Discovering relative importance of skyline attributes

D. Mindolin & J. Chomicki

Department of Computer Science and Engineering
University at Buffalo, SUNY

VLDB 2009

Main contributions

1. generalizing skylines to p-skylines to capture relative attribute importance
2. discovering p-skylines on the basis of user feedback: algorithms and complexity

Skylines [Börzsönyi et al., ICDE'01]

Skyline preferences

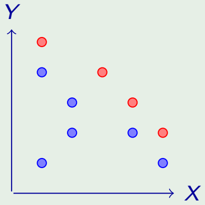
- Atomic preferences ($\mathcal{H}$): total orders over attribute values
- Skyline preference relation ($sky_{\mathcal{H}}$): t_1 preferred to t_2 if
  - t_1 equal or better than t_2 in every attribute, and
  - t_1 strictly better than t_2 in at least one attribute
- Skyline: the set $w_{sky_{\mathcal{H}}}(\mathcal{O})$ of best tuples (according to $sky_{\mathcal{H}}$) in a set of tuples $\mathcal{O}$

Skylines [Börzsönyi et al., ICDE'01]

Skyline preferences

- Atomic preferences ($\mathcal{H}$): total orders over attribute values
- Skyline preference relation ($sky_{\mathcal{H}}$): t_1 preferred to t_2 if
  - t_1 equal or better than t_2 in every attribute, and
  - t_1 strictly better than t_2 in at least one attribute
- Skyline: the set $w_{sky_{\mathcal{H}}}(\mathcal{O})$ of best tuples (according to $sky_{\mathcal{H}}$) in a set of tuples $\mathcal{O}$

Example

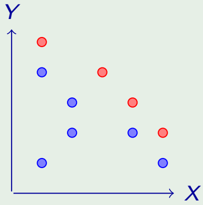


Skylines [Börzsönyi et al., ICDE'01]

Skyline preferences

- Atomic preferences ($\mathcal{H}$): total orders over attribute values
- Skyline preference relation ($sky_{\mathcal{H}}$): t_1 preferred to t_2 if
  - t_1 equal or better than t_2 in every attribute, and
  - t_1 strictly better than t_2 in at least one attribute
- Skyline: the set $w_{sky_{\mathcal{H}}}(\mathcal{O})$ of best tuples (according to $sky_{\mathcal{H}}$) in a set of tuples $\mathcal{O}$

Example



Skyline properties

- Simple, unique way of composing atomic preferences
- Equal attribute importance
- Skyline of exponential size

p-skylines

p-skyline relation $\succ$

- Induced by an atomic preference relation $>_A \in \mathcal{H}$

$$\succ \; = \; \{(t, t') \mid t.A >_A t'.A\}$$

- Pareto accumulation ("$\succ_1$ equally important as $\succ_2$")

$$\succ = \succ_1 \otimes \succ_2$$

- Prioritized accumulation ("$\succ_1$ more important than $\succ_2$")

$$\succ = \succ_1 \,\&\, \succ_2$$

p-skylines

p-skyline relation $\succ$

- Induced by an atomic preference relation $>_A \in \mathcal{H}$

$$\succ \; = \; \{(t, t') \mid t.A >_A t'.A\}$$

- Pareto accumulation ("$\succ_1$ equally important as $\succ_2$")

$$\succ \; = \; \succ_1 \; \otimes \; \succ_2$$

- Prioritized accumulation ("$\succ_1$ more important than $\succ_2$")

$$\succ \; = \; \succ_1 \; \& \; \succ_2$$

Each atomic preference must be used exactly once in $\succ$

Pareto accumulation [Kießling'02]

Definitions

$Var(\succ)$ - set of attributes used in definition of $\succ$

$E_S = \{(t.A, t'.A) \mid A \in S \wedge t.A = t'.A\}$ - pairs of tuples equal in every attribute in S

Pareto accumulation: $\succ_1$ as important as $\succ_2$

$$\succ_1 \otimes \succ_2 = (\succ_1 \cap E_{Var(\succ_2)}) \cup (\succ_2 \cap E_{Var(\succ_1)}) \cup (\succ_1 \cap \succ_2)$$

Pareto accumulation [Kießling'02]

Definitions

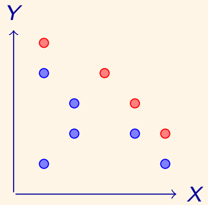
$Var(\succ)$ - set of attributes used in definition of $\succ$

$E_S = \{(t.A, t'.A) \mid A \in S \wedge t.A = t'.A\}$ - pairs of tuples equal in every attribute in S

Pareto accumulation: $\succ_1$ as important as $\succ_2$

$$\succ_1 \otimes \succ_2 = (\succ_1 \cap E_{Var(\succ_2)}) \cup (\succ_2 \cap E_{Var(\succ_1)}) \cup (\succ_1 \cap \succ_2)$$

$\succ_X \otimes \succ_Y$



Prioritized accumulation [Kießling'02]

Definitions

$Var(\succ)$ - set of attributes used in definition of $\succ$

$E_S = \{(t.A, t'.A) \mid A \in S \wedge t.A = t'.A\}$ - pairs of tuples equal in every attribute in S

Prioritized accumulation: $\succ_1$ more important than $\succ_2$

$$\succ_1 \ \& \ \succ_2 = \succ_1 \cup (\succ_2 \cap E_{Var(\succ_2)})$$

Prioritized accumulation [Kießling'02]

Definitions

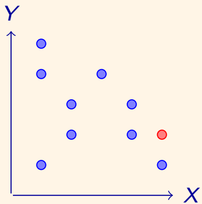
$Var(\succ)$ - set of attributes used in definition of $\succ$

$E_S = \{(t.A, t'.A) \mid A \in S \wedge t.A = t'.A\}$ - pairs of tuples equal in every attribute in S

Prioritized accumulation: $\succ_1$ more important than $\succ_2$

$$\succ_1 \ \& \ \succ_2 = \succ_1 \cup (\succ_2 \cap E_{Var(\succ_2)})$$

$\succ_X \ \& \ \succ_Y$



p-skyline properties

p-skyline properties

- Many different ways of composing atomic preferences (different combinations of $\otimes$ and $\&$)
- Reduction in query result size w.r.t. skylines
- Differences in attribute importance

Representing attribute importance with p-graphs

p-graph

$\Gamma_{\succ}$ represents attribute importance induced by a p-skyline relation $\succ$

- Nodes: attributes $Var(\succ)$
- Edges: from more important to less important attributes

Representing attribute importance with p-graphs

p-graph

$\Gamma_{\succ}$ represents attribute importance induced by a p-skyline relation $\succ$

- Nodes: attributes $Var(\succ)$
- Edges: from more important to less important attributes

$$\succ' = \succ_A \otimes \succ_B \otimes \succ_C$$

$\boxed{A}$ $\boxed{B}$ $\boxed{C}$

Representing attribute importance with p-graphs

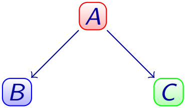
p-graph

$\Gamma_{\succ}$ represents attribute importance induced by a p-skyline relation $\succ$

- Nodes: attributes $Var(\succ)$
- Edges: from more important to less important attributes

$\succ' = \succ_A \otimes \succ_B \otimes \succ_C$

$\succ'' = \succ_A \,\&\, (\succ_B \otimes \succ_C)$



Properties of p-graphs

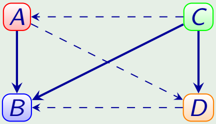
Necessary and sufficient conditions for p-graphs

Γ is a p-graph of a p-skyline relation iff Γ is

- SPO
- satisfies Envelope property

Envelope

$$\forall A, B, C, D \in \mathcal{A}, \text{ all different}$$
$$(A, B) \in \Gamma \wedge (C, D) \in \Gamma \wedge (C, B) \in \Gamma \Rightarrow$$
$$(C, A) \in \Gamma \vee (A, D) \in \Gamma \vee (D, B) \in \Gamma$$



Dominance testing using p-graphs

Is o preferred to o' by $\succ$?

$o \succ o'$ iff

- $o \neq o'$, and
- for every attribute B in which o is worse than o', there is a parent A of B in which o is better than o'

Dominance testing using p-graphs

Is o preferred to o' by $\succ$?

$o \succ o'$ iff

- $o \neq o'$, and
- for every attribute B in which o is worse than o', there is a parent A of B in which o is better than o'

$$\succ \; = \; \succ_A \;\&\; (\succ_B \;\otimes\; \succ_C)$$

$$\begin{tikzcd}
 & A \arrow[dl] \arrow[dr] & \\
B & & C
\end{tikzcd}$$

Example

A : b better than w
B : b better than w
C : b better than w

Then

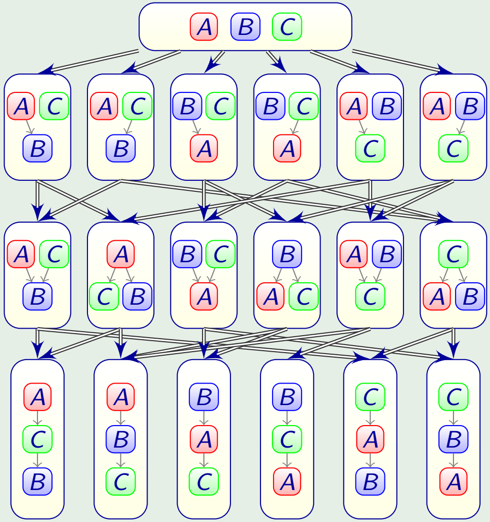
$(\mathrm{b}, \mathrm{w}, \mathrm{b}) \succ (\mathrm{w}, \mathrm{b}, \mathrm{b})$
$(\mathrm{b}, \mathrm{w}, \mathrm{b}) \not\succ (\mathrm{b}, \mathrm{b}, \mathrm{w})$

Containment of p-skyline relations

Using p-graphs for checking containment

$$\succ \subset \succ' \Leftrightarrow E(\Gamma_{\succ}) \subset E(\Gamma_{\succ'})$$

Containment hierarchy



Containment of p-skyline relations

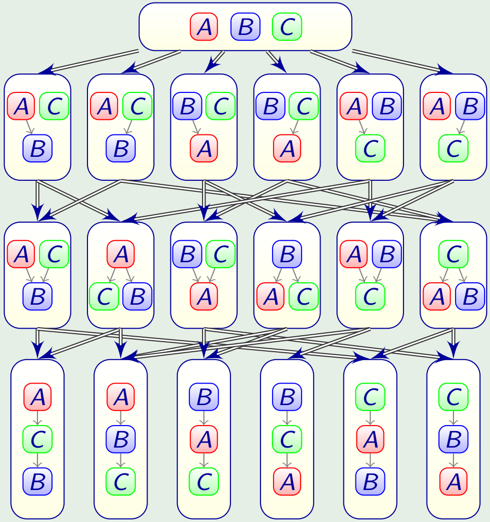
Using p-graphs for checking containment

$$\succ \subset \succ' \Leftrightarrow E(\Gamma_{\succ}) \subset E(\Gamma_{\succ'})$$

Minimal extensions of $\succ$

- Correspond to immediate children of $\Gamma_{\succ}$ in the hierarchy
- Obtained using rewriting rules applied to syntax trees of p-skyline formulas

Containment hierarchy



Minimal extension rewriting rules

Rules to compute minimal extensions of p-skyline relation

- Applied to syntax trees of p-skyline formulas
- Every minimal extension computed by a single rule application in PTIME
- Full set consists of four rule templates
- All minimal extensions of p-skyline relation can be computed in PTIME

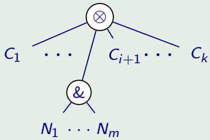
Minimal extension rewriting rules

Rules to compute minimal extensions of p-skyline relation

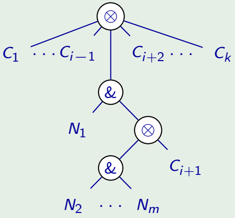
- Applied to syntax trees of p-skyline formulas
- Every minimal extension computed by a single rule application in PTIME
- Full set consists of four rule templates
- All minimal extensions of p-skyline relation can be computed in PTIME

$Rule_1$ template

Original tree part



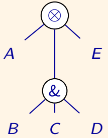
Transformed tree part



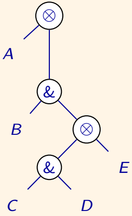
Minimal extension rewriting rules

Example

$A \otimes (B \ \& \ C \ \& \ D) \otimes E$

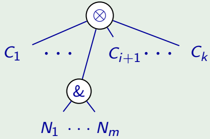


$A \otimes (B \ \& \ (E \otimes (C \ \& \ D)))$

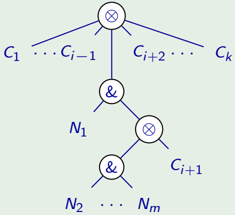


$Rule_1$ template

Original tree part



Transformed tree part



Discovery of p-skyline relations from user feedback

Problem

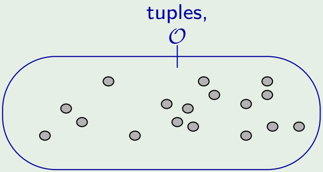
Given a set $\mathcal{A}$ of relevant attributes and a set $\mathcal{H}$ of atomic preferences over $\mathcal{A}$, discover the relative importance of attributes [in the form of a p-skyline relation $\succ$], based on user feedback.

Discovery of p-skyline relations from user feedback

Problem

Given a set $\mathcal{A}$ of relevant attributes and a set $\mathcal{H}$ of atomic preferences over $\mathcal{A}$, discover the relative importance of attributes [in the form of a p-skyline relation $\succ$], based on user feedback.

User Feedback

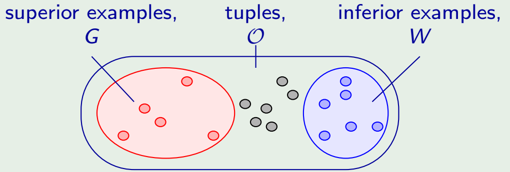


Discovery of p-skyline relations from user feedback

Problem

Given a set $\mathcal{A}$ of relevant attributes and a set $\mathcal{H}$ of atomic preferences over $\mathcal{A}$, discover the relative importance of attributes [in the form of a p-skyline relation $\succ$], based on user feedback.

User Feedback



Superior examples

Tuples in $\mathcal{O}$ which user confidently likes

Inferior examples

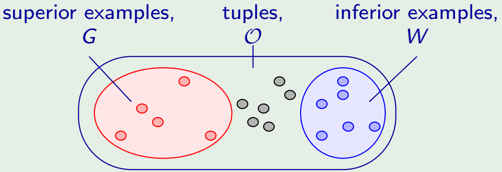
Tuples in $\mathcal{O}$ which user confidently dislikes

Discovery of p-skyline relations from user feedback

Problem

Given a set $\mathcal{A}$ of relevant attributes and a set $\mathcal{H}$ of atomic preferences over $\mathcal{A}$, discover the relative importance of attributes [in the form of a p-skyline relation $\succ$], based on user feedback.

User Feedback



Superior examples

Tuples in $\mathcal{O}$ which user confidently likes

Inferior examples

Tuples in $\mathcal{O}$ which user confidently dislikes

$\succ$ favors G/disfavors W in $\mathcal{O}$

1. G are among the best tuples in $\mathcal{O}$ according to $\succ$
2. W are not among the best tuples in $\mathcal{O}$ according to $\succ$

Complexity of p-skyline relation discovery

	Arbitrary W	$W = \emptyset$
Checking existence of $\succ$ favoring G and disfavoring W in O	NP-complete	PTIME
Computing maximal $\succ$ favoring G and disfavoring W in O	FNP-complete	PTIME

Computing maximal $\succ$ favoring G in $\mathcal{O}$ $(W = \emptyset)$

Approach

1. Construct a system $\mathcal{N}$ of negative constraints from G and O
2. Apply minimal extension rules to find maximal $\succ$ satisfying $\mathcal{N}$
3. Various optimizations possible

Computing maximal $\succ$ favoring G in $\mathcal{O}$ $(W = \emptyset)$

Approach

1. Construct a system $\mathcal{N}$ of negative constraints from G and O
2. Apply minimal extension rules to find maximal $\succ$ satisfying $\mathcal{N}$
3. Various optimizations possible

Algorithm complexity

$$\mathcal{O}(|\mathcal{O}| \cdot |G| \cdot |\mathcal{A}|^3)$$

Negative constraints

$\succ$ favors G in $\mathcal{O}$: what does it mean?

1. $\succ$ favors G in $\mathcal{O}$: for every $o \in O, o' \in G$, $o \not\succ o'$
2. $o \not\succ o'$: use the dominance testing rule
   - $\mathcal{L}_\tau$: attributes in which o is better than o'
   - $\mathcal{R}_\tau$: attributes in which o is worse than o'
   - negative constraint $\tau =< \mathcal{L}_\tau, \mathcal{R}_\tau >$: some attribute in $\mathcal{R}_\tau$ is not a child in $\Gamma_\succ$ of (i.e., not less important than) any attribute in $\mathcal{L}_\tau$
3. $o \not\succ o'$ iff corresponding τ is satisfied

Negative constraints

$\succ$ favors G in $\mathcal{O}$: what does it mean?

1. $\succ$ favors G in $\mathcal{O}$: for every $o \in O, o' \in G$, $o \not\succ o'$
2. $o \not\succ o'$: use the dominance testing rule
   - $\mathcal{L}_\tau$: attributes in which o is better than o'
   - $\mathcal{R}_\tau$: attributes in which o is worse than o'
   - negative constraint $\tau =< \mathcal{L}_\tau, \mathcal{R}_\tau >$: some attribute in $\mathcal{R}_\tau$ is not a child in $\Gamma_\succ$ of (i.e., not less important than) any attribute in $\mathcal{L}_\tau$
3. $o \not\succ o'$ iff corresponding τ is satisfied

Example

id	A	B	C
o	b	w	w
o'	w	b	b

$o \not\succ o'$ represented by $\tau =< \{A\}, \{B, C\} >$

Negative constraints

$\succ$ favors G in $\mathcal{O}$: what does it mean?

1. $\succ$ favors G in $\mathcal{O}$: for every $o \in O, o' \in G$, $o \not\succ o'$
2. $o \not\succ o'$: use the dominance testing rule
   - $\mathcal{L}_\tau$: attributes in which o is better than o'
   - $\mathcal{R}_\tau$: attributes in which o is worse than o'
   - negative constraint $\tau =< \mathcal{L}_\tau, \mathcal{R}_\tau >$: some attribute in $\mathcal{R}_\tau$ is not a child in $\Gamma_\succ$ of (i.e., not less important than) any attribute in $\mathcal{L}_\tau$
3. $o \not\succ o'$ iff corresponding τ is satisfied

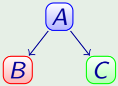
Example

id	A	B	C
o	b	w	w
o'	w	b	b

$o \not\succ o'$ represented by $\tau =< \{A\}, \{B, C\} >$

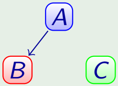
Example 2

$\succ_A \ \& \ (\succ_B \ \otimes \ \succ_C)$



does not satisfy τ

$(\succ_A \ \& \ \succ_B) \otimes \ \succ_C$



satisfy τ

Using $\mathcal{N}$ in algorithm

Rule application strategy

- Three out of four rule templates used to compute

$$\succ_{sky} \subset \succ_1 \subset \ \ldots \ \subset \succ_k$$

- Each $\succ_i$ satisfies $\mathcal{N}$
- Each $\succ_i$ is a minimal extension of $\succ_{i-1}$ $(\succ_{sky} = \succ_0)$
- No minimal extension of $\succ_k$ satisfies $\mathcal{N}$
- Each rule only addes edges to $\Gamma_{\succ_i}$ going to/from set of attributes to distinguished attribute E

Using $\mathcal{N}$ in algorithm

Rule application strategy

- Three out of four rule templates used to compute

$$\succ_{sky} \subset \succ_1 \subset \ \ldots \ \subset \succ_k$$

- Each $\succ_i$ satisfies $\mathcal{N}$
- Each $\succ_i$ is a minimal extension of $\succ_{i-1}$ ($\succ_{sky} = \succ_0$)
- No minimal extension of $\succ_k$ satisfies $\mathcal{N}$
- Each rule only addes edges to $\Gamma_{\succ_i}$ going to/from set of attributes to distinguished attribute E

Bottleneck: checking satisfaction of $\mathcal{N}$ by $\succ$

- Efficiently check every $\tau \in \mathcal{N}$ against $\succ$
- Reduce the size of $\mathcal{N}$

Efficient checking satisfaction to $\mathcal{N}$

Minimization of $\mathcal{N}$

- $\mathcal{N}$ is minimal w.r.t. $\succ$ iff every $\tau \in \mathcal{N}$ is minimal w.r.t. $\succ$
- τ is minimal w.r.t. $\succ$ iff $\neg\exists X \in \mathcal{L}_\tau, Y \in \mathcal{R}_\tau : (X, Y) \in \Gamma_\succ$

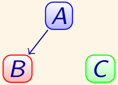
Efficient checking satisfaction to $\mathcal{N}$

Minimization of $\mathcal{N}$

- $\mathcal{N}$ is minimal w.r.t. $\succ$ iff every $\tau \in \mathcal{N}$ is minimal w.r.t. $\succ$
- τ is minimal w.r.t. $\succ$ iff $\neg\exists X \in \mathcal{L}_\tau, Y \in \mathcal{R}_\tau : (X, Y) \in \Gamma_\succ$

Minimization of τ

$$(\succ_A \ \& \ \succ_B) \otimes \succ_C$$



satisfy τ

$$\tau =< \{A\}, \{B, C\} >$$

Minimal $\tau =< \{A\}, \{C\} >$

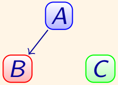
Efficient checking satisfaction to $\mathcal{N}$

Minimization of $\mathcal{N}$

- $\mathcal{N}$ is minimal w.r.t. $\succ$ iff every $\tau \in \mathcal{N}$ is minimal w.r.t. $\succ$
- τ is minimal w.r.t. $\succ$ iff $\neg\exists X \in \mathcal{L}_\tau, Y \in \mathcal{R}_\tau : (X, Y) \in \Gamma_\succ$

Minimization of τ

$$(\succ_A \ \& \ \succ_B) \otimes \succ_C$$



satisfy τ

$\tau =< \{A\}, \{B, C\} >$

Minimal $\tau =< \{A\}, \{C\} >$

Checking minimal τ

- τ - minimal w.r.t $\succ_i$
- $\succ_{i+1}$ - minimal extension of $\succ_i$
- $\Gamma_{\succ_{i+1}} - \Gamma_{\succ_i} =$ $\{(X, E) \mid X \in P_E\} \cup \{(E, Y) \mid Y \in C_E\}$

Then $\succ_{i+1}$ violates τ iff

- $\mathcal{R}_\tau = \{E\} \wedge P_E \cap \mathcal{L}_\tau \neq \emptyset$, or
- $\mathcal{R}_\tau \subseteq C_E \wedge E \in \mathcal{L}_\tau$

Methods to reduce $|\mathcal{N}|$

Motivation

$\mathcal{N}$ of $|G| \cdot (|\mathcal{O}| - 1)$ constraints checked agains $\succ_i$ in every iteration

Methods to reduce $|\mathcal{N}|$

Motivation

$\mathcal{N}$ of $|G| \cdot (|\mathcal{O}| - 1)$ constraints checked agains $\succ_i$ in every iteration

Apply skyline to $\mathcal{O}$

- Need only to compare G with skyline of O instead of $\mathcal{O}$
- $\mathcal{N}(G, \mathcal{O})$ equivalent to $\mathcal{N}(G, w_{sky_{\mathcal{H}}}(O))$

Methods to reduce $|\mathcal{N}|$

Motivation

$\mathcal{N}$ of $|G| \cdot (|\mathcal{O}| - 1)$ constraints checked agains $\succ_i$ in every iteration

Apply skyline to $\mathcal{O}$

- Need only to compare G with skyline of O instead of $\mathcal{O}$
- $\mathcal{N}(G, \mathcal{O})$ equivalent to $\mathcal{N}(G, w_{sky_{\mathcal{H}}}(O))$

Apply skyline to $\mathcal{N}$

Represent $\tau \in \mathcal{N}$ as bitmap, e.g.,

$\mathcal{L}_\tau = \{A\}, \mathcal{R}_\tau = \{C\}$

$\mathcal{L}_\tau$			$\mathcal{R}_\tau$		
A	**B**	**C**	**A**	**B**	**C**
1	0	0	1	1	0

$\mathcal{N}(G, \mathcal{O})$ equivalent to bitmap skyline of $\mathcal{N}(G, \mathcal{O})$

Experiments: Accuracy

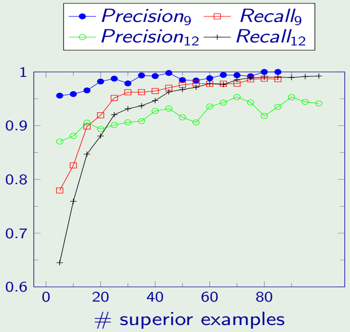
Setup

- $\mathcal{O}$: NHL player stats of $\sim 10k$ tuples
- $|\mathcal{A}| \in \{9, 12\}$
- $\succ_{fav}$ generated randomly
- G drawn from $w_{\succ_{fav}}(\mathcal{O})$

Accuracy measures

- $Precision = \frac{|w_{\succ}(\mathcal{O}) \cap w_{\succ_{fav}}(\mathcal{O})|}{|w_{\succ}(\mathcal{O})|}$
- $Recall = \frac{|w_{\succ}(\mathcal{O}) \cap w_{\succ_{fav}}(\mathcal{O})|}{|w_{\succ_{fav}}(\mathcal{O})|}$

Results



Conclusions

- *Precision* is consistently high
- *Recall* is low for small G (due to the maximality of $\succ$) but grows fast

Experiments: Performance

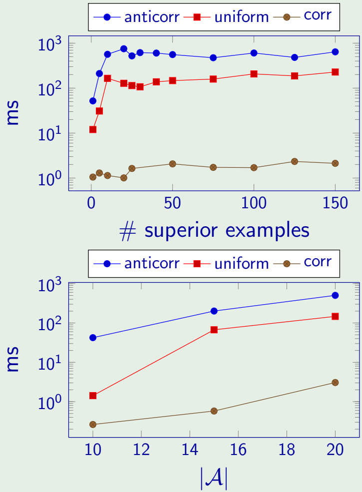
Setup

- Three datasets (anticorrelated, uniform, correlated) of 50k tuples
- $|\mathcal{A}| \in \{10, 15, 20\}$

Conclusions

Algorithm is scalable w.r.t. # superior examples and $|A|$

Results



Related work

1. [Börzsönyi et al., ICDE'01]
   - Skylines
2. [Kießling et al., VLDB'02]
   - Pareto and prioritized accumulation
3. [Holland et al., PKDD'03]
   - Mining p-skyline-like preferences (atomic preferences, operators)
   - Web server logs used as input
   - Heuristics used
4. [Jiang et al., KDD'08]
   - Mining atomic preference relations using superior/inferior examples [skyline semantics]
   - Intractable problems, heuristics used
5. [Lee et al., DEXA'08]
   - Mining of [Skyline+equivalence] preference relations
   - Answers to simple comparison questions used as feedback

Future work

- Attribute importance relationships between sets of attributes
- Selecting "good" superior examples
- Other scenarios of discovery (various forms of feedback, various result criteria)
- p-skylines: expressiveness, algorithms