Interesting Event Detection through Hall of Fame Rankings

Alvanaki, Foteini, et al.
Saarland University
ACM SIGMOD Workshop on Databases and Social Networks, 2013

11 August, 2014
Jaehwan Lee
Outline

- Introduction
- Study on Entity Rankings
- Framework
- Event Ranking
- Experiments
- Conclusion
Introduction

- Some rankings are highly subjective
  - E.g. top-10 movies through user votes

- A large amount of rankings can be generated and refreshed fully automatically, also

- We call the considered rankings “Halls of Fame”
  - not only the top portion of a ranking
  - but entity centric
  - have meaningful constraints and ranking criteria
  - competitive to be of interest to users
Key Idea

- If there is changes on ranking, there would be an interesting event
Study on Entity Rankings

- Ranking size
  - crawled from ranker.com on Nov 11\textsuperscript{th} and 12\textsuperscript{th}, 2012
  - intended ranking size
    - usually specified in the title like “top-10 movies”
    - no strict control, however
  - a shift toward rankings longer than the authors intention
    - users can propose own entities to be included

Figure 1: Log-log plots of the number of rankings with respect to their real (left) and intended (right) size.
Study on Entity Rankings

- Ranking popularity
  - the most popular ranking received around 6 million views
  - 27 rankings received over 1 million views in total
  - 118 rankings with less than 100 views
  - further inspection of the creation time showed
    - a linear dependency between the age of a ranking and the number of views it received, on average

Figure 2: Log-log plot of the number of rankings with respect to their views.
Study on Entity Rankings

- Specificity
  - a drastic decline in views with increasing number of constraints
  - E.g. the best actors in the world vs. in France

**Figure 3:** The impact of the number of constraints on the number of views by users, for different topics.
Framework

- Queries
  - executing standard OLAP-style SQL aggregation queries
  - expert user annotate entity types, numerical or categorical attributes

```sql
SELECT entity attribute, aggregate(numeric attribute)
FROM dataTable
WHERE predicate
GROUP BY entity attribute
ORDER BY aggregate(numeric attribute) ASC|DESC
LIMIT K
```

- Maintenance
  - requires monitoring Hall of Fame rankings for changes
  - use of techniques from materialized view maintenance
Event Ranking

- Scoring model
  - consider small rank improvements of a player throughout an entire season
    - each individual rank improvement of, say 1 rank each,
    - might not be noteworthy compared to its overall improvement of many ranks
  - hence, accumulation of individual rank improvements is considered

- Two Characteristics
  - Dynamic Characteristics
  - Static Characteristics
    - Entropy
    - Selectivity
Dynamic Characteristics

- to compute a score describing the quality of the jump from rank \( r \) to \( r' \)
- E.g. ‘from 84 to 65’ not ‘from 100 to 65’ on \( e_2 \) of Figure 4

**Figure 4:** Information at hand for computing the dynamic score component
Event Ranking

- Dynamic Characteristics Score
  - rank $b$ is a parameter
  - gives weight 1 to all rank improvements above rank $b$
  - punishes lower ranks with a weight of $1/\log_b(.)$
  - results in values between $1/\log_b(K)$ and $K$
    - for the smallest noticeable rank improvement from rank $K+1$ to $K$
    - for the biggest noticeable improvement from rank $K+1$ to 1

\[
rs(\{(r_i, r'_i)| i \in \mathbb{N}\}) := \sum_{r'_i \leq b} (r_i - r'_i) + \sum_{r'_i > b} \frac{(r_i - r'_i)}{\log_b(r'_i)}
\]
Event Ranking

- Static Characteristics
  - Entropy
    - each Hall of Fame predicate consists of a set of attributes used in categorical attributes bindings (E.g. team='Phoenix')
    - inspect all possible instantiations $I_i$ of the set $(\text{attr}_1, \text{attr}_2, \ldots \text{attr}_n)$
    - for each instantiation $I_i$, count the number of tuples that satisfy it
    - and divide with the total number of tuples in the table
  - Selectivity
    - the quality of the specific instantiation used in a query
    - the smaller the fraction of the table that qualifies for a query result is the lower the score of such a query is
Event Ranking

- Putting it All Together
  - use lexicographical ordering (not plain version)
  - $u \prec_{lt} v$, iff $(u + 1/2n) \prec_{lt} v$ (n is the number of coarse groups)
  - $u, v$ are considered as normalized in the range $[0,1]$
    - 0 denote the lowest possible score
    - 1 denote the largest possible score
  - final score vector goes in order like below
    - (SELECTIVITY, DYNAMIC SCORE, ENTROPY)

![Diagram showing lexicographic tradeoffs]

Figure 5: Ranges in Lexicographic Tradeoffs
Event Ranking

- Lexicographical ordering
  - in case, \( u \prec_{lt} v \text{ iff } 2u \leq v \text{ and } u \succ_{lt} v \text{ if } v > 2u \)
  - then, \((7, 3, 6)\) is larger than \((3, 8, 4)\)
  - on the other hand, \((7, 2, 6)\) is smaller than \((5, 6, 2)\)
Experiments

- uses Java 1.7 and Postgresql 9.1
- Basketball statistics obtained from databasebasketball.com
- 4,000 players in the last 65 years
- entity attributes: player, team
- categorical attributes: league, team, age

<table>
<thead>
<tr>
<th>column name</th>
<th>aggr. function</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>turnovers</td>
<td>sum</td>
<td>ascending</td>
</tr>
<tr>
<td>rebounds</td>
<td>sum</td>
<td>descending</td>
</tr>
<tr>
<td>assists</td>
<td>sum</td>
<td>descending</td>
</tr>
<tr>
<td>field goals percentage</td>
<td>avg</td>
<td>both</td>
</tr>
<tr>
<td>points per game</td>
<td>avg</td>
<td>descending</td>
</tr>
<tr>
<td>points</td>
<td>sum</td>
<td>descending</td>
</tr>
<tr>
<td>three-points per game</td>
<td>sum</td>
<td>descending</td>
</tr>
<tr>
<td>three throws made</td>
<td>sum</td>
<td>descending</td>
</tr>
</tbody>
</table>

Table 1: Sample user generated description of how to use numeric attributes. Used in the experimental evaluation.
Experiments

- Update Modelling
  - it lacks “live” data
  - create 10 update statements where the intermediate values are calculated differently depending on whether summation or averaging is used
    - for the summation, generate values using $N \sim (0.5, 0.2^2)$ until have 9 values in range $(0,1)$ values are sorted in ascending order
    - for the averaging, generate values using $N \sim (0, 0.1^2)$ until have 9 values in range $(-1,1)$
    - the value of the $i^{th}$ update equals to the $i^{th}$ generated value times the final (actual) value
  - to keep the evaluation tractable, used the years 2005 to 2011
  - generated a total of 275,580 updates
Experiments

- User Study on Ranking Quality
  - present to users 9 rankings of events
  - Hall of Fame size $K = 20$, the parameter $b = 5$
  - for each user in a couple of minutes
  - each ranking consists of 3 events
    - one put by our algorithm at rank one, rank five and rank ten
  - for each such event, manually created a statement
    - E.g. “Team Philadelphia advanced from position 19 to position 17 in the NBA top field game points scored list.”
  - randomly order these sentences
  - ask users to order them using a scale from 1 to 3, allowing ties
Experiments

- Results
  - a good trend of diagonal elements being the largest

<table>
<thead>
<tr>
<th></th>
<th>Assigned</th>
<th>Ratings by Users</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rating 1</td>
<td>Rating 2</td>
</tr>
<tr>
<td>Event at Rank 1</td>
<td>49</td>
<td>34</td>
</tr>
<tr>
<td>Event at Rank 5</td>
<td>24</td>
<td>52</td>
</tr>
<tr>
<td>Event at Rank 10</td>
<td>29</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 2: Raw results of our user study

- for three pairs : (1, 5), (1, 10), (5, 10)
  - right order +1, reverser order -1, otherwise 0 point
  - sum up this scores over all pairs, over all event rankings and over all users and achieve 66 points
  - as for the best 148 points, for the worst 148 points
  - it goes 72.39% accuracy (66+148+1/148+148+1)
Conclusion

- We conducted a carefully designed experimental evaluation using real-world data obtained from a basketball statistics website.
- Conducted user study showed that the ordering of events using the proposed lexicographic-tradeoff-based ranking is in line with user expectations.