Expressing and Combining Flexible Recommendations in CourseRank

Georgia Koutrika
Stanford University
Introduction

- Social sites on the Web thrive

- Open - user participation - simple interactions/tasks
  uploading, reviewing, poking, socializing, sharing ...

- CourseRank in Stanford:
  A Closed-Community, Special-Purpose Social Site
Key Highlights

A  Courserank’s special features

B  Flexible recommendations

C  Demo
An **educational** and **social** site for Stanford students to:

- Evaluate courses
- Browse courses, instructors, books
- Plan their academic program
- Interact with each other
- Ask and answer questions
Special features

1. Well-defined closed community
2. Multiple constituencies
3. Special-purpose tools
4. Official and social data
CourseRank in numbers:

- 1.5 year
- 14,557 students
- 9,532 instructors
- 18,951 courses
- 102,152 ratings
- 3,172 text reviews
Challenges and Opportunities

In CourseRank, interacting with rich data is a great challenge!
Recommendation systems

Basic Idea

- Two types of entities: **Users** × **Items**
- Utility of item \( i \) for user \( u \) is represented by some rating \( r \)
- Estimate unknown ratings based on the known ratings: \( R: \text{Users} \times \text{Items} \rightarrow \text{Rating} \)

How?

- **Content-based approaches:**
  Find items similar to those \( u \) has liked in the past
- **Collaborative filtering approaches:**
  Find items similar people have liked
- **Hybrid approaches**
Recommendations in CourseRank

Typical recommendations

- courses similar to those the user has liked
- courses that students with similar ratings have liked

But, we have lots of rich data!

Courses (CourseID, DeplID, Title, Description, Units)
Students (SuID, Name, Class, GPA)
Comments (SuID, CourseID, Year, Term, Text, Rating, Date)
StudentHistory (SuID, CourseID, Year, Term, Grade)
And many more tables:

Instructors, StudentStudies, Departments, StudentPlans, Books, ...
Recommendations in CourseRank

More recommendations

- courses based on my grades
- the best quarter to take an artificial intelligence course
- a major based on my grades
- dance courses that CS students with similar tastes take
- math courses that second-year students with similar grades and tastes take

Current Limitations

- Limited expressivity
- Fixed recommendations
- Hard-wired methods
Roadmap

1. Recommendation Systems
2. FleXRecs Framework
3. Recommendation Engine
4. Experiments
5. Summary
Expressing Flexible Recommendations

Our approach

User Interface

Monolithic system

Recommendations

Data Access

database
Expressing Flexible Recommendations

Our approach

- **Decouple** the definition of a recommendation strategy from the execution

- **Declaratively express** a recommendation approach as a high-level workflow

- Execute any recommendation workflow **using the same engine**
Recommendations as Queries

A recommendation approach

- is defined declaratively as a high-level workflow over relational data
- combines traditional relational operators and new operators
  - recommend operator
  - extend operator
  - blend operator
Special Operators

R \triangleright_{\text{cf}, a} S := \{ (r, v) \mid r \in R \land v := a[\text{cf}](r, S) \}

Comparison function

tuple 1 \xrightarrow{\text{cf}} \text{score}
tuple 2

Aggregation function

score 1
... score n

score 1
... score n

score
Special Operators

Example 1: Find similar courses (to-1 comparisons)

Courses A

<table>
<thead>
<tr>
<th>CourseID</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2</td>
<td>Programming: Part 2</td>
</tr>
<tr>
<td>c3</td>
<td>Advanced Programming Techniques</td>
</tr>
<tr>
<td>c4</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>c5</td>
<td>Reasoning in Artificial Intelligence</td>
</tr>
</tbody>
</table>

Courses B

<table>
<thead>
<tr>
<th>CourseID</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>Programming: Part 1</td>
</tr>
</tbody>
</table>

R_Courses

<table>
<thead>
<tr>
<th>CourseID</th>
<th>Title</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2</td>
<td>Programming: Part 2</td>
<td>0.5</td>
</tr>
<tr>
<td>c3</td>
<td>Advanced Programming Techniques</td>
<td>0.2</td>
</tr>
<tr>
<td>c4</td>
<td>Artificial Intelligence</td>
<td>0</td>
</tr>
<tr>
<td>c5</td>
<td>Reasoning in Artificial Intelligence</td>
<td>0</td>
</tr>
</tbody>
</table>

$\text{Jaccard}(r, s)[A] = \frac{|r[A] \cap s[A]|}{|r[A] \cup s[A]|}$

e.g., $\text{Jaccard}(c1, c2)[\text{Title}] = \frac{2}{4} = 0.5$
Example 2: Find similar courses (to-many comparisons)

Courses A

<table>
<thead>
<tr>
<th>CourseID</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2</td>
<td>Programming: Part 2</td>
</tr>
<tr>
<td>c3</td>
<td>Advanced Programming Techniques</td>
</tr>
<tr>
<td>c4</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>c5</td>
<td>Reasoning in Artificial Intelligence</td>
</tr>
</tbody>
</table>

Courses B

<table>
<thead>
<tr>
<th>CourseID</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>Programming: Part 1</td>
</tr>
<tr>
<td>c6</td>
<td>Advanced Java Programming</td>
</tr>
</tbody>
</table>

R_Courses

<table>
<thead>
<tr>
<th>CourseID</th>
<th>Title</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2</td>
<td>Programming: Part 2</td>
<td>0.66</td>
</tr>
<tr>
<td>c3</td>
<td>Advanced Programming Techniques</td>
<td>0.7</td>
</tr>
<tr>
<td>c4</td>
<td>Artificial Intelligence</td>
<td>0</td>
</tr>
<tr>
<td>c5</td>
<td>Reasoning in Artificial Intelligence</td>
<td>0</td>
</tr>
</tbody>
</table>

Jaccard[Title], sum
### Special Operators

Moving towards more complex comparisons

#### Students

<table>
<thead>
<tr>
<th>SuID</th>
<th>Name</th>
<th>Class</th>
<th>GPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John</td>
<td>2010</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Mary</td>
<td>2010</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>Alan</td>
<td>2010</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>Kim</td>
<td>2010</td>
<td>7.5</td>
</tr>
</tbody>
</table>

#### Comments

<table>
<thead>
<tr>
<th>SuID</th>
<th>CourseID</th>
<th>Year</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>c1</td>
<td>2007</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>c2</td>
<td>2007</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>c3</td>
<td>2008</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>c1</td>
<td>2007</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>c2</td>
<td>2007</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>c6</td>
<td>2006</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>c2</td>
<td>2008</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>c2</td>
<td>2008</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>c6</td>
<td>2006</td>
<td>9</td>
</tr>
</tbody>
</table>
Moving towards more complex comparisons

<table>
<thead>
<tr>
<th>Student</th>
<th>Name</th>
<th>Class</th>
<th>GPA</th>
<th>Comments(CourseID, Year, Rating)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John</td>
<td>2010</td>
<td>10</td>
<td>c1: 2007, 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>c2: 2007, 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>c3: 2008, 9</td>
</tr>
<tr>
<td>2</td>
<td>Mary</td>
<td>2010</td>
<td>7</td>
<td>c2: 2008, 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>c6: 2006, 9</td>
</tr>
<tr>
<td>3</td>
<td>Alan</td>
<td>2010</td>
<td>9</td>
<td>c2: 2008, 5</td>
</tr>
<tr>
<td>4</td>
<td>Kim</td>
<td>2010</td>
<td>7.5</td>
<td>c1: 2007, 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>c2: 2007, 8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>c6: 2006, 8</td>
</tr>
</tbody>
</table>
## Students

<table>
<thead>
<tr>
<th>SuID</th>
<th>Name</th>
<th>Class</th>
<th>GPA</th>
<th>Comments(CourseID, Year, Rating)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Mary</td>
<td>2010</td>
<td>7</td>
<td>c2 2008 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>c6 2006 9</td>
</tr>
<tr>
<td>3</td>
<td>Alan</td>
<td>2010</td>
<td>9</td>
<td>c2 2008 5</td>
</tr>
<tr>
<td>4</td>
<td>Kim</td>
<td>2010</td>
<td>7.5</td>
<td>c1 2007 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>c2 2007 8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>c6 2006 8</td>
</tr>
</tbody>
</table>

Euclidean[Comments]

## Special Operators

<table>
<thead>
<tr>
<th>SuID</th>
<th>Name</th>
<th>Class</th>
<th>GPA</th>
<th>Comments(CourseID, Year, Rating)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John</td>
<td>2010</td>
<td>10</td>
<td>c1 2007 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>c2 2007 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>c3 2008 9</td>
</tr>
</tbody>
</table>
Special Operators

**Extend**

\[ R \cdot \varepsilon \cdot S := \{ (r, v) \mid r \in R \land v := \pi(r) \} \]

\[
\begin{array}{cccc}
A & B & C & D \\
\hline
\cdot & \cdot & \cdot & \cdot \\
\end{array}
\]

\[ \varepsilon \]

\[
\begin{array}{cccc}
A & E & F & G \\
\hline
\cdot & \cdot & \cdot & \cdot \\
\end{array}
\]

\[
\begin{array}{cccc}
A & B & C & D \\
\hline
\cdot & \cdot & \cdot & \cdot \\
\end{array}
\]

\[ S(E, F, G) \]
### Blend

**Definition:**

\[ R \xrightarrow{\beta_M} S := \{(r, v) | r \in R \cup S \land v := M[R,S](r)\} \]

**Diagram:**

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| A   | B   | C   | D   |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     |     |     |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     |     |     |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     |     |     |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+

\( R \xrightarrow{\beta_M} \)

```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| A   | B   | C   | D   |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     |     |     |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     |     |     |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     |     |     |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+

\( R \cup S \)

```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| A   | B   | C   | D   |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     |     |     |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     |     |     |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     |     |     |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     |     |     |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+

\( M[R,S](r) \)

```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| A   | B   | C   | D   | bscore |     |     |     |
+-----+-----+-----+-----+---------+-----+-----+-----+
|     |     |     |     |         |     |     |     |
+-----+-----+-----+-----+---------+-----+-----+-----+
|     |     |     |     |         |     |     |     |
+-----+-----+-----+-----+---------+-----+-----+-----+
|     |     |     |     |         |     |     |     |
+-----+-----+-----+-----+---------+-----+-----+-----+
|     |     |     |     |         |     |     |     |
+-----+-----+-----+-----+---------+-----+-----+-----+

\( \beta_M \)

```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| A   | B   | C   | D   |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     |     |     |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     |     |     |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     |     |     |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     |     |     |     |     |     |     |
+-----+-----+-----+-----+-----+-----+-----+-----+
```
## Special Operators

**Example:**

### Similar_Courses

<table>
<thead>
<tr>
<th>CourseID</th>
<th>Title</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2</td>
<td>Programming: Part 2</td>
<td>10</td>
</tr>
<tr>
<td>c5</td>
<td>AI Techniques</td>
<td>7</td>
</tr>
<tr>
<td>c10</td>
<td>Graphics</td>
<td>9</td>
</tr>
</tbody>
</table>

### Blended_Courses

<table>
<thead>
<tr>
<th>CourseID</th>
<th>Title</th>
<th>bscore</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2</td>
<td>Programming: Part 2</td>
<td>10</td>
</tr>
<tr>
<td>c5</td>
<td>AI Techniques</td>
<td>7</td>
</tr>
<tr>
<td>c10</td>
<td>Graphics</td>
<td>8</td>
</tr>
<tr>
<td>c22</td>
<td>Compilers</td>
<td>8</td>
</tr>
</tbody>
</table>

### Required_Courses

<table>
<thead>
<tr>
<th>CourseID</th>
<th>Title</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2</td>
<td>Programming: Part 2</td>
<td>10</td>
</tr>
<tr>
<td>c10</td>
<td>Graphics</td>
<td>7</td>
</tr>
<tr>
<td>c22</td>
<td>Compilers</td>
<td>8</td>
</tr>
</tbody>
</table>

$\beta_{avg[\text{score}]}$
Recommendation queries

Recommendation query :=
new operators + traditional relational operators
Recommendation queries

Recommendation query :=
new operators + traditional relational operators

Content-based recommendations

Collaborative filtering recommendations
Recommendation queries

Recommendation query := new operators + traditional relational operators

Content-based recommendations

Collaborative filtering recommendations
Recommendation queries
Parameterized queries allow on-the-fly personalization
Roadmap

Recommendation Systems

FleXRecs Framework

Recommendation Engine

Experiments

Summary
Recommendation Engine

users

invoke

define

designer

Query Manager → Query Parser → Plan Generator → Recommendation Generator

Functions

DB Engine

X

Courses

Y

inv_Euclidean[Comments, CourseID, Rating]

Z

Identify[CourseID, Comments, Rating], W_Avg[Score]

T\{(StudenID, CourseID, Rating)\}

Students

Comments
CREATE TEMPORARY TABLE temp
SELECT t1.SuID, 1/SQRT(SUM((t1.Rating - t2.Rating) * (t1.Rating - t2.Rating))) as score
FROM Comments t1, Comments t2
WHERE t1.CourseID = t2.CourseID AND t2.SuID = 444 AND t1.SuID <> 444
GROUP BY t1.SuID

CREATE TEMPORARY TABLE temp2
SELECT t1.*, score FROM Comments t1, temp
WHERE t1.SuID = temp.SuID;
SELECT Courses.*, SUM(score*rating)/SUM(score) AS CScore
FROM temp2, Courses
WHERE temp2.CourseID=Courses.CourseID
GROUP BY CourseID ORDER BY CScore
Recommendation Engine

Plan Generator

1. “Chop” the tree
2. Each subtree maps to a query block
Recommendation Engine

Plan Generator

Block A

CREATE TEMPORARY TABLE temp

SELECT  t1.SuID, 1/SQRT(SUM((t1.Rating − t2.Rating) ∗ (t1.Rating − t2.Rating)))  as  score
FROM     Comments t1, Comments t2
WHERE   t1.CourseID = t2.CourseID AND t2.SuID = 444 AND t1.SuID <> 444
GROUP BY t1.SuID
CREATE TEMPORARY TABLE temp2

SELECT t1.*, score
FROM Comments t1, temp
WHERE t1.SuID = temp.SuID;

SELECT Courses.*, SUM(score*rating)/SUM(score) AS CScore
FROM temp2, Courses
WHERE temp2.CourseID=Courses.CourseID
GROUP BY CourseID
ORDER BY CScore
Roadmap

- Recommendation Systems
- FleXRecs Framework
- Recommendation Engine
- Experiments
- Summary
Recommendations in use

The engine supports multiple recommendations in CourseRank.

**Collaborative filtering**

**Majors recommendations**

**Related courses**
Recommendations in use

Collaborative filtering

![Collaborative filtering graph](image)

Related courses

![Related courses graph](image)
In summary ...

- Recommendation processes as **queries over relational data**
- **Operators** for expressing and composing recommendations
- Recommendations **leverage the database query optimizer**
- System prototype **in production use**
- **Novel** recommendation paradigms
Research challenges

- **Optimization** of multiple recommendations
- **Different implementations** of operators
- Flexible **recommendation query language**
- Flexible recommendation **user interface**