Top-\(k\) Querying of Incomplete Data under Order Constraints

Antoine Amarilli\(^1\) \hspace{1em} Yael Amsterdamer\(^2\)
Tova Milo\(^2\) \hspace{1em} Pierre Senellart\(^{1,3}\)

\(^1\)Télécom ParisTech, Paris, France
\(^2\)Tel Aviv University, Tel Aviv, Israel
\(^3\)National University of Singapore

April 20th, 2015
Introduction

Taxonomy of items for a store

- **Products**
  - **Electronics**
    - TVs
    - Cell Phones
  - **Clothing**
    - Shoes
  - **Sports**
    - Watches
    - Diving Gear
  - **Smartphones**
  - **Wearable Devices**
    - Diving Watches
Introduction

Taxonomy of items for a store with categories.
Introduction

Taxonomy of items for a store with categories.
Ask the crowd to classify items
Introduction

Taxonomy of items for a store with categories.
Ask the crowd to classify items

- Products
  - Electronics
    - TVs
    - Smartphones
  - Clothing
    - Cell Phones
    - Wearable Devices
  - Sports
    - Shoes
    - Watches
    - Diving Gear
    - Diving Watches
Taxonomy of items for a store with *categories*. Ask the crowd to classify items.
Taxonomy of items for a store with *categories*. Ask the crowd to classify items.
Introduction

Taxonomy of items for a store with categories. Ask the crowd to classify items

- Products
  - Electronics
    - TVs
    - Cell Phones
  - Clothing
    - Shoes
  - Sports
    - Watches
    - Diving Gear
  - Wearable Devices
    - Smartphones
    - Diving Watches
Introduction

Taxonomy of items for a store with categories. Ask the crowd to classify items.
Introduction

Taxonomy of items for a store with categories. Ask the crowd to classify items
Introduction

Taxonomy of items for a store with categories. Ask the crowd to classify items
Introduction

Taxonomy of items for a store with categories. Ask the crowd to classify items.

- Products
  - Electronics
    - TVs
    - Cell Phones
  - Clothing
    - Shoes
  - Sports
    - Watches
    - Diving Gear
      - Diving Watches

Monotonicity: compatibility increases as we go up.

Best categories?

Naive answer...

Clever answer...
Introduction

Taxonomy of items for a store with categories.
Ask the crowd to classify items

- Products
  - Electronics
    - TVs
    - Cell Phones
    - Smartphones
  - Clothing
    - Shoes
    - Wearable Devices
  - Sports
    - Watches
    - Diving Gear
    - Diving Watches
Monotonicity: compatibility increases as we go up.
Monotonicity: compatibility increases as we go up.
Best categories?
**Monotonicity:** compatibility increases as we go up.

Best categories? **Naive** answer...
Monotonicity: compatibility increases as we go up.
Best categories? Naive answer...
Monotonicity: compatibility increases as we go up.
Best categories? **Naive** answer... **Clever** answer...
Monotonicity: compatibility increases as we go up. Best categories? Naive answer... Clever answer...
Introduction

Products

- Electronics
  - TVs
  - Cell Phones
    - Smartphones
- Clothing
  - Shoes
    - Wearable Devices
- Sports
  - Watches
  - Diving Gear
    - Diving Watches

Monotonicity: compatibility increases as we go up.
Best categories? Naive answer... Clever answer...
Monotonicity: compatibility increases as we go up.
Best categories? Naive answer... Clever answer...
Monotonicity: compatibility increases as we go up.
Best categories? Naive answer... Clever answer...
Problem statement

- **Taxonomy:**
  - Partial order, i.e., directed acyclic graph
  - Some end categories distinguished

- **Compatibility values:**
  - To simplify, assume $0 \leq \bullet \leq 1$
  - Monotonicity with respect to the taxonomy
  - Some values known, other unknown
Problem statement

- **Taxonomy:**
  - Partial order, i.e., directed acyclic graph
  - Some end categories distinguished

- **Compatibility values:**
  - To simplify, assume $0 \leq \bullet \leq 1$
  - Monotonicity with respect to the taxonomy
  - Some values known, other unknown

→ How to complete the missing values?
→ What are the top-$k$ and their expected values?
→ What is our confidence in the answer?
# Table of contents

1. Introduction
2. Approach
3. Complexity results
4. Conclusion
Admissible polytope

- Each **unknown value** has one variable
- Consider the **space** of all possible assignments
- It is a **polytope** (linear constraints)
Admissible polytope

- Each unknown value has one variable
- Consider the space of all possible assignments
- It is a polytope (linear constraints)

Example:
- \(0 \leq x \leq .8, \ .2 \leq y \leq 1\)
Admissible polytope

- Each **unknown value** has one variable
- Consider the **space** of all possible assignments
- It is a **polytope** (linear constraints)

**Example:**
- \( 0 \leq x \leq .8, \ .2 \leq y \leq 1 \)
- \( y \leq x \)
Probabilistic formalization

- Consider the **admissible polytope**
- Take the **uniform distribution** on it
  (Intuitively, all possible assignments are **equiprobable**)
Probabilistic formalization

- Consider the **admissible polytope**
- Take the **uniform distribution** on it
  (Intuitively, all possible assignments are **equiprobable**)
→ **What is the average value of each variable?**
Probabilistic formalization

- Consider the **admissible polytope**
- Take the **uniform distribution** on it
  (Intuitively, all possible assignments are **equiprobable**)
  → What is the **average value** of each variable?
  (Possible extensions: variance, marginal distribution...)

**Introduction**

**Approach**

**Complexity results**

**Conclusion**
Easy case: total order

\[ 0 \leq \bullet \leq \bullet \leq 0.3 \leq \bullet \leq 1 \]
Easy case: total order

\[ 0 \leq \bullet \leq \bullet \leq 0.3 \leq \bullet \leq 1 \]

- How to complete this? Any ideas? ...
Easy case: total order

\[ 0 \leq \bullet \leq \bullet \leq 0.3 \leq \bullet \leq 1 \]

- How to complete this? Any ideas? ...

→ Linear interpolation!
Easy case: total order

\[ 0 \leq .1 \leq .2 \leq .3 \leq .65 \leq 1 \]

- How to complete this? Any ideas? ...

→ Linear interpolation!
Easy case: total order

0 \leq 0.1 \leq 0.2 \leq 0.3 \leq 0.65 \leq 1

- How to complete this? Any ideas? ...

→ Linear interpolation!

- (For marginal distribution: order statistics, Beta distribution)
General case

- Consider the **taxonomy**: partial order
General case

- Consider the **taxonomy**: partial order
- Consider all possible **total orders**
General case

- Consider the taxonomy: partial order
- Consider all possible total orders
  (Ties can be made negligible)
General case

- Consider the **taxonomy**: partial order
- Consider all possible **total orders**
  (Ties can be made **negligible**)
- Solve each **total order** as before
General case

- Consider the **taxonomy**: partial order
- Consider all possible **total orders**
  (Ties can be made **negligible**)
- Solve each **total order** as before
- Take the **weighted average** of the orders
General case

- Consider the **taxonomy**: partial order
- Consider all possible **total orders**
  (Ties can be made **negligible**)
- Solve each **total order** as before
- Take the **weighted average** of the orders
- Total order weight: **probability** of this order
Consider the **taxonomy**: partial order
Consider all possible **total orders**
(Ties can be made **negligible**)
Solve each **total order** as before
Take the **weighted average** of the orders
Total order weight: **probability** of this order
→ Gives the **average** for the **actual taxonomy**!
Example

Possibility 1:
Expected values:
\( x = 1 \), \( y = 2 \), \( z = 65 \)
Probability:
Volume of \( x \cdot y \cdot z \) is 3 times volume of \( 3 \cdot z \)

Possibility 2:
Expected values of \( y \):
\( 2 \) and \( 533 \)
Normalized probabilities:
3 and 7
Final result:
\( y \) has expected value \( \frac{43}{16} \)
Possibility 1: $0 \leq x \leq y \leq y' \leq z \leq 1$
**Possibility 1:** \(0 \leq x \leq y \leq y' \leq z \leq 1\)

\[\text{Expected values: } x = .1, \ y = .2, \ z = .65\]
Example

- **Possibility 1**: $0 \leq x \leq y \leq y' \leq z \leq 1$
  - Expected values: $x = .1$, $y = .2$, $z = .65$
  - Probability:
Example

- **Possibility 1:** $0 \leq x \leq y \leq y' \leq z \leq 1$
  - Expected values: $x = .1$, $y = .2$, $z = .65$
  - Probability:
    - Volume of $0 \leq x \leq y \leq .3$
    - times volume of $.3 \leq z \leq 1$
Example

- **Possibility 1:** $0 \leq x \leq y \leq y' \leq z \leq 1$
  - Expected values: $x = .1$, $y = .2$, $z = .65$
  - Probability:
    - Volume of $0 \leq x \leq y \leq .3$
    - times volume of $.3 \leq z \leq 1$
    - $\frac{.3^2}{2!}$ and $\frac{1-.3}{1!}$
**Example**

- **Possibility 1:** $0 \leq x \leq y \leq y' \leq z \leq 1$
  
  → Expected values: $x = .1$, $y = .2$, $z = .65$

- **Probability:**
  
  → Volume of $0 \leq x \leq y \leq .3$
  
  times volume of $0.3 \leq z \leq 1$

  → $\frac{.3^2}{2!}$ and $\frac{1-.3}{1!}$

- **Possibility 2:** $0 \leq x \leq y' \leq y \leq z \leq 1$
Example

- **Possibility 1:** $0 \leq x \leq y \leq y' \leq z \leq 1$
  - Expected values: $x = .1$, $y = .2$, $z = .65$
  - Probability:
    - Volume of $0 \leq x \leq y \leq .3$
      - times volume of $0.3 \leq z \leq 1$
    - $\frac{.3^2}{2!}$ and $\frac{1-.3}{1!}$

- **Possibility 2:** $0 \leq x \leq y' \leq y \leq z \leq 1$
  - Expected values of $y$: .2 and .533
Example

- **Possibility 1:** $0 \leq x \leq y \leq y' \leq z \leq 1$
  
  → Expected values: $x = .1$, $y = .2$, $z = .65$

  - Probability:
    
    → Volume of $0 \leq x \leq y \leq .3$
      
      times volume of $.3 \leq z \leq 1$

    → $\frac{3^2}{2!}$ and $\frac{1-.3}{1!}$

- **Possibility 2:** $0 \leq x \leq y' \leq y \leq z \leq 1$

  → Expected values of $y$: .2 and .533

  → Normalized probabilities: .3 and .7
Example

- **Possibility 1:** $0 \leq x \leq y \leq y' \leq z \leq 1$
  - Expected values: $x = .1, y = .2, z = .65$
  - Probability:
    - Volume of $0 \leq x \leq y \leq .3$ times volume of $.3 \leq z \leq 1$
    - $\frac{.3^2}{2!}$ and $\frac{1-.3}{1!}$

- **Possibility 2:** $0 \leq x \leq y' \leq y \leq z \leq 1$
  - Expected values of $y$: .2 and .533
  - Normalized probabilities: .3 and .7
  - Final result: $y$ has expected value $.43$
Table of contents

1 Introduction
2 Approach
3 Complexity results
4 Conclusion
Complexity of the brute-force algorithm

- **Complexity** of the previous algorithm:
  PTIME in the number of compatible total orders (aka. linear extensions)
- They can be enumerated in PTIME in their number
Complexity of the brute-force algorithm

- **Complexity** of the previous algorithm: PTIME in the number of compatible total orders (aka. linear extensions)
- They can be enumerated in PTIME in their number
- However there may be exponentially many
Complexity of the bruteforce algorithm

- **Complexity** of the previous algorithm: PTIME in the number of compatible total orders (aka. linear extensions)
- They can be enumerated in PTIME in their number
- However there may be exponentially many
- → Volume computation for convex polytopes is \#P-hard
Complexity of the bruteforce algorithm

- **Complexity** of the previous algorithm:
  PTIME in the number of compatible total orders (aka. *linear extensions*)
- They can be enumerated in PTIME in their number
- However there may be exponentially many
  → Volume computation for convex polytopes is \#P-hard
  → Can we show hardness of our problems?
Completeness results

- **Existing results** [Brightwell and Winkler, 1991]
  - Counting the number of linear extensions is \#P-hard
  - Expected rank computation is \#P-hard
Completeness results

- **Existing results** [Brightwell and Winkler, 1991]
  - Counting the number of linear extensions is \#P-hard
  - Expected rank computation is \#P-hard
  - Computing the expected value in our setting is \#P-hard
  - Connection between expected rank and value
Completeness results

- **Existing results** [Brightwell and Winkler, 1991]
  - Counting the number of linear extensions is #P-hard
  - Expected rank computation is #P-hard
  - Computing the expected value in our setting is #P-hard
    - Connection between expected rank and value
  - Computing the top-\(k\) is #P-hard even without values!
    - Binary search against known values to find expected value
    - Uses scheme for rational search [Papadimitriou, 1979]
Completeness results

- **Existing results** [Brightwell and Winkler, 1991]
  - Counting the number of linear extensions is \#P-hard
  - Expected rank computation is \#P-hard
- Computing the expected value in our setting is \#P-hard
  - Connection between expected rank and value
- Computing the top-\(k\) is \#P-hard even without values!
  - Binary search against known values to find expected value
  - Uses scheme for rational search [Papadimitriou, 1979]
- FP\#P-membership of our problems
  - Non-trivial as polytope volume computation is not in FP\#P!
Tractable cases

- Intractable for arbitrary taxonomies
- Are there tractable subcases?
Tractable cases

- Intractable for arbitrary taxonomies
- Are there tractable subcases?
- Common situation: taxonomy is a tree
  \[\rightarrow\] PTIME expected value computation
  (Compute the marginal distributions as piecewise polynomials)
Table of contents

1 Introduction

2 Approach

3 Complexity results

4 Conclusion
Conclusion

- **Formal definition** of top-
  \( k \) queries on incomplete data
- Also generalizes **linear interpolation** to partial orders
Conclusion

- **Formal definition** of top-$k$ queries on incomplete data
- Also generalizes **linear interpolation** to partial orders
- **Principled algorithm** for top-$k$
Conclusion

- **Formal definition** of top-$k$ queries on incomplete data
- Also generalizes **linear interpolation** to partial orders
- **Principled algorithm** for top-$k$
- **Hardness results** for these problems
Conclusion

- **Formal definition** of top-$k$ queries on incomplete data
- Also generalizes **linear interpolation** to partial orders
- **Principled algorithm** for top-$k$
- **Hardness results** for these problems
- **Tractable subcases** for tree-shaped taxonomies
Conclusion

- **Formal definition** of top-$k$ queries on incomplete data
- Also generalizes **linear interpolation** to partial orders
- **Principled algorithm** for top-$k$
- **Hardness results** for these problems
- **Tractable subcases** for tree-shaped taxonomies
- **Open questions:**
  - Is this the right definition?
  - Are there other tractable cases?
  - What about choosing the next queries?
Conclusion

- **Formal definition** of top-\(k\) queries on incomplete data
- Also generalizes **linear interpolation** to partial orders
- **Principled algorithm** for top-\(k\)
- **Hardness results** for these problems
- **Tractable subcases** for tree-shaped taxonomies
- **Open questions:**
  - Is this the **right definition**?
  - Are there **other tractable cases**?
  - What about choosing the next queries?

Thanks for your attention!

Smartwatch photo on slide 2: Bostwickenator, CC-BY-SA 3.0