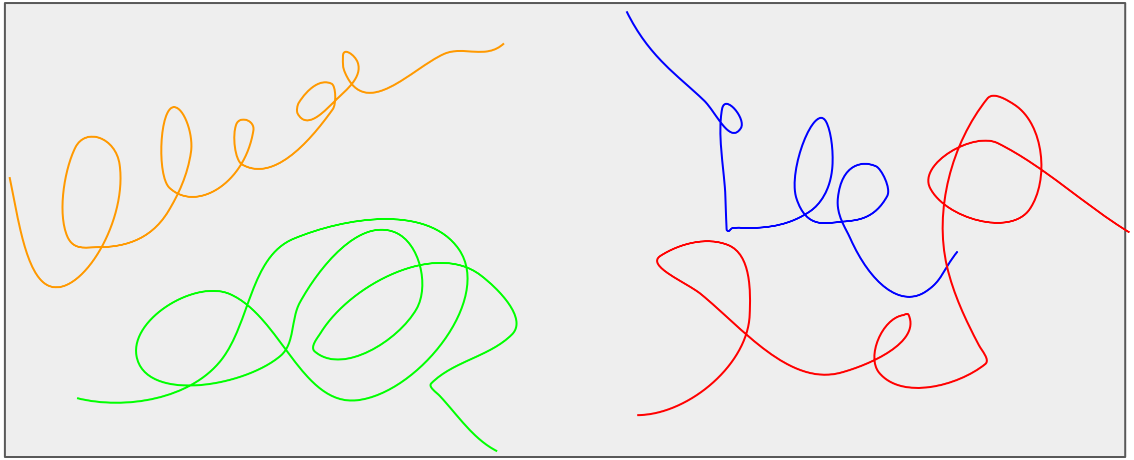
A Statistical Modeling Framework For Understanding Recommenders

David Yang
xdyang70@gmail.com

Challenge!

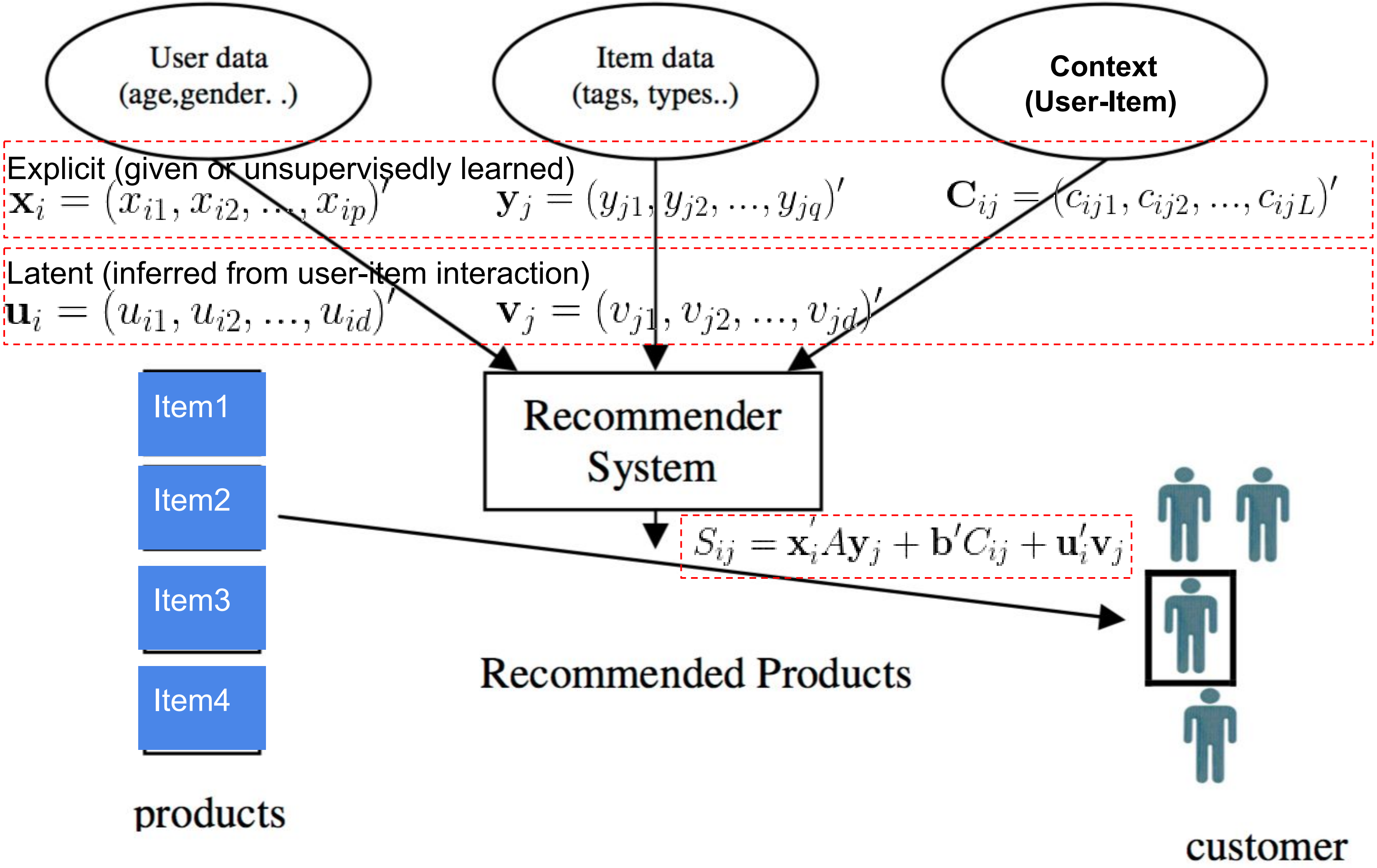
Recommender Options: Popularity, BPR-FM, NMF, etc.
(Today's Focus)

Metrics: Accuracy, Novelty,...



Goal: CTR, $Rev, etc

Use Cases (News, Magazine,...: email, web, mobile, …)


User data
(age,gender. .)
Item data
(tags, types..)
Context
(User-Item)
Explicit (given or unsupervisedly learned)
$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})'$
$\mathbf{y}_j = (y_{j1}, y_{j2}, \ldots, y_{jq})'$
$\mathbf{C}_{ij} = (c_{ij1}, c_{ij2}, \ldots, c_{ijL})'$
Latent (inferred from user-item interaction)
$\mathbf{u}_i = (u_{i1}, u_{i2}, \ldots, u_{id})'$
$\mathbf{v}_j = (v_{j1}, v_{j2}, \ldots, v_{jd})'$
Recommender System
$S_{ij} = \mathbf{x}_i' A \mathbf{y}_j + \mathbf{b}' C_{ij} + \mathbf{u}_i' \mathbf{v}_j$
Item1
Item2
Item3
Item4
products
Recommended Products
customer

Recommender via Statistical Models

Using Rating Data with Gaussian Distribution for Illustration

Likelihood:

$$R_{ij} \sim Normal(S_{ij}, \sigma^2)$$
$$S_{ij} = \mathbf{x}_i^{'} A \mathbf{y}_j + \mathbf{b}' C_{ij} + \mathbf{u}_i' \mathbf{v}_j$$

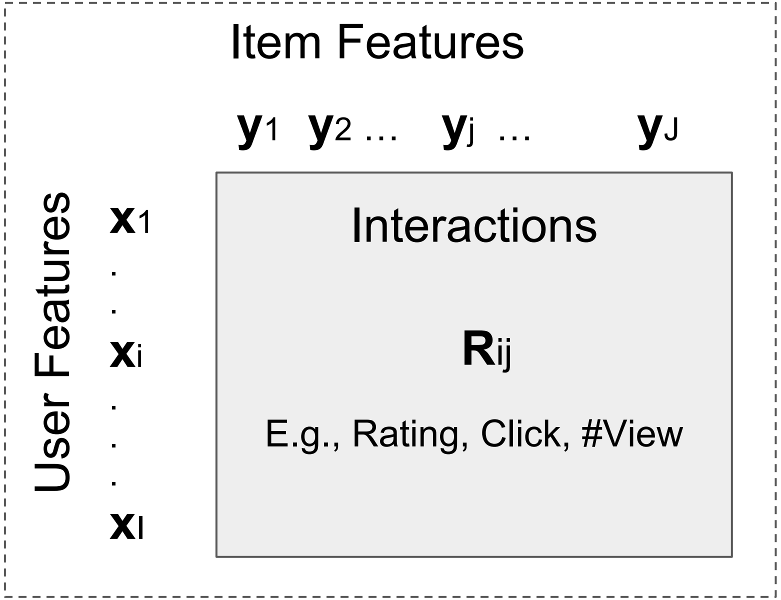
Prior Distribution:

$$Pr(\Theta) = Pr(\{\mathbf{A}, \mathbf{b}, \mathbf{U}, \mathbf{V}, \sigma^2\})$$

A Framework for Recommenders

Recommendations jointly made based on (Hybrid Solution)

- What we know about the items (Item-based)
- What we know about the user (User-based)
- How users interact with items (Collaborative)



Features

- User Attributes $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})'$
- Item Attributes $\mathbf{y}_j = (y_{j1}, y_{j2}, \ldots, y_{jq})'$

Interactions (Explicit: e.g., rating, Implicit:e.g., click):

- $\mathbf{R}_{ij} = (R_{ij1}, R_{ij2}, \ldots, R_{ijK})'$
  [default K=1]

Contexts of interactions (e.g., time, position):

- $\mathbf{C}_{ij} = (c_{ij1}, c_{ij2}, \ldots, c_{ijL})'$

Scores - Modeling Result:

- $S_{ij} = f(\mathbf{x}_i, \mathbf{y}_j, \mathbf{c}_{ij}, \mathbf{\Theta})$

Modeling:

- Minimize $\sum_{i,j} Dist(R_{ij}, S_{ij}) + \lambda r(\mathbf{\Theta})$

Matrix-Vector Format (using Gaussian for illustration)

$$y_{11}, \ldots, y_{1q} \qquad \ldots \qquad y_{J1}, \ldots, y_{Jq}$$

$$\begin{array}{l|l} x_{11}, \ldots, x_{1p} & R_{11}, \; . \; . \; . \; , R_{1J} \\ x_{i1}, \ldots, x_{ip} & R_{i1}, \; . \; . \; . \; , R_{iJ} \\ x_{I1}, \ldots, x_{Ip} & R_{I1}, \; . \; . \; . \; , R_{IJ} \end{array}$$

$$R_{ij} = \overset{\text{Fixed Effects}}{\boxed{\mathbf{x}_i \mathbf{A} \mathbf{y}_j + \mathbf{b}\mathbf{c}_{ij}}} + \overset{\text{Random Effects}}{\boxed{\mathbf{u}_i' \mathbf{v}_j + \epsilon_{ij}}}$$

$$\mathbf{u}_i \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_{d \times d})$$

$$\mathbf{v}_j \sim N(\mathbf{0}, \sigma_v^2 \mathbf{I}_{d \times d})$$

$$\epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$$

Alternative Formulation

Predictors (I.V.)	DV
$x_{11}, \ldots, x_{1p}; y_{11}, \ldots, y_{1q}; x_{11}y_{11}, \ldots, x_{1p}y_{1q}; c_{111}, \ldots, c_{11l}; \ldots$	R_{11}
....	.
$x_{i1}, \ldots, x_{ip}; y_{j1}, \ldots, y_{jq}; x_{i1}y_{j1}, \ldots, x_{ip}y_{jq}; c_{ij1}, \ldots, c_{ijl}; \ldots$	R_{ij}
....	.
$x_{I1}, \ldots, x_{Ip}; y_{J1}, \ldots, y_{Jq}; x_{I1}y_{J1}, \ldots, x_{Ip}y_{Jq}; c_{IJ1}, \ldots, c_{IJl}; \ldots$	R_{IJ}

$$R_{ij} = \underbrace{\beta_1 x_{11} + \ldots + \beta_m c_{ijL}}_{\text{Fixed Effects}} + \underbrace{\mathbf{u}_i'\mathbf{v}_j + \epsilon_{ij}}_{\text{Random Effects}}$$

I.V. = Independent Variable; DV = Dependent Variable

Item/User Feature Vectors

Example of Item Features	**Yj**
Category: Business	0.0
Category: Entertainment	0.4
…	…
Category: Science	0.1
Words: best	0.0
Words: worst	0.2
…	…
Words: Surprise	0.3
Doc2Vec	
Vec_1	0.9
…	...
Vec_100	-0.5
Other: Length	100
…	…
Other: Aging	20

Common Features

Example of User Features	**Xi**
Interest: Business	1.0
Interest: Entertainment	1.0
…	…
Interest: Science	0.0
Words: best	0.3
Words: worst	0.1
…	…
Words: Surprise	0.1
Profile2Vec	
Vec_1	0.7
…	...
Vec_100	-0.3
Other: #views	80
…	…
Other: Demographics	30

Approaches Creating Item Feature Vector: **Y**j

- Categorization & Description
  Industry Coding; Man-made Tags; Sources; Location; Image/Audio Features

- Bag-of-Words:
  TF, TF-IDF; Phrase & Entities; Synonym Expansion; Dimension Reduction (Feature Selection with L1/L2 Regularization; Singular Value Decomposition; Random Projection)

- Topic Modeling
  Latent Dirichlet Allocation (LDA); Hierarchical Dirichlet Process (LDA); lda2Vec

- Semantics Modeling:
  Word2Vec; lda2Vec

Approaches Creating User Feature Vector: $\mathbf{X}_i$

- Declared Profile
  Demographics; Declared/implied Interest (e.g., Netflix and StitchFix)

- Past Interaction with Content
  $\mathbf{X}_i = F(\{ \mathbf{Y}_j:\ j \in \overline{J}_i \})$ where $\overline{J}_i$ contains all items interacted by user i. F can be simple or weighted averaging (give higher weights to items recently interacted).

- Other User-related information
  Current Location; Usage-based features (web visits; devices used to access web); Search History; Item Set (similar to Bag of Words).

User/Item-based Methods

Unsupervised

- User-Item Similarity
  $S(\mathbf{X}_i, \mathbf{Y}_j)$: i = 1,...,I; j = 1,...,J.

- Item-Item Similarities
  $S(\mathbf{Y}_{j1}, \mathbf{Y}_{j2})$: j1, j2 = 1,…,J.

- User-User Similarity
  $S(\mathbf{X}_{i1}, \mathbf{X}_{i2})$: i1, i2 = 1,...,I.

Similarity:

- Cosine, Pearson/Spearman Correlation Coefficient, Okapi BM25, Jaccard.
- Weighted: $S(X_i, Y_j) = X'_i\mathbf{W}Y_j$
  Problem: how to determine W?

Supervised

$$S_{ij} = S(\mathbf{x}_i, \mathbf{y}_j) = \mathbf{x}_i^{'} A \mathbf{y}_j + \mathbf{b}' C_{ij}$$

- Binary Ratings (logistic model)

$$R_{ij} \sim Bernoulli((1 + exp\{-S_{ij}\})^{-1})$$

- Numerical Ratings (Gaussian Model)

$$R_{ij} \sim Normal(S_{ij}, \sigma^2)$$

- Ordinal Ratings (Cumulative Logit)

$$R_{ij} \sim Multinomial(\pi_{ij1}, ..., \pi_{ijK})$$

- Pairwise Preference Scores

$$R_{ijk} \sim Bernoulli((1 + exp\{-(S_{ij} - S_{ik}))\})^{-1}))$$

- Regularized M.L.E.

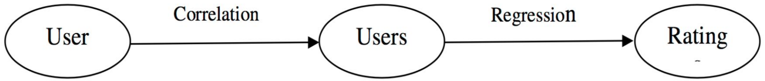
$$\underset{A,\Theta}{arg\ max}(logPr(R|A, \Theta) - \lambda r(A))$$

Collaborative Filtering

User-User Similarity

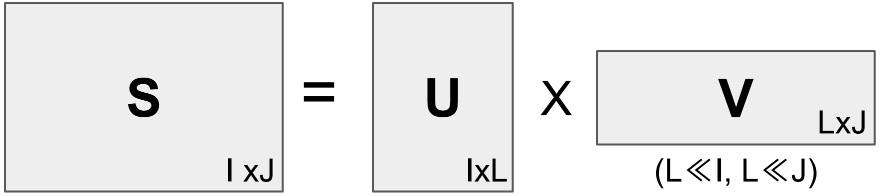
$$S_{ij} = \bar{R}_{i.} + \frac{\sum_{l \in I_j(i)} w(i,l)(R_{lj} - \bar{R}_{l.})}{\sum_{l \in I_j(i)} |w(i,l)|}$$

- Similarity Function (e.g., Pearson Correlation)
- Neighborhood Selection
- Weighting



Item-Item Similarity

Matrix Factorization: $S_{ij} = \mathbf{u}_i^{'}\mathbf{v}_j$



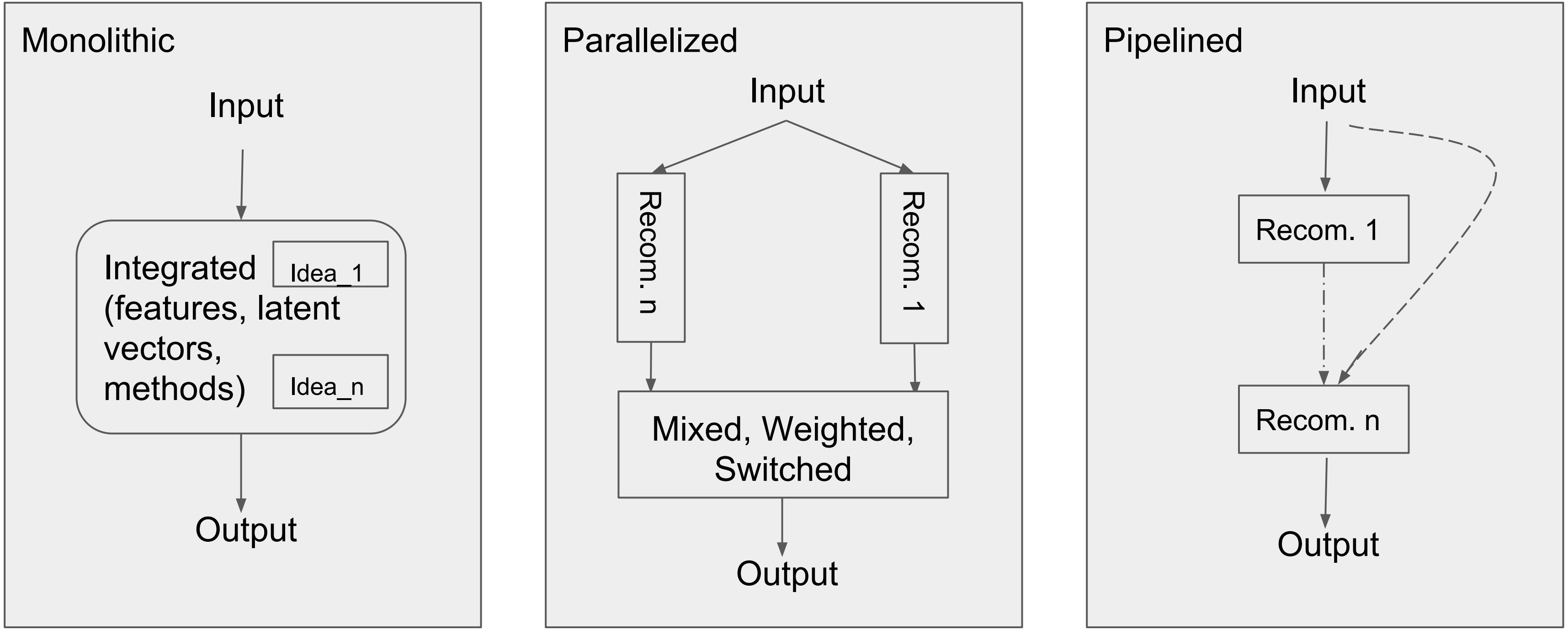
E.g., for numerical ratings

$$\underset{\mathbf{u}_i, \mathbf{v}_j}{arg\ min} \sum_{i,j} (R_{ij} - S_{ij})^2$$

Optimization Methods

- Alternating Least Square
- Stochastic Gradient Descent

Putting Them Together - Hybridization (Ensemble)



More on Hybridizations

Weighted	The ratings of several recommendation techniques are combined together to produce a single recommendation
Switching	The system switches between recommendation techniques depending on the current situation
Mixed	Recommendations from several different recommenders are presented at the same time
Feature Combination	Features from different recommendation data sources are thrown together into a single recommendation algorithm
Cascade	One recommender refines the recommendations given by another
Feature Augmentation	Output from one technique is used as an input feature to another
Metal-Level	The model learned by one recommender is used as input to another

Integrative Approaches

- Treating collaborative filtering as features: e.g., Fast Online Bilinear Factor Model (FOBFM)

Offline

$$S_{ijt} = \mathbf{c}'_{ijt}b + \mathbf{x}'_{it}\mathbf{A}\mathbf{y}_j + \mathbf{x}'_{it}\mathbf{v}_{jt}$$

$$\mathbf{v}_{jt} = \mathbf{B}\delta_j \qquad \delta_j \sim N(\mathbf{0}, \delta^2\mathbf{I})$$

\+

Online

$$S_{ijt} = \mathbf{c}'_{ijt}b + \mathbf{x}'_{it}\mathbf{A}\mathbf{y}_j + \mathbf{x}'_{it}\mathbf{B}\delta_{jt}$$

δ_{jt} leant online

- Using feature-based similarity in similarity-based collaborative filtering

- Collaborative filtering augmented by artificial feature-based ratings: e.g., Regression-Based Latent Factor Model (RLFM)

Likelihood

$$R_{ij} \sim N(S_{ij}, \sigma^2)$$

(Or $R_{ij} \sim Bern((1 + exp\{-S_{ij}\})^{-1})$)

$$S_{ij} = \mathbf{x}'_i\mathbf{A}\mathbf{y}_j + \alpha_i + \beta_j + \mathbf{u}'_i\mathbf{v}_j$$

Priors

$$\alpha_i \sim N(g(\mathbf{x}_i), \sigma_\alpha^2) \quad \mathbf{u}_i \sim N(G(\mathbf{x}_i), \sigma_u^2\mathbf{I})$$

$$\beta_j \sim N(h(\mathbf{y}_j), \sigma_\beta^2) \quad \mathbf{v}_j \sim N(H(\mathbf{y}_j), \sigma_v^2\mathbf{I})$$

Other Methods and Considerations

- Factorization through Latent Dirichlet Allocation

$$S_{ij} = \alpha_i + \beta_j + \mathbf{u}_i'\bar{\mathbf{z}}_j \qquad \bar{z}_j = \sum_{n=1}^{W_j} \frac{\mathbf{z}_{jn}}{W_j}$$

- Context-Dependent Recommendation

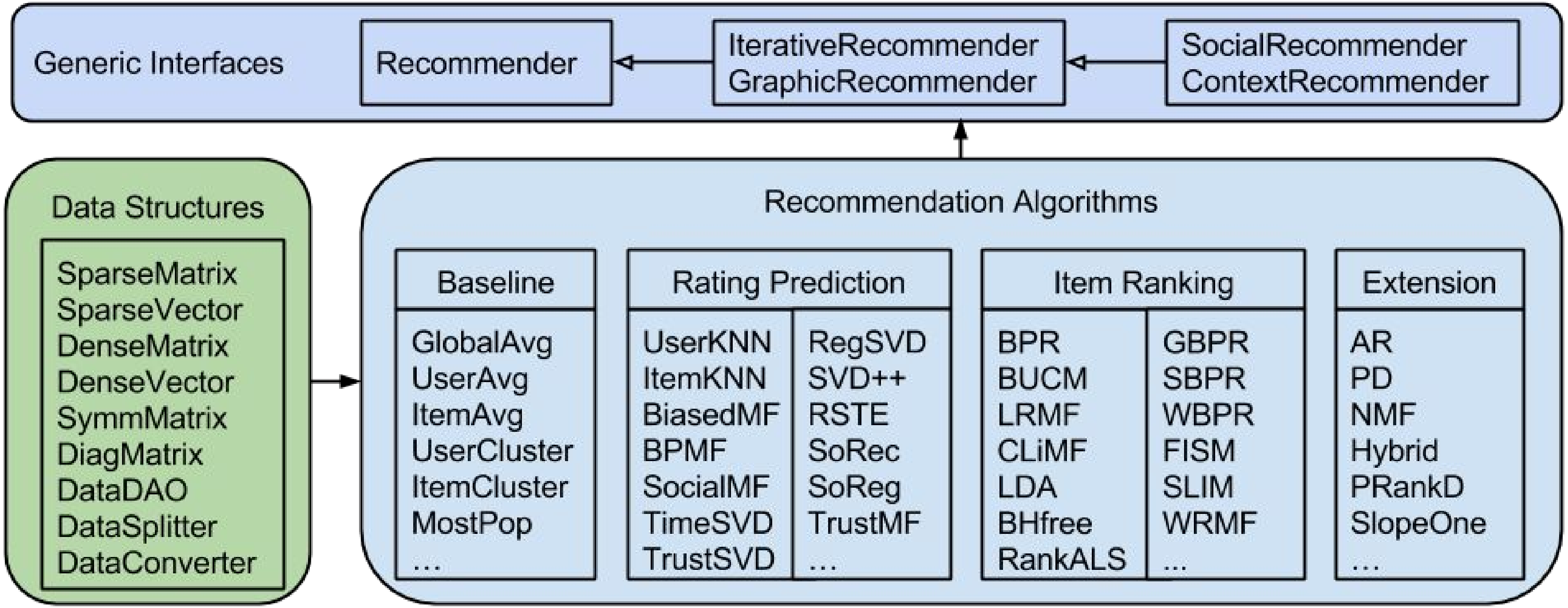
$$\begin{aligned} S_{ijk} = &\, b(\mathbf{x}_{ijk}) + \alpha_i + \beta_j + \gamma_k \\ &+ < \mathbf{u}_i, \mathbf{v}_j > + < \mathbf{u}_i, \mathbf{w}_k > + < \mathbf{v}_j, \mathbf{w}_k > \\ &+ < \mathbf{u}_i, \mathbf{v}_j, \mathbf{w}_k > \end{aligned}$$

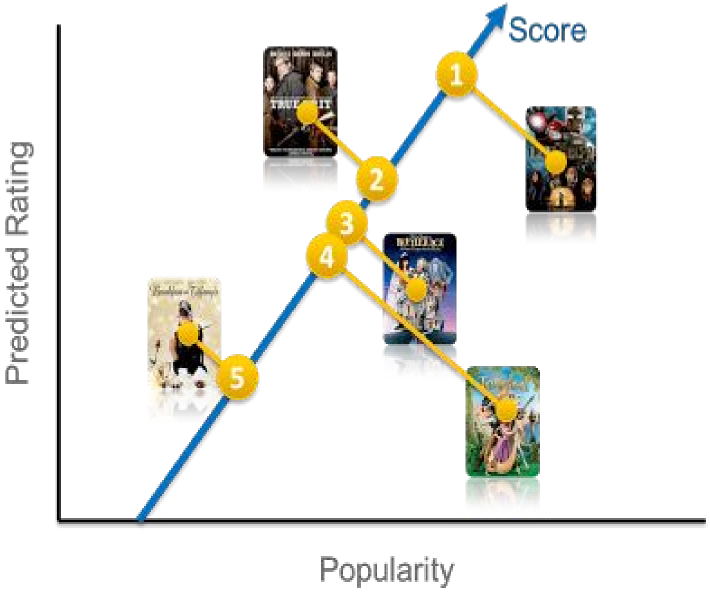
- Multi-objective Optimization
  - Segment approach
  - Personalized approach
  - Approximation approach
- Knowledge-based Recommender
  - Constraint-based
  - Case-based (similarity between requirement and items)

Discussion

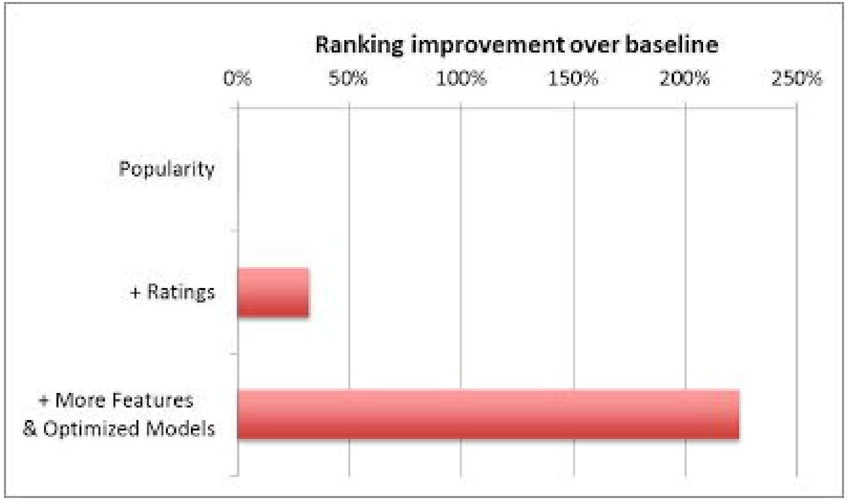
- Our Implemented Methods (pros & cons)
  - Random
  - Jaccard Similarity
  - Nonnegative Matrix Factorization
  - Popularity-based
  - BPR-FM
  - LDA
  - Filtering (constraint)
  - TBD
- Success Criteria
  - Relevancy (accuracy and coverage)
  - Novelty (diversity)
  - Serendipity (unexpectedness and Usefulness)
  - Privacy and Trust Management

Recommender Algorithms in LibRec





$f_{rank}(u,v) = w_1\, p(v) + w_2\, r(u,v) + b$, where u=user, v=video item, p=popularity and r=predicted rating



A new feature does not show value because the model cannot learn it? Or, a more powerful model is not useful simply because you don't have the feature space that exploits its benefits?