

Recommender Systems

Introduction

Radek Pelánek

2014

Today

- motivation
- main notions
- course organization
- project discussion

Motivation

- information overload
 - many choices available
 - “the paradox of choice”
- recommender system
 - provide aid
 - set of items + user “context” \Rightarrow selection of items (predicted to be “good” for the user)

(definition?)

Motivation

- 1 What recommender systems do you know?
- 2 What recommender systems would you like to have?

Examples of Applications

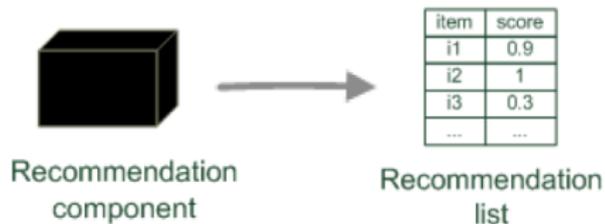
- movies
- music
- books
- software (apps)
- products in general
- research articles
- people (dating)
- services (restaurants, accommodation, ...)
- jokes

Value of Recommendations

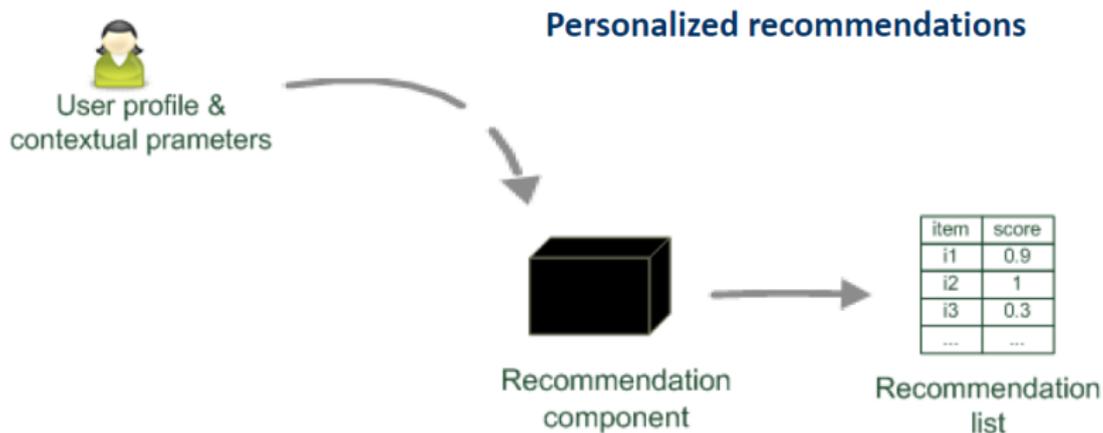
- Netflix: 2/3 of the movies watched
- Amazon: 35% sales
- Google news: recommendations \Rightarrow 38% more clickthrough

Types of Recommender Systems

Recommender systems reduce information overload by estimating relevance



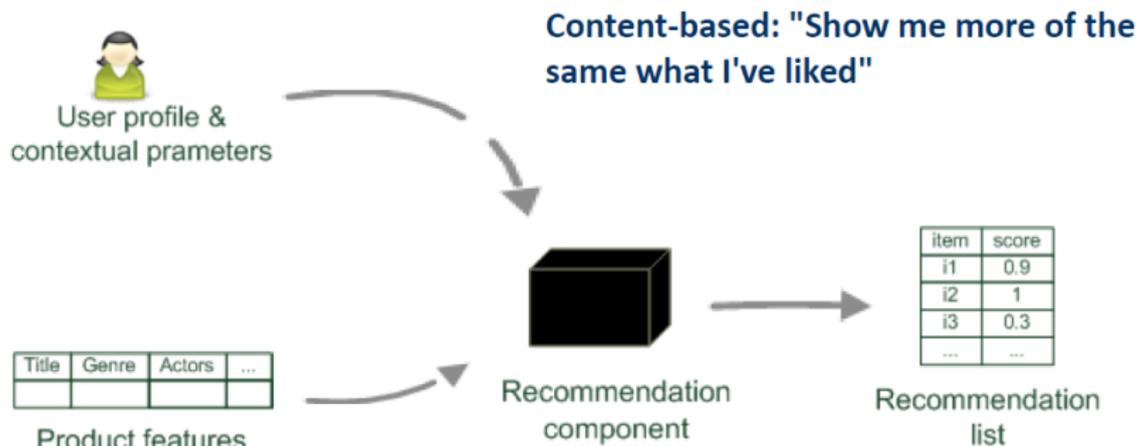
Types of Recommender Systems



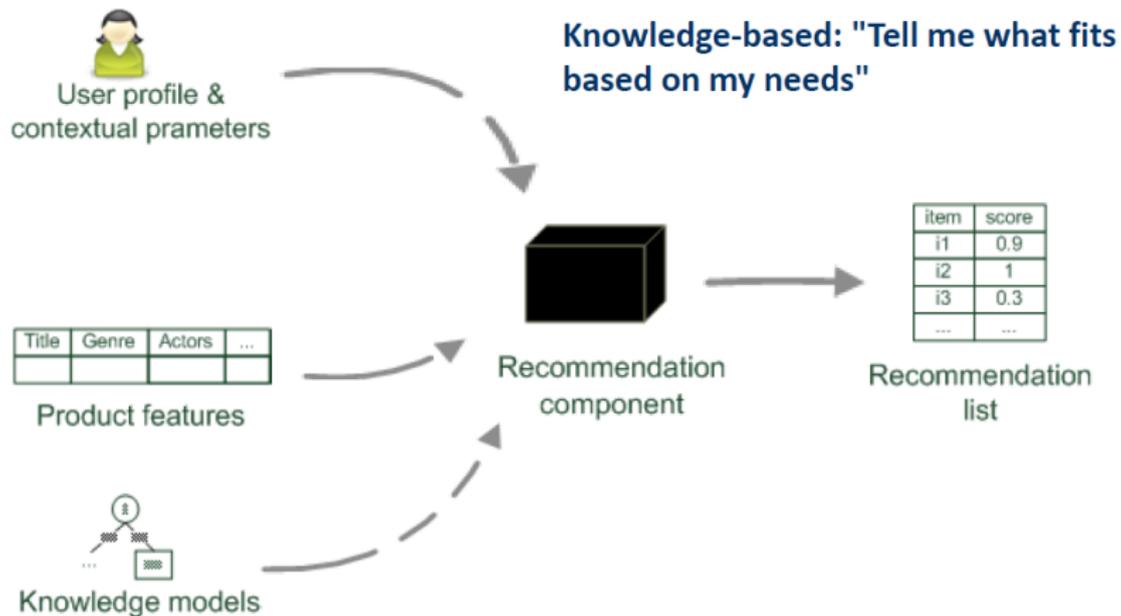
Types of Recommender Systems



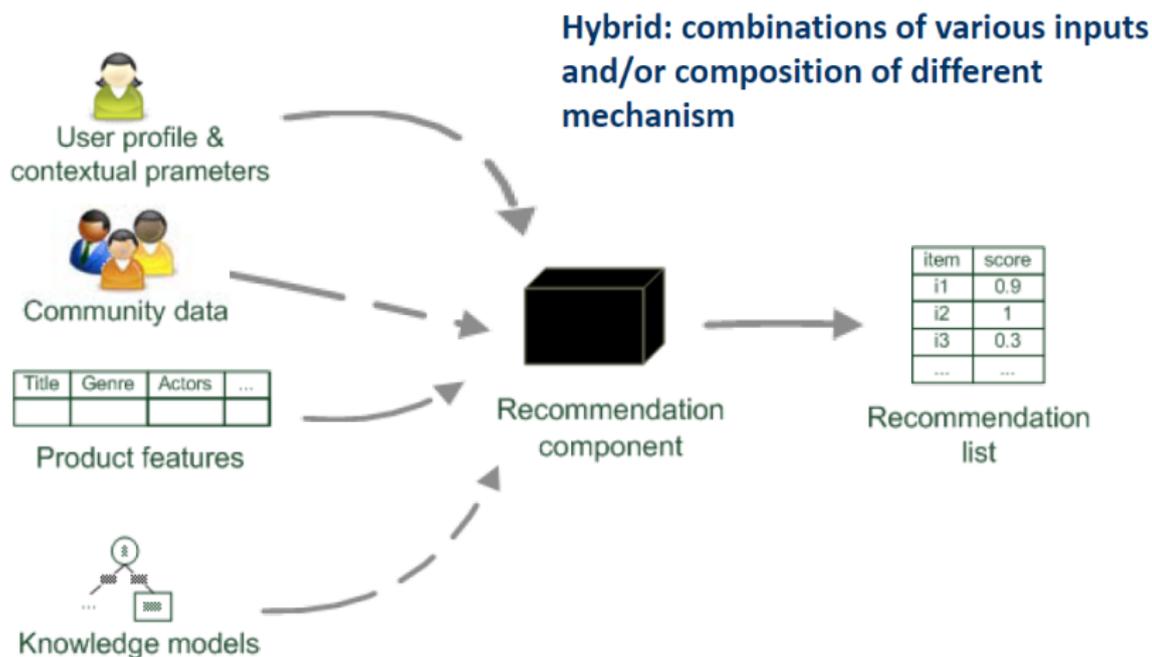
Types of Recommender Systems



Types of Recommender Systems



Types of Recommender Systems



Types of Recommender Systems

- non-personalized
- demographic
- collaborative filtering
- content based
- knowledge-based
- hybrid

Recommender System Functions

Provider's point of view:

- sell more items
- sell more diverse items (long tail)
- increase user satisfaction, fidelity
- better understand what users want

Long Tail



source: Wikipedia

Recommender System Functions

User's point of view:

- looking for something:
 - find some good items
 - find all good items (closer to IR)
 - recommend a sequence, a bundle
- just browsing
- side-effects (collaborative filtering systems):
 - express self
 - help others
 - influence others

RecSys and Information Retrieval

Information retrieval is the activity of obtaining information resources relevant to an information need from a collection of information resources. (Wikipedia)

- RecSys and IR closely connected (many similar or analogical techniques)
- different goals:
 - IR – “I know what I’m looking for”
 - RecSys – “I’m not sure what I’m looking for”

Serendipity

- unsought finding
- unexpected, but useful result
- do not recommend items the user already knows or would find anyway, try something more interesting
- example – books:
 - I like books by Remarque, Potok, Skácel
 - recommending another book by Remarque not very useful
 - recommending Munro = serendipity

A Brief History

- 1990s' – first systems (e.g., GroupLens), basic algorithms
- 1995-2000 – rapid commercialization, challenges of scale
- 2000-2005 – research explosion, mainstream applications
- 2006 – Netflix prize
- 2007 – the first Recommender Systems conference
- now – very active research, many applications

Netflix Prize

- Netflix – video rental company
- contest: 10% improvement of the quality of recommendations
- collaborative filtering
- prize: 1 million dollars
- data: user ID, movie ID, time, rating

Collaborative Filtering

- one of the most often and successfully used techniques
- widely applicable, does not need any domain knowledge
- interesting analogies, metaphors, questions
 - ants, social insect: communication via pheromone
 - recommender systems: people \sim ants, pheromone \sim ratings (clicks)
 - between human intelligence and (good old-fashioned) artificial intelligence

Ratings

- explicit
 - Likert scale (5 stars), like/dislike
 - require additional effort from users
- implicit
 - buying an item, visiting a page, viewing a video
 - easier to collect, less precise
 - more “honest”

RecSys and Educational Domain

- learning materials – direct application
- problems, exercises:
 - users \sim students
 - items \sim problems
 - ratings \sim performance (correctness of answers, problem solving times)

Our Projects at FI

educational systems:

- tutor.fi.muni.cz
- slepemapy.cz

Course Organization

(preliminary)

- ~ 6 weeks
 - lectures: main notions of the field
 - discussions: relations of notions to your projects
- ~ 6 weeks
 - work on projects
 - consultations
- final 2 weeks
 - presentation of projects

Focus of This Course

- practical experience
- collaborative filtering
- educational applications

more focus on consultations / discussions than on lectures
(good lectures available online)

Prerequisites

- programming
- math (basic linear algebra, statistics)
- (basics of machine learning – not strictly necessary)

(depends also on the choice of project)

Materials, Sources

- Introduction to Recommender Systems book
 - <http://www.recommenderbook.net/>
 - slides freely available – more details than in course slides
- Recommender Systems Handbook
- Video lectures: Coursera, Machine learning summer school

(links at course web page)

Projects

2 main options:

- “application”: development of a simple recommender systems
recommended for AP, INS, SSME students
- “research”: development and experimental evaluation of algorithms used by recommender systems
recommended for TEI, UMI students

“Application”: System Development

build a simple recommender system, 1-4 students

- “short text” recommendations: jokes, quotes, poetry, baby names, recipes, ...
- “local” recommendations (Brno): restaurants, cultural events, places, ...
- educational recommendations: courses (MU, MOOC), foreign language vocabulary, learning materials, ...
- product recommendation (specialized for a particular domain): board games, books for children, ...

requirements: simple web portal implementation (PHP or Python / MySQL / JavaScript)

“Research” : Models, Evaluation

individual project

- develop a model for predicting user ratings / student performance
- evaluate the model, visualize results
- provided: specifics datasets (movies, slepemapy.cz data), guidelines, baseline model implementations (in Python)

requirements: data analysis (Python recommended),
implementation of machine learning techniques

Alternatives Possible

- creating good Wikipedia articles in Czech
- creating educational video (tutorial) in Czech
- giving a presentation about some specific topic
- own proposal

(high demands, only in case of strong interest, not as an “escape” from programming)

Colloquium – Requirements

- interesting project
- active participation during semester or solid knowledge of covered topics (discussion at the end of the semester)