Discovering relative importance of skyline attributes

D. Mindolin & J. Chomicki

Department of Computer Science and Engineering
University at Buffalo, SUNY

August 26, 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

Skyline preferences

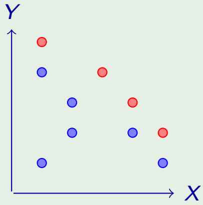
- Atomic preferences ($\mathcal{H}$): total orders over (single) attributes
- Skyline preference relation ($sky_{\mathcal{H}}$): t_1 preferred to t_2 if
  - t_1 equal or better than t_2 in every attribute, and
  - t_1 strictly better than t_2 in at least one attribute
- Skyline: the set $w_{sky_{\mathcal{H}}}(\mathcal{O})$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

Skylines

Skyline preferences

- Atomic preferences ($\mathcal{H}$): total orders over (single) attributes
- 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

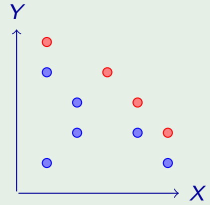


Skylines

Skyline preferences

- Atomic preferences ($\mathcal{H}$): total orders over (single) attributes
- 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 $>_A \in \mathcal{H}$

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

- 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 $>_A \in \mathcal{H}$

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

- 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 $\succ$

p-skyline properties

p-skyline properties

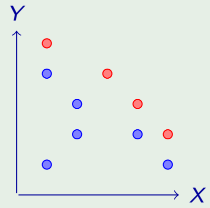
- Many different ways of composing atomic preferences (different combinations of $\otimes$ and $\&$)
- Differences in attribute importance
- Reduction in query result size

p-skyline properties

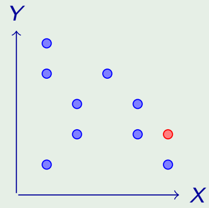
p-skyline properties

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

Pareto: $\succ_{\mathbf{X}} \otimes \succ_{\mathbf{Y}}$



Prioritized: $\succ_{\mathbf{X}} \& \succ_{\mathbf{Y}}$



p-graphs

p-graph

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

- Nodes: attributes
- Edges: from more important to less important attributes

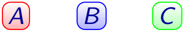
p-graphs

p-graph

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

- Nodes: attributes
- Edges: from more important to less important attributes

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



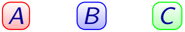
p-graphs

p-graph

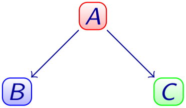
$\Gamma_{\succ}$ represents attribute importance induced by a p-skyline relation $\succ$

- Nodes: attributes
- Edges: from more important to less important attributes

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



$\succ'' = \succ_A \mathbin{\&} (\succ_B \otimes \succ_C)$

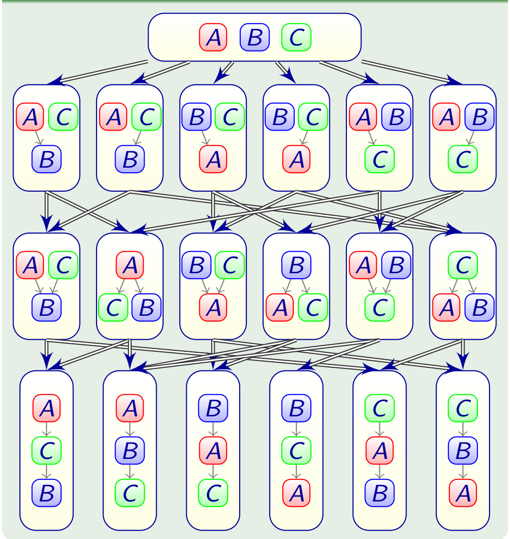


Containment of p-skyline relations

Containment

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

Containment hierarchy



Containment of p-skyline relations

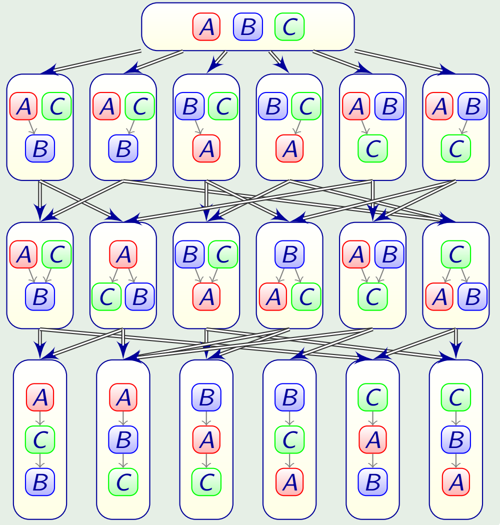
Containment

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

Minimal extensions of $\succ$

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

Containment hierarchy



Discovery of p-skyline relations from user feedback

Problem

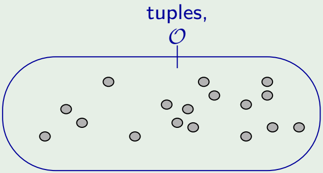
Given a set $\mathcal{A}$ of relevant attributes and a set $\mathcal{H}$ of atomic preferences over $\mathcal{A}$, discover the relative importance of attributes [in the form of a p-skyline relation $\succ$], based on user feedback.

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

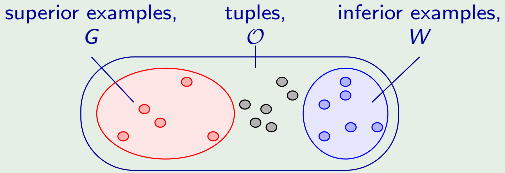


Discovery of p-skyline relations from user feedback

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

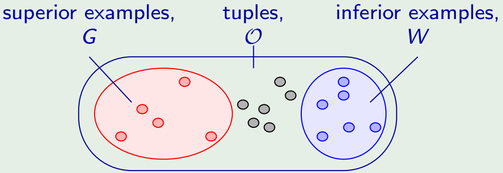
Tuples in $\mathcal{O}$ which user confidently dislikes

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 $\mathcal{O}$	NP-complete	PTIME
Computing maximal $\succ$ favoring G and disfavoring W in $\mathcal{O}$	FNP-complete	PTIME

Computing maximal $\succ$ favoring G in $\mathcal{O}$

Approach

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

Computing maximal $\succ$ favoring G in $\mathcal{O}$

Approach

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

Negative constraint

represents $t_1 \not\succ t_2$

- Syntax: $\tau =< \mathcal{L}_\tau, \mathcal{R}_\tau >$
- Semantics:
  - some attr in $\mathcal{R}_\tau$ is not a child (in $\Gamma_\succ$) of any attr in $\mathcal{L}_\tau$
  - $\mathcal{L}_\tau$ = attrs in which t_1 is better
  - $\mathcal{R}_\tau$ = attrs in which t_2 is better

Computing maximal $\succ$ favoring G in $\mathcal{O}$

Approach

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

Negative constraint

represents $t_1 \not\succ t_2$

- Syntax: $\tau = < \mathcal{L}_\tau, \mathcal{R}_\tau >$
- Semantics:
  - some attr in $\mathcal{R}_\tau$ is not a child (in $\Gamma_\succ$) of any attr in $\mathcal{L}_\tau$
  - $\mathcal{L}_\tau$ = attrs in which t_1 is better
  - $\mathcal{R}_\tau$ = attrs in which t_2 is better

Example

id	make	price	year
t_1	*bmw*	$20k$	2006
t_2	*kia*	$10k$	2007

$t_1 \not\succ t_2$ represented by
$< \{make\}, \{price, year\} >$

Computing maximal $\succ$ favoring G in $\mathcal{O}$

Approach

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

Negative constraint

represents $t_1 \not\succ t_2$

- Syntax: $\tau =< \mathcal{L}_\tau, \mathcal{R}_\tau >$
- Semantics:
  - some attr in $\mathcal{R}_\tau$ is not a child (in $\Gamma_\succ$) of any attr in $\mathcal{L}_\tau$
  - $\mathcal{L}_\tau$ = attrs in which t_1 is better
  - $\mathcal{R}_\tau$ = attrs in which t_2 is better

Example

id	make	price	year
t_1	*bmw*	$20k$	2006
t_2	*kia*	$10k$	2007

$t_1 \not\succ t_2$ represented by $< \{make\}, \{price, year\} >$

Algorithm complexity

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

Experiments: Accuracy

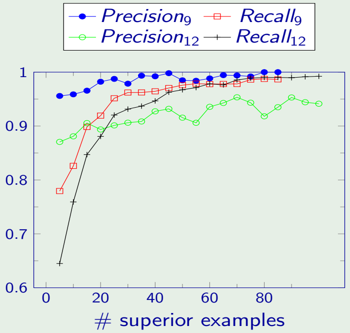
Setup

- $\mathcal{O}$: 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

Due to the maximality of $\succ$:

- *Precision* is consistently high
- *Recall* is low for small G but grows fast

Experiments: Performance

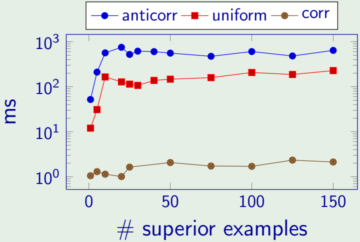
Setup

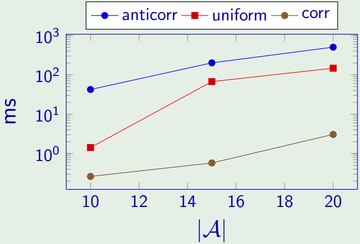
- Three datasets (anticorrelated, uniform, correlated) of 50k tuples
- $|\mathcal{A}| \in \{10, 15, 20\}$

Conclusions

Algorithm is scalable w.r.t. the number of superior examples and $|A|$

Results





Related work

1. [Holland et al, PKDD'2003]
   - Mining p-skyline-like preferences (atomic preferences, operators)
   - Web server logs used as input
   - Heuristics used
2. [Jiang et al, KDD'2008]
   - Mining atomic preference relations using superior/inferior examples [skyline semantics]
   - Intractable problems, heuristics used
3. [Lee et al, DEXA'2008]
   - 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