Capturing knowledge of user preferences with recommender systems

Stuart E. Middleton
David C. De Roure, Nigel R. Shadbolt
Intelligence, Agents and Multimedia Research Group
Dept of Electronics and Computer Science
University of Southampton
United Kingdom

Email: sem99r@ecs.soton.ac.uk

Capturing knowledge of user preferences with recommender systems

- Introduction to recommender systems
- Knowledge capture of user profiles
- Quickstep architecture and approach
- Issues arising from Quickstep evaluation
- Foxtrot architecture and approach
- Future work

Capturing knowledge of user preferences with recommender systems

- **Introduction to recommender systems**

  WWW information overload
  Recommender systems
  - Collaborative filters (several commercial examples)
  - Content-based filters
  - Hybrid filters

  A real world problem domain
  - On-line research paper recommendation for researchers
  - Evaluation of users in a real work setting

  Knowledge acquisition must be unobtrusive
  - System must not interfere with normal work practice
  - Monitoring should be unobtrusive
  - Feedback requested only when recommendations checked

Capturing knowledge of user preferences with recommender systems

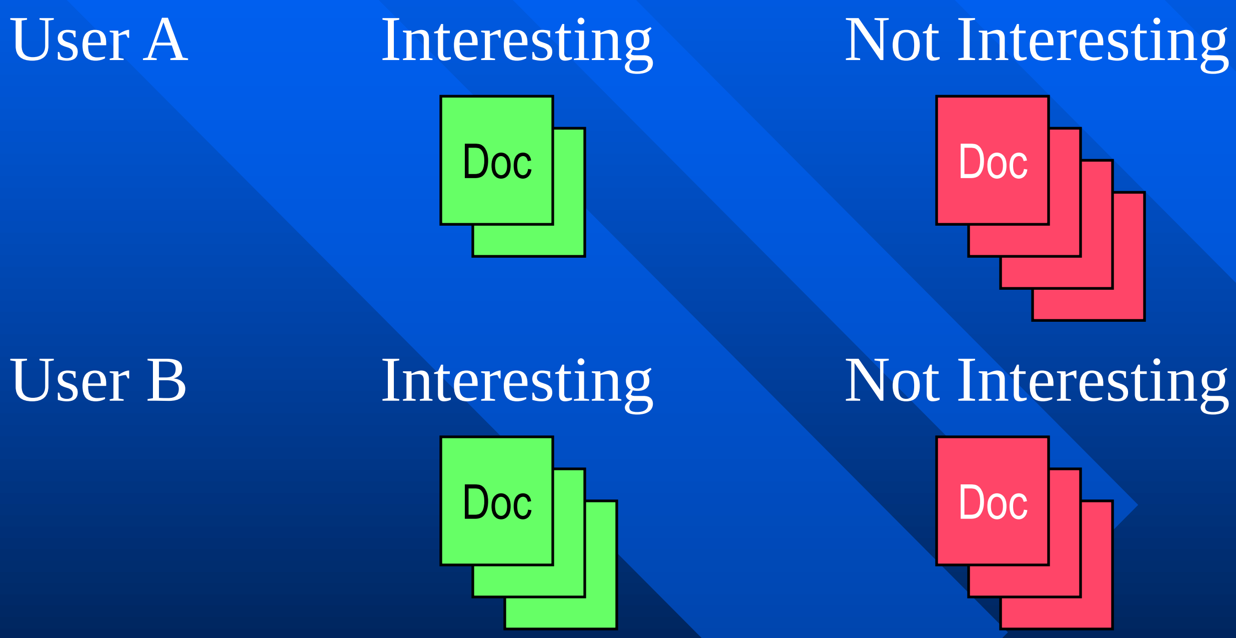
- **Knowledge capture of user profiles**

  Binary class profile representation
  - 'Interesting' and 'not interesting' examples
  - Time-decay function favours recent examples
  - Machine learning classifies new information (e.g. TF-IDF)

Capturing knowledge of user preferences with recommender systems

- **Knowledge capture of user profiles**

Binary class profile representation



Capturing knowledge of user preferences with recommender systems

- **Knowledge capture of user profiles**

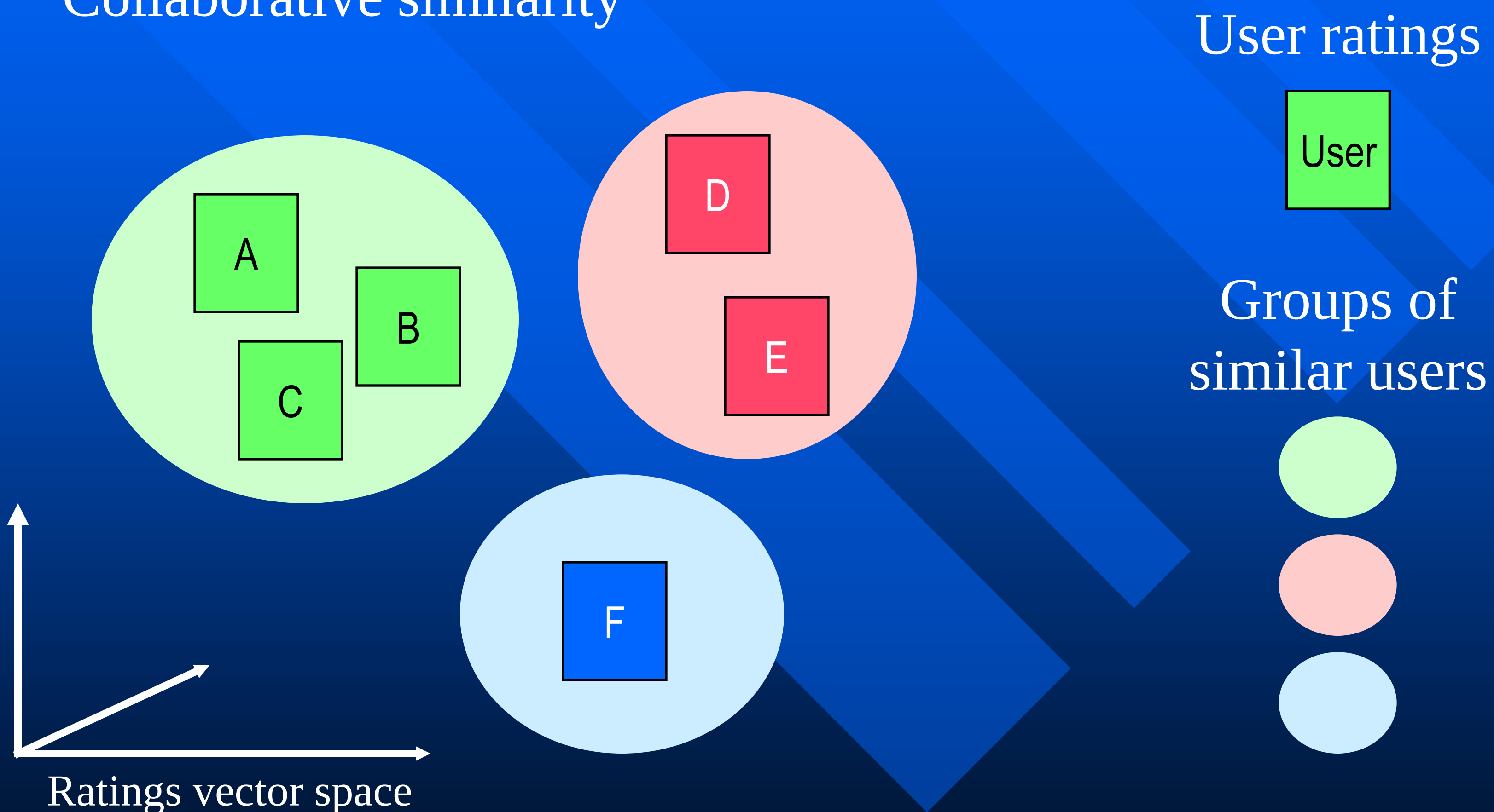
  Binary class profile representation
  - ‘Interesting’ and ‘not interesting’ examples
  - Time-decay function favours recent examples
  - Machine learning classifies new information (e.g. TF-IDF)

  Collaborative similarity
  - Behaviour correlation finds similar users (e.g. Pearson r)
  - New information comes from similar users

Capturing knowledge of user preferences with recommender systems

- **Knowledge capture of user profiles**

Collaborative similarity



Capturing knowledge of user preferences with recommender systems

- **Knowledge capture of user profiles**

Binary class profile representation
- 'Interesting' and 'not interesting' examples
- Time-decay function favours recent examples
- Machine learning classifies new information (e.g. TF-IDF)

Collaborative similarity
- Behaviour correlation finds similar users (e.g. Pearson r)
- New information comes from similar users

Our approach - Multi-class profile
- Classes explicitly represent using domain ontology
- Domain knowledge can enhance profiling
- Examples of classes can be shared
- Accuracy decreases with number of classes

Capturing knowledge of user preferences with recommender systems

- **Knowledge capture of user profiles**

Multi-class profile representation

Topic A　　Topic B　　Topic C

User A

Interesting	Topic A,B
Not interesting	Topic C

User B

Interesting	Topic B,C
Not interesting	Topic A

Capturing knowledge of user preferences with recommender systems

- **Quickstep architecture and approach**

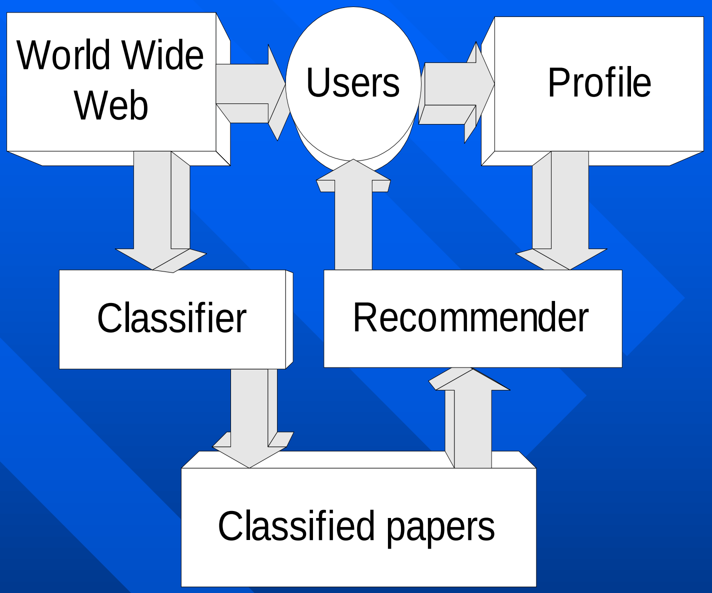
Research papers

TF vector representation

Classifier

k-nearest neighbour

Users can add examples



Capturing knowledge of user preferences with recommender systems

- **Quickstep architecture and approach**

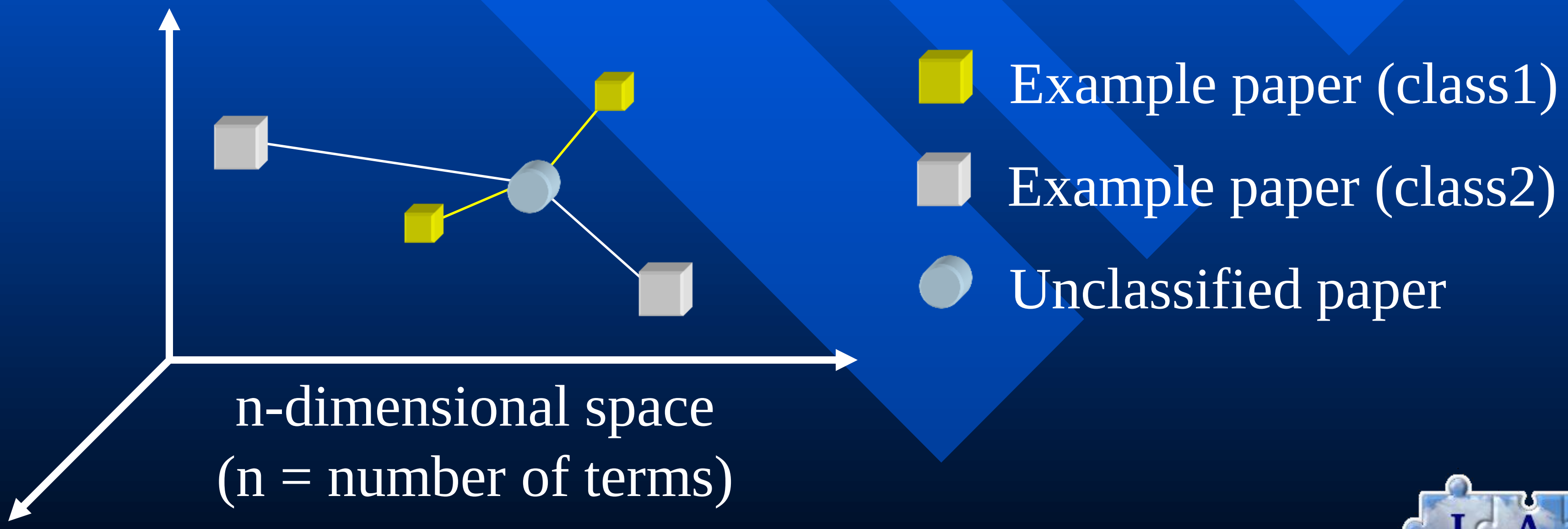
K-Nearest Neighbour - kNN

TF vector representation

Examples exist in an n dimensional space

New papers are added to this space

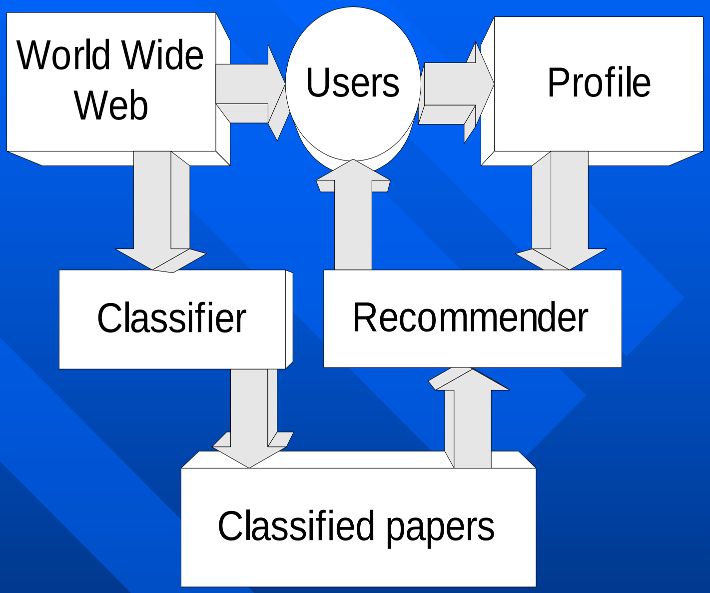
Classification is a function of its 'closeness' to examples



Capturing knowledge of user preferences with recommender systems

- **Quickstep architecture and approach**

Research papers
    TF vector representation
Classifier
    k-nearest neighbour
    Users can add examples
Classified paper database
    Grows as users browse
Profiler
    Feedback and browsed papers give time/interest profile
    Time decay function computes current interests



Capturing knowledge of user preferences with recommender systems

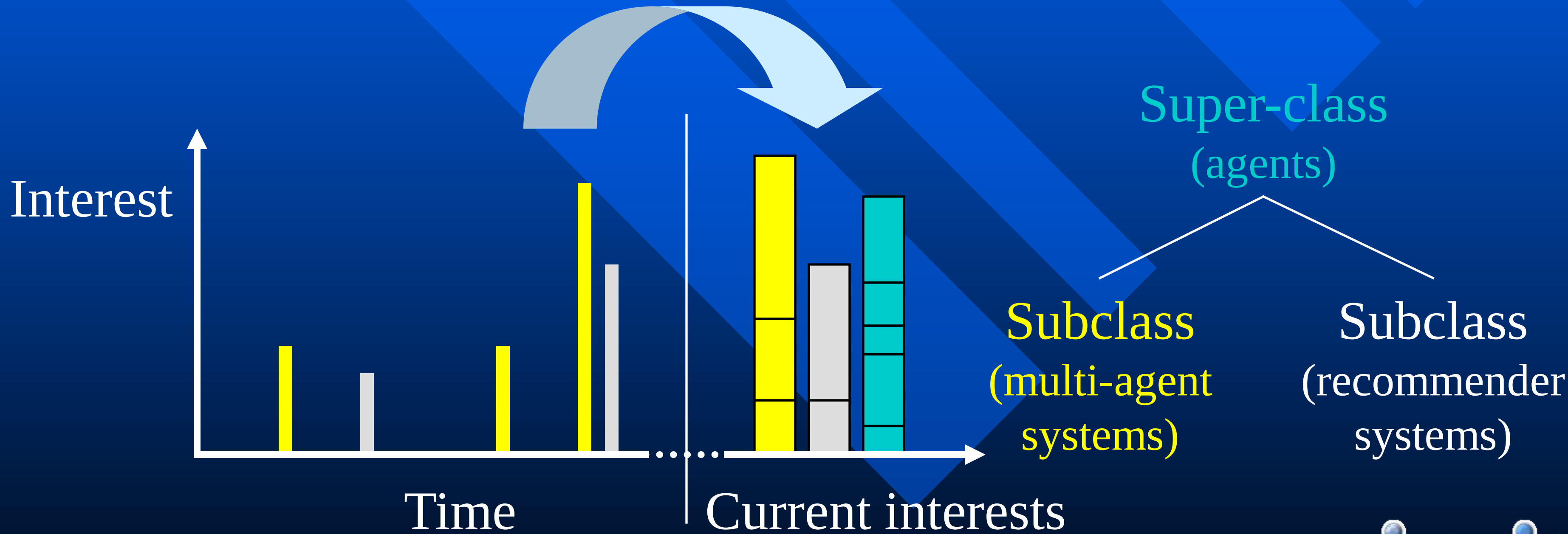
- **Quickstep architecture and approach**

Profiling

Time/Interest profile

Is-a hierarchy infers topic interest in super-classes

Time decay function biases towards recent interests



Capturing knowledge of user preferences with recommender systems

- **Quickstep architecture and approach**

Research papers

  TF vector representation

Classifier

  k-nearest neighbour

  Users can add examples

Classified paper database

  Grows as users browse

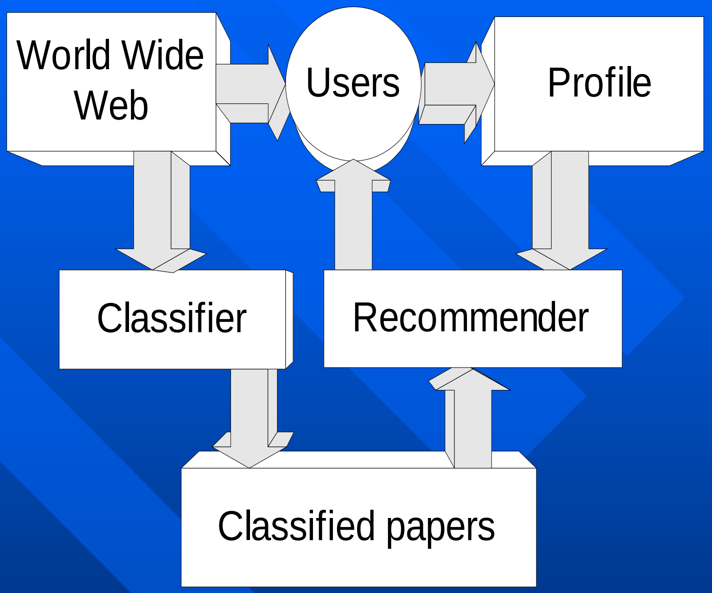
Profiler

  Feedback and browsed papers give time/interest profile

  Time decay function computes current interests

Recommender

  Recommends new papers on current topics of interest



Capturing knowledge of user preferences with recommender systems

- **Issues arising from our empirical evaluation**

  Experimental evaluation
  - Two trials, 24 and 14 users, 1.5 months each trial
  - Evaluate use of an is-a hierarchy and dynamic flat-list

  What advantages does an ontology bring to the system?
  - Adding super-classes ‘rounded’ out profiles
  - Ontology gave a consistent conceptual model to users
  - Ontology users had more interesting recommendations

  Does using domain knowledge compensate for the reduced accuracy of the multi-class classifier?
  - Classifier accuracy was lower than a typical binary classifier
  - When wrong, k-NN chose a topic in a related area
  - Recommendations best for reading around an area

Capturing knowledge of user preferences with recommender systems

- **Issues arising from our empirical evaluation**

Is the recommender system useful as a workplace tool?

- About 10% of recommendations led to good jumps
- Users felt system was moderately useful
- Topic classes were too broad for some users

How does Quickstep compare to other recommender systems?

- There is a lack of trials with real users
- There is no standard metric to measure ‘usefulness’
- Performance compared reasonably with other systems

Work published in the K-CAP2001 conference

- http://sern.ucalgary.ca/ksi/K-CAP/K-CAP2001/

Capturing knowledge of user preferences with recommender systems

- **Foxtrot architecture and approach**

Searchable database of papers

  Title, content, topic, quality and date search supported

  HTML support in addition to PS,PDF and zip,gz,Z

Ontology and training set

  96 classes, based on CORA paper database hierarchy

  5-10 example papers per class (714 training examples)

More collaborative recommendation

  Quality feedback used to rank recommendations

  Pearson r correlation to find similar users

Profile visualization

  Users can provide explicit feedback on their interest profile

Capturing knowledge of user preferences with recommender systems

- **Foxtrot empirical evaluation**

Experiment currently running

Run over this academic year

All 3rd and 4th year UG's, staff and PG's can use Foxtrot

70+ registered users

15,000+ research papers

Two groups, random subject selection

One group can provide explicit profile feedback

One group cannot (just relevance feedback)

Sign up!

Just email me with your username and I will register you

sem99r@ecs.soton.ac.uk

Capturing knowledge of user preferences with recommender systems

- **Future work**

Short paper for WWW conference with Harith

Looking at synergies between Quickstep and COP

Could result in a full paper

Foxtrot experiment

Full results in July, written up in a journal article

Will also appear in my Thesis

Profile algorithm analysis on log data

Run profile algorithms on 1 year’s worth of URL logs

Log data could become an IAM resource