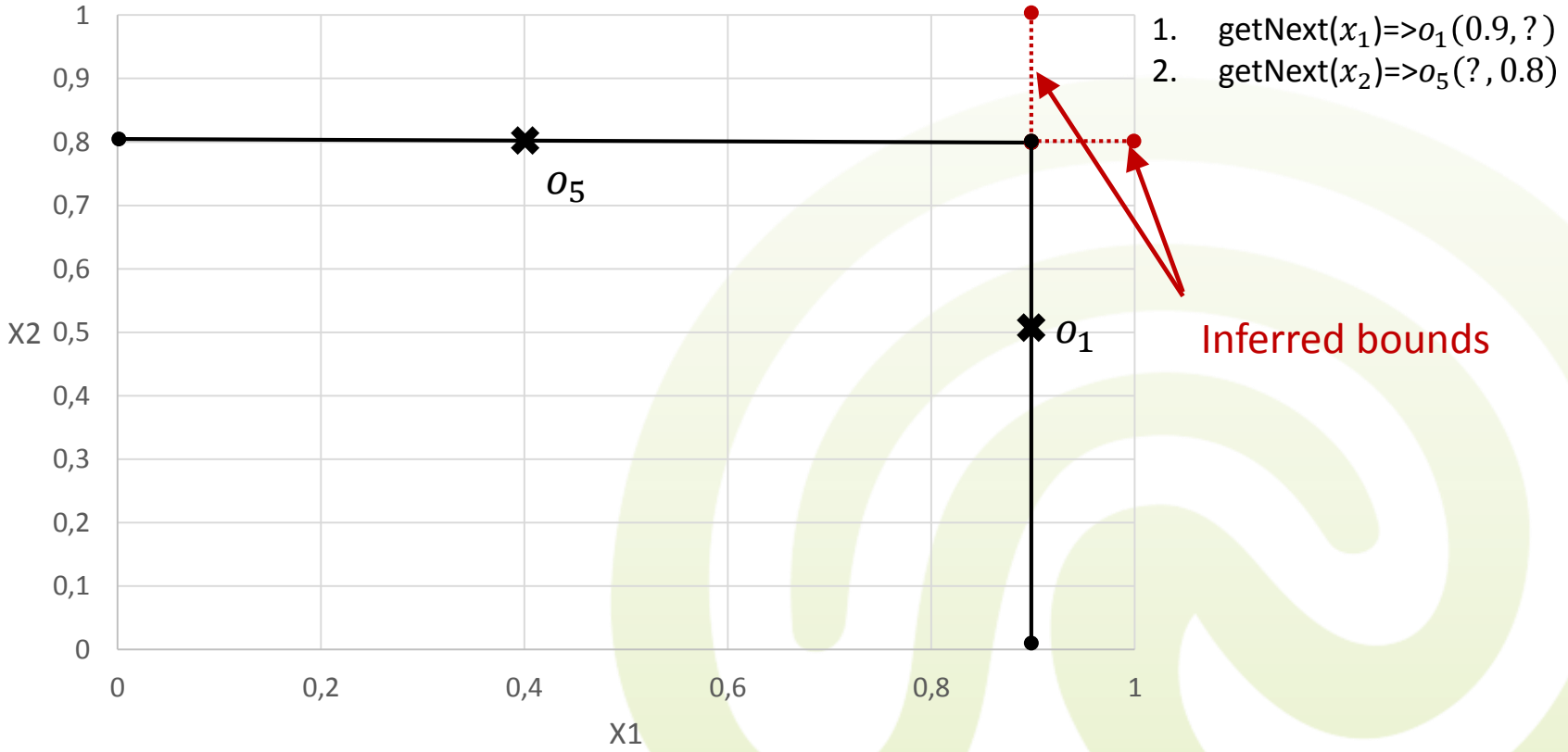
# Revisiting Top-k Retrieval

- Classical Top-k Algorithms:
  - Example: Top 1, $score = (x_1 + x_2)/2$



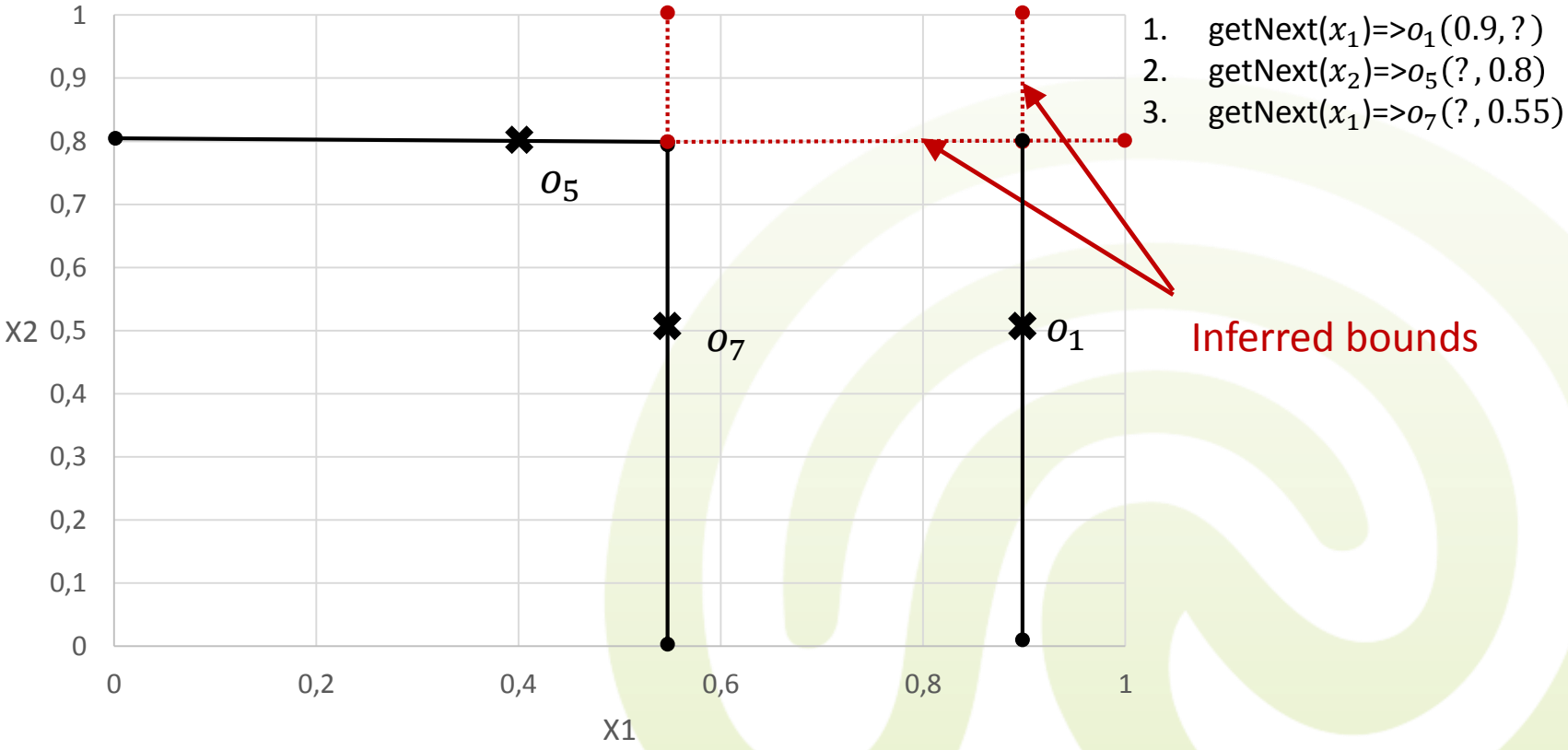
# Revisiting Top-k Retrieval

- Classical Top-k Algorithms:
  - Example: Top 1, $score = (x_1 + x_2)/2$



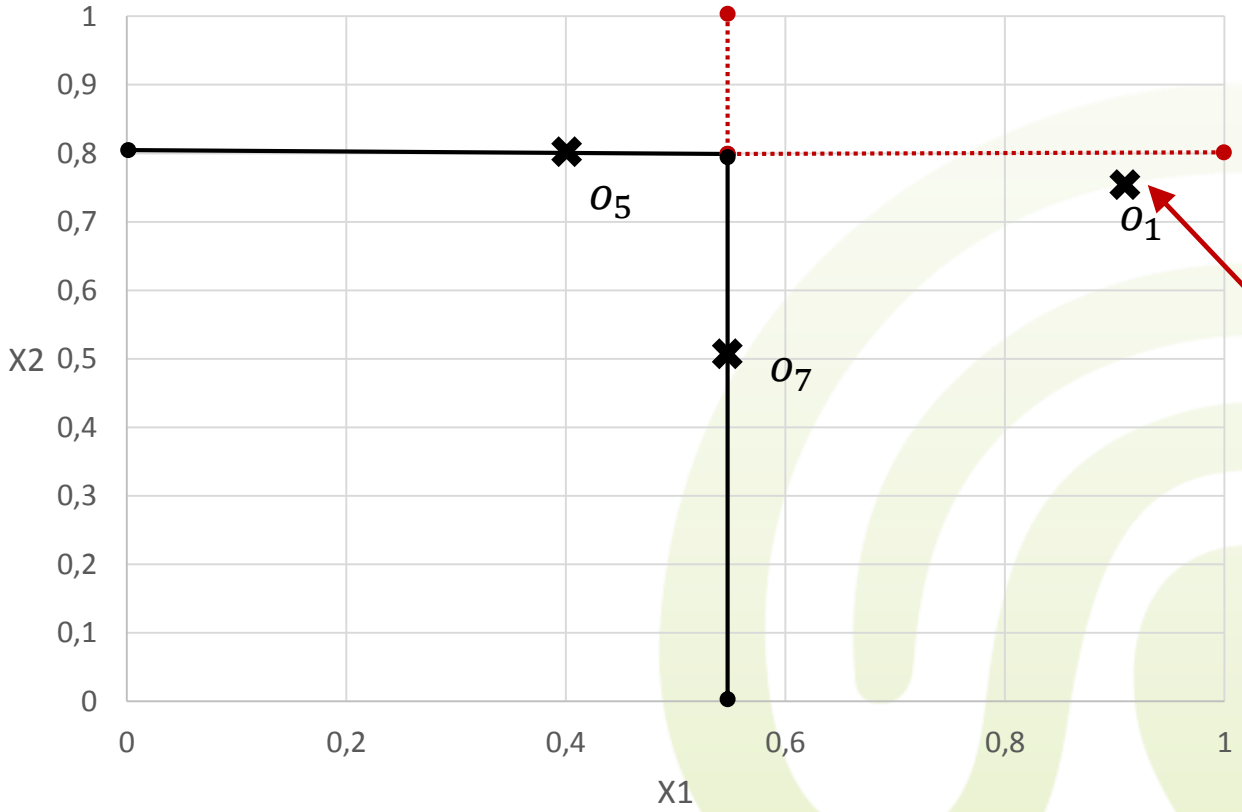
# Revisiting Top-k Retrieval

- Classical Top-k Algorithms:
  - Example: Top 1, $score = (x_1 + x_2)/2$



# Revisiting Top-k Retrieval

- Classical Top-k Algorithms:
  - Example: Top 1, $score = (x_1 + x_2)/2$




1. getNext($x_1$)=>$o_1(0.9, ?)$
2. getNext($x_2$)=>$o_5(?, 0.8)$
3. getNext($x_1$)=>$o_7(?, 0.55)$
4. get($o_1$)=>$o_1(0.9, 0.7)$

Top 1 found
without knowing $o_5$ and $o_7$

# The Challenge

- Incomplete Data
  - Example: Ebay

**Item specifics**

| | |
|---|---|
| Condition: | **Used :** |
| Seller Notes: | "good shape , no cracks or dents to case, screen is in good condition with no dead pixels." |

| | | | |
|---|---|---|---|
| Type: | Notebook | Processor Type: | Intel Core Duo |
| Brand: | ASUS | Processor Speed: | 2.53 GHz |
| Product Line: | G Series | Graphics Processing Type: | Dedicated Graphics |
| Model: | g50vt | Memory: | 6 GB |
| MPN: | Alienware | Hard Drive Capacity: | 750 GB |
| Operating System: | Windows 7 | Operating System Edition: | Home Premium |
| Screen Size: | 15.6" | | |

Vs.

**Item specifics**

| | | | |
|---|---|---|---|
| Condition: | New: A brand-new, unused, unopened, undamaged item in its original packaging (where packaging is ... Read more | Brand: | ASUS |
| Processor Speed: | 2.16GHz - 2.41GHz | Model: | X551MAV-RCLN06 |
| Memory: | 4 GB | Operating System: | Windows 8.1 |
| Hard Drive Capacity: | 500 GB | Screen Size: | 15.6" |

???

# The Challenge

- Incomplete Data
  - Due to user provided content, optional fields, integrated/federated data sources, meta searches,…
  - Some attribute values for some objects are missing
    - Values are not "NULL" (not applicable)
    - …just currently unknown
    - If ignored, our result set will likely be wrong
  - How to get the missing values?

# Crowdsourcing

- Humans can do it!
  - Ask a human to fill in the missing information
    - Contact the vendor (car, house)
    - Lookup the product on the web (laptop)
  - Service personal or Crowdsourcing platform
    - E.g. Crowdflower, Amazon Mechanical Turk,…
  - *Crowd access*
    - Instead of random access

# Managing Costs

- Crowdsourcing comes with a price
  - Delay for *crowd requests*
    - ~several minutes for humans to start a task
    - ➢Make many requests in parallel

  - Payment for the crowd workers
    - Each request costs money
    - ➢Avoid asking for objects not in the Top-k

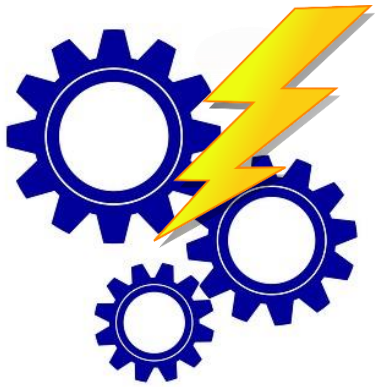
# What changes?

Can't we use classical Top-k?

*Crowd access* is just a more expensive *random access*, right?

# What changes?

- Sorted access
  - Some values are unknown
    - ➢ No complete ranking
    - ➢ No inferred bounds



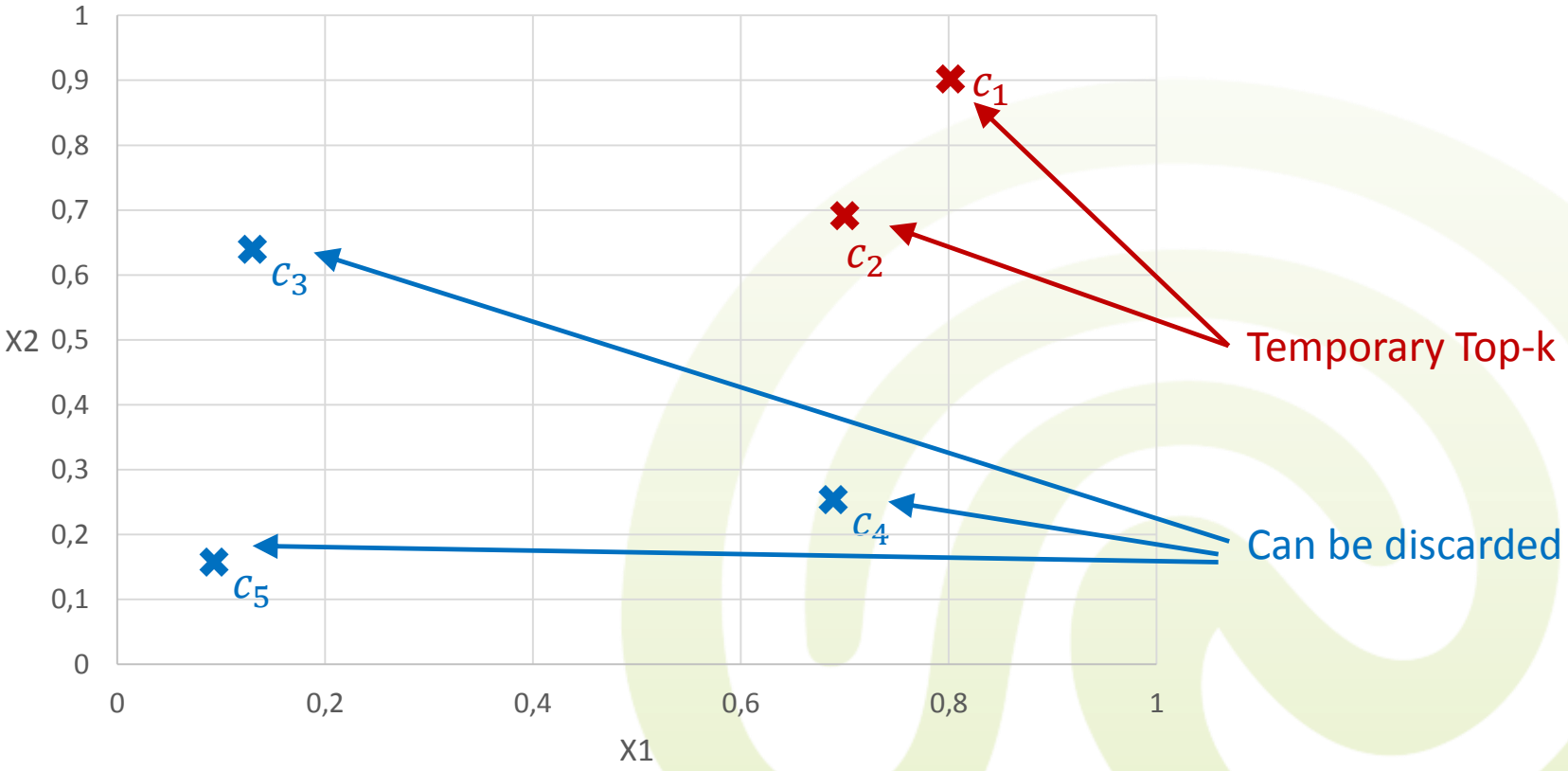
- Classical Top-k breaks
  - Underlying assumptions no longer hold

# TopCrowd Approach

1. Classical Top-k on complete objects
2. Optimal Safe Pruning of incomplete objects
3. Probabilistic ranking of incomplete objects
4. Crowd access cost control

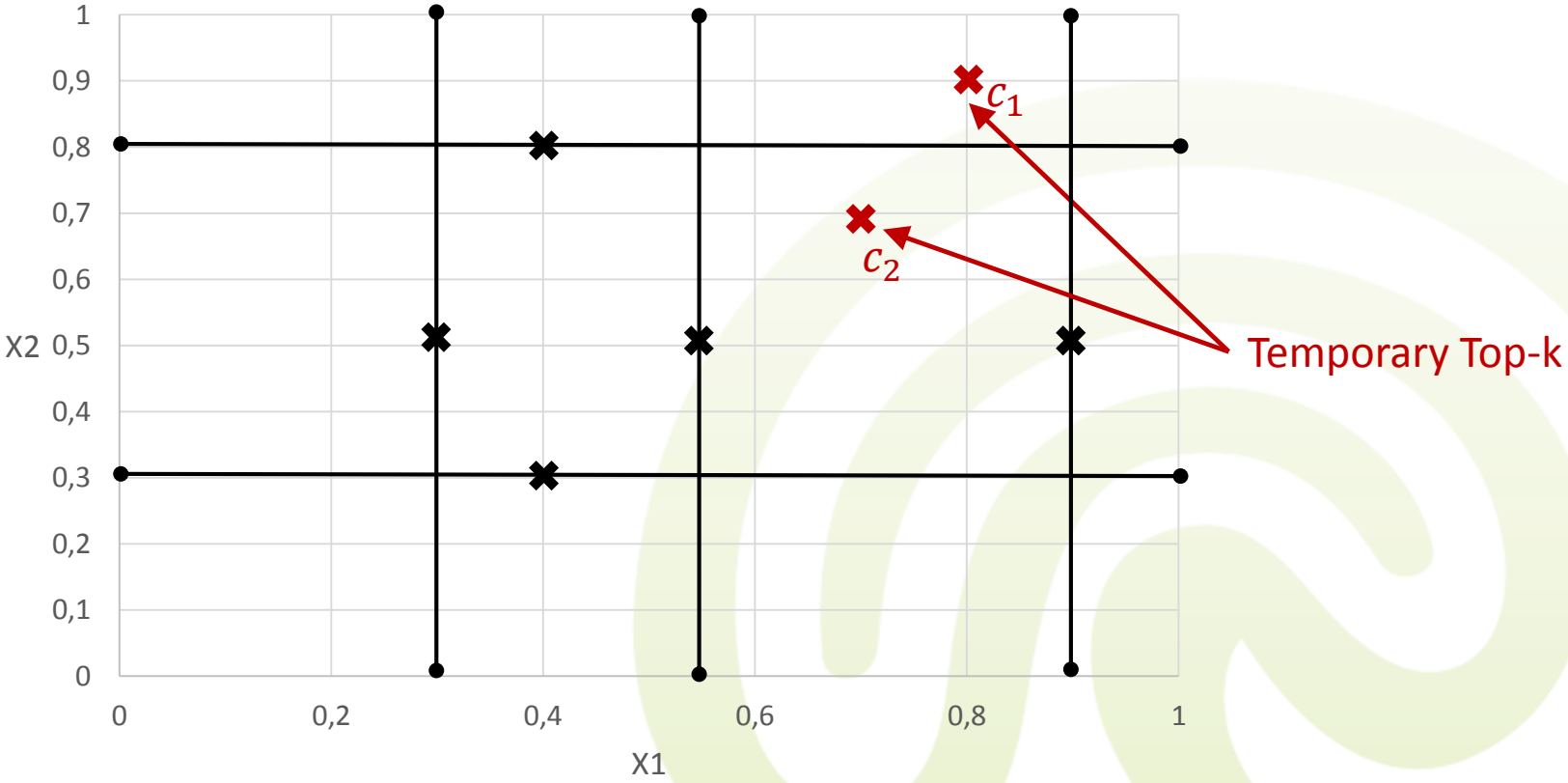
# TopCrowd Approach

- Classical Top-k on complete objects
  - Example: Top 2, $score = (x_1 + x_2)/2$



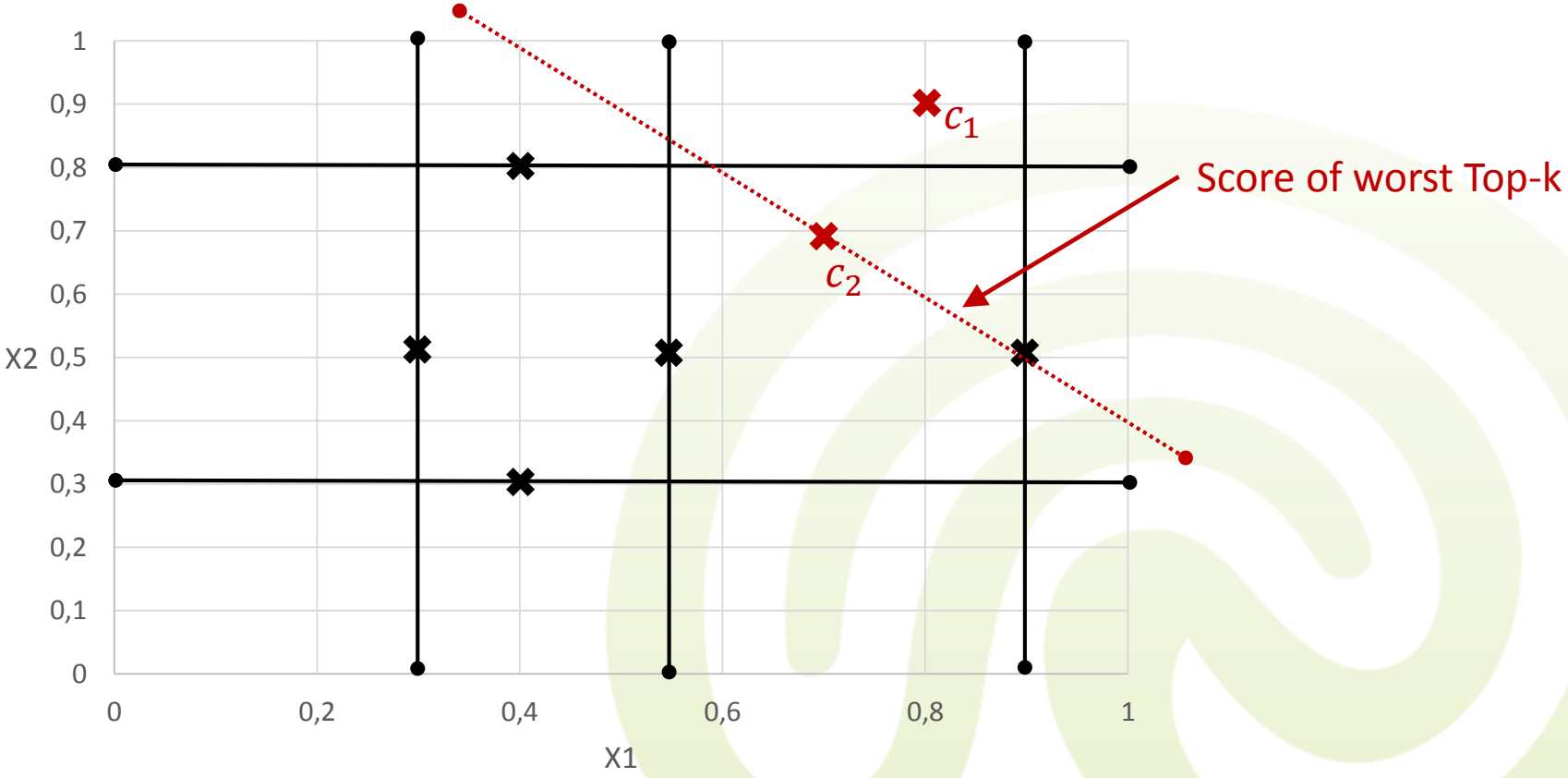
# TopCrowd Approach

- Incomplete objects are left
  - Example: Top 2, $score = (x_1 + x_2)/2$



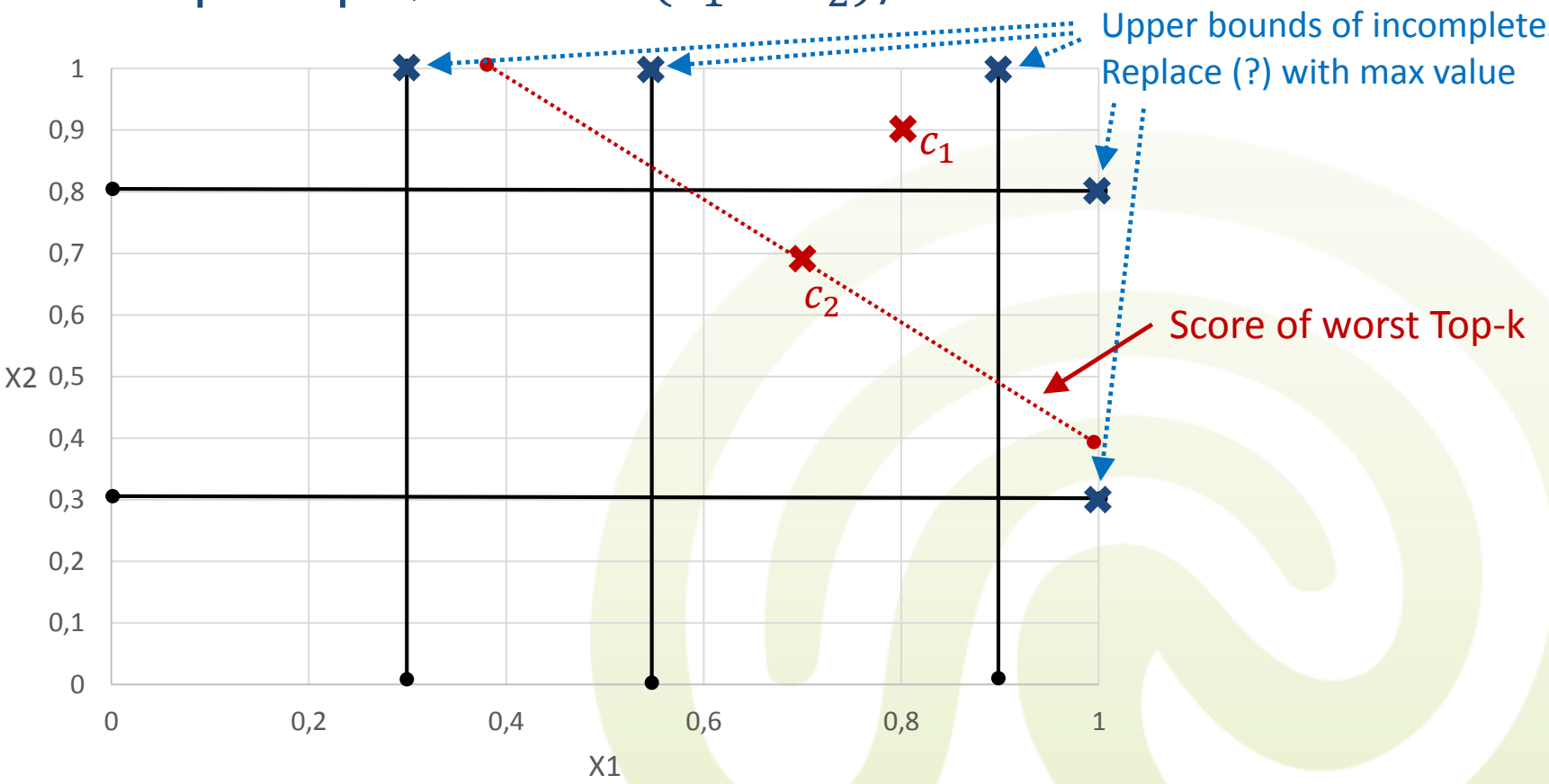
# TopCrowd Approach

- Optimal Safe Pruning
  - Example: Top 2, $score = (x_1 + x_2)/2$



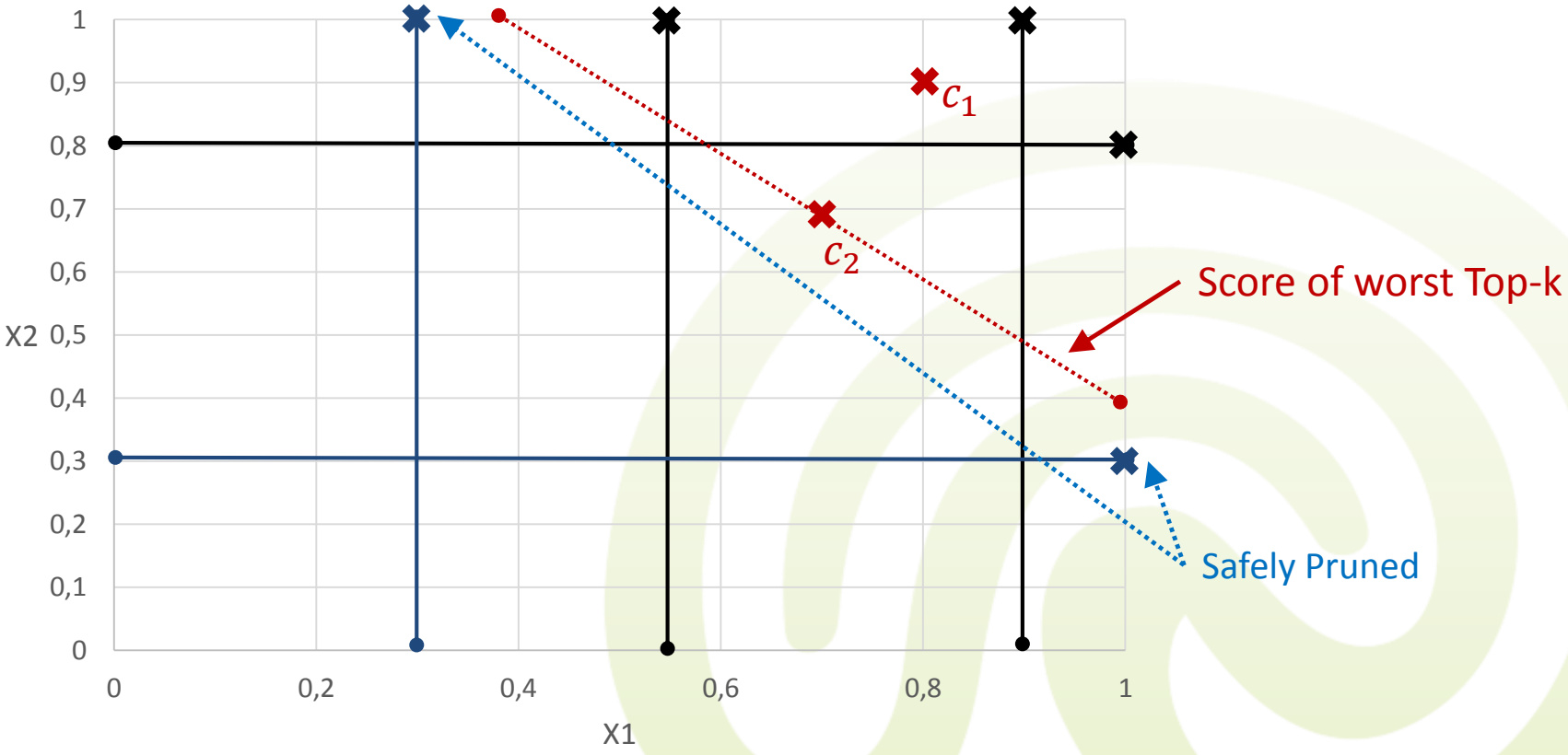
# TopCrowd Approach

- Optimal Safe Pruning
  - Example: Top 2, $score = (x_1 + x_2)/2$



# TopCrowd Approach

- Optimal Safe Pruning
  - Example: Top 2, $score = (x_1 + x_2)/2$



# Next Step?

- Optimal Safe Pruning is done
  - Crowdsource all remaining incompletes?

# Next Step?

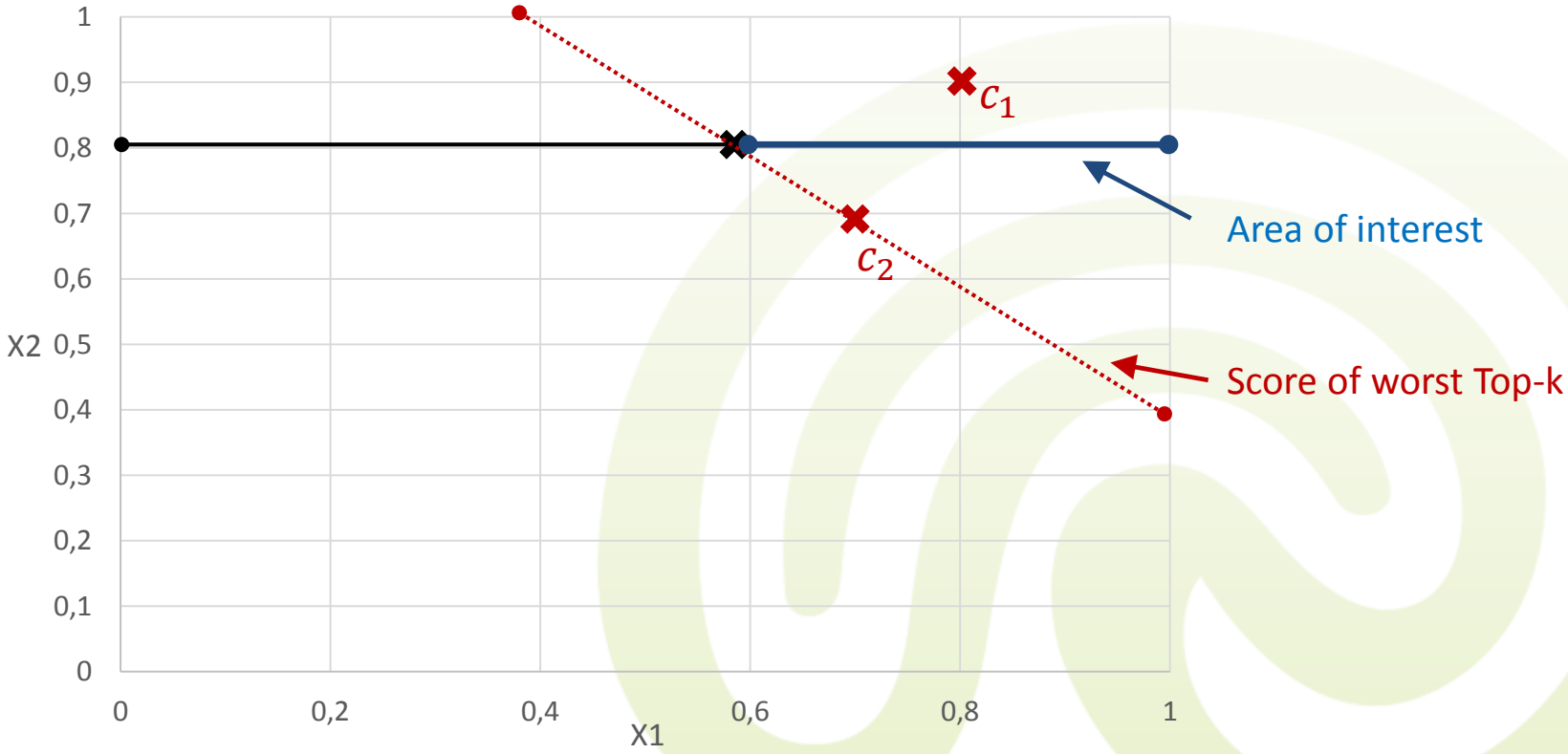
- Optimal Safe Pruning is done
  - Crowdsource all remaining incompletes?
- Try to find best candidates first
  - New threshold score
  - Perform additional pruning

# TopCrowd Approach

- Probabilistic Ranking of Incomplete Objects
  - Highest upper bound first?
    - (?,?,?,0.1) has upper bound: (1,1,1,0.1)
    - Unknowns' scores are highly overestimated

# TopCrowd Approach

- Probabilistic Ranking of Incomplete Objects
  - Estimate probability of unknown being in space above threshold score

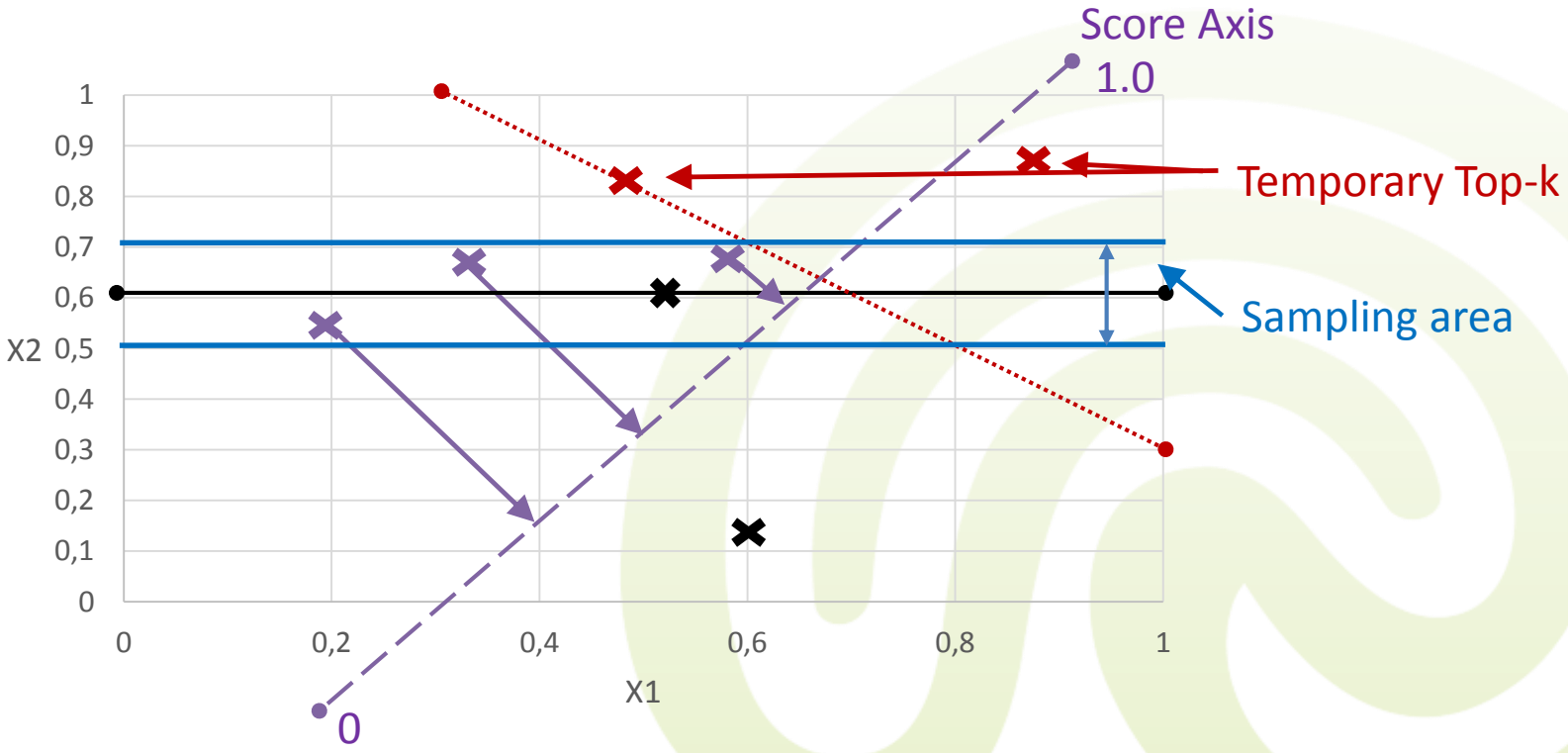


# TopCrowd Approach

- Estimating Probability
  - Probability of Object in Area of Space
  - Typical solution: Kernel Density Estimation
    - Basically "Smoothed Histogram"
    - Using complete objects as sample
  - Problem:
    - Sampling of distribution infeasible for high dimensions
    - 100 samples per Dimensions -> $100^D$ samples

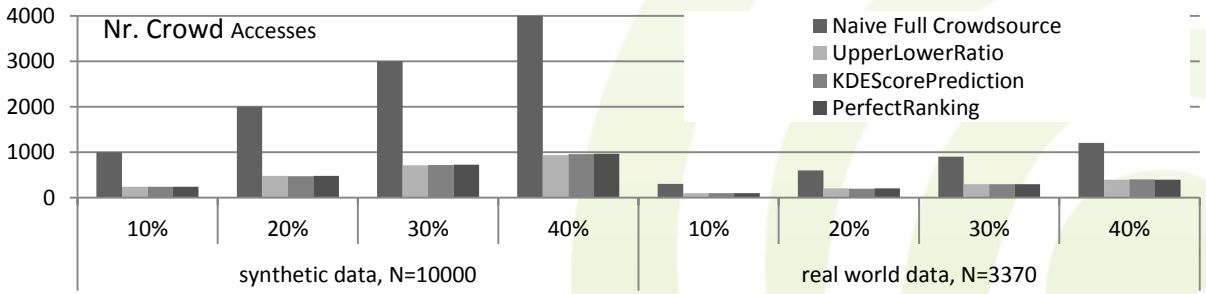
# TopCrowd Approach

- Solution:
  - Projection on score space
  - Then one dimensional estimation



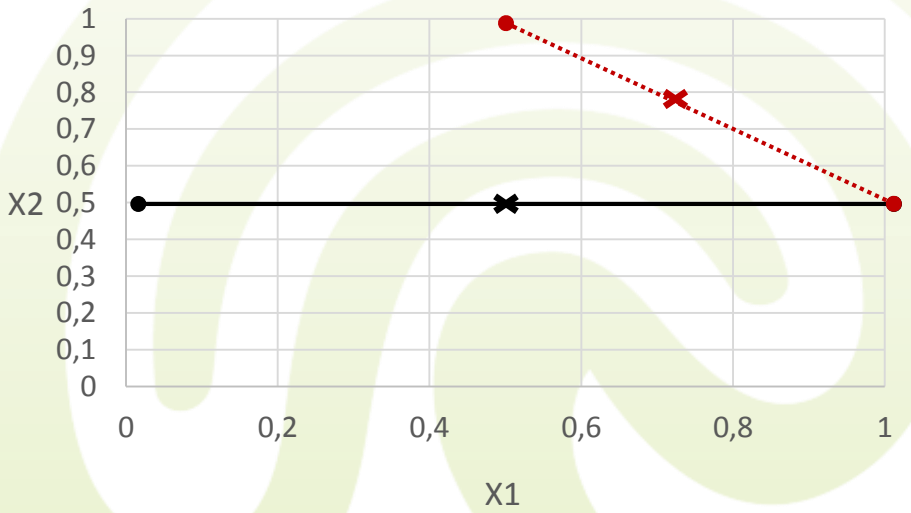
# TopCrowd Approach

- So far:
  - Optimal safe pruning
  - Ranking of promising candidates for crowdsourcing
- Performance:
  - Selection strategy has no effect



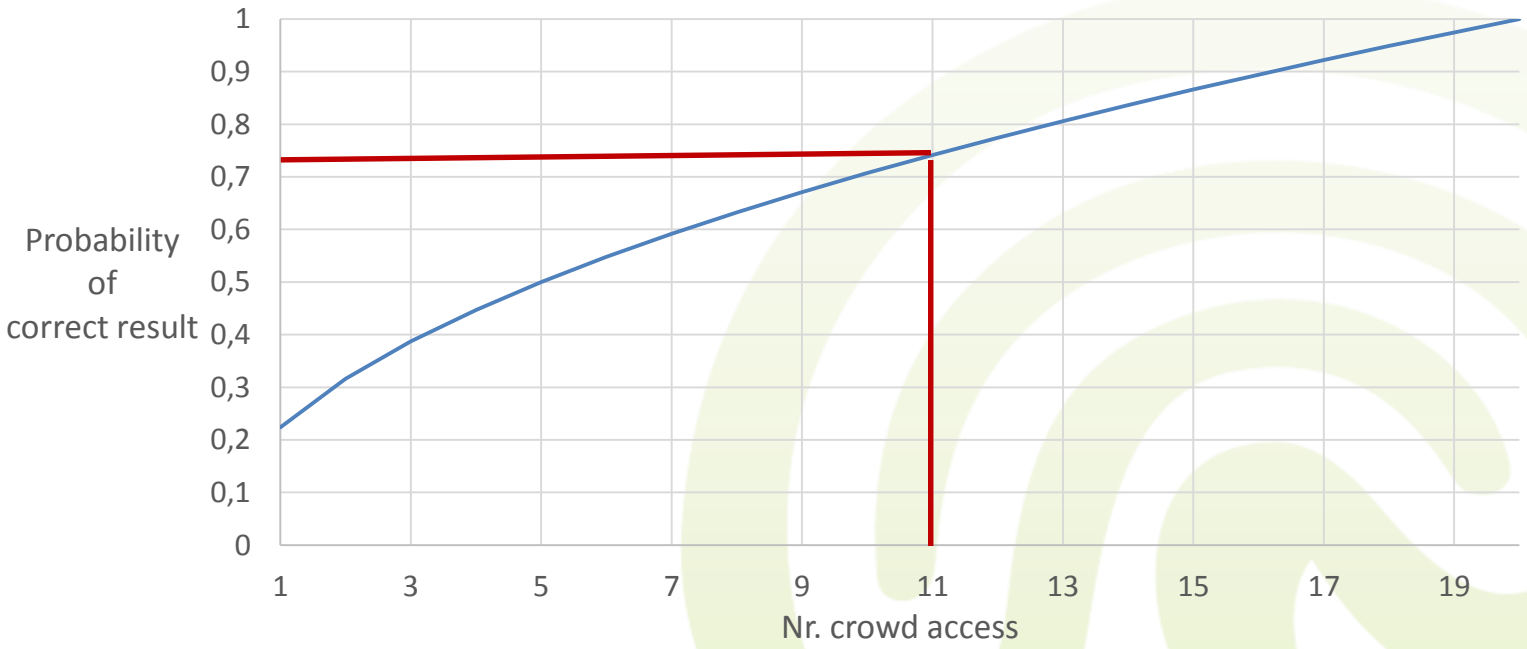
# TopCrowd

- Probabilistic Approach
  - Safe pruning is too restrictive!
    - We keep items with very low probability
  - Allow for uncertainties
    - But in a controlled way



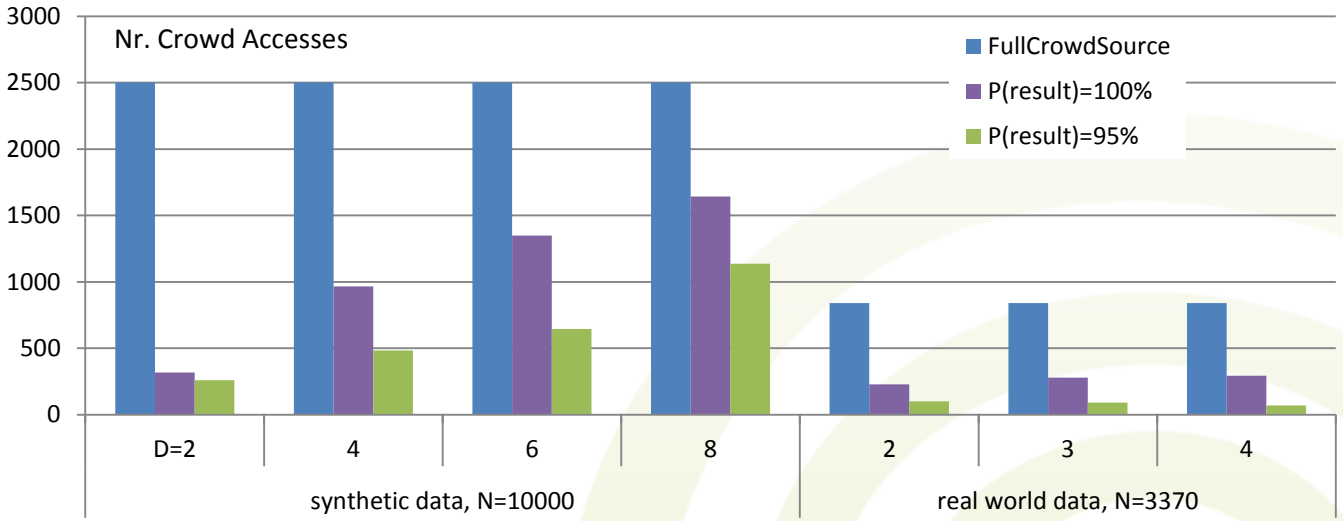
# Where to stop?

- Crowd access cost control
  - Balance costs
    - Financial, time (delay), result quality



# Results

- Performance of the Algorithm




$missing\_rate \in (10\%, 20\%, 30\%, 40\%)$ and $k \in (10,20,40)$

# Discussion

Questions?