It turns out that this is a popularity contest after all

Mike Rylander
2016 Evergreen Conference
Thursday, April 21

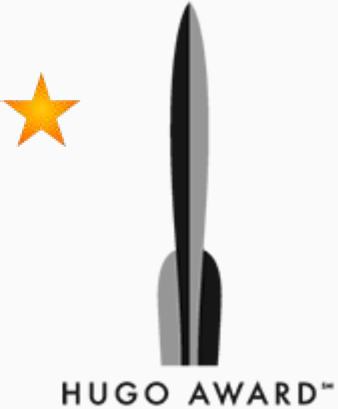
The problem

Query Relevance cannot read the user's mind.

The problem

Non-bibliographic data, some of which we don't have access to, affect a patron's thinking, consciously or unconsciously, about searching and relevance. Some factors are:

- Recency of publication
- Awards and Best-Seller lists
- Current events
- Word of mouth from other patrons
- Educational needs
- ...



The goal

1. Model the effects of these outside elements ...
2. Predict future impact of outside elements ...

... and let this data inform search result order

The goal

Popularity Ranking

The solution*

*For the subset of the mind reading problem we chose to attack

Record Badges

What to care about

Only **exceptional** records

What to care about

Only **exceptional** records

- Within a scope

What to care about

Only **exceptional** records

- Within a scope
- With comparable attributes

What to care about

Only **exceptional** records

- Within a scope
- With comparable attributes
- That matter to patrons (or, at least, seem to do so)

Record Populations

- Scope -- "where" the record earns a badge
- Population Filters -- grouping "comparable" records
- Discard Value Count -- Chop off the long tail (different kind of special)
- Inclusion Threshold Percentile -- How "special" a record is within a group

Record Populations

- Scope -- "where" the record earns a badge
  - Has copies at or below the badge owner
  - Has located URIs in scope at the badge owner
  - Circulated at or below the badge owner
  - Hold fulfilled at or below the badge owner
  - Hold placed for pickup at or below the badge owner

Record Populations

- Population Filters -- grouping "comparable" records
  - Particular bib source
  - Bibliographic attributes (anything from CCVM)
  - Has copies with a particular circulation modifier
  - Has copies that live in a particular copy location group

Record Populations

- Discard Value Count -- Chop off the long tail (different kind of special)
    - Ignore records with low values -- for most populations this removes a long tail of noise

Record Populations

- Inclusion Threshold Percentile -- How "special" a record is within a group
  - Assumes a normal distribution -- this is why "Discard Common" is important!
  - 99% or higher is useful for very large populations
  - 90% or higher for small, general populations

Primary indicators* for popularity

* For which we collect data, today

- Bibliographic Record Age (days)
- Publication Age (years, really)
- On-Line Availability
- Percent of Time Circulating

Primary indicators* for popularity

* For which we **don't** have or collect data, today

- Club and association awards
- 3rd party ratings
- Patron or staff ratings
- Purchasing reasons and decisions

Primary indicators* for popularity-ish metrics

* For which we **don't** have data, today

- Local educational needs
- Subject locality
- Outreach efforts

Secondary indicators* for popularity

* For which we collect data, today

- Circulating/Total Ratio
- Current Hold Count
- Circulations Over Time
- Current Circulation Count
- Holds/Total Ratio
- Holds/Holdable Ratio
- Holds Filled Over Time
- Holds Requested Over Time

Popularity Parameters ... look familiar?

- Circulating/Total Ratio
- Current Hold Count
- Circulations Over Time
- Current Circulation Count
- Bibliographic Record Age (days)
- Publication Age (years, really)
- Holds/Total Ratio
- Holds/Holdable Ratio
- Holds Filled Over Time
- Holds Requested Over Time
- On-Line Availability
- Percent of Time Circulating

Popularity Parameters ... look familiar?

Temporal(-ish)

- Circulations Over Time
- Current Circulation Count
- Bibliographic Record Age (days)
- Publication Age (years, really)
- Holds Filled Over Time
- Holds Requested Over Time
- Percent of Time Circulating

Point in time(-ish)

- Holds/Total Ratio
- Holds/Holdable Ratio
- On-Line Availability
- Circulating/Total Ratio
- Current Hold Count

Recency Scaling

For temporal parameters, recent == important

- Age horizon
- Importance horizon
- Importance interval
- Importance scale

Fixed ratings

For staff-curated sets:

- Copy location groups
- Specialized CCVM values
- etc...

... configuration can supply a fixed rating to every record in the population.

Global Knobs

- Relevance adjustment scale global flag
- Default sort selection global flag

Let's add a new one

Copy Count

- Raw value collection
- Define the population
- Configure the badge

Let's add a new one

Copy Count - Raw value collection

- Implemented as a stored procedure
- Simple API
  - **Input:** badge ID
  - **Output:** set of bibliographic record ID, raw popularity value

Required slide full of code: 1

Let's add a new one

```
CREATE OR REPLACE FUNCTION rating.copy_count(badge_id INT)
    RETURNS TABLE (record INT, value NUMERIC) AS $f$
DECLARE
    badge   rating.badge_with_orgs%ROWTYPE;
BEGIN

    SELECT * INTO badge FROM rating.badge_with_orgs WHERE id = badge_id;

    PERFORM rating.precalc_bibs_by_copy(badge_id);

    DELETE FROM precalc_copy_filter_bib_list WHERE id NOT IN (
        SELECT id FROM precalc_filter_bib_list
            INTERSECT
        SELECT id FROM precalc_bibs_by_copy_list
    );
    ANALYZE precalc_copy_filter_bib_list;

    RETURN QUERY
     SELECT f.id::INT AS bib,
            COUNT(f.copy)::NUMERIC
      FROM  precalc_copy_filter_bib_list f
            JOIN asset.copy cp ON (f.copy = cp.id)
            JOIN asset.call_number cn ON (cn.id = cp.call_number)
      WHERE cn.owning_lib = ANY (badge.orgs) GROUP BY 1;

END;
$f$ LANGUAGE PLPGSQL STRICT;
```

Required slide full of code: 2

Let's add a new one

```
CREATE OR REPLACE FUNCTION rating.copy_count(badge_id INT)
    RETURNS TABLE (record INT, value NUMERIC) AS $f$
DECLARE
    badge   rating.badge_with_orgs%ROWTYPE;
BEGIN
    -- Most raw value calculation procedures will need the badge scope org units
    SELECT * INTO badge FROM rating.badge_with_orgs WHERE id = badge_id;

    PERFORM rating.precalc_bibs_by_copy(badge_id); -- Bibs with copies for this badge's scope.

    DELETE FROM precalc_copy_filter_bib_list WHERE id NOT IN ( -- Ignore copies not on bibs in the population
        SELECT id FROM precalc_filter_bib_list -- We get this from an earlier step...
            INTERSECT
        SELECT id FROM precalc_bibs_by_copy_list -- and this is from the PERFORM above.
    );
    ANALYZE precalc_copy_filter_bib_list; -- Correct stats so we get a good plan.

    RETURN QUERY -- And, finally, get the copy count per bib of copies in-scope to the badge.
     SELECT f.id::INT AS bib,
            COUNT(f.copy)::NUMERIC
      FROM  precalc_copy_filter_bib_list f -- This is our precalculated bib+copy list to consider.
            JOIN asset.copy cp ON (f.copy = cp.id)
            JOIN asset.call_number cn ON (cn.id = cp.call_number)
      WHERE cn.owning_lib = ANY (badge.orgs) GROUP BY 1; -- We use owning_lib instead of circ_lib ... floating!

END;
$f$ LANGUAGE PLPGSQL STRICT;
```

Required slide full of code: 3

Let's add a new one

```
CREATE OR REPLACE FUNCTION rating.copy_count(badge_id INT)
    RETURNS TABLE (record INT, value NUMERIC) AS $f$
DECLARE
    badge   rating.badge_with_orgs%ROWTYPE;
BEGIN

    SELECT * INTO badge FROM rating.badge_with_orgs WHERE id = badge_id;

    PERFORM rating.precalc_bibs_by_copy(badge_id);

    DELETE FROM precalc_copy_filter_bib_list WHERE id NOT IN (
        SELECT id FROM precalc_filter_bib_list
            INTERSECT
        SELECT id FROM precalc_bibs_by_copy_list
    );
    ANALYZE precalc_copy_filter_bib_list;

    RETURN QUERY
     SELECT f.id::INT AS bib,
            COUNT(f.copy)::NUMERIC
      FROM  precalc_copy_filter_bib_list f
            JOIN asset.copy cp ON (f.copy = cp.id)
            JOIN asset.call_number cn ON (cn.id = cp.call_number)
      WHERE cn.owning_lib = ANY (badge.orgs) GROUP BY 1;

END;
$f$ LANGUAGE PLPGSQL STRICT;
```

```
INSERT INTO rating.popularity_parameter (name, func, require_percentile) VALUES ('Copy Count', 'rating.copy_count', TRUE);
```

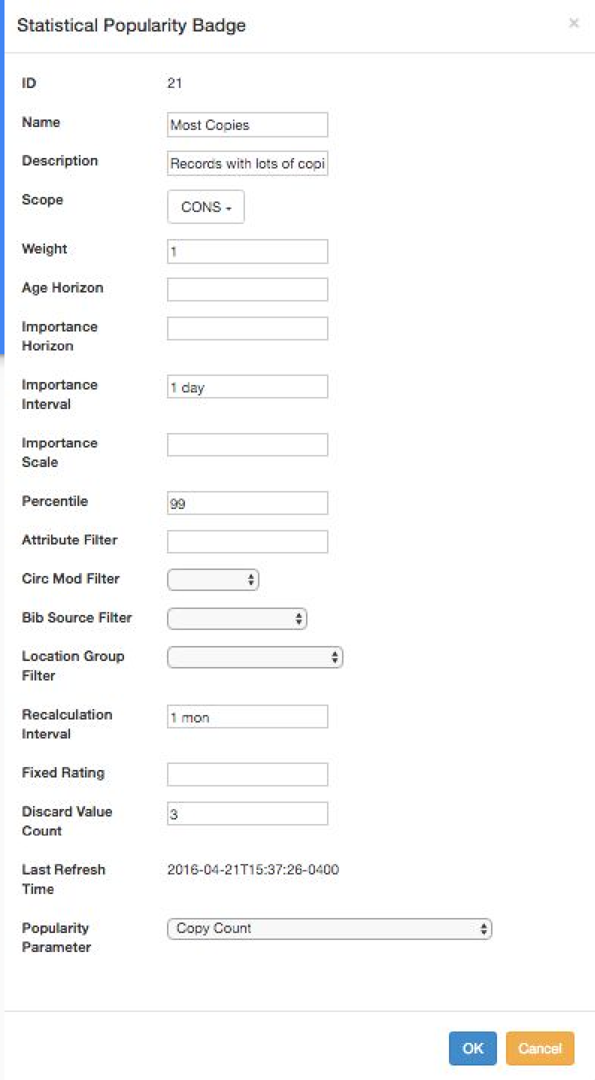
Let's add a new one

Copy Count - Define population

- What is the org unit scope of the badge? Everywhere.
- Bib attribute, circ modifier, or copy location filters? Nah...
- Ignore bibs with the three lowest number of copies: 3 (probably 1, 2, and 3)
- Limit to just those with lots: 99th percentile

Let's add a new one

Copy Count - Configure badge

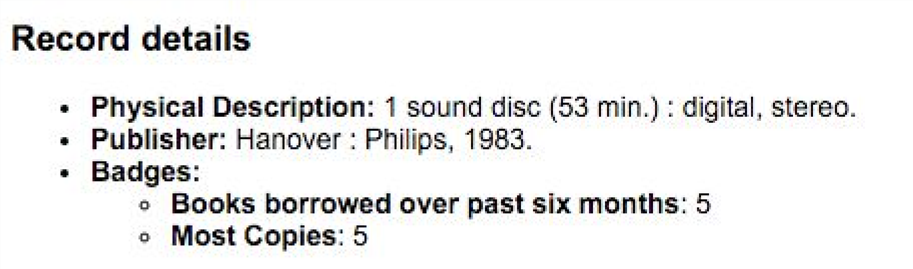


Statistical Popularity Badge

ID	21
Name	Most Copies
Description	Records with lots of copi
Scope	CONS
Weight	1
Age Horizon	
Importance Horizon	
Importance Interval	1 day
Importance Scale	
Percentile	99
Attribute Filter	
Circ Mod Filter	
Bib Source Filter	
Location Group Filter	
Recalculation Interval	1 mon
Fixed Rating	
Discard Value Count	3
Last Refresh Time	2016-04-21T15:37:26-0400
Popularity Parameter	Copy Count

Let's add a new one

Copy Count - See it work!



Record details

- **Physical Description:** 1 sound disc (53 min.) : digital, stereo.
- **Publisher:** Hanover : Philips, 1983.
- **Badges:**
  - **Books borrowed over past six months**: 5
  - **Most Copies**: 5

Thanks!

Mike Rylander
2016 Evergreen Conference
Thursday, April 21