Ranking Result Sets

EURO VO Markus Dolensky, Bruno Rino

Acknowledgment:

Thomas Boch, Johan Lindroos, Arnold Rots, Tobias Scholl, Andreas Wicenec, Noel Winstanley

Full Document:

`http://www.ivoa.net/internal/IVOA/IvoaDAL/ranking-v01.pdf`

The Problem

The problem to solve occurs when a query returns a large result set but ...

- **user seeks the (top) few for manual inspection**
- **and/or a service is not capable of returning all**
- **...**

Requirements

R1 express score as a number

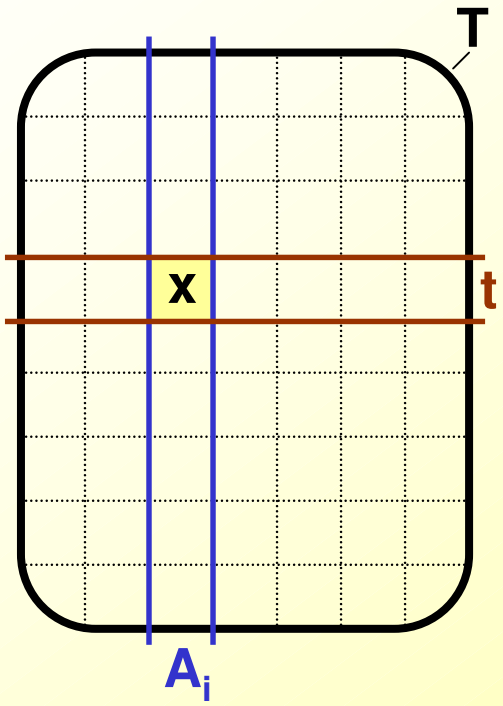
R2 context sensitivity

R3 generality

R4 no quality assessment

R5 VO compliance

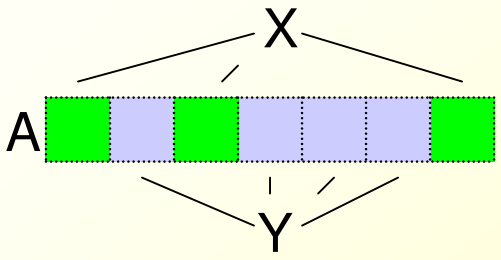
Definitions(I)



Symbol	Meaning
T	Table
t	tuple (record)
A	Attribute (column)
x	datum (cell)
i	Attribute (column) index

Definitions(II)

$$Q := C_1 \, AND \ldots AND \, C_n \Rightarrow S \subseteq T \quad (1)$$



$$X \subseteq A \quad (2)$$

$$Y = A - X \quad (3)$$

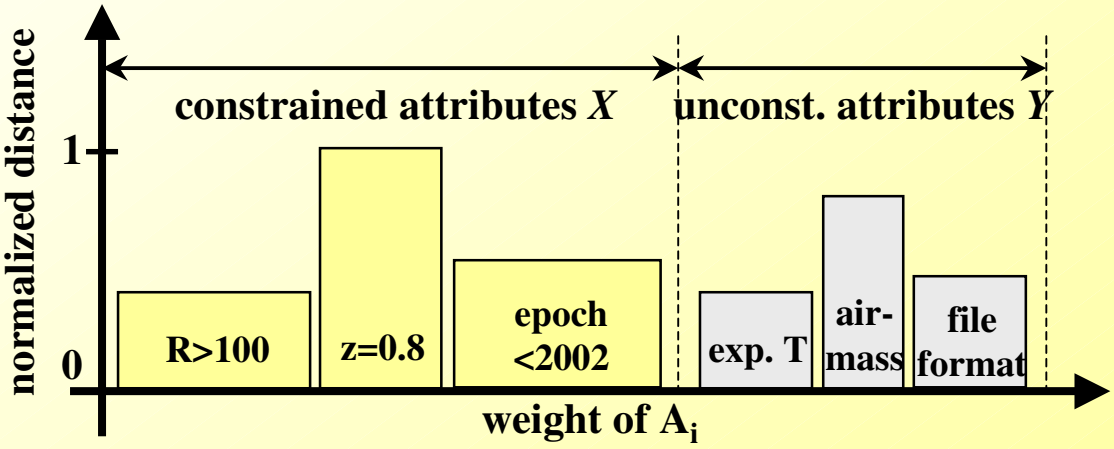
$$d_{Ai}(X_i, x) \quad (4)$$

Symbol	Meaning
X	set of constrained attributes
Y	set of unconstrained attributes
Q	Query; set of constraints
C	Constraint
S	Set of resulting tuples
d	normalized distance function

Ranking Recipes (I)

Q := { R>100, 0.6<z<1.0, epoch<2002 }

t := { R=80, z=0.6, epoch=2001, expT=400, airmass=5.7, format=FITS }



$$\forall t \in S: score = \sum_{i=1}^{n} d_{Ai}(X_i, x_i) \qquad (5)$$

Ranking Recipes (II)

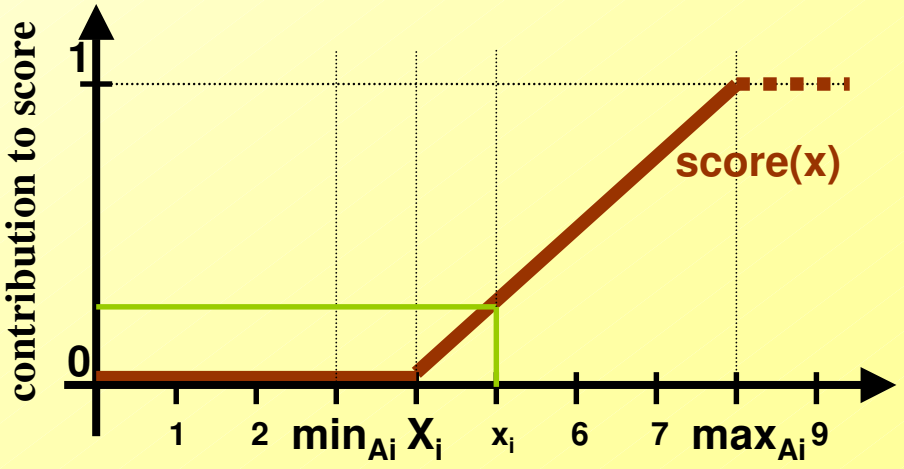
Sample recipe to measure success of constraint ‚*bigger than*‘:

$$d_{Ai}(X_i, x_i) = \begin{cases} 0 & \text{for } xi < Xi \\ \left(\dfrac{1}{\dfrac{\max(A_i) - X_i}{abs(Xi - xi)}} \right) & \text{for } Xi \le xi \le \max(A_i) \end{cases} \quad (6)$$

recipe normalizes score to range between Xi and max(Ai)

Example:

$$\{\min(A_i), \max(A_i), C_i, x_i\} = \{3,8,\geq 4,5\}$$



Ranking Recipes (III)

Further basic recipes:

$$choice(Y_i) = \begin{cases} 0.1 & for\ image / jpeg \\ 0.2 & for\ image / png \\ 0.8 & for\ application / fits \\ 1.0 & for\ application / x-votable+xml \end{cases} \quad (7)$$

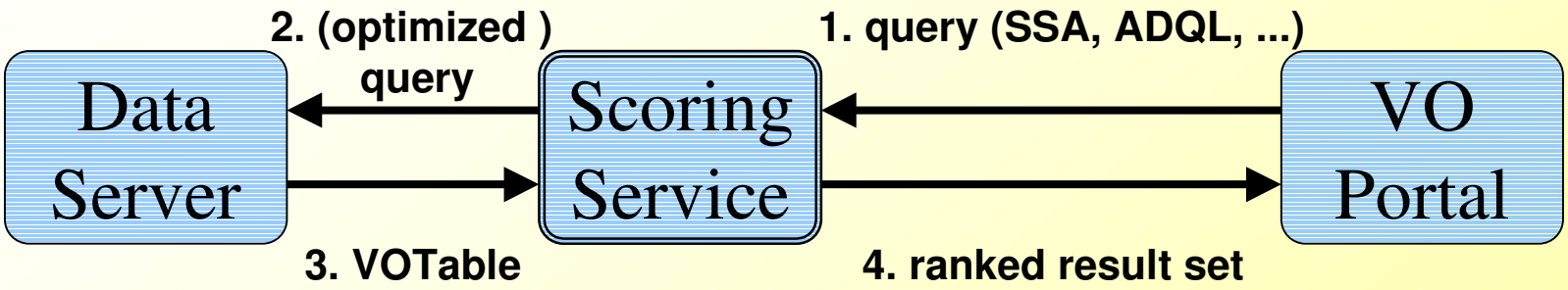
Note: choice(mime) is used for an unconstrained attribute Y_i

$$stringmatch(X_i, x_i) = \left\{ norm\left(\frac{length\ of\ search\ string(X_i)}{length\ of\ text(x_i)}\right) \right. \quad (8)$$

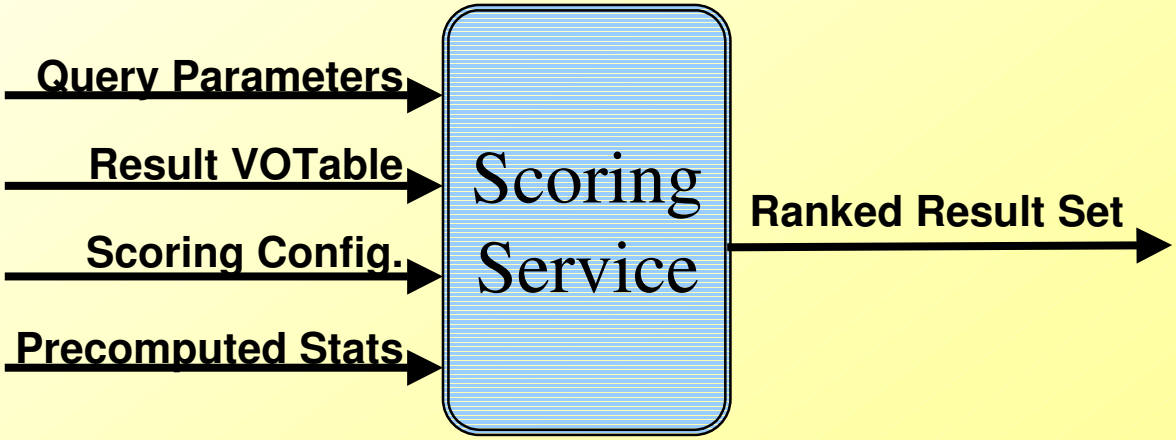
etc., etc., ...

Design (I)

Typical Runtime Scenario:



I/O of Scoring Service:



Design (II)

- domain expert to set global weights and selects d_{Ai}
- define d_{Ai} for re-occuring patterns
- modular algorithm
- shortcuts in *trivial* cases:
  no match, one perfect match, ...
- pre-compute (e.g. stats) & cache; for efficiency
- ?, ...

Advanced (future) Design Considerations

- feedback loop to adjust weights
- refine Q to reduce S
- support multiple configs. depending on context
- ...

Prototyping Efforts

Return column as proposed in SSA IF draft:

```
<FIELD name="score" datatype="float“
       ucd="meta.code.number;stat.likelihood"
       uytpe="ssa:Query.Score"/>
```

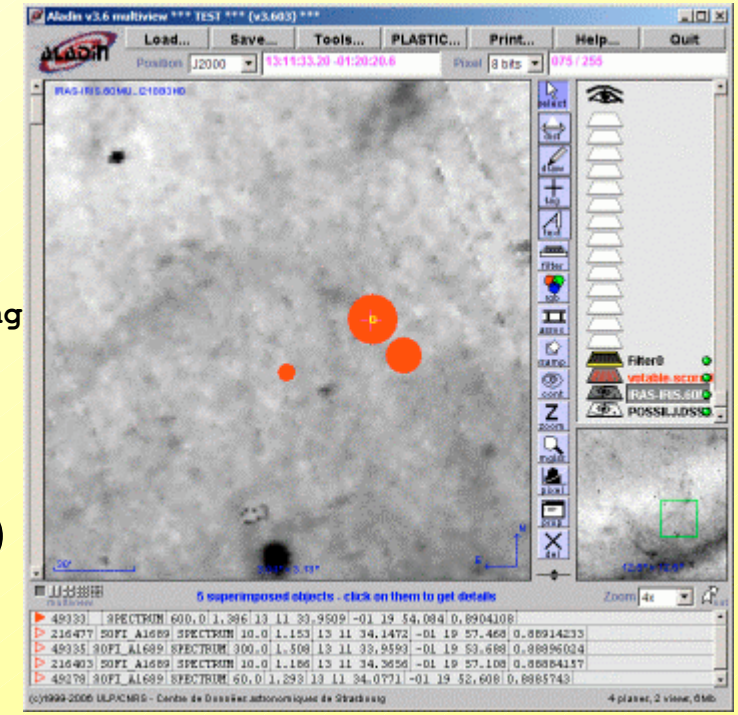
Implementation of Algorithm:

Lindroos (2005), Rino (2006+)

`http://eurovotech.org/twiki/bin/view/VOTech/SsaRanking`

Visualization of Score:

Boch (Aladin), Winstanley (Astroscope)



Plans

- define set of recipes d_{Ai} (your feedback?)
- implement ranking library (VOTech, others?)
- publish experience as IVOA note