Using Order Constraints in Crowd Data Sourcing

Antoine Amarilli[1,2], Yael Amsterdamer[3,4], Tova Milo[4], Pierre Senellart[1,2]

February 12th, 2018

[1]Télécom ParisTech

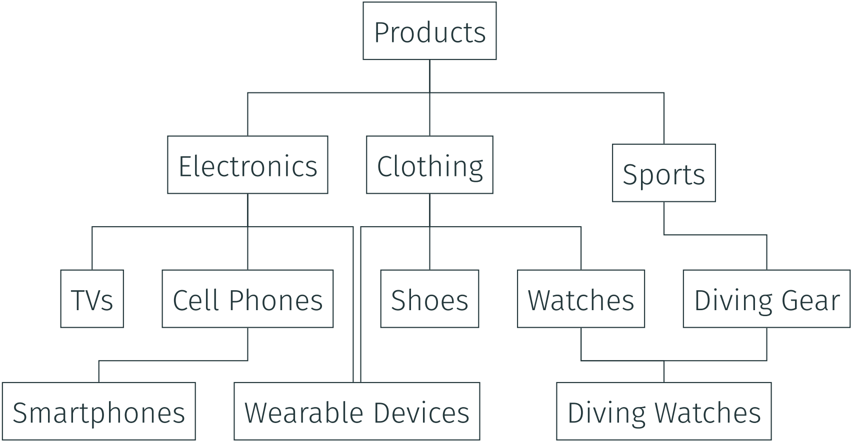
[2]École normale supérieure

[3]Bar Ilan University

[4]Tel Aviv University

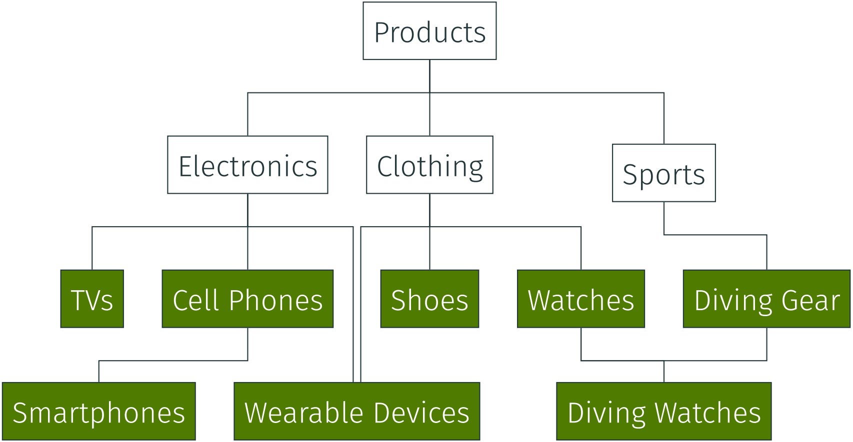
Introduction

Example 1: Classifying products



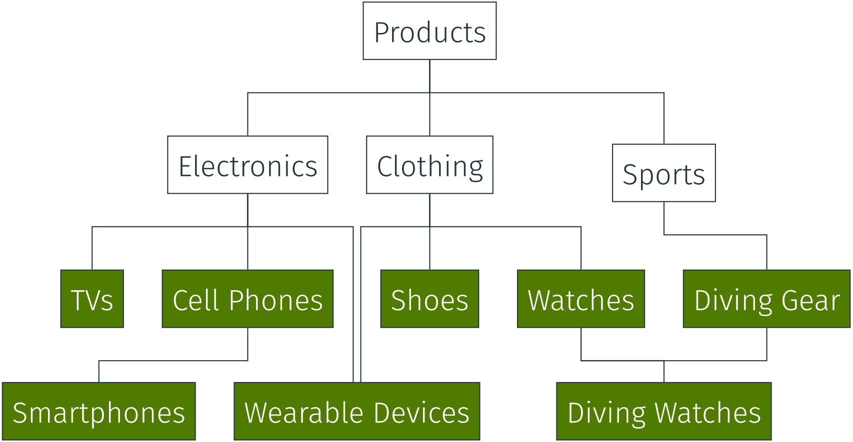
Taxonomy of items for a store

Example 1: Classifying products



Taxonomy of items for a store with categories.

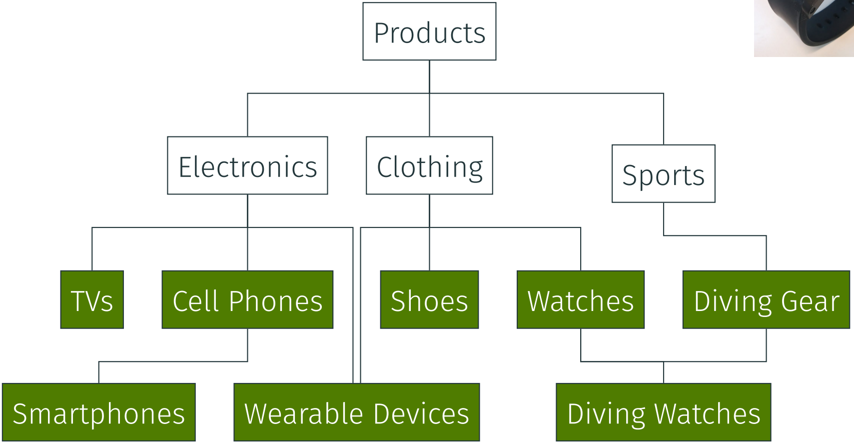
Example 1: Classifying products



Taxonomy of items for a store with categories.

Ask the crowd to classify items

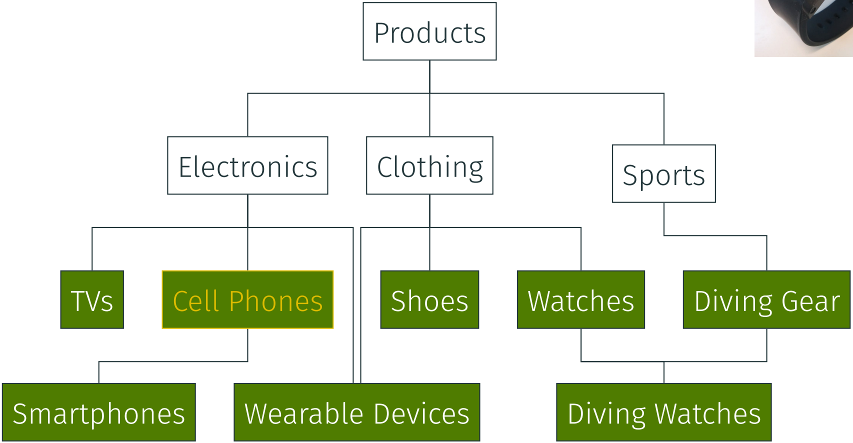
Example 1: Classifying products



Taxonomy of items for a store with categories.

Ask the crowd to classify items

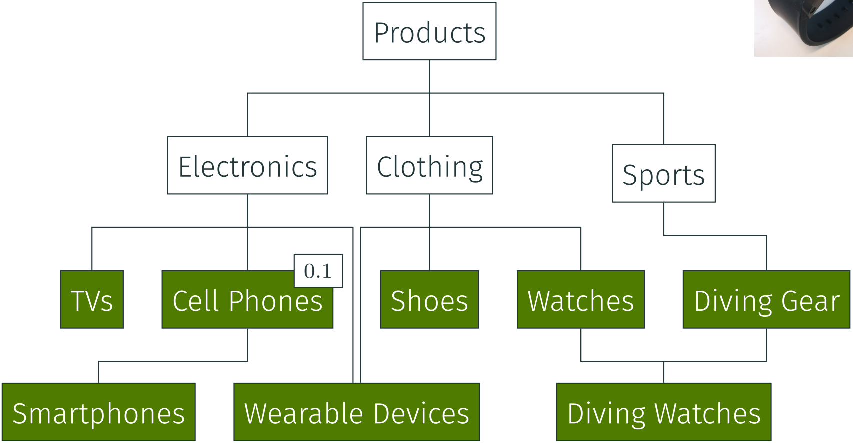
Example 1: Classifying products



Taxonomy of items for a store with categories.

Ask the crowd to classify items

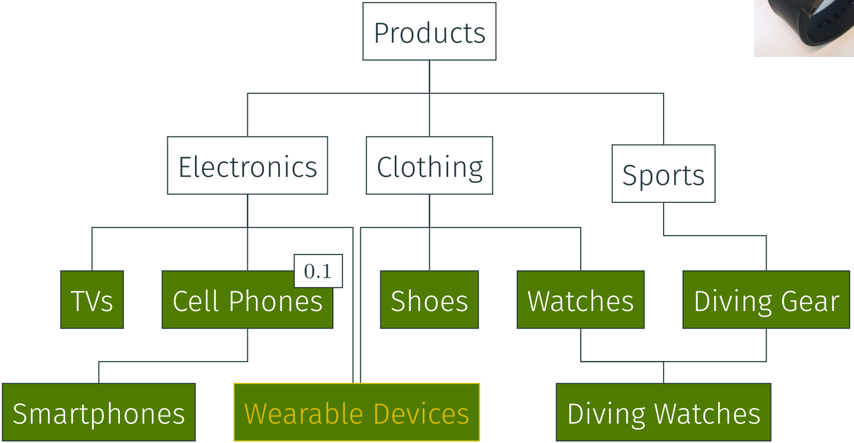
Example 1: Classifying products



Taxonomy of items for a store with categories.

Ask the crowd to classify items

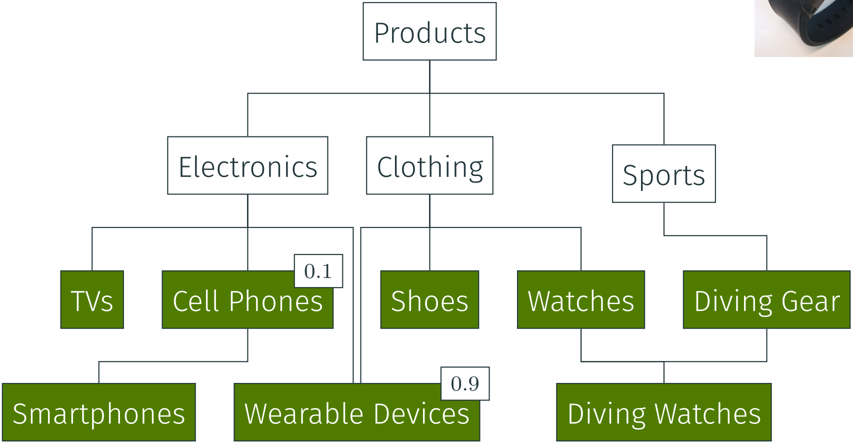
Example 1: Classifying products



Taxonomy of items for a store with categories.

Ask the crowd to classify items

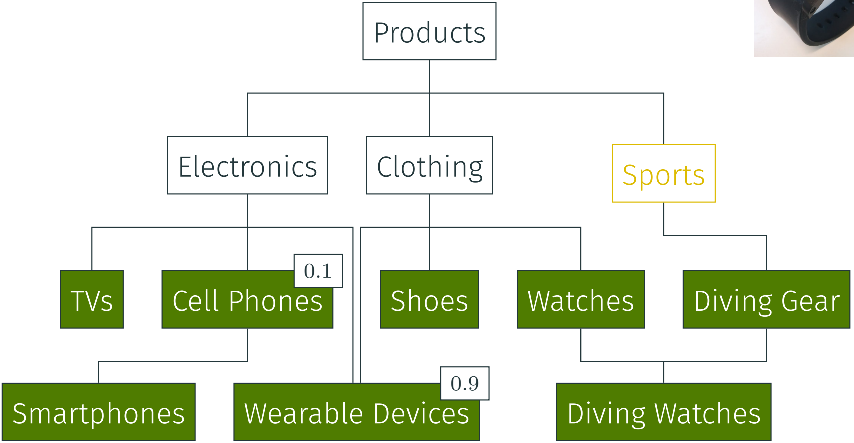
Example 1: Classifying products



Taxonomy of items for a store with categories.

Ask the crowd to classify items

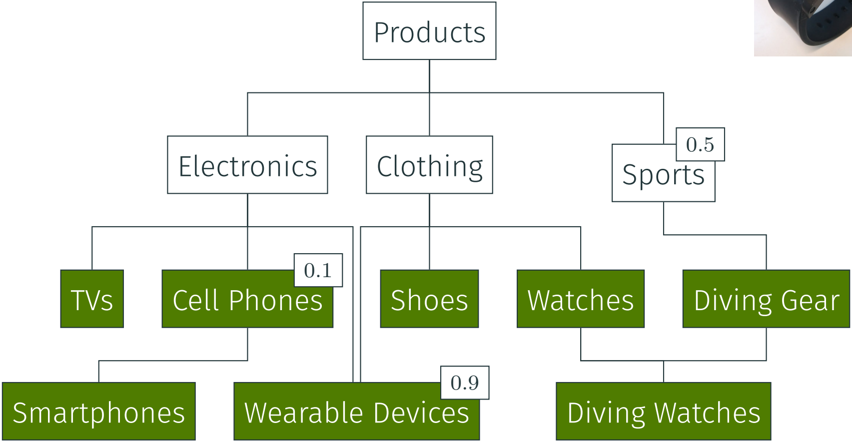
Example 1: Classifying products



Taxonomy of items for a store with categories.

Ask the crowd to classify items

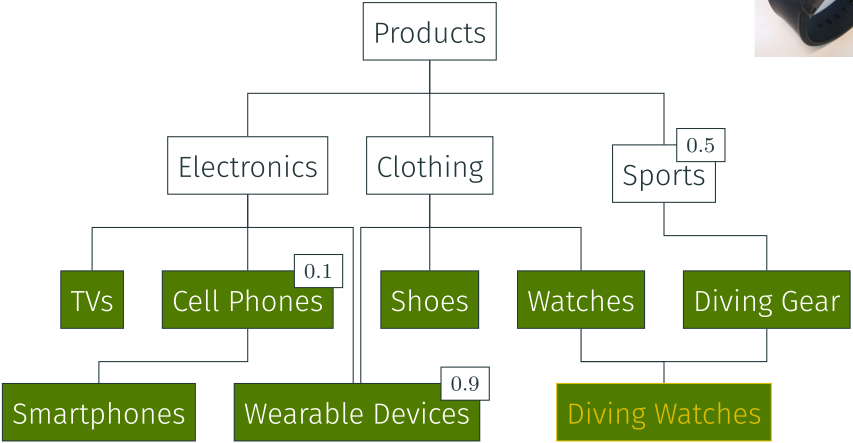
Example 1: Classifying products



Taxonomy of items for a store with categories.

Ask the crowd to classify items

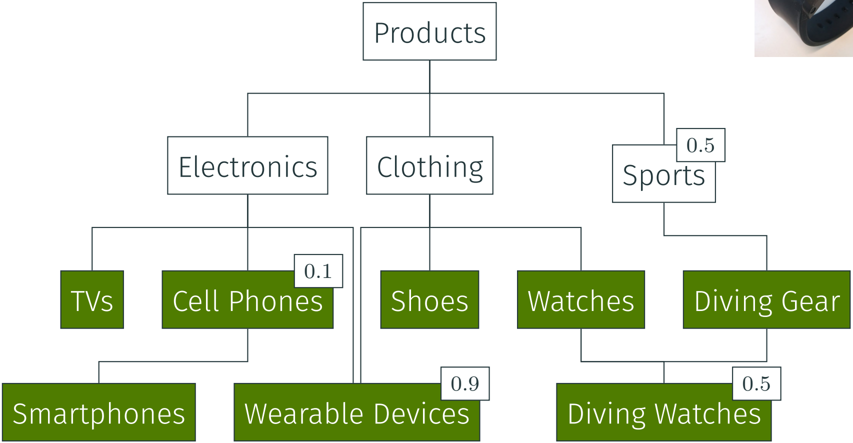
Example 1: Classifying products



Taxonomy of items for a store with categories.

Ask the crowd to classify items

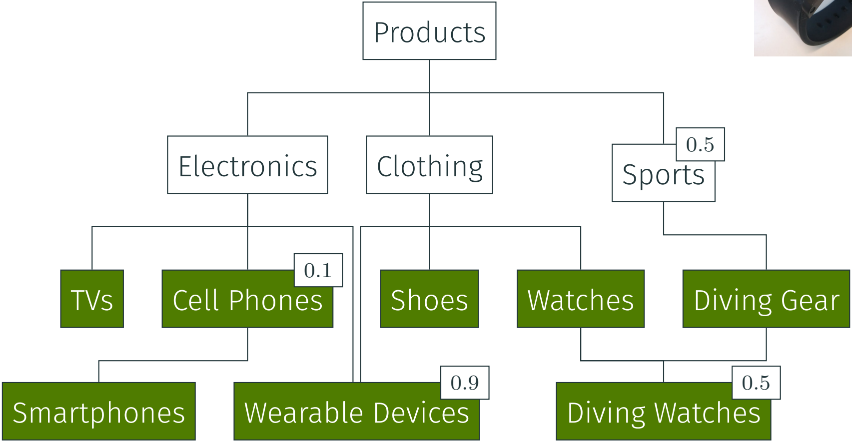
Example 1: Classifying products



Taxonomy of items for a store with categories.

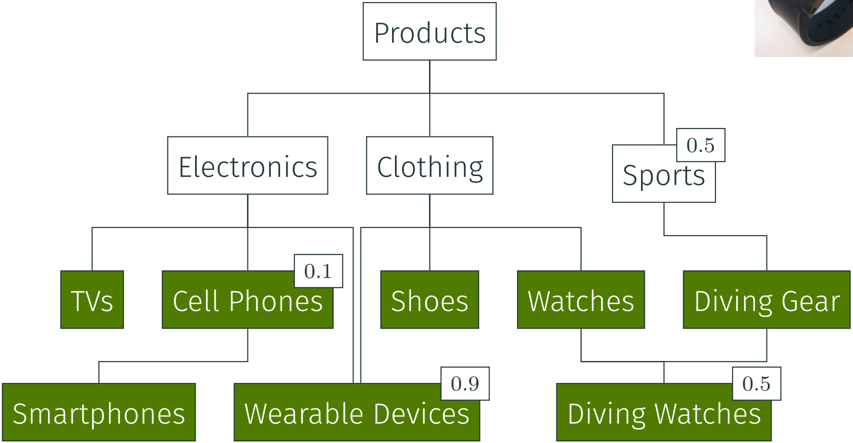
Ask the crowd to classify items

Example 1: Classifying products



Monotonicity: compatibility increases as we go up.

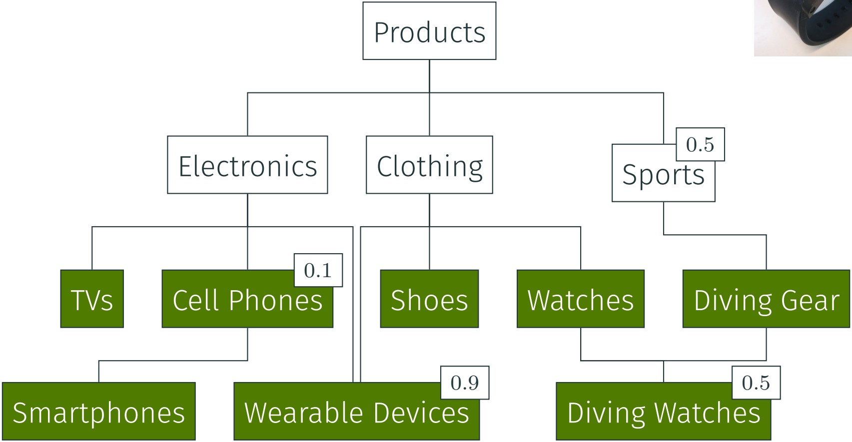
Example 1: Classifying products



Monotonicity: compatibility increases as we go up.

Best categories?

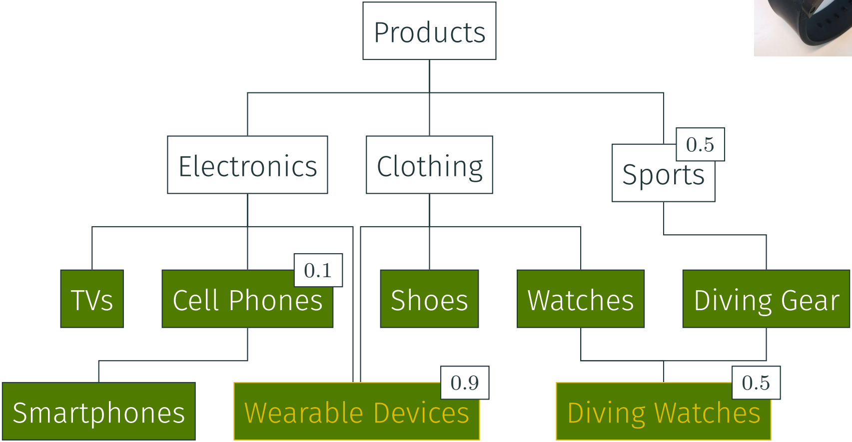
Example 1: Classifying products



Monotonicity: compatibility increases as we go up.

Best categories? Naive answer...

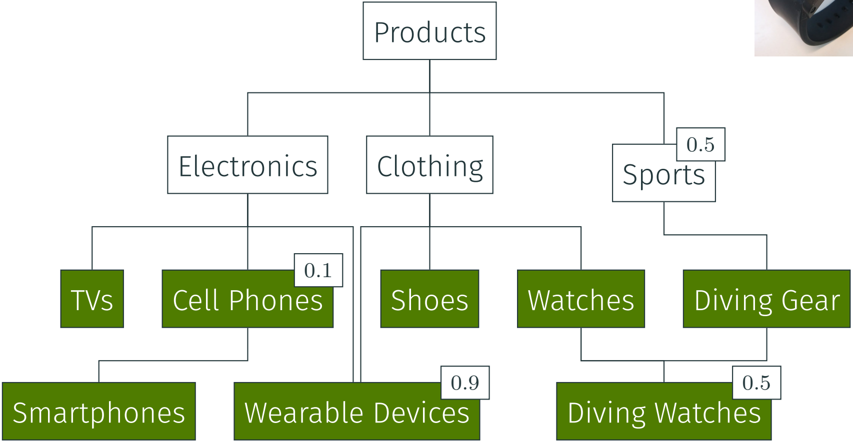
Example 1: Classifying products



Monotonicity: compatibility increases as we go up.

Best categories? Naive answer...

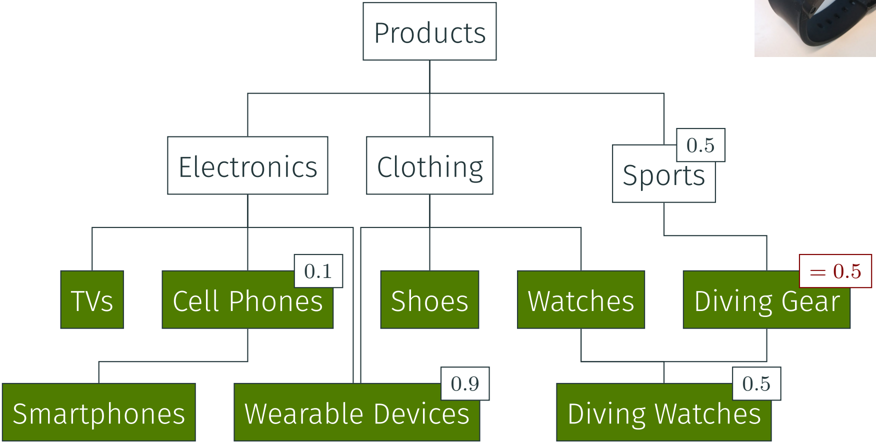
Example 1: Classifying products



Monotonicity: compatibility increases as we go up.

Best categories? Naive answer... Clever answer...

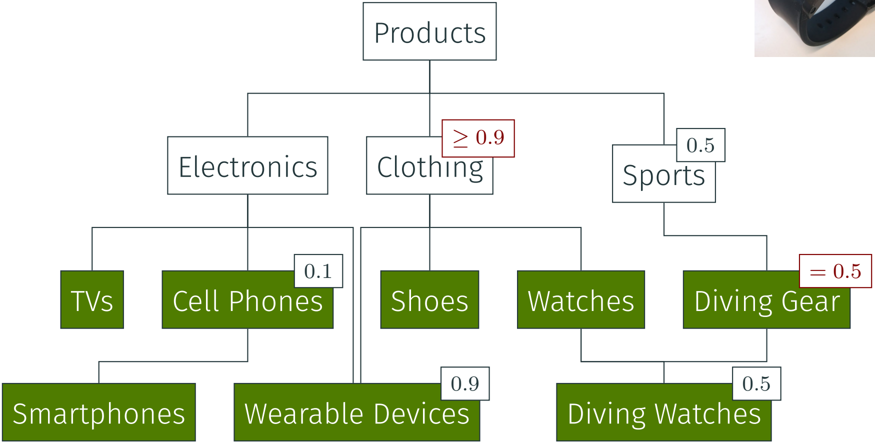
Example 1: Classifying products



Monotonicity: compatibility increases as we go up.

Best categories? Naive answer… Clever answer…

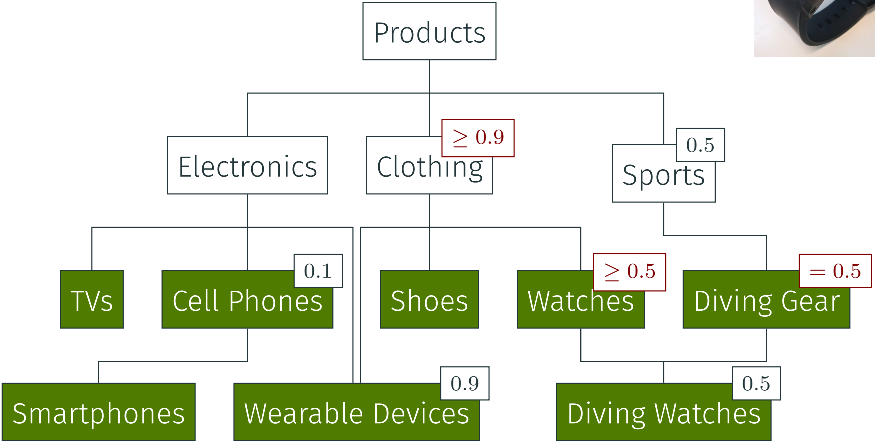
Example 1: Classifying products



Monotonicity: compatibility increases as we go up.

Best categories? Naive answer... Clever answer...

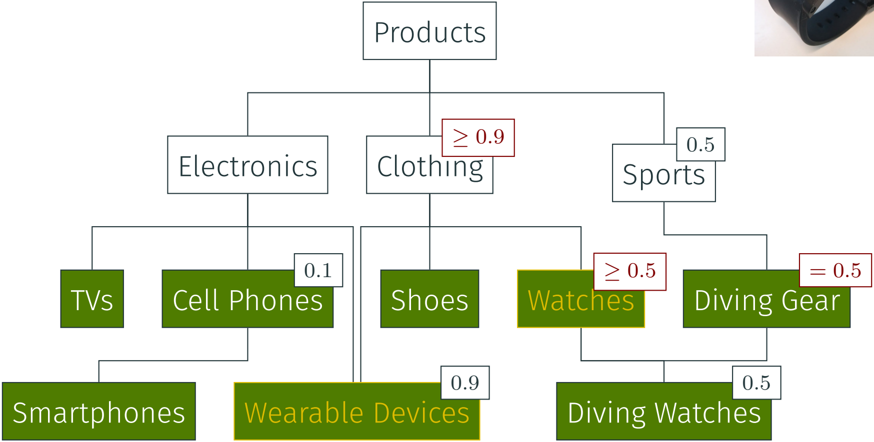
Example 1: Classifying products



Monotonicity: compatibility increases as we go up.

Best categories? Naive answer... Clever answer...

Example 1: Classifying products

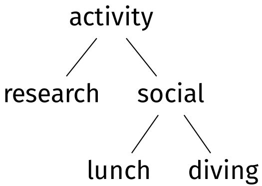


Monotonicity: compatibility increases as we go up.

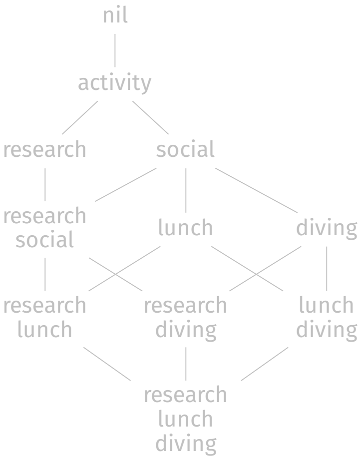
Best categories? Naive answer... Clever answer...

Example 2: Finding popular activity combinations

Taxonomy of activities

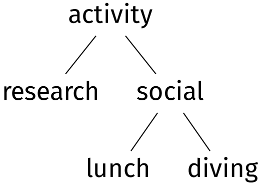


Taxonomy of activity combinations

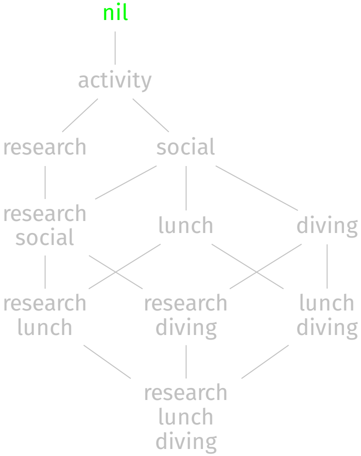


Example 2: Finding popular activity combinations

Taxonomy of activities

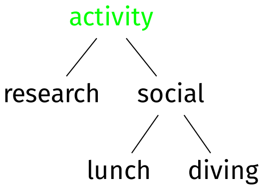


Taxonomy of activity combinations

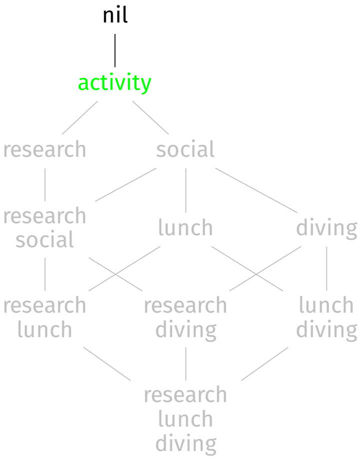


Example 2: Finding popular activity combinations

Taxonomy of activities

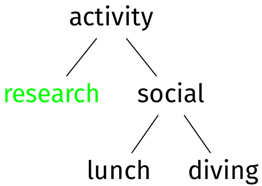


Taxonomy of activity combinations

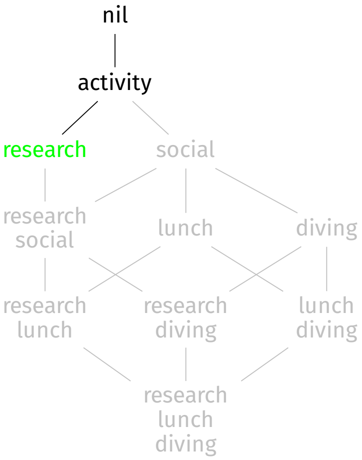


Example 2: Finding popular activity combinations

Taxonomy of activities

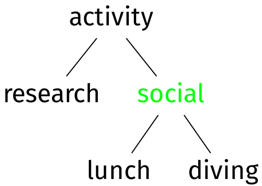


Taxonomy of activity combinations

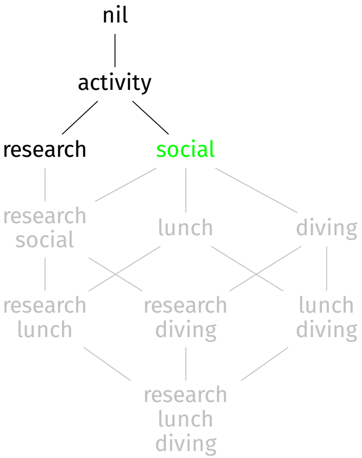


Example 2: Finding popular activity combinations

Taxonomy of activities

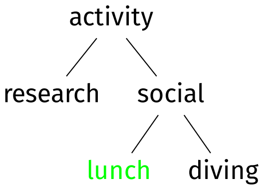


Taxonomy of activity combinations

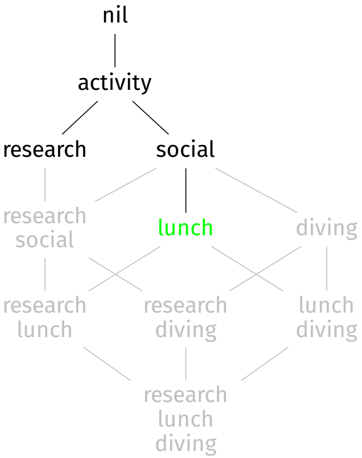


Example 2: Finding popular activity combinations

Taxonomy of activities

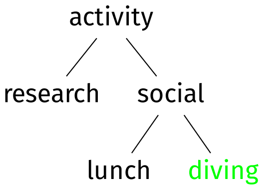


Taxonomy of activity combinations

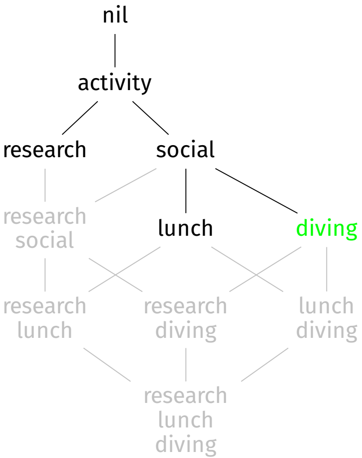


Example 2: Finding popular activity combinations

Taxonomy of activities

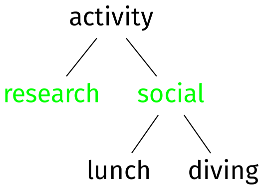


Taxonomy of activity combinations

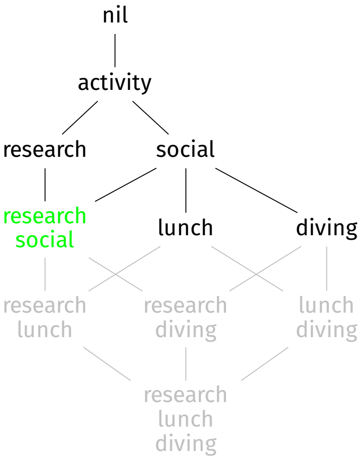


Example 2: Finding popular activity combinations

Taxonomy of activities

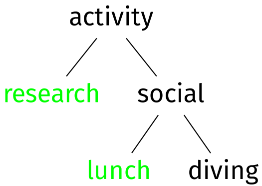


Taxonomy of activity combinations

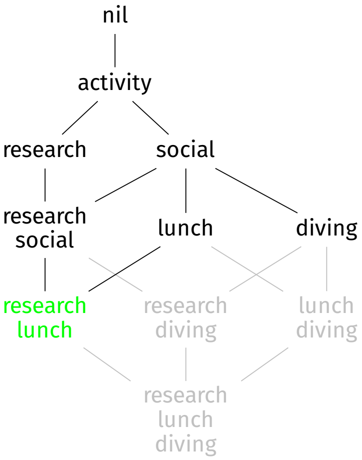


Example 2: Finding popular activity combinations

Taxonomy of activities

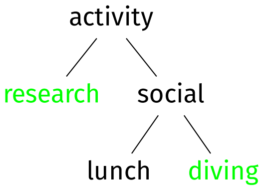


Taxonomy of activity combinations

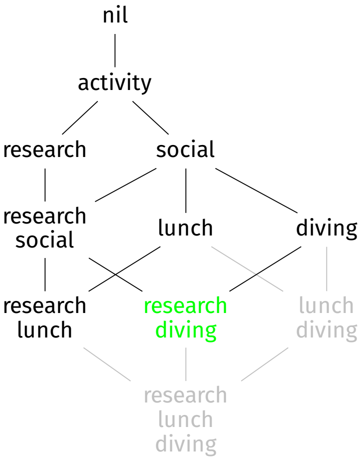


Example 2: Finding popular activity combinations

Taxonomy of activities

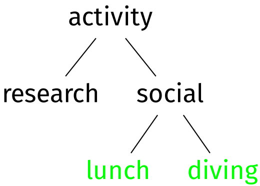


Taxonomy of activity combinations

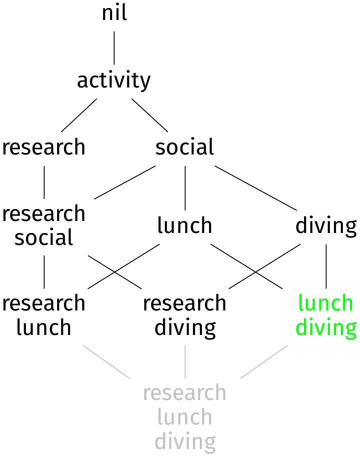


Example 2: Finding popular activity combinations

Taxonomy of activities

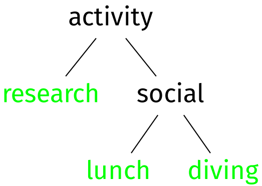


Taxonomy of activity combinations

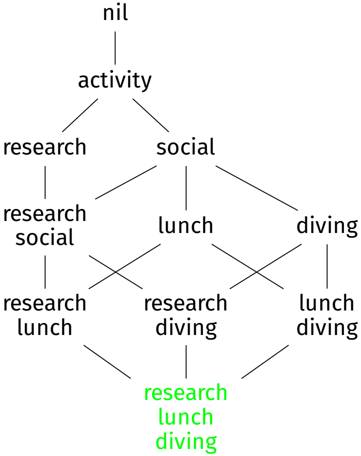


Example 2: Finding popular activity combinations

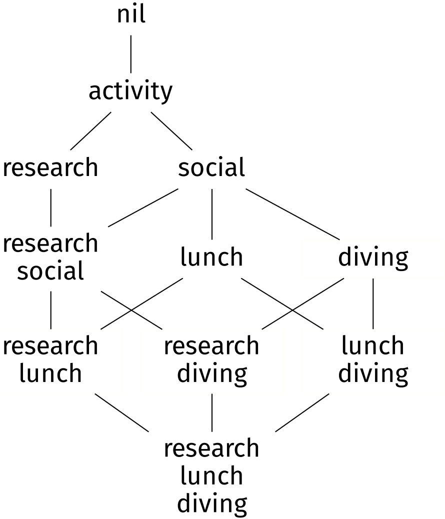
Taxonomy of activities



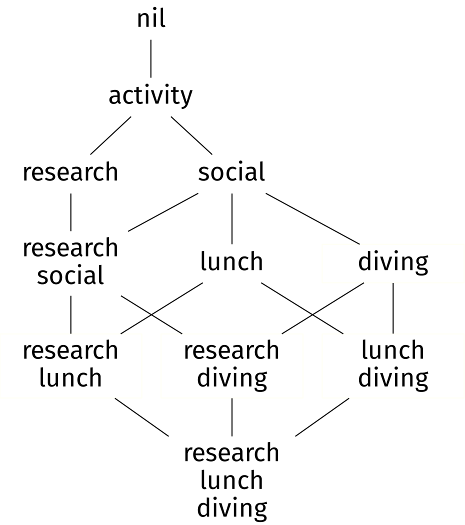
Taxonomy of activity combinations



Example 2: Finding popular activity combinations

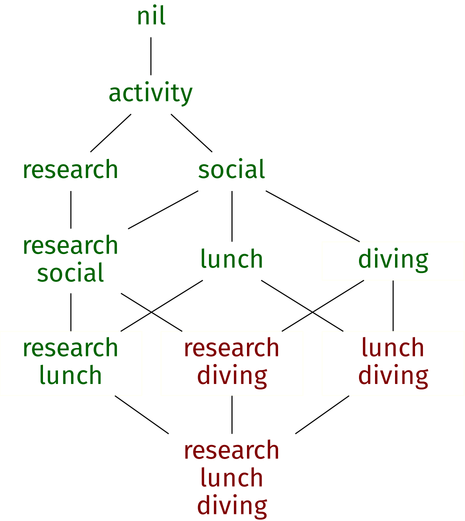


Example 2: Finding popular activity combinations



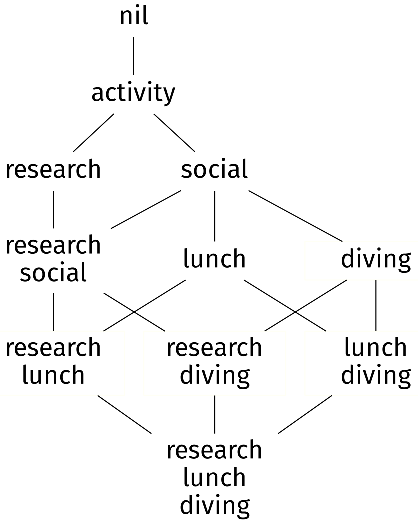
- More specific combinations are less frequent

Example 2: Finding popular activity combinations



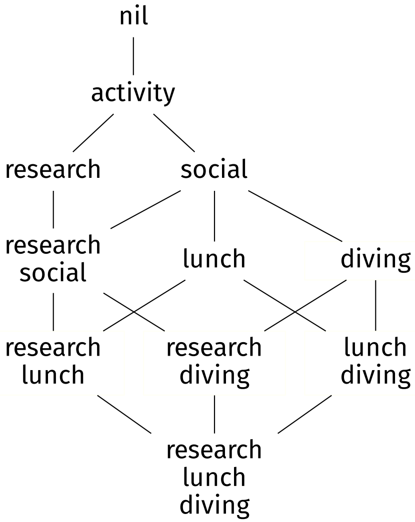
- More specific combinations are less frequent

Example 2: Finding popular activity combinations



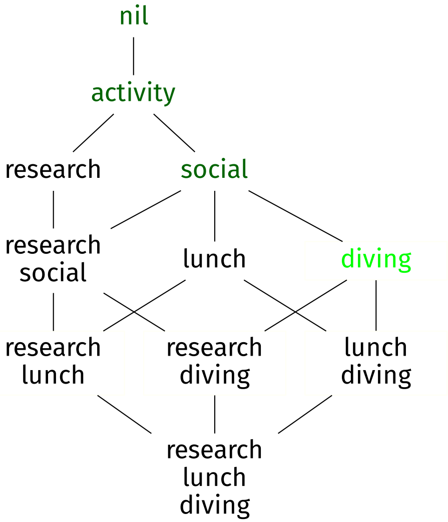
- More specific combinations are less frequent
- Ask crowd questions:

Example 2: Finding popular activity combinations



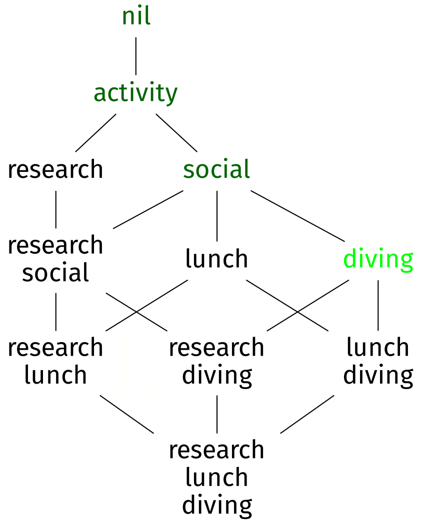
- More specific combinations are less frequent
- Ask crowd questions:
- Is {diving} frequent?

Example 2: Finding popular activity combinations



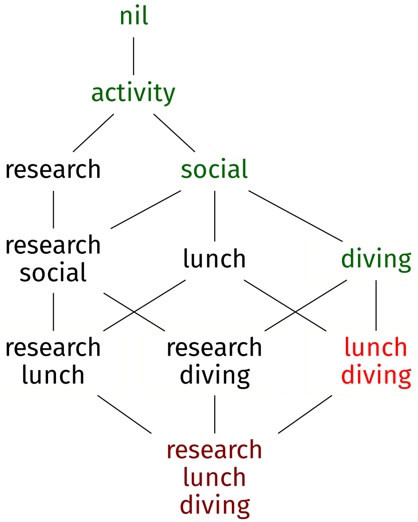
- More specific combinations are less frequent
- Ask crowd questions:
- Is {diving} frequent?
  - ⇒ Yes!

Example 2: Finding popular activity combinations



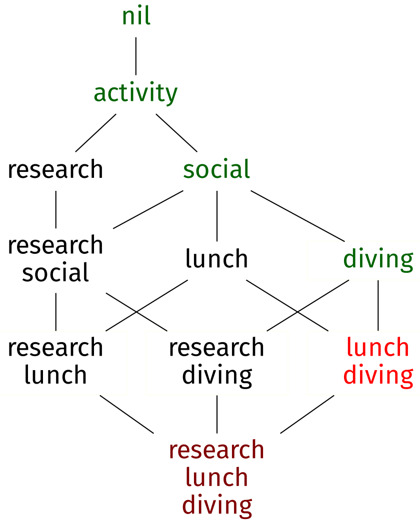
- More specific combinations are less frequent
- Ask crowd questions:
- Is {diving} frequent?
  - ⇒ Yes!
- {lunch, diving}?

Example 2: Finding popular activity combinations



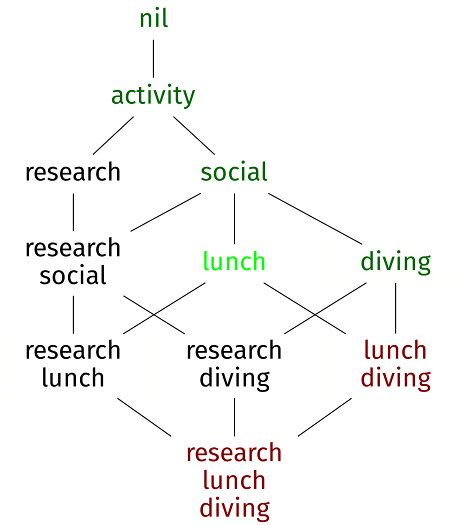
- More specific combinations are less frequent
- Ask crowd questions:
- Is {diving} frequent?
  - $\Rightarrow$ Yes!
- {lunch, diving}?
  - $\Rightarrow$ No!

Example 2: Finding popular activity combinations



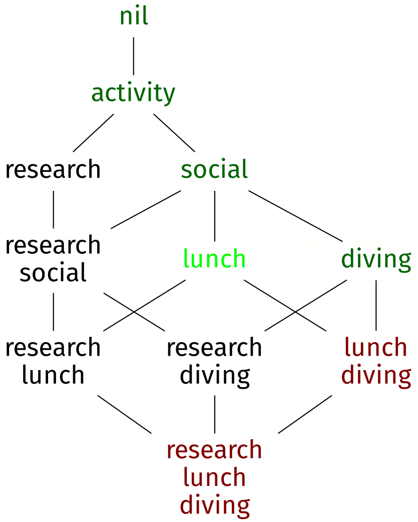
- More specific combinations are less frequent
- Ask crowd questions:
- Is {diving} frequent?
  - $\Rightarrow$ Yes!
- {lunch, diving}?
  - $\Rightarrow$ No!
- {lunch}?

Example 2: Finding popular activity combinations



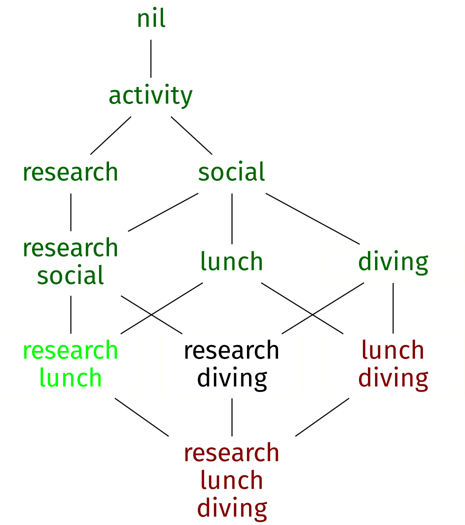
- More specific combinations are less frequent
- Ask crowd questions:
- Is {diving} frequent?
  - $\Rightarrow$ Yes!
- {lunch, diving}?
  - $\Rightarrow$ No!
- {lunch}?
  - $\Rightarrow$ Yes!

Example 2: Finding popular activity combinations



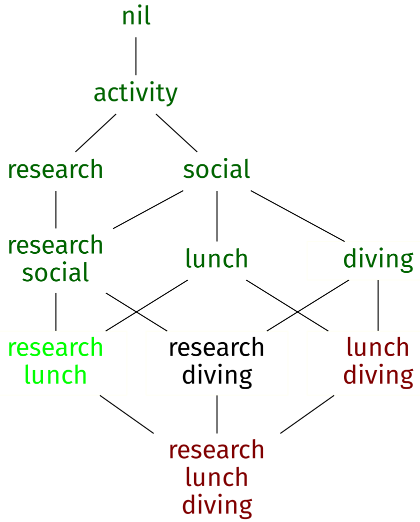
- More specific combinations are less frequent
- Ask crowd questions:
- Is {diving} frequent?
  - ⇒ Yes!
- {lunch, diving}?
  - ⇒ No!
- {lunch}?
  - ⇒ Yes!
- {research, lunch}?

Example 2: Finding popular activity combinations



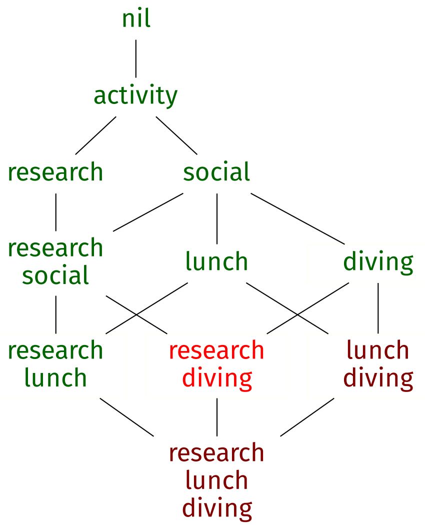
- More specific combinations are less frequent
- Ask crowd questions:
- Is {diving} frequent?
  - ⇒ Yes!
- {lunch, diving}?
  - ⇒ No!
- {lunch}?
  - ⇒ Yes!
- {research, lunch}?
  - ⇒ Yes!

Example 2: Finding popular activity combinations



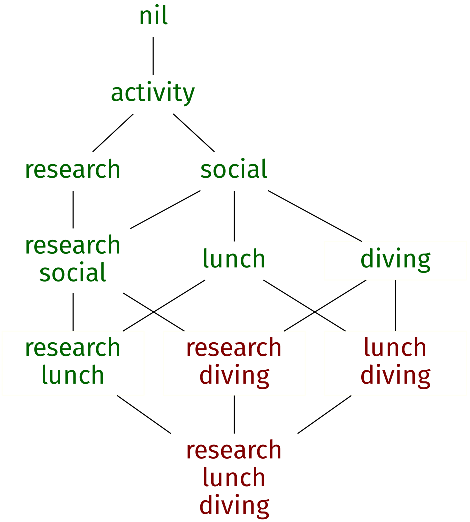
- More specific combinations are less frequent
- Ask crowd questions:
- Is {diving} frequent?
  - ⇒ Yes!
- {lunch, diving}?
  - ⇒ No!
- {lunch}?
  - ⇒ Yes!
- {research, lunch}?
  - ⇒ Yes!
- {research, diving}?

Example 2: Finding popular activity combinations



- More specific combinations are less frequent
- Ask crowd questions:
- Is {diving} frequent?
  - ⇒ Yes!
- {lunch, diving}?
  - ⇒ No!
- {lunch}?
  - ⇒ Yes!
- {research, lunch}?
  - ⇒ Yes!
- {research, diving}?
  - ⇒ No!

Example 2: Finding popular activity combinations



- More specific combinations are less frequent
- Ask crowd questions:
- Is {diving} frequent?
  - $\Rightarrow$ Yes!
- {lunch, diving}?
  - $\Rightarrow$ No!
- {lunch}?
  - $\Rightarrow$ Yes!
- {research, lunch}?
  - $\Rightarrow$ Yes!
- {research, diving}?
  - $\Rightarrow$ No!

Example 3: Estimating unknown values

- How much food do people eat at conference buffets?

Example 3: Estimating unknown values

- How much food do people eat at conference buffets?
- Let's ask fellow organizers!

Example 3: Estimating unknown values

small sweet	tiny both	small salty
medium sweet	small both	medium salty
large sweet	medium both	large salty
	large both	

- How much food do people eat at conference buffets?
- Let’s ask fellow organizers!

Example 3: Estimating unknown values

small sweet	tiny both	small salty
medium sweet	small both	medium salty
large sweet	medium both	large salty
	large both	

- How much food do people eat at conference buffets?
- Let's ask fellow organizers!

Example 3: Estimating unknown values

small sweet

tiny both

small salty

medium sweet

small both

medium salty

large sweet

medium both

large salty

large both

- How much food do people eat at conference buffets?
- Let's ask fellow organizers!
- How to estimate quantities for my own conference?

Example 3: Estimating unknown values

small sweet	tiny both	small salty
medium sweet	small both	medium salty
large sweet	**medium both**	large salty
	large both	

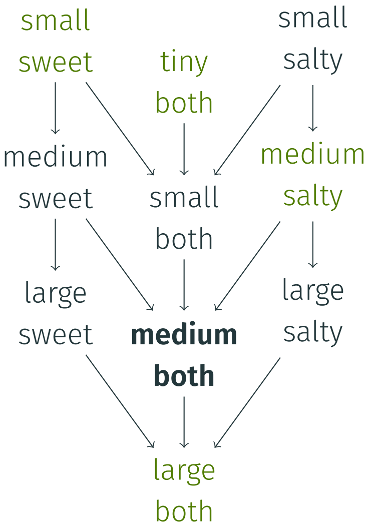
- How much food do people eat at conference buffets?
- Let's ask fellow organizers!
- How to estimate quantities for my own conference?

Example 3: Estimating unknown values

small sweet	tiny both	small salty
medium sweet	small both	medium salty
large sweet	**medium both**	large salty
	large both	

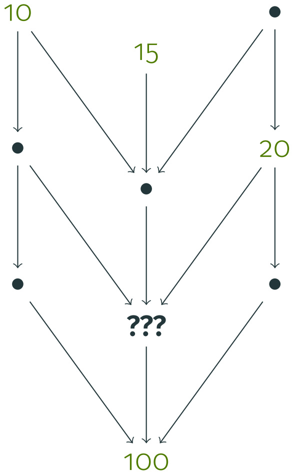
- How much food do people eat at conference buffets?
- Let's ask fellow organizers!
- How to estimate quantities for my own conference?
- Some order relations are implied

Example 3: Estimating unknown values



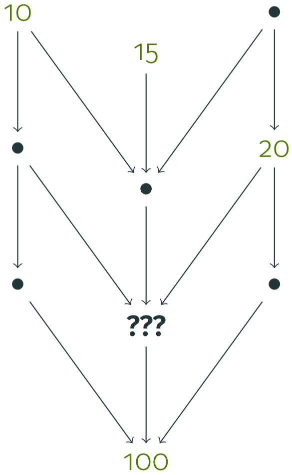
- How much food do people eat at conference buffets?
- Let's ask fellow organizers!
- How to estimate quantities for my own conference?
- Some order relations are implied

Example 3: Estimating unknown values



- How much food do people eat at conference buffets?
- Let's ask fellow organizers!
- How to estimate quantities for my own conference?
- Some order relations are implied

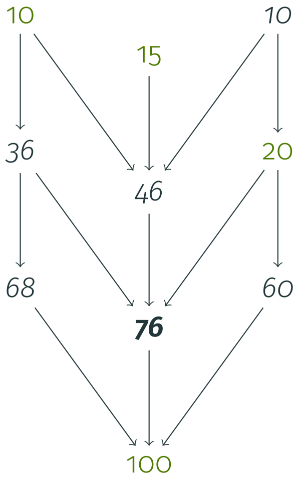
Example 3: Estimating unknown values



- How much food do people eat at conference buffets?
- Let's ask fellow organizers!
- How to estimate quantities for my own conference?
- Some order relations are implied

→ How to complete the missing values?

Example 3: Estimating unknown values



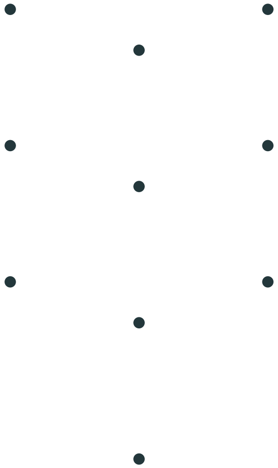
- How much food do people eat at conference buffets?
- Let's ask fellow organizers!
- How to estimate quantities for my own conference?
- Some order relations are implied

$\rightarrow$ How to complete the missing values?

General problem

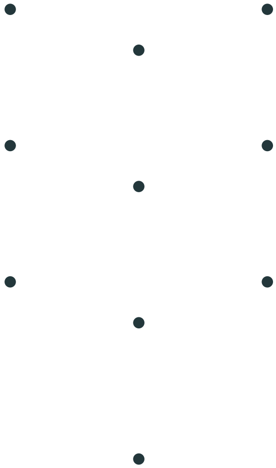
- Several items with values

General problem



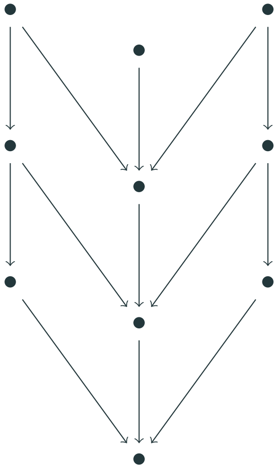
- Several items with values

General problem



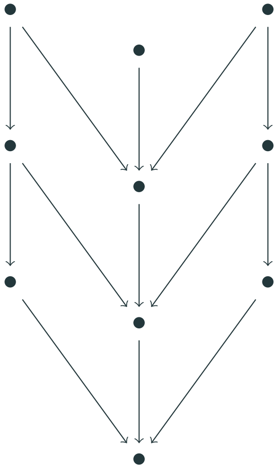
- Several items with values
- Order relations on the values

General problem



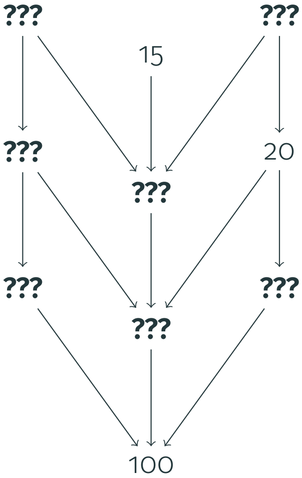
- Several items with values
- Order relations on the values

General problem



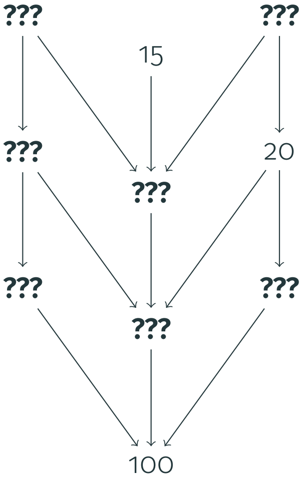
- Several items with values
- Order relations on the values
- Some values are known
  and the others are unknown

General problem



- Several items with values
- Order relations on the values
- Some values are known
  and the others are unknown

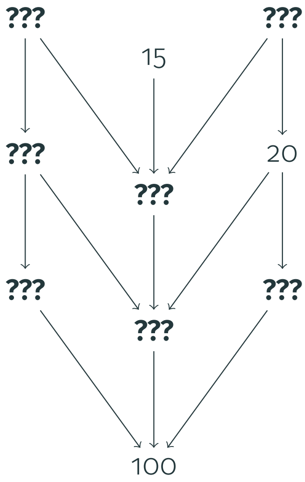
General problem



- Several items with values
- Order relations on the values
- Some values are known
  and the others are unknown

→ Estimate the unknown values

General problem



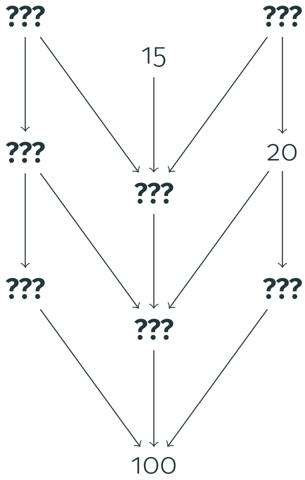
- Several items with values
- Order relations on the values
- Some values are known
  and the others are unknown

$\rightarrow$ Estimate the unknown values

$\rightarrow$ Find the next crowd question to ask
to obtain more known values

Estimating Missing Values

Estimating missing values (Antoine, Yael, Pierre, Tova, ICDT'17)



- Known and unknown values with order relation
- Estimate the unknown values without asking more crowd questions

Easy case: total order

$$\begin{tikzcd} 0 \arrow[d] \\ \textbf{???} \arrow[d] \\ 70 \arrow[d] \\ \textbf{???} \arrow[d] \\ \textbf{???} \arrow[d] \\ 100 \end{tikzcd}$$

- If the items are totally ordered, what can we do?

Easy case: total order

0

$\downarrow$

???

$\downarrow$

70

$\downarrow$

???

$\downarrow$

???

$\downarrow$

100

- If the items are totally ordered, what can we do?

$\rightarrow$ Linear interpolation!

Easy case: total order

$$\begin{tikzcd} 0 \arrow[d] \\ 35 \arrow[d] \\ 70 \arrow[d] \\ 80 \arrow[d] \\ 90 \arrow[d] \\ 100 \end{tikzcd}$$

- If the items are totally ordered, what can we do?

$\rightarrow$ Linear interpolation!

Easy case: total order

$$\begin{array}{c} 0 \\ \downarrow \\ 35 \\ \downarrow \\ 70 \\ \downarrow \\ 80 \\ \downarrow \\ 90 \\ \downarrow \\ 100 \end{array}$$

- If the items are totally ordered, what can we do?

$\rightarrow$ Linear interpolation!

$\rightarrow$ Can we generalize this if the order is not total?

Polytope interpretation

- Introduce one variable per item:

  $\rightarrow$ x, y, z, w

Polytope interpretation

- Introduce one variable per item:
  $\rightarrow$ x, y, z, w
- Write the order constraints:
  $\rightarrow$ $x \geq y, w \leq y, y \leq z$

Polytope interpretation

- Introduce one variable per item:
  - $\rightarrow$ x, y, z, w
- Write the order constraints:
  - $\rightarrow$ $x \geq y, w \leq y, y \leq z$
- Write the known values:
  - $\rightarrow$ $w = 0.3, z = 0.9$

Polytope interpretation

- Introduce one variable per item:
  - $\rightarrow$ x, y, z, w
- Write the order constraints:
  - $\rightarrow$ $x \geq y$, $w \leq y$, $y \leq z$
- Write the known values:
  - $\rightarrow$ $w = 0.3$, $z = 0.9$
- These linear constraints define an admissible polytope

Polytope interpretation

- Introduce one variable per item:
  $\rightarrow$ x, y, z, w
- Write the order constraints:
  $\rightarrow$ $x \geq y, w \leq y, y \leq z$
- Write the known values:
  $\rightarrow$ $w = 0.3, z = 0.9$
- These linear constraints define an admissible polytope

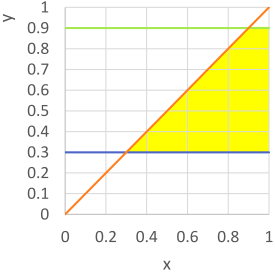
$x \geq y$

$w \leq y$

$y \leq z$

$w = 0.3$

$z = 0.9$



Polytope interpretation

- Introduce one variable per item:
  - $\rightarrow$ x, y, z, w
- Write the order constraints:
  - $\rightarrow$ $x \geq y$, $w \leq y$, $y \leq z$
- Write the known values:
  - $\rightarrow$ $w = 0.3$, $z = 0.9$
- These linear constraints define an admissible polytope

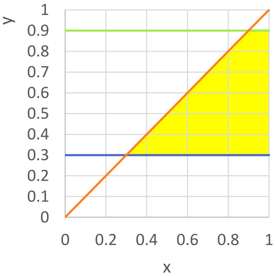
$x \geq y$

$w \leq y$

$y \leq z$

$w = 0.3$

$z = 0.9$



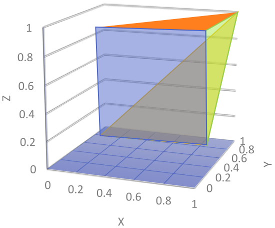
$x \geq y$

$w \leq y$

$y \leq z$

$w = 0.3$

$z \leq 1$



Polytope interpretation

- Introduce one variable per item:
  - $\rightarrow$ x, y, z, w
- Write the order constraints:
  - $\rightarrow$ $x \geq y$, $w \leq y$, $y \leq z$
- Write the known values:
  - $\rightarrow$ $w = 0.3$, $z = 0.9$
- These linear constraints define an admissible polytope

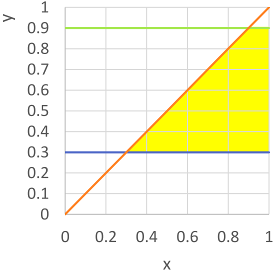
$x \geq y$

$w \leq y$

$y \leq z$

$w = 0.3$

$z = 0.9$



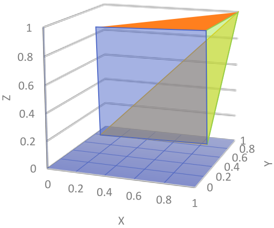
$x \geq y$

$w \leq y$

$y \leq z$

$w = 0.3$

$z \leq 1$

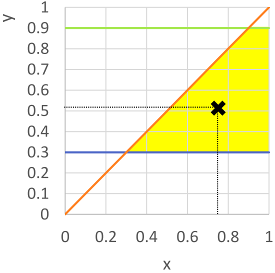


$\rightarrow$ Estimate unknown values as the center of mass of the polytope!

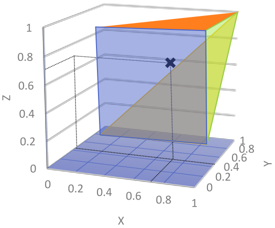
Polytope interpretation

- Introduce one variable per item:
  - $\rightarrow$ x, y, z, w
- Write the order constraints:
  - $\rightarrow$ $x \geq y, w \leq y, y \leq z$
- Write the known values:
  - $\rightarrow$ $w = 0.3, z = 0.9$
- These linear constraints define an admissible polytope

$$x \geq y$$
$$w \leq y$$
$$y \leq z$$
$$w = 0.3$$
$$z = 0.9$$



$$x \geq y$$
$$w \leq y$$
$$y \leq z$$
$$w = 0.3$$
$$z \leq 1$$



$\rightarrow$ Estimate unknown values as the center of mass of the polytope!

Results of our ICDT'17 paper

- For total orders, the polytope method gives the same result as linear interpolation

Results of our ICDT'17 paper

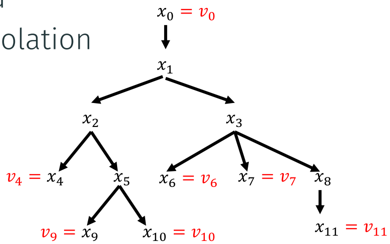
- For total orders, the polytope method gives the same result as linear interpolation
- Brute-force algorithm to compute the center of mass

Results of our ICDT'17 paper

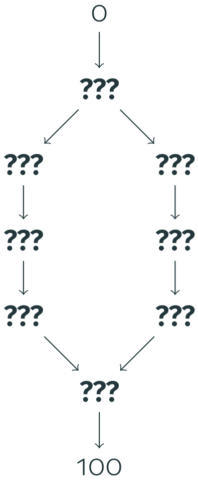
- For total orders, the polytope method gives the same result as linear interpolation
- Brute-force algorithm to compute the center of mass
- Intractability results for this task and for computing the top-k items

Results of our ICDT'17 paper

- For total orders, the polytope method gives the same result as linear interpolation
- Brute-force algorithm to compute the center of mass
- Intractability results for this task and for computing the top-k items
- Tractable cases when the order is a tree

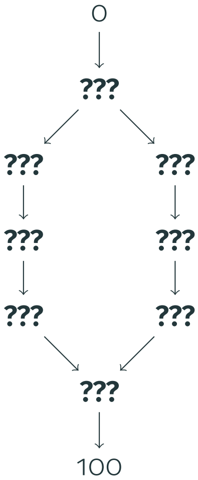


Open problems



- Are there more general posets/polytopes where we can interpolate efficiently? (generalizing trees, e.g., treelike posets)

Open problems

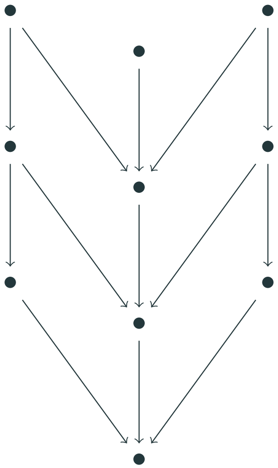


- Are there more general posets/polytopes where we can interpolate efficiently? (generalizing trees, e.g., treelike posets)

- Fixing one value to its interpolated value can change other interpolated values! (unlike linear interpolation)

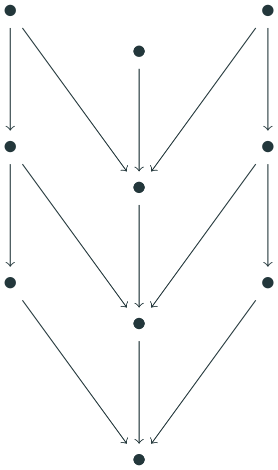
Asking Questions

Asking questions (Antoine, Yael, Tova, ICDT'14)



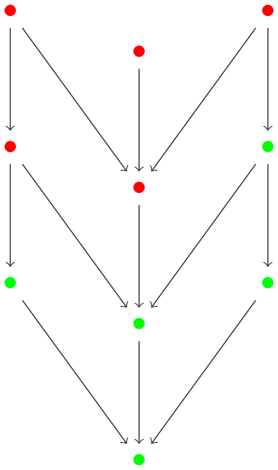
- Unknown values are Boolean: either 0 or 1

Asking questions (Antoine, Yael, Tova, ICDT'14)



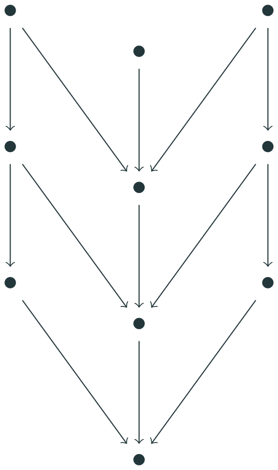
- Unknown values are Boolean: either 0 or 1
- The Boolean function is monotone with respect to the order

Asking questions (Antoine, Yael, Tova, ICDT'14)



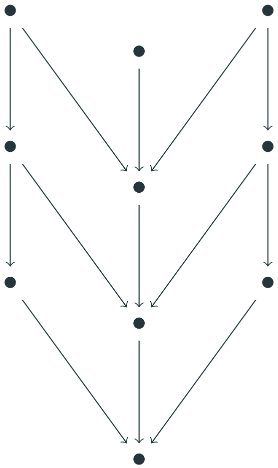
- Unknown values are Boolean: either 0 or 1
- The Boolean function is monotone with respect to the order

Asking questions (Antoine, Yael, Tova, ICDT'14)



- Unknown values are Boolean: either 0 or 1
- The Boolean function is monotone with respect to the order

Asking questions (Antoine, Yael, Tova, ICDT'14)



- Unknown values are Boolean: either 0 or 1
- The Boolean function is monotone with respect to the order
- Ask the right questions to determine the Boolean function completely

Easy case: total order

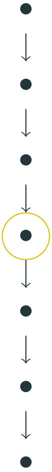
- If the items are totally ordered, what can we do?

Easy case: total order

- If the items are totally ordered, what can we do?

→ Binary search!

Easy case: total order



- If the items are totally ordered, what can we do?

$\rightarrow$ Binary search!

Easy case: total order

- If the items are totally ordered, what can we do?

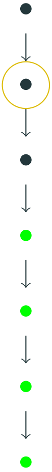
$\rightarrow$ Binary search!

Easy case: total order

- If the items are totally ordered, what can we do?

$\rightarrow$ Binary search!

Easy case: total order



- If the items are totally ordered, what can we do?

→ Binary search!

Easy case: total order

- If the items are totally ordered, what can we do?

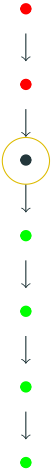
→ Binary search!

Easy case: total order

- If the items are totally ordered, what can we do?

$\rightarrow$ Binary search!

Easy case: total order



- If the items are totally ordered, what can we do?

→ Binary search!

Easy case: total order

- If the items are totally ordered, what can we do?

→ Binary search!

Easy case: total order

- If the items are totally ordered, what can we do?

→ Binary search!

→ Can we generalize this if the order is not total?

Results of our ICDT'14 paper

- Lower bound: each crowd query gives one bit so we need logarithmically many crowd queries in the number of possible Boolean functions

Results of our ICDT'14 paper

- Lower bound: each crowd query gives one bit so we need logarithmically many crowd queries in the number of possible Boolean functions
- Matching upper bound: we can always query an item that splits the remaining Boolean functions evenly

Results of our ICDT'14 paper

- Lower bound: each crowd query gives one bit so we need logarithmically many crowd queries in the number of possible Boolean functions
- Matching upper bound: we can always query an item that splits the remaining Boolean functions evenly
- Output-sensitive complexity: we can determine the function in a number of crowd queries linear in the "frontier"

Results of our ICDT'14 paper

- Lower bound: each crowd query gives one bit so we need logarithmically many crowd queries in the number of possible Boolean functions
- Matching upper bound: we can always query an item that splits the remaining Boolean functions evenly
- Output-sensitive complexity: we can determine the function in a number of crowd queries linear in the "frontier"
- Computational hardness, e.g., of finding the best-split element

Open problems

- We do not have a strategy that generalizes binary search? (i.e., computationally tractable and makes few queries)

Open problems

- We do not have a strategy that generalizes binary search? (i.e., computationally tractable and makes few queries)
- What is the performance of making queries at random?

Open problems

- We do not have a strategy that generalizes binary search? (i.e., computationally tractable and makes few queries)
- What is the performance of making queries at random?
- Many technical questions left open!

Open problems

- We do not have a strategy that generalizes binary search? (i.e., computationally tractable and makes few queries)
- What is the performance of making queries at random?
- Many technical questions left open!

Is it #P-hard to compute the number of antichains of a distributive lattice?

11

An antichain of a poset $(P, <)$ is a subset of pairwise incomparable elements, namely, a subset $A \subseteq P$ such that there are no $x, y \in A$ with $x < y$. By a result of Provan and Ball, it is known that it is #P-hard, given a poset, to compute its number of antichains

asked Nov 11 '15 at 19:28
a3nm
2,906 • 10 • 46

Open problems

- We do not have a strategy that generalizes binary search? (i.e., computationally tractable and makes few queries)
- What is the performance of making queries at random?
- Many technical questions left open!

Is it #P-hard to compute the number of antichains of a distributive lattice?

11

An antichain of a poset $(P, <)$ is a subset of pairwise incomparable elements, namely, a subset $A \subseteq P$ such that there are no $x, y \in A$ with $x < y$. By a result of Provan and Ball, it is known that it is #P-hard, given a poset, to compute its number of antichains

asked Nov 11 '15 at 19:28
a3nm
2,906 • 10 • 46

Worst number of questions needed to learn a monotonic predicate over a poset

15

Consider $(X, \leq)$ a finite poset over n items, and P an unknown monotonic predicate over X (i.e., for any $x, y \in X$, if $P(x)$ and $x \leq y$ then $P(y)$). I can evaluate P by providing one node $x \in X$ and finding out if $P(x)$ holds or not. My goal is to determine exactly the set of nodes $x \in X$ such that $P(x)$ holds, using as

asked Jan 28 '13 at 14:58
a3nm
2,906 • 10 • 46

Open problems

- We do not have a strategy that generalizes binary search? (i.e., computationally tractable and makes few queries)
- What is the performance of making queries at random?
- Many technical questions left open!

Is it #P-hard to compute the number of antichains of a distributive lattice?

11

An antichain of a poset $(P, <)$ is a subset of pairwise incomparable elements, namely, a subset $A \subseteq P$ such that there are no $x, y \in A$ with $x < y$. By a result of Provan and Ball, it is known that it is #P-hard, given a poset, to compute its number of antichains

asked Nov 11 '15 at 19:28
a3nm
2,906 • 10 • 46

Worst number of questions needed to learn a monotonic predicate over a poset

15

Consider $(X, \leq)$ a finite poset over n items, and P an unknown monotonic predicate over X (i.e., for any $x, y \in X$, if $P(x)$ and $x \leq y$ then $P(y)$). I can evaluate P by providing one node $x \in X$ and finding out if $P(x)$ holds or not. My goal is to determine exactly the set of nodes $x \in X$ such that $P(x)$ holds, using as

asked Jan 28 '13 at 14:58
a3nm
2,906 • 10 • 46

Minimal elements of a monotonic predicate over the powerset $2^{|n|}$

12

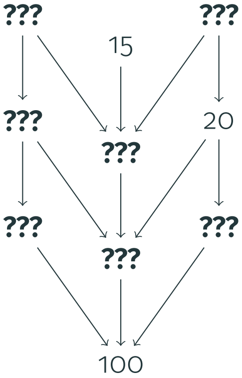
Consider a *monotonic* predicate P over the powerset $2^{|n|}$ (ordered by inclusion). By "monotonic" I mean: $\forall x, y \in 2^{|n|}$ such that $x \subset y$, if $P(x)$ then $P(y)$. I am looking for an algorithm to find all the minimal elements of P, i.e., the $x \in 2^{|n|}$ such that $P(x)$ but $\forall y \subset x, \neg P(y)$. Since the width of $2^{|n|}$ is $\binom{n}{\cdot}$, there could be

asked Dec 20 '12 at 11:41
a3nm
2,906 • 10 • 46

Conclusion

Summary and conclusion

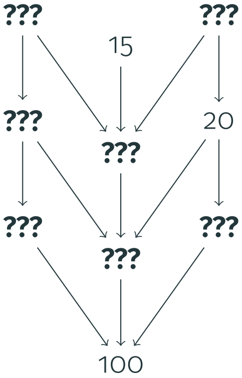
- General problem:
    - Known and unknown values
    - Order relation between them



Summary and conclusion

- General problem:
  - Known and unknown values
  - Order relation between them

$\rightarrow$ Estimate the missing values

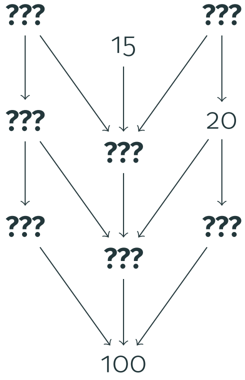


Summary and conclusion

- General problem:
  - Known and unknown values
  - Order relation between them

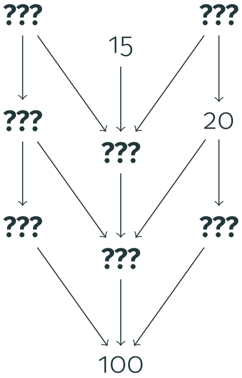
→ Estimate the missing values

→ Choose the next question to ask



Summary and conclusion

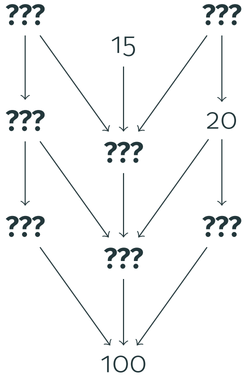
- General problem:
  - Known and unknown values
  - Order relation between them
  - → Estimate the missing values
  - → Choose the next question to ask



→ Can we combine the two approaches?

Summary and conclusion

- General problem:
  - Known and unknown values
  - Order relation between them
  - → Estimate the missing values
  - → Choose the next question to ask

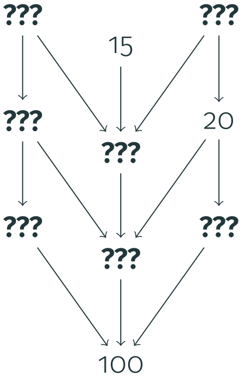


→ Can we combine the two approaches?

→ What about uncertainty on crowd answers?
(ask the same question multiple times to get a better estimate)

Summary and conclusion

- General problem:
  - Known and unknown values
  - Order relation between them
  - → Estimate the missing values
  - → Choose the next question to ask



→ Can we combine the two approaches?

→ What about uncertainty on crowd answers?
(ask the same question multiple times to get a better estimate)

Thanks for your attention!

References

Amarilli, A., Amsterdamer, Y., and Milo, T. (2014).
On the Complexity of Mining Itemsets from the Crowd Using Taxonomies.
In *ICDT*.

Amarilli, A., Amsterdamer, Y., Milo, T., and Senellart, P. (2017).
Top-k Queries on Unknown Values under Order Constraints.
In *ICDT*.

Image credits

- Title slide, Bar-Ilan logo, https://en.wikipedia.org/wiki/File:Bar_Ilan_logo2.svg
- Slide 9: picture by Yael Amsterdamer, https://a3nm.net/work/talks/icdt2017/amarilli2017top_slides.pdf, slide 7
- Slide 10: picture by Yael Amsterdamer, https://a3nm.net/work/talks/icdt2017/amarilli2017top_slides.pdf, slide 14