Manipulating the Capacity of Recommendation Models in Recall-Coverage Optimization

Ing. Tomáš ŘEHOŘEK

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Motivation

- Recommender Systems = actively studied topic in the last decades
  - Large number of novel algorithms continuously published
- Confusion in:
  - Evaluating and comparing the models
    - Netflix Prize = RMSE
    - Recall, Precision, F-Measure
  - Tasks solved by the models
  - Long-tail vs. popular items recommendation
- Missing systematic framework for long tail recommendations
  - Novelty and Serendipity of the models
    - Evaluation
    - Control

State of the Art

- Before 2006: GroupLens era
  - Bardul Sarwar: ItemKnn, Associaton Rules, SVD
  - Optimizing MAE, NMAE
- 2006-2014: Netflix Prize, Matrix Factorization era
  - $1M Prize, 40K teams from 186 countries
  - Optimizing RMSE
  - Robert Bell, Yehuda Koren – Top-N recommendation
  - Harald Steck [100,101] – MNAR, popularity-stratified recall
- 2015-2019: Deep learning approaches
  - Collaborative Deep Learning – optimizing Recall@N, mAP
  - AutoRec – RMSE
  - Wide & Deep Recommendation – optimizing AUC

Contributions of This Thesis

- Capacity manipulating hyperparameters for different learning algorithms
  - Showing shared nature of most recommendation models
  - Generalizing large number of published algorithms under a common framework
- Popularity-biasing coefficient β as universal capacity hyperparameter
  - Thanks to putting models under common score(u,i) framework
  - Generalizing approach already proposed in [101] to full scale of models
- Recall-Coverage Optimization (RCO)
  - Searching for Pareto-Optimal states while manipulating the capacity
  - Using **Recall@N coupled with Catalog Coverage** to optimize models

Formal Framework for Hyperparametrized Learning

Learning Algorithm

$$\mathcal{A}\colon \mathcal{P}_{\mathcal{A}} \times 2^{\mathcal{X}\times\mathcal{Y}} \to \mathcal{Y}^{\mathcal{X}}$$

Recommendation Model

Hyperparametrization

Training Dataset

Rating Prediction

$$\mathcal{A}^{\mathrm{RP}}\colon \mathcal{P}_{\mathcal{A}^{\mathrm{RP}}} \times (\mathbb{R}\cup\{?\})^{U\times I} \to \mathbb{R}^{(U\times I)}$$

Binary Classification

$$\mathcal{A}^{\mathrm{BC}}\colon \mathcal{P}_{\mathcal{A}^{\mathrm{BC}}} \times \{0,1,?\}^{U\times I} \to \{0,1\}^{(U\times I)}$$

Learning to Rank

$$\mathcal{A}^{\mathrm{Rank}}\colon \mathcal{P}_{\mathcal{A}^{\mathrm{Rank}}} \times (\mathbb{R}\cup\{?\})^{U\times I} \to \mathfrak{S}(I)^{U}$$

Top-N Recommendation

$$\mathcal{A}^{\mathrm{Top}\text{-}N}\colon \mathcal{P}_{\mathcal{A}^{\mathrm{Top}\text{-}N}} \times (\mathbb{R}\cup\{?\})^{U\times I} \to \{I'\subset I \mid |I'|=N\}^{U}$$

Solving Recommendation Tasks Using score(u,i)

$$score\colon U \times I \to \mathbb{R}$$

Rating Prediction

$$\text{model}(u,i) = \alpha \cdot score(u,i) - \theta$$

Binary Classification

$$\text{model}(u,i) = \begin{cases} 0 & \text{if } score(u,i) < \theta \\ 1 & \text{if } score(u,i) \geq \theta \end{cases}$$

Learning to Rank

$$\text{model}(u) = \left(i_1, \ldots, i_{|I|}\right)$$
$$\text{w.r.t. } \forall k \in \{1, \ldots, |I|-1\} : score(u, i_k) \geq score(u, i_{k+1})$$

Top-N Recommendation

$$\begin{aligned} &\text{model}(u) = \{i_1, \ldots i_N\} \\ \text{w.r.t.} \quad &\forall i_k \in \text{model}(u): \quad i_k \in I \wedge \\ &\qquad |\{j \in I \setminus \{i_k\} \mid score(u,j) > score(u,i_k)\}| < N \end{aligned}$$

Existing Recommendation Models as score(u,i)

Popularity (“Bestseller”) Model

$$score(u,i) = \sum_{\substack{v \in U \setminus \{u\} \\ r_{v,i} \neq ?}} r_{v,i}$$

Existing Recommendation Models as score(u,i)

UserKnn Model

$$\mathrm{NN}_k(u) = \{v_1, \ldots, v_k\}$$
$$\text{w.r.t.} \quad \forall v \in \mathrm{NN}_k(u)\colon \; v \in U \setminus \{u\} \wedge |w \in U \setminus \{u\} \mid sim(u,w) > sim(u,v)| < k$$

Unweighted

$$score(u,i) = \begin{cases} \dfrac{\sum_{\substack{v \in \mathrm{NN}_k(u) \\ r_{v,i} \neq ?}} r_{v,i}}{|\{v \in \mathrm{NN}_k(u) \mid r_{v,i} \neq ?\}|} & \text{if } \exists v \in \mathrm{NN}_k(u)\colon r_{v,i} \neq ? \\ 0 & \text{otherwise} \end{cases}$$

Weighted

$$score(u,i) = \begin{cases} \dfrac{\sum_{\substack{v \in \mathrm{NN}_k(u) \\ r_{v,i} \neq ?}} sim(u,v) \cdot r_{v,i}}{\sum_{\substack{v \in \mathrm{NN}_k(u) \\ r_{v,i} \neq ?}} sim(u,v)} & \text{if } \exists v \in \mathrm{NN}_k(u)\colon r_{v,i} \neq ? \\ 0 & \text{otherwise} \end{cases}$$

With Non-Normalized Neighborhood

$$score(u,i) = \begin{cases} \sum_{\substack{v \in \mathrm{NN}_k(u) \\ r_{v,i} \neq ?}} sim(u,v) \cdot r_{v,i} & \text{if } \exists v \in \mathrm{NN}_k(u)\colon r_{v,i} \neq ? \\ 0 & \text{otherwise} \end{cases}$$

Using Cosine Similarity

$$sim(u,v) = \frac{\mathbf{r}_{u,*} \cdot \mathbf{r}_{v,*}^T}{\|\mathbf{r}_{u,*}\| \cdot \|\mathbf{r}_{v,*}\|} = \frac{\sum_{\substack{i \in I \\ r_{u,i} \neq ? \\ r_{v,i} \neq ?}} r_{u,i} \cdot r_{v,i}}{\sqrt{\sum_{\substack{i \in I \\ r_{u,i} \neq ?}} r_{u,i}^2} \cdot \sqrt{\sum_{\substack{i \in I \\ r_{v,i} \neq ?}} r_{v,i}^2}}$$

Using Pearson Correlation Coefficient

$$sim(u,v) = \frac{\sum_{\substack{i \in I \\ r_{u,i} \neq ? \\ r_{v,i} \neq ?}} (r_{u,i} - \overline{\mathbf{r}_{u,*}}) \cdot (r_{v,i} - \overline{\mathbf{r}_{v,*}})}{\sqrt{\sum_{\substack{i \in I \\ r_{u,i} \neq ? \\ r_{v,i} \neq ?}} (r_{u,i} - \overline{\mathbf{r}_{u,*}})^2} \cdot \sqrt{\sum_{\substack{i \in I \\ r_{u,i} \neq ? \\ r_{v,i} \neq ?}} (r_{v,i} - \overline{\mathbf{r}_{v,*}})^2}}$$

Ochiai for Binary Matrices

$$sim(u,v) = \frac{|\hat{\mathbf{r}_u} \cap \hat{\mathbf{r}_v}|}{|\hat{\mathbf{r}_u}| \cdot |\hat{\mathbf{r}_v}|}$$

Existing Recommendation Models as score(u,i)

ItemKnn Model

$$\forall i \in \{i' \in I \mid r_{u,i'} \neq ?\}: \quad \mathrm{NN}_k(i) = \{j_1, \ldots, j_k\}$$
$$\text{w.r.t.} \quad \forall j \in \mathrm{NN}_k(i): \quad j \in I \setminus \{i\} \wedge$$
$$|\ell \in I \setminus \{i\} \mid sim(i,\ell) > sim(i,j)| < k$$

Weighted

$$score(u,i) = \begin{cases} \dfrac{\sum\limits_{\substack{j \in I \\ r_{u,j} \neq ? \\ i \in \mathrm{NN}_k(j)}} r_{u,j}}{|\{j \in I \mid r_{u,j} \neq ? \wedge i \in \mathrm{NN}_k(j)\}|} & \text{if } \exists j \in I\colon r_{u,j} \neq ? \wedge i \in \mathrm{NN}_k(j) \\ 0 & \text{otherwise} \end{cases}$$

Unweighted

$$score(u,i) = \begin{cases} \dfrac{\sum\limits_{\substack{j \in I \\ r_{u,j} \neq ? \\ i \in \mathrm{NN}_k(j)}} sim(i,j) \cdot r_{u,j}}{\sum\limits_{\substack{j \in I \\ r_{u,j} \neq ? \\ i \in \mathrm{NN}_k(j)}} sim(i,j)} & \text{if } \exists j \in I\colon r_{u,j} \neq ? \wedge i \in \mathrm{NN}_k(j) \\ 0 & \text{otherwise} \end{cases}$$

With Non-Normalized Neighborhood

$$score(u,i) = \begin{cases} \sum\limits_{\substack{j \in I \\ r_{u,j} \neq ? \\ i \in \mathrm{NN}_k(j)}} sim(i,j) \cdot r_{u,j} & \text{if } \exists j \in I\colon r_{u,j} \neq ? \wedge i \in \mathrm{NN}_k(j) \\ 0 & \text{otherwise} \end{cases}$$

With rating similarity functions as in UserKnn (transposed)

With attribute (content) similarity

- **tokenization**

$$attrsim(a_i, a_j) = \frac{|tokenization(a_i) \cap tokenization(a_j)|}{|tokenization(a_i) \cup tokenization(a_j)|}$$

- **embedding**

$$attrsim(a_i, a_j) = \frac{embedding(a_i)^T \cdot embedding(a_j)}{\|embedding(a_i)\| \cdot \|embedding(a_j)\|}$$

Existing Recommendation Models as score(u,i)

Association Rules (Survey and Novel Framework)

$$T(u) = \{i \in I \mid r_{u,i} \neq ? \wedge r_{u,i} \geq \theta\} \quad supp(A) = \frac{|\{u \in U \mid A \subseteq T(u)\}|}{|U|} \quad \mathcal{R} = \{X \Rightarrow Y \mid supp\,(X \cup Y) \geq s_{\min}\}$$

Multiple Rule-Quality Measures

- **confidence**

$$conf\,(X \Rightarrow Y) = \frac{supp\,(X \cup Y)}{supp(X)}$$

- **lift**

$$lift\,(X \Rightarrow Y) = \frac{conf\,(X \Rightarrow Y)}{supp(Y)}$$

- **conviction**

$$conv\,(X \Rightarrow Y) = \frac{1 - supp(Y)}{1 - conf\,(X \Rightarrow Y)}$$

Different Ways of Combining Rules Together for Recommendation

- **best-rule**

$$score(u,i) = \begin{cases} \max\limits_{\substack{(X \Rightarrow Y) \in \mathcal{R} \\ T(u) \subseteq X \\ i \in Y}} (q\,(X \Rightarrow Y)) & \text{if } \exists\,(X \Rightarrow Y) \in \mathcal{R}\colon X \subseteq T(u) \wedge i \in Y \\ 0 & \text{otherwise} \end{cases}$$

- **weighted voting**

$$score(u,i) = \begin{cases} \sum\limits_{\substack{(X \Rightarrow Y) \in \mathcal{R} \\ T(u) \subseteq X \\ i \in Y}} q\,(X \Rightarrow Y) & \text{if } \exists\,(X \Rightarrow Y) \in \mathcal{R}\colon X \subseteq T(u) \wedge i \in Y \\ 0 & \text{otherwise} \end{cases}$$

Association Rules

Existing Recommendation Models as score(u,i)

Matrix Factorization (Survey)

Simple $score(u,i) = \mathbf{q}_{*,i}^T \cdot \mathbf{p}_{*,u}$

$$\min_{\substack{\mathbf{P}\in\mathbb{R}^{f\times|U|}\\ \mathbf{Q}\in\mathbb{R}^{f\times|I|}}} \sum_{\substack{u\in U\\ i\in I\\ r_{u,i}\neq ?}} \left(r_{u,i} - \mathbf{q}_{*,i}^T \cdot \mathbf{p}_{*,u}\right)^2 + \lambda\left(\|\mathbf{q}_{*,i}\|^2 + \|\mathbf{p}_{*,u}\|^2\right)$$

With biases

$score(u,i) = \mu + b_u^U + b_i^I + \mathbf{q}_{*,i}^T \cdot \mathbf{p}_{*,u}$

$$\min_{\substack{\mathbf{P}\in\mathbb{R}^{f\times|U|}\\ \mathbf{Q}\in\mathbb{R}^{f\times|I|}\\ \mathbf{b}^U\in\mathbb{R}^{|U|}\\ \mathbf{b}^I\in\mathbb{R}^{|I|}\\ \mu\in\mathbb{R}}} \sum_{\substack{u\in U\\ i\in I\\ r_{u,i}\neq ?}} \left(r_{u,i} - \mu - b_i^I - b_u^U - \mathbf{q}_{*,i}^T \cdot \mathbf{p}_{*,u}\right)^2 + \lambda\left(\|\mathbf{q}_{*,i}\|^2 + \|\mathbf{p}_{*,u}\|^2 + {b_i^I}^2 + {b_u^U}^2\right)$$

With mixed implicit and explicit feedback data

$score(u,i) = b_{u,i} + \sum_{\substack{j\in I\\ r_{u,j}^R\neq ?}} \left(r_{u,j}^R - b_{u,j}\right)\cdot w_{i,j} + \sum_{\substack{j\in I\\ r_{u,j}^N\neq ?}} c_{i,j}$

$$\min_{\substack{\mathbf{B}\in\mathbb{R}^{|U|\times|I|}\\ \mathbf{W}\in\mathbb{R}^{|I|\times|I|}\\ \mathbf{C}\in\mathbb{R}^{|I|\times|i|}\\ \mathbf{b}^U\in\mathbb{R}^{|U|}\\ \mathbf{b}^I\in\mathbb{R}^{|I|}\\ \mu\in\mathbb{R}}} \sum_{\substack{u\in U\\ i\in I\\ r_{u,i}^R\neq ?}} \left(\left(r_{u,i}^R - \mu - b_u^U - b_i^I - \sum_{\substack{j\in I\\ r_{u,j}^R\neq ?}} \left(r_{u,j}^R - b_{u,j}\right)\cdot w_{i,j} - \sum_{\substack{j\in I\\ r_{u,j}^N\neq ?}} c_{i,j}\right)^2 + \lambda\left({b_u^U}^2 + {b_i^I}^2 + \sum_{\substack{j\in I\\ r_{u,j}^R\neq ?}} w_{i,j}^2 + \sum_{\substack{j\in I\\ r_{u,j}^N\neq ?}} c_{i,j}^2\right)\right)$$

With imputed unobserved ratings

- **Yifan Hu, Yehuda Koren** $$\min_{\substack{\mathbf{P}\in\mathbb{R}^{f\times U}\\ \mathbf{Q}\in\mathbb{R}^{f\times I}}} \sum_{\substack{u\in U\\ i\in I}} \left(c_{u,i}\cdot\left(y_{u,i} - \mathbf{q}_{*,i}^T \cdot \mathbf{p}_{*,u}\right)^2\right) + \lambda\left(\sum_{u\in U}\|\mathbf{p}_{*,u}\|^2 + \sum_{i\in I}\|\mathbf{q}_{*,i}\|^2\right)$$
- **Harald Steck** $$\min_{\substack{\mathbf{P}\in\mathbb{R}^{f\times|U|}\\ \mathbf{Q}\in\mathbb{R}^{f\times|I|}}} \sum_{\substack{u\in U\\ i\in I}} w_{u,i}\cdot\left(\left(\mathbf{r}_{u,i}^{\text{o\&i}} - r_m - \mathbf{q}_{*,i}^T \cdot \mathbf{p}_{*,u}\right)^2 + \lambda\cdot\sum_{\ell=0}^{f}\left(p_{\ell,u}^2 + q_{\ell,i}^2\right)\right)$$

...trained using different optimization algorithms (Stochastic Gradient Descent, Alternating Least Squares)

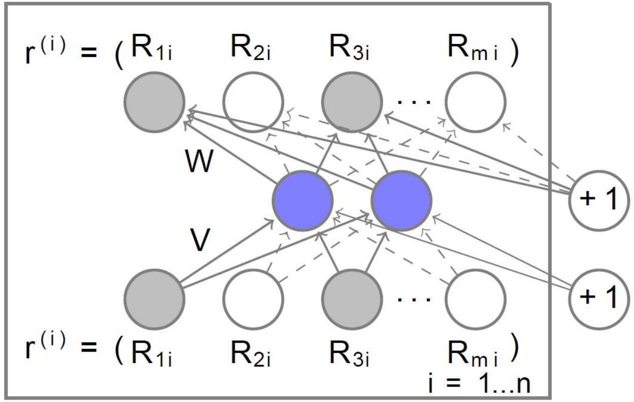
Existing Recommendation Models as score(u,i)

AutoRec

U-AutoRec

$$score(u,i) = \left(h\left(\mathbf{W} \cdot g\left(\mathbf{V} \cdot \hat{\mathbf{r}}_{u,*}^T + \boldsymbol{\mu}\right) + \mathbf{b}\right)\right)_i$$

$$\min_{\substack{\mathbf{V}\in\mathbb{R}^{f\times|I|}\\ \mathbf{W}\in\mathbb{R}^{|I|\times f}\\ \boldsymbol{\mu}\in\mathbb{R}^f\\ \mathbf{b}\in\mathbb{R}^{|I|}}} \sum_{\substack{u\in U\\ i\in I\\ r_{u,i}\neq ?}} \left(r_{u,i} - \left(h\left(\mathbf{W} \cdot g\left(\mathbf{V} \cdot \hat{\mathbf{r}}_{u,*}^T + \boldsymbol{\mu}\right) + \mathbf{b}\right)\right)_i\right)^2 + \frac{\lambda}{2} \cdot \left(\|\mathbf{V}\|^2 + \|\mathbf{W}\|^2\right)$$



I-AutoRec

$$score(u,i) = \left(h\left(\mathbf{W} \cdot g\left(\mathbf{V} \cdot \hat{\mathbf{r}}_{*,i} + \boldsymbol{\mu}\right) + \mathbf{b}\right)\right)_u$$

$$\min_{\substack{\mathbf{V}\in\mathbb{R}^{f\times|U|}\\ \mathbf{W}\in\mathbb{R}^{|U|\times f}\\ \boldsymbol{\mu}\in\mathbb{R}^f\\ \mathbf{b}\in\mathbb{R}^{|U|}}} \sum_{\substack{u\in U\\ i\in I\\ r_{u,i}\neq ?}} \left(r_{u,i} - \left(h\left(\mathbf{W} \cdot g\left(\mathbf{V} \cdot \hat{\mathbf{r}}_{*,i} + \boldsymbol{\mu}\right) + \mathbf{b}\right)\right)_u\right)^2 + \frac{\lambda}{2} \cdot \left(\|\mathbf{V}\|^2 + \|\mathbf{W}\|^2\right)$$

Generalized Validation Loss and Validation Reward

$$\mathcal{L}\colon \mathcal{Y}^{\mathcal{X}} \times 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}$$

$$P^*_{\mathcal{A}} = \arg\min_{P\in\mathcal{P}_{\mathcal{A}}} \mathcal{L}\left(\mathcal{A}\left(P, \mathcal{T}_{\text{train}}\right), \mathcal{T}_{\text{val}}\right)$$

$$\mathcal{F}\colon \mathcal{Y}^{\mathcal{X}} \times 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}$$

$$P^*_{\mathcal{A}} = \arg\max_{P\in\mathcal{P}_{\mathcal{A}}} \mathcal{F}\left(\mathcal{A}\left(P, \mathcal{T}_{\text{train}}\right), \mathcal{T}_{\text{val}}\right)$$

Rating Prediction $\mathbf{R} \in (\mathbb{R} \cup \{?\})^{U\times I}$

$$\mathcal{T} = \{((u,i), r_{u,i}) \mid u \in U \wedge i \in I \wedge r_{u,i} \neq ?\}$$

$$\text{MAE}\left(m_{\text{RP}}, \mathcal{T}_{\text{val}}\right) = \frac{1}{|\mathcal{T}_{\text{val}}|} \sum_{((u,i),r_{u,i})\in\mathcal{T}_{\text{val}}} |m_{\text{RP}}(u,i) - r_{u,i}|$$

$$\text{NMAE}\left(m_{\text{RP}}, \mathcal{T}_{\text{val}}\right) = \frac{\text{MAE}\left(m_{\text{RP}}, \mathcal{T}_{\text{val}}\right)}{\max\limits_{((u,i),r_{u,i})\in\mathcal{T}} (r_{u,i}) - \min\limits_{((u,i),r_{u,i})\in\mathcal{T}} (r_{u,i})}$$

$$\text{MSE}\left(m_{\text{RP}}, \mathcal{T}_{\text{val}}\right) = \frac{1}{|\mathcal{T}_{\text{val}}|} \sum_{((u,i),r_{u,i})\in\mathcal{T}_{\text{val}}} \left(m_{\text{RP}}(u,i) - r_{u,i}\right)^2$$

$$\text{RMSE}\left(m_{\text{RP}}, \mathcal{T}_{\text{val}}\right) = \sqrt{\text{MAE}\left(m_{\text{RP}}, \mathcal{T}_{\text{val}}\right)}$$

Binary Classification $\mathbf{R} \in \{0, 1, ?\}^{U\times I}$

$$\mathcal{T} = \{((u,i), r_{u,i}) \mid u \in U \wedge i \in I \wedge r_{u,i} \neq ?\}$$

$$\mathit{precision}\left(m, \mathcal{T}_{\text{val}}\right) = \frac{|\{((u,i), r_{u,i}) \in \mathcal{T}_{\text{val}} \mid m(u,i) = 1 \wedge r_{u,i} = 1\}|}{|\{((u,i), r_{u,i}) \in T_{\text{val}} \mid m(u,i) = 1\}|}$$

$$\mathit{recall}\left(m, \mathcal{T}_{\text{val}}\right) = \frac{|\{((u,i), r_{u,i}) \in \mathcal{T}_{\text{val}} \mid m(u,i) = 1 \wedge r_{u,i} = 1\}|}{|\{((u,i), r_{u,i}) \in T_{\text{val}} \mid r_{u,i} = 1\}|}$$

$$F_1\left(m, \mathcal{T}_{\text{val}}\right) = \frac{2 \cdot \mathit{precision}\left(m, \mathcal{T}_{\text{val}}\right) \cdot \mathit{recall}\left(m, \mathcal{T}_{\text{val}}\right)}{\mathit{precision}\left(m, \mathcal{T}_{\text{val}}\right) + \mathit{recall}\left(m, \mathcal{T}_{\text{val}}\right)}$$

Generalized Validation Loss and Validation Reward

Top-N Recommendation

$$\mathcal{T} = \left\{ ((u, obs), target) \mid u \in U \land obs \cup target = r^+(u) \land obs \cap target = \emptyset \land obs \neq \emptyset \right\}$$

Accuracy Measures

$$precision@N\,(m, \mathcal{T}_{\text{val}}) = \frac{1}{|\mathcal{T}_{\text{val}}|} \sum_{((u,obs),target) \in \mathcal{T}_{\text{val}}} \frac{|m(u, obs) \cap target|}{|m(u, obs)|}$$

$$recall@N\,(m, \mathcal{T}_{\text{val}}) = \frac{1}{|\mathcal{T}_{\text{val}}|} \sum_{((u,obs),target) \in \mathcal{T}_{\text{val}}} \frac{|m(u, obs) \cap target|}{|target|}$$

$$recall_{\text{PS}}^{\beta,w}@N\,(m, \mathcal{T}_{\text{val}}) = \sum_{((u,obs),target)} w^{\beta}(u) \cdot \frac{\sum_{i \in target \cap m(u,obs)} p(i)^{-\beta}}{\sum_{i \in target} p(i)^{-\beta}}$$

$$p(i) = |\{u \in U \mid r_{u,i} = 1\}|$$

$$recall@N_{\text{LOO}}\,(m, \mathcal{T}_{\text{val}}) = \frac{|\{(u, i) \mid (u, r^+(u)) \in \mathcal{T}_{\text{val}} \land i \in r^+(u) \land i \in m\,(u, r^+(u) \setminus \{i\})\}|}{|\{(u, i) \mid (u, r^+(u)) \in \mathcal{T}_{\text{val}} \land i \in r^+(u)\}|}$$

Coverage Measures

$$catalog\text{-}coverage\,(m, \mathcal{T}_{\text{val}}) = \frac{1}{|I|} \left| \bigcup_{((u,obs),target) \in \mathcal{T}_{\text{val}}} m(u, obs) \right|$$

$$user\text{-}coverage(m, \mathcal{T}_{\text{val}}) = \frac{1}{|U|} \sum_{((u,obs),target) \in \mathcal{T}_{\text{val}}} \begin{cases} 0 & \text{if } m(u, obs) \neq \emptyset \\ 1 & \text{if } m(u, obs) = \emptyset \end{cases}$$

Validation Reward Function
=
Generalization of Accuracy and Coverage

Model Capacity Hyperparameters

For Individual Learning Algorithms

$$
\begin{aligned}
\mathcal{P}_{\mathcal{A}_{\mathrm{UserKnn}}} &= [k \in \mathbb{N}, \beta \in \mathbb{R}] \\
\mathcal{P}_{\mathcal{A}_{\mathrm{ItemKnn}}} &= [k \in \mathbb{N}, \beta \in \mathbb{R}] \\
\mathcal{P}_{\mathcal{A}_{\mathrm{AR}}} &= [s_{\min} \in [0,1], \beta \in \mathbb{R}] \\
\mathcal{P}_{\mathcal{A}_{\mathrm{MF}}} &= [f \in \mathbb{N}, \lambda \in \mathbb{R}, \beta \in \mathbb{R}] \\
\mathcal{P}_{\mathcal{A}_{\mathrm{U\text{-}AutoRec}}} &= [f \in \mathbb{N}, \lambda \in \mathbb{R}, \beta \in \mathbb{R}] \\
\mathcal{P}_{\mathcal{A}_{\mathrm{I\text{-}AutoRec}}} &= [f \in \mathbb{N}, \lambda \in \mathbb{R}, \beta \in \mathbb{R}]
\end{aligned}
$$

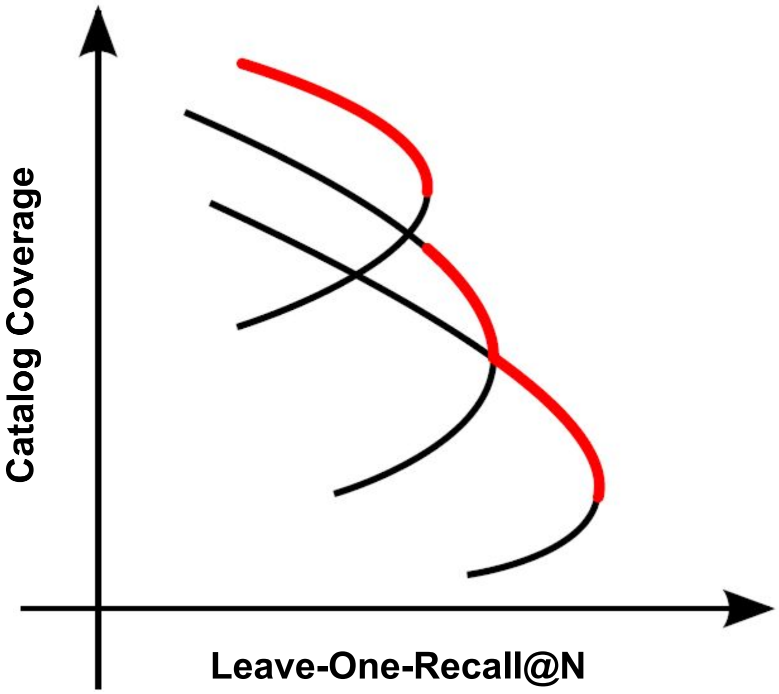
β = Universal Capacity Hyperparameter

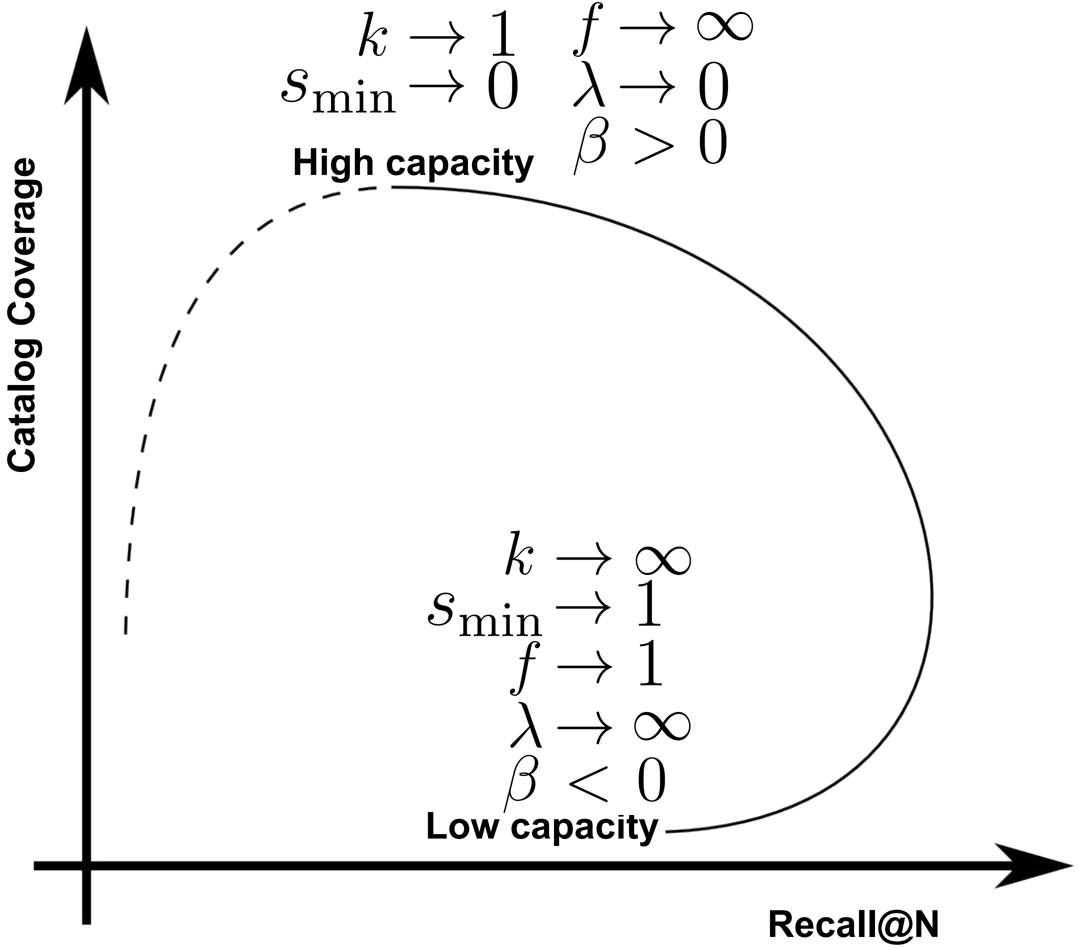
Using score(u,i)

$$score^{\beta}_{\mathrm{PS}}(u,i) = \begin{cases} \dfrac{score(u,i)}{|\{v \in U \mid r_{v,i} > 0\}|^{\beta}} & \text{if } \exists v \in U \colon r_{v,i} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Recall-Coverage Optimization

Searching for Pareto-Optimal States




$k \to 1$ $f \to \infty$
$s_{\min} \to 0$ $\lambda \to 0$
$\beta > 0$
High capacity
Catalog Coverage
$k \to \infty$
$s_{\min} \to 1$
$f \to 1$
$\lambda \to \infty$
$\beta < 0$
Low capacity
Recall@N

Experiments

- Different learning algorithms
- Mixture of academic and industrial datasets

Dataset Name	Total Interactions	Interaction Types	Item Attributes
MovieLens 1M	1,000,209	Explicit ratings	Genres, Title, Year
MovieLens 20M	20,000,263	Explicit ratings	Genres, Title, Year
Last.FM 2K	92,834	Plays	$\emptyset$
BUBB.Store	8,697,556	Detail-views, Cart-additions, Purchases, Explicit ratings	Store ID, Brand, Categories, Name
Casa Cenina	9,891,560	Detail-views, Bookmarks, Cart-additions, Purchases, Explicit ratings	Categories, Product tags, Manufacturer, Product model
Just Spotted	875,013	Detail-views, Bookmarks, Cart-additions, Purchases, Explicit ratings	Product name, Vendor, Type
Moodings	2,884,636	Detail-views, Bookmarks, Cart-additions, Purchases, Explicit ratings	Product name, Vendor, Type

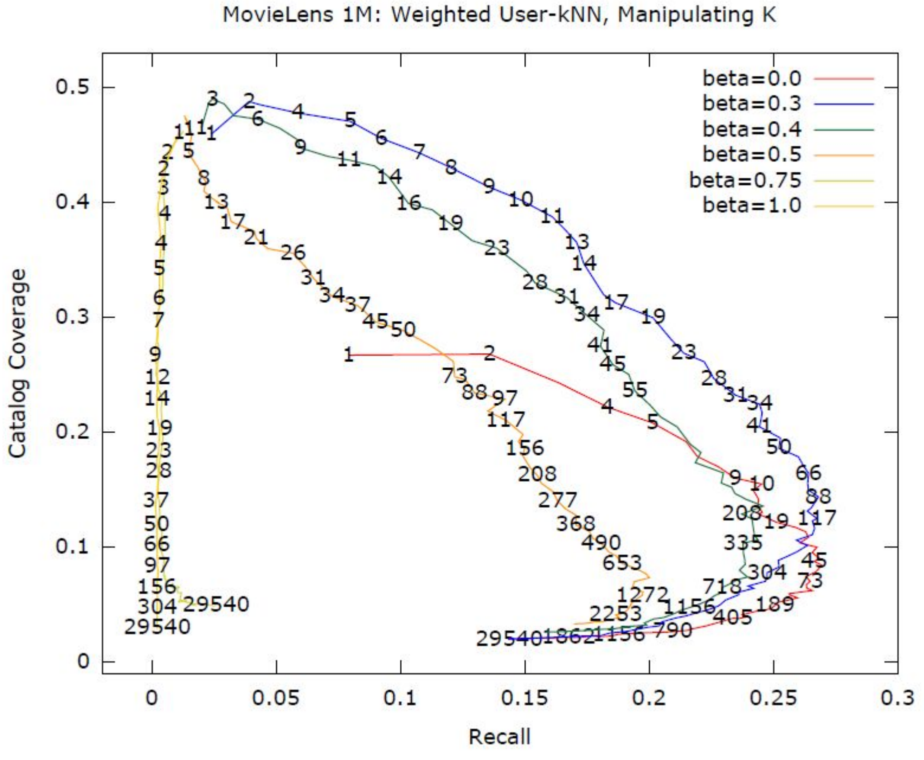
Experiments

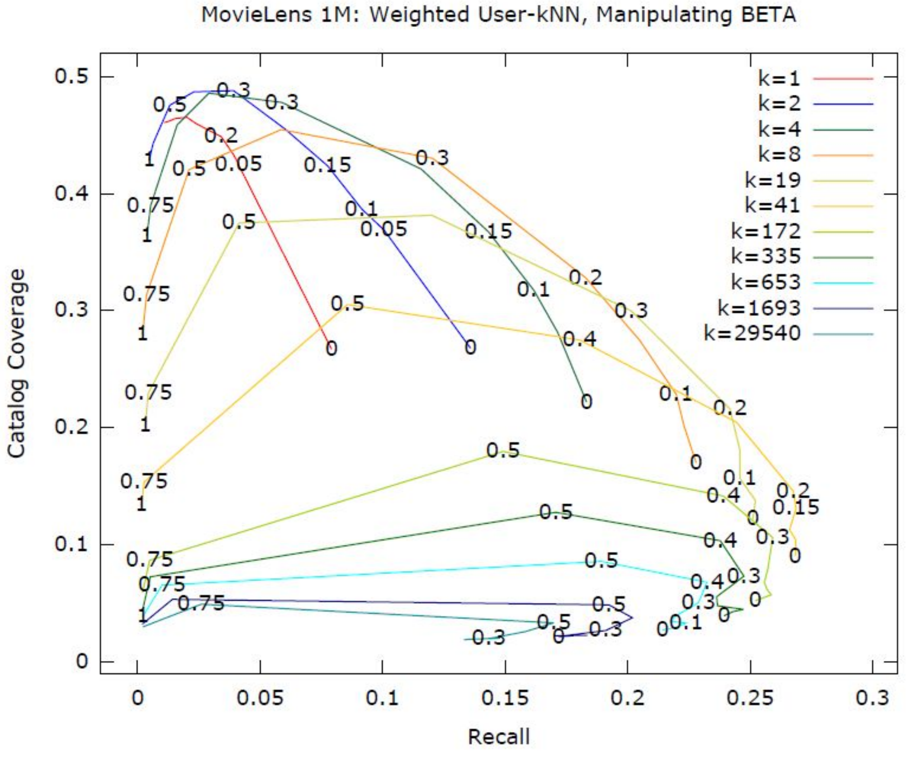
- Different characteristics of academic and industrial datasets

Dataset Name	$\|U\|$	Interactions per User				
		P_1	Q_1	Q_2	Q_3	P_{99}
MovieLens1M	6040	20	44	96	208	909
MovieLens20M	138493	20	35	68	155	1114
Last.FM 2K	1892	10	50	50	50	50
BUBB.Store	6119542	1	1	1	1	12
Casa Cenina	465519	1	1	2	6	335
Just Spotted	340842	1	1	2	4	41
Moodings	562170	1	1	2	5	75

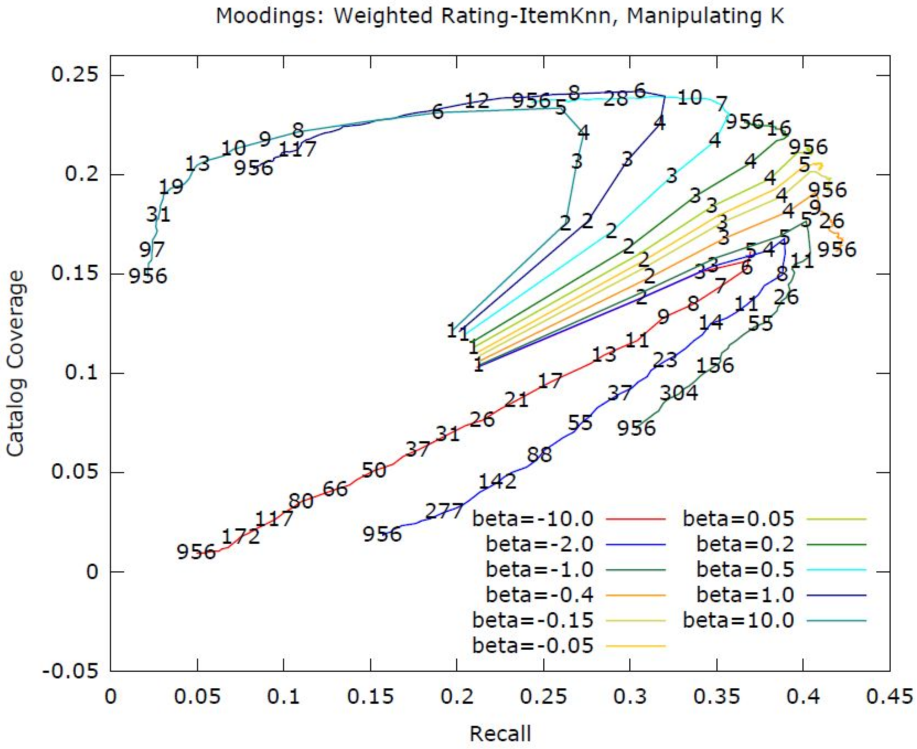
Dataset Name	$\|I\|$	Interactions per Item				
		P_1	Q_1	Q_2	Q_3	P_{99}
MovieLens1M	3883	0	26	109	330	1783
MovieLens20M	27278	0	3	16	194	14309
Last.FM 2K	17632	1	1	1	3	81
BUBB.Store	51380	0	0	2	32	1906
Casa Cenina	124559	0	4	23	82	837
Just Spotted	4179	0	14	48	143	3063
Moodings	11975	0	29	77	211	2721

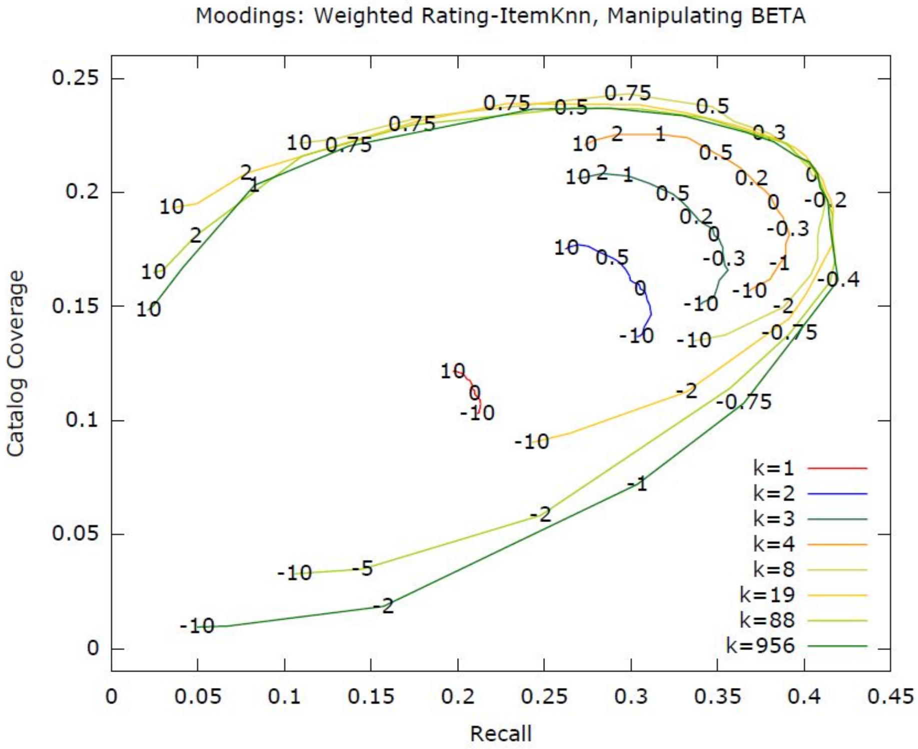
UserKnn


MovieLens 1M: Weighted User-kNN, Manipulating K
Catalog Coverage
Recall
beta=0.0
beta=0.3
beta=0.4
beta=0.5
beta=0.75
beta=1.0

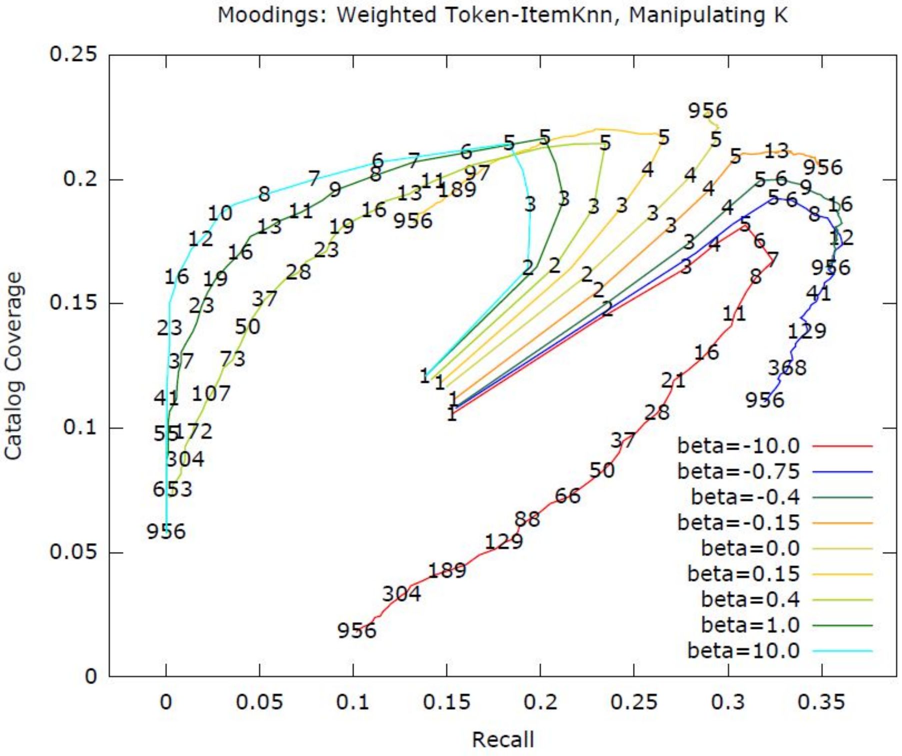

MovieLens 1M: Weighted User-kNN, Manipulating BETA
Catalog Coverage
Recall
k=1
k=2
k=4
k=8
k=19
k=41
k=172
k=335
k=653
k=1693
k=29540

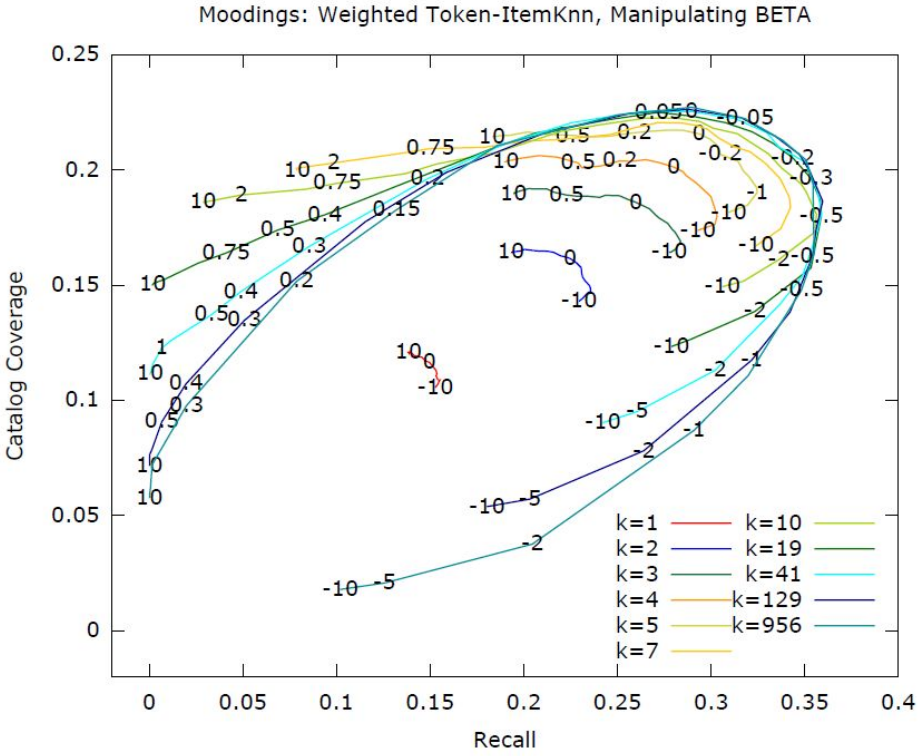
ItemKnn: Rating Similarity



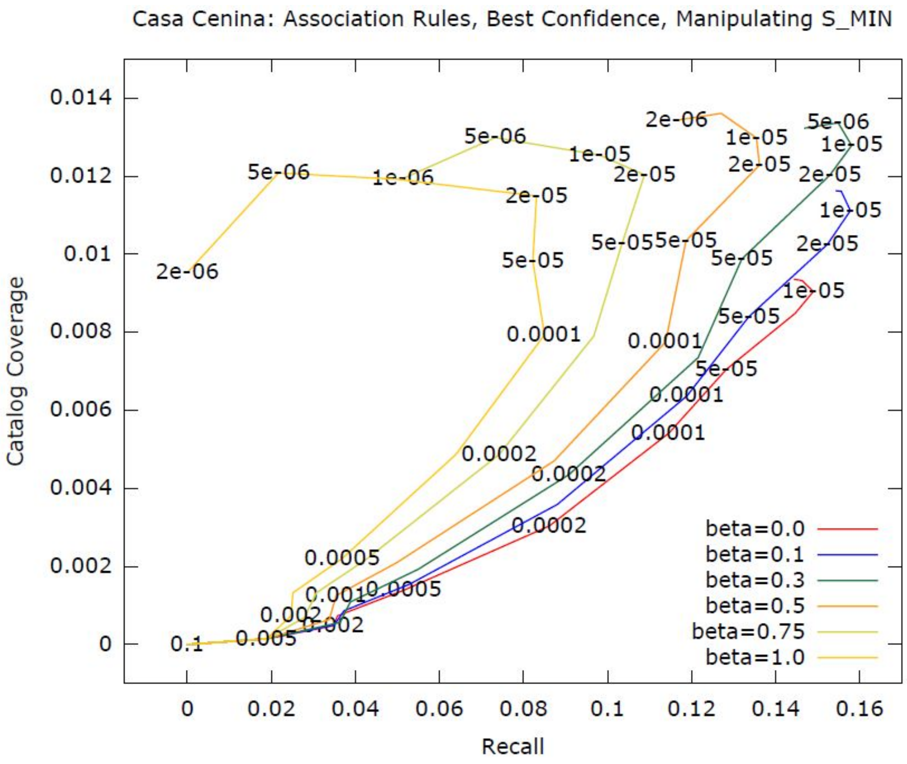


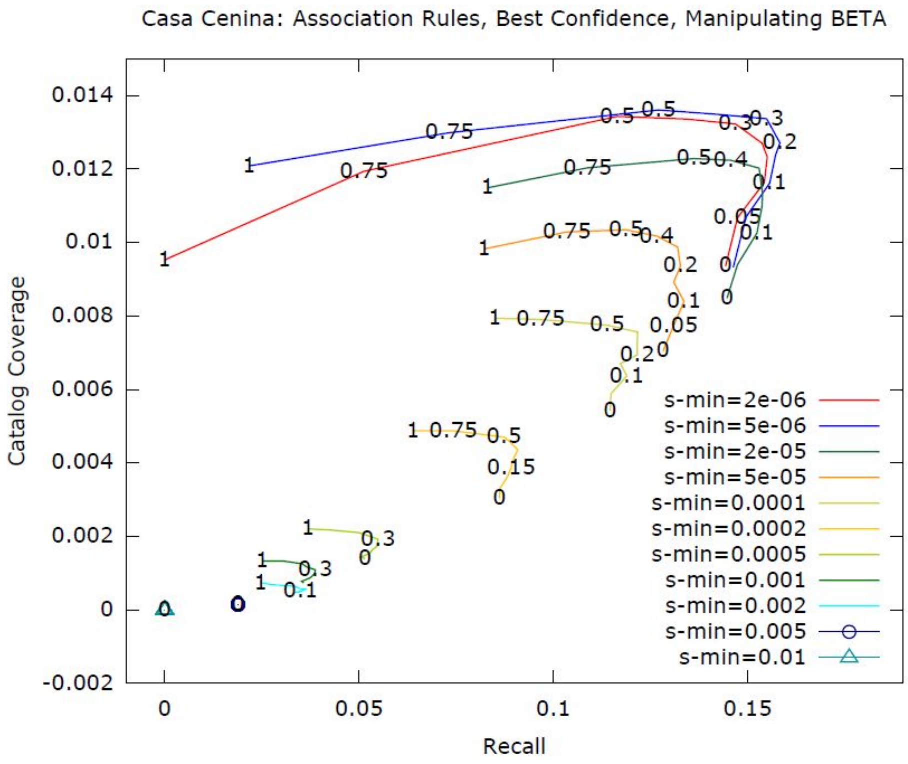
ItemKnn: Attribute Similarity



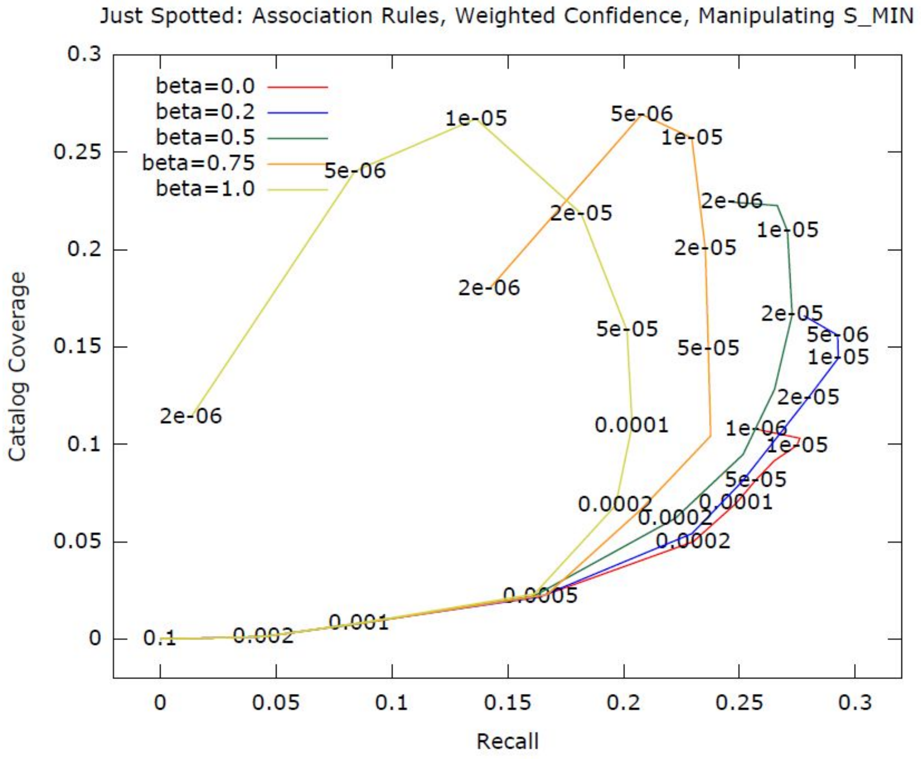


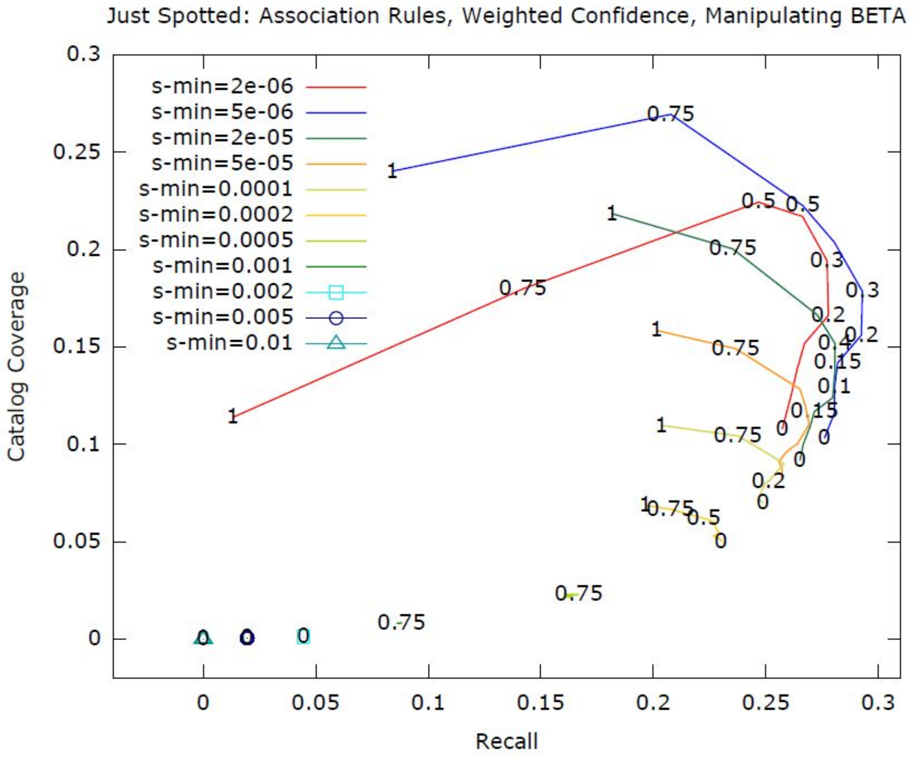
Association Rules: Best-Rule Method



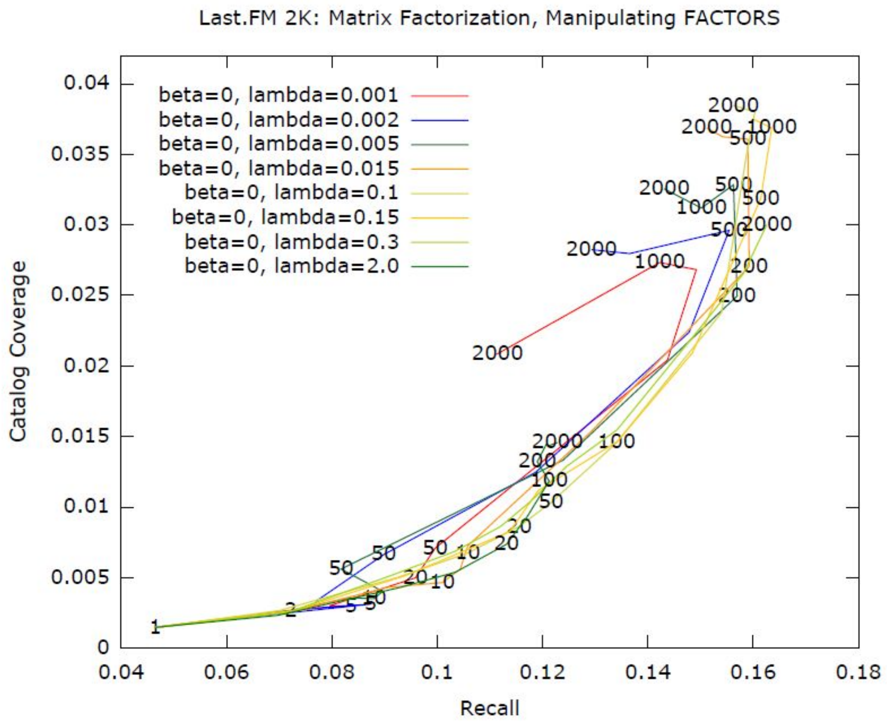


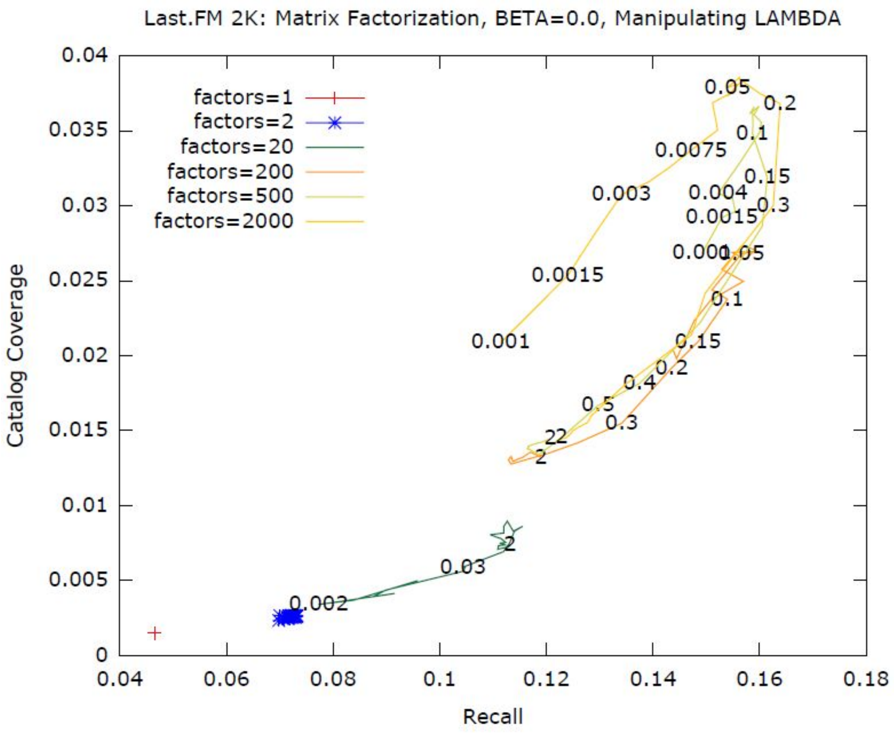
Association Rules: Weighted Voting Method



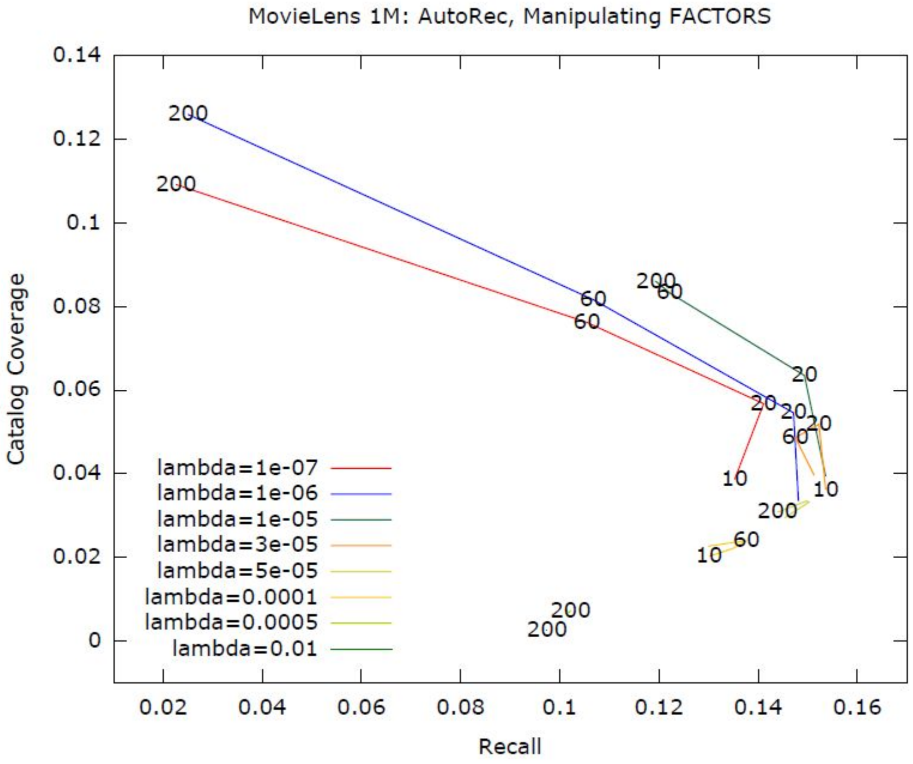


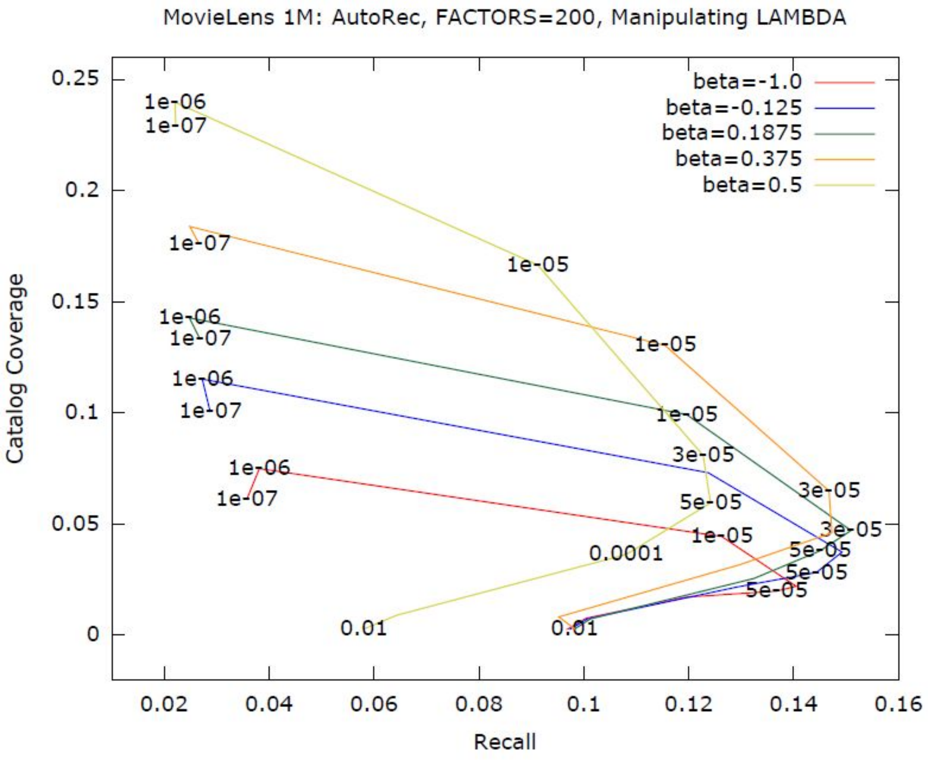
Matrix Factorization





AutoRec


MovieLens 1M: AutoRec, Manipulating FACTORS
Catalog Coverage
Recall
lambda=1e-07
lambda=1e-06
lambda=1e-05
lambda=3e-05
lambda=5e-05
lambda=0.0001
lambda=0.0005
lambda=0.01


MovieLens 1M: AutoRec, FACTORS=200, Manipulating LAMBDA
Catalog Coverage
Recall
beta=-1.0
beta=-0.125
beta=0.1875
beta=0.375
beta=0.5

Related Activities of the Author

- Supervised 20 Bachelor+Master theses
  - 7× received the Dean’s award
- Created and lectured Fundamentals of Artificial Intelligence course
  - 200+ students / 1 run
  - created full course from scratch, giving lectures and seminars for 5 years
- Co-Founded Recombee, designed and implemented Recombee Engine
  - Supervising demand-inspired research in embeddings, deep learning, auto ML
  - SaaS, 3000 online registrations,
  - 50 recurrently paying customers from 30 countries,
  - running real-time service on 120 servers, 1000 CPUs, 10TB RAM
  - currently maintained and improved by team of 10 people
  - customers with
    - 200 M interactions / month,
    - 300 recommendations / second
    - 20 M items with text descriptions and average 5 images per item

Thank you for your attention!

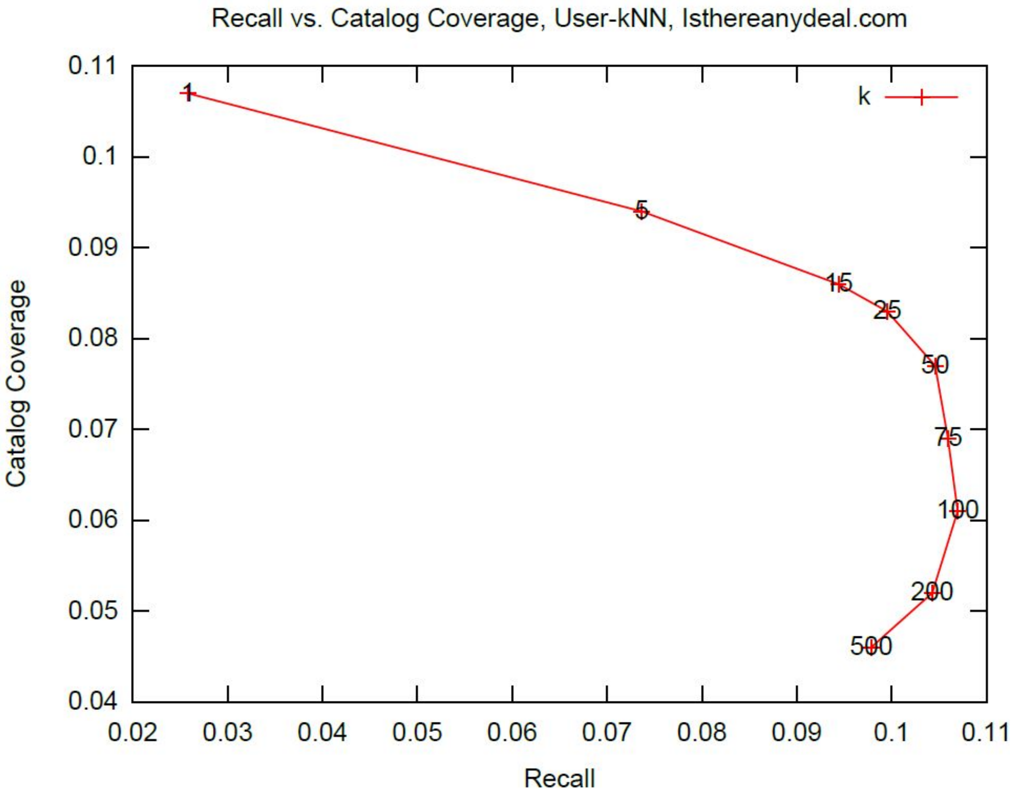
Questions?

Performance in Production

Recombee

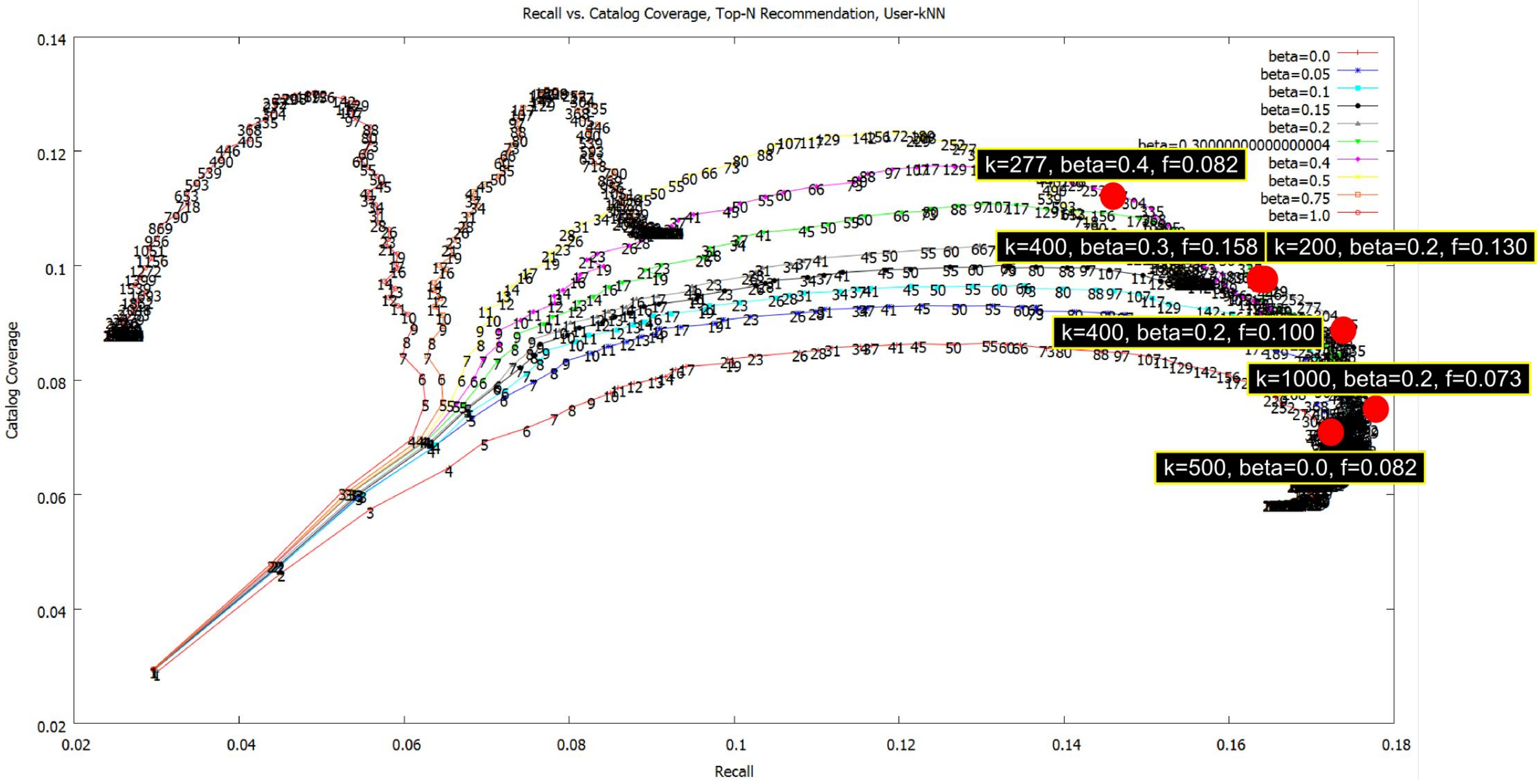
- Presented models are running live in Recombee
  - Incremental implementation (response to new data)
  - Dozens of use-cases (E-Commerce, news, classifieds, movies, job boards, cultural events...)
  - Initial implementation done by author of this thesis
- Capacity manipulation from dissertation frequently A/B tested in production
- Using $\beta > 0$ by default for some models
- Maximizing recall: better than random, but sometimes suboptimal
- RCO:
  - sometimes better than recall,
  - sometimes both recall and RCO are deceiving for user response (CTR, CR)

IsThereAnyDeal.com Experiment

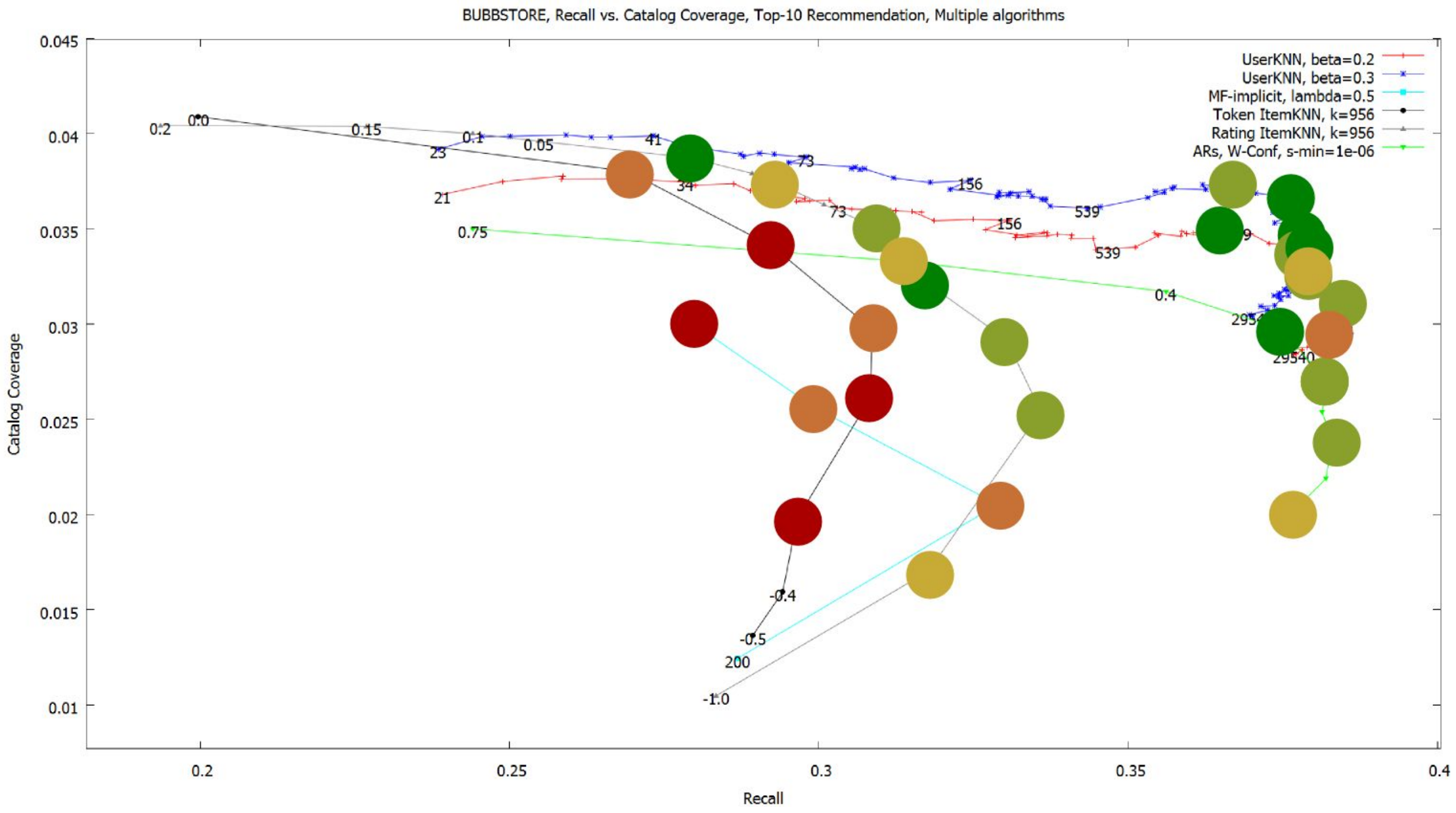


Method	# Clicks	Improvement over old method
old	64	0%
5-NN	93	45.31%
25-NN	**263**	**310.94%**
100-NN	155	142.19%
500-NN	154	140.63%

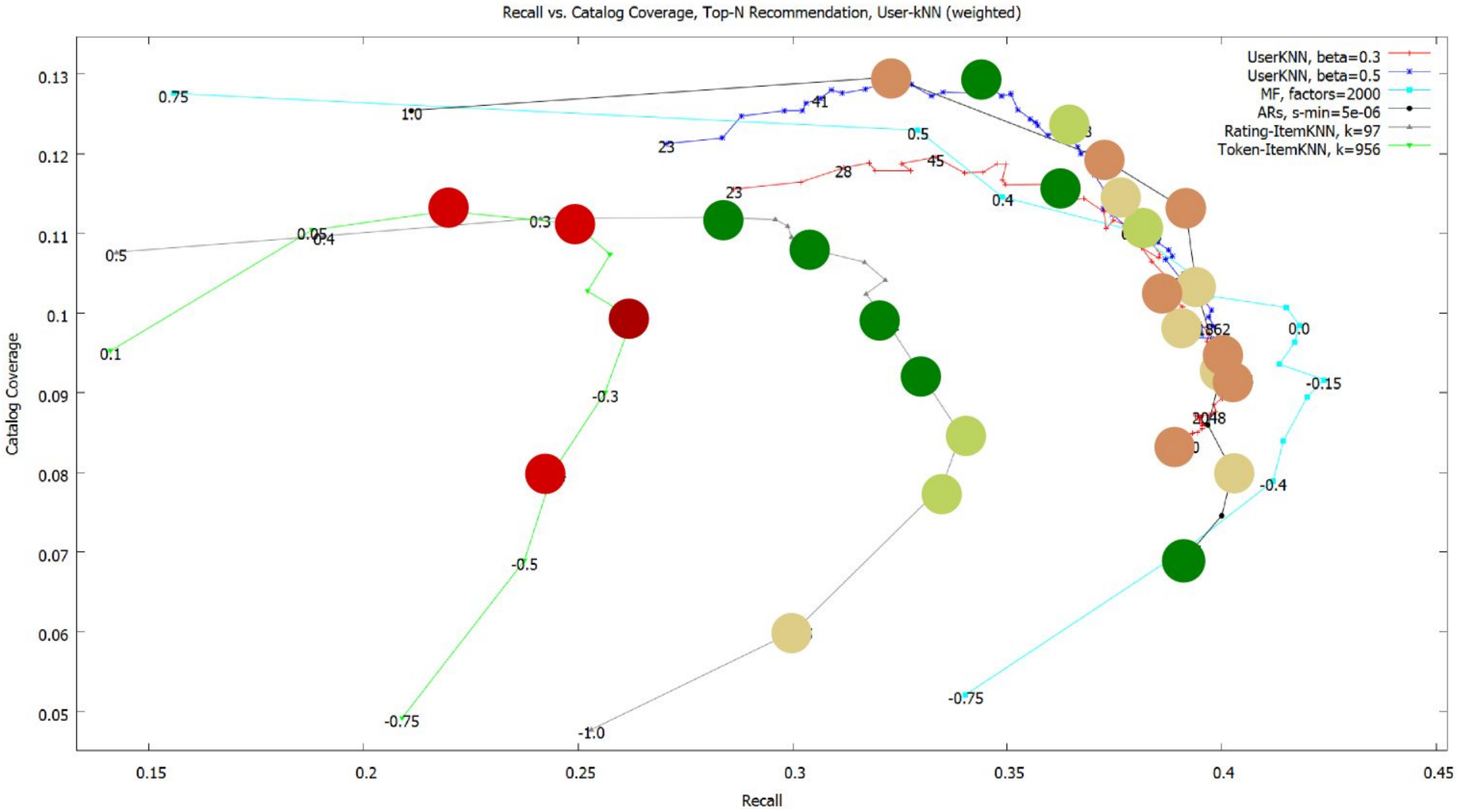
Online Experiment: Large Czech Job Board



Other Experiments (BUBB.Store)

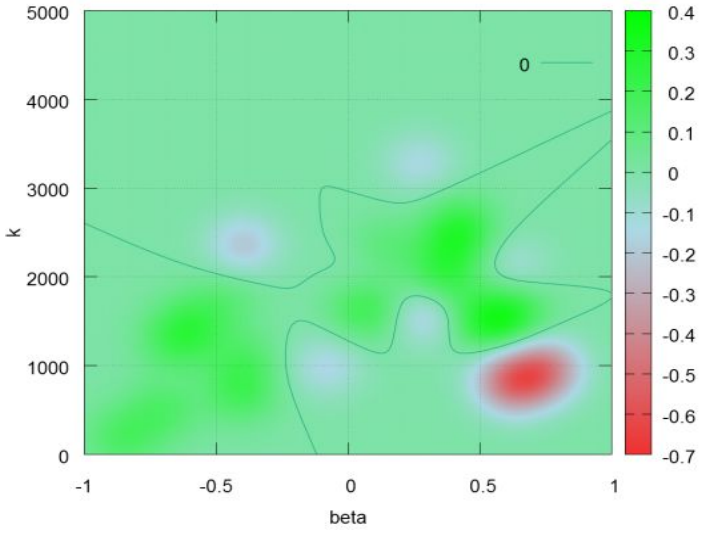


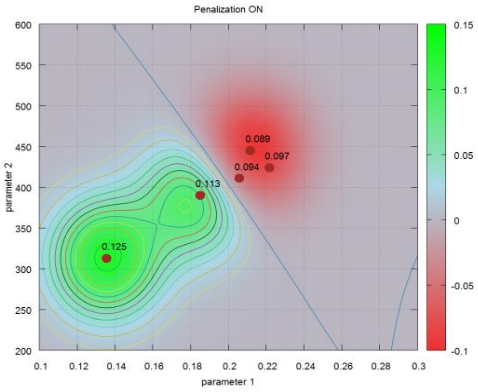
Other Experiments (Moodings)

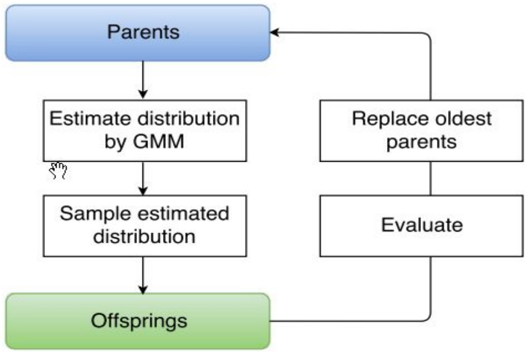


Online Model Optimization (CMA-ES)

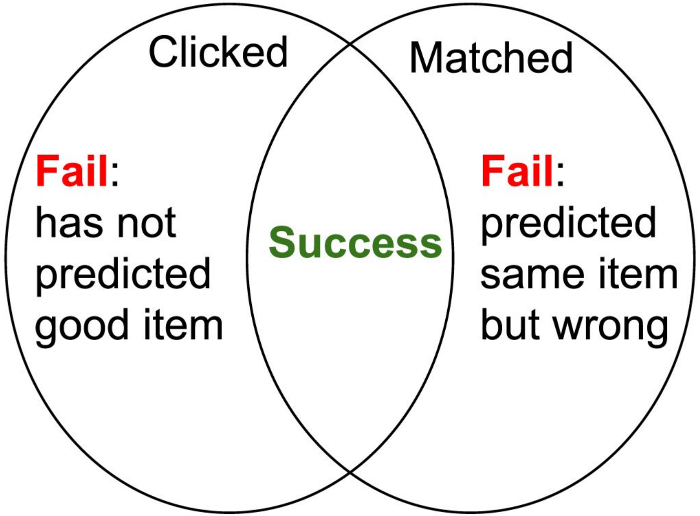
- Bc. Radek Bartyzal



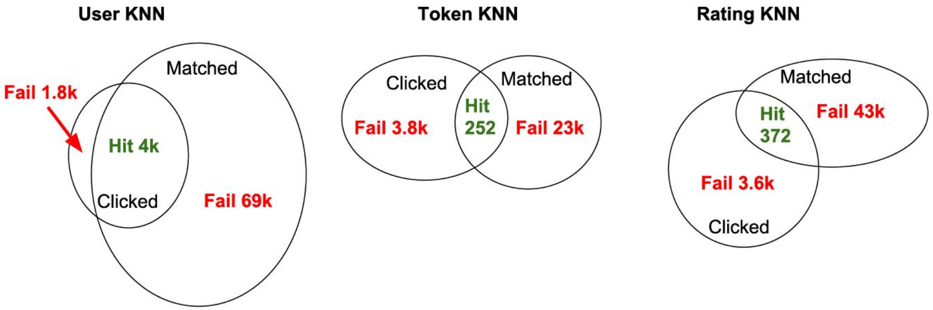




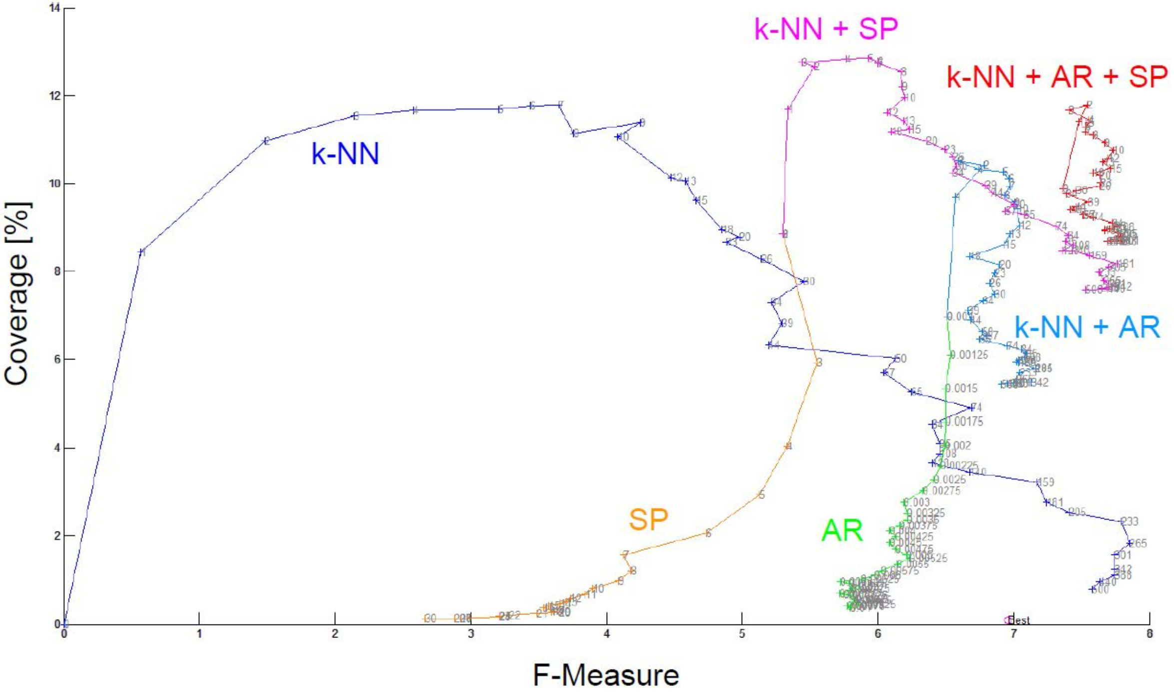
Comparing Offline and Online Evaluation Results of Recommender Systems (RecSys Paper)

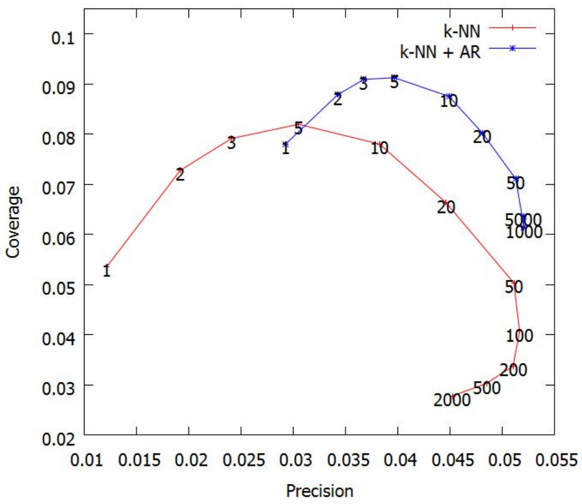


$$JIE = \frac{\text{clicked} \cap \text{matched}}{\text{clicked} \cup \text{matched}}$$



Model Ensembles in RCO

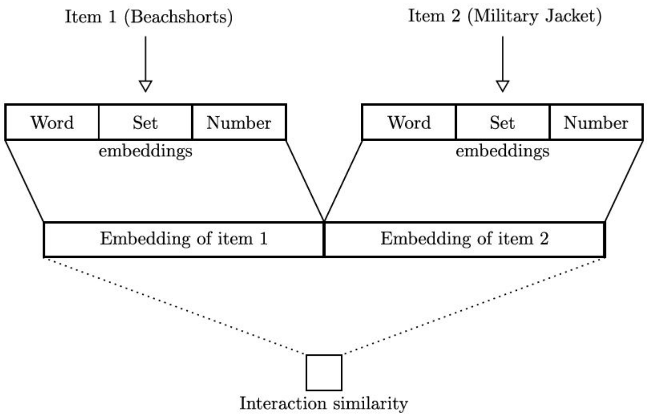


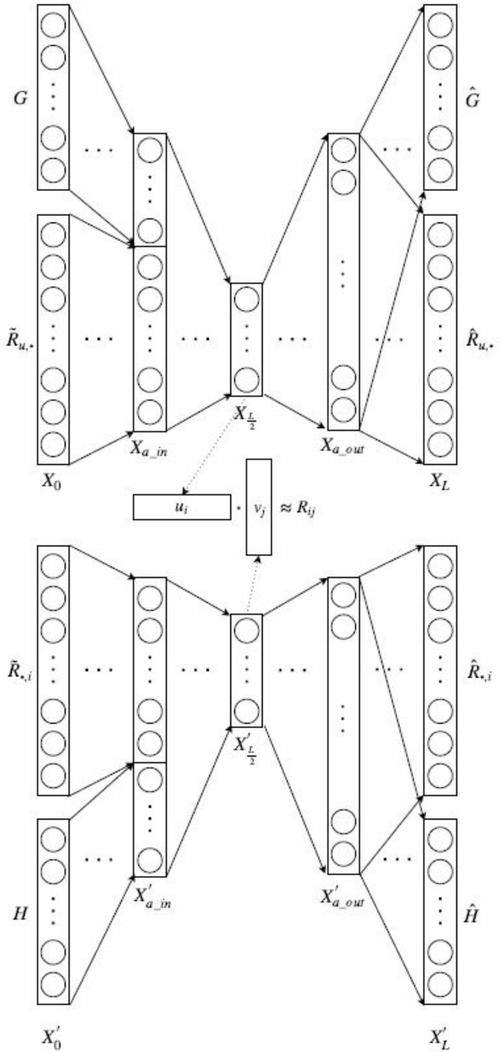


Other Research Supervised by Author

Cold-Start

- Rating data = high quality, but limited
- Recommendation based on attributes needed
- Researched topics:
    - Feature selection by Semi-supervised learning (Bc. Michal Režnický)
    - Neural prediction of interaction similarity from attributes (Bc. Petr Kasalický)
    - Stacked denoising autoencoders (Bc. Radek Bartyzal)

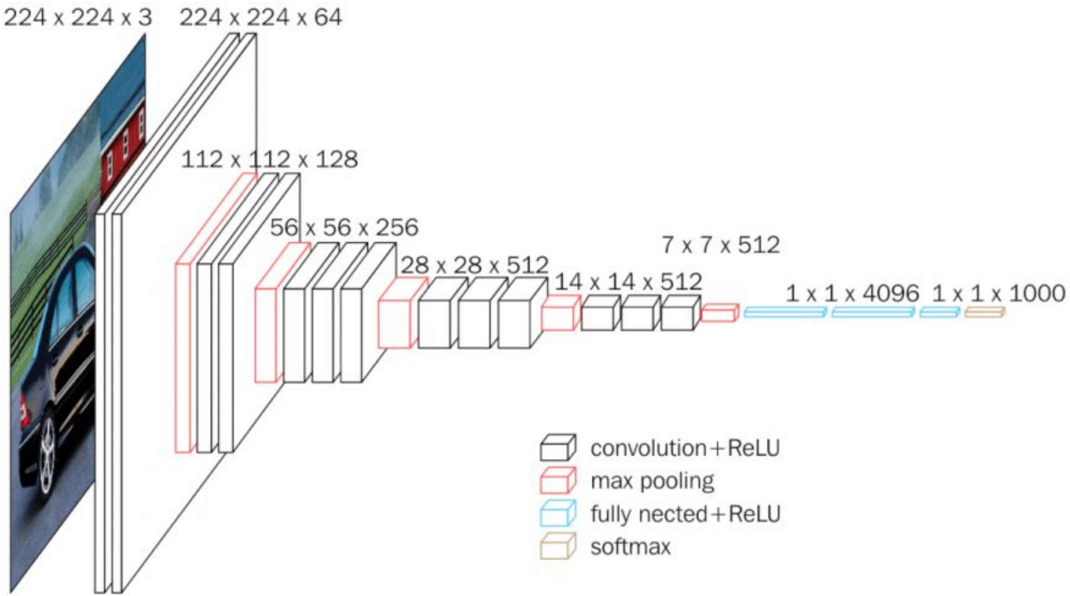


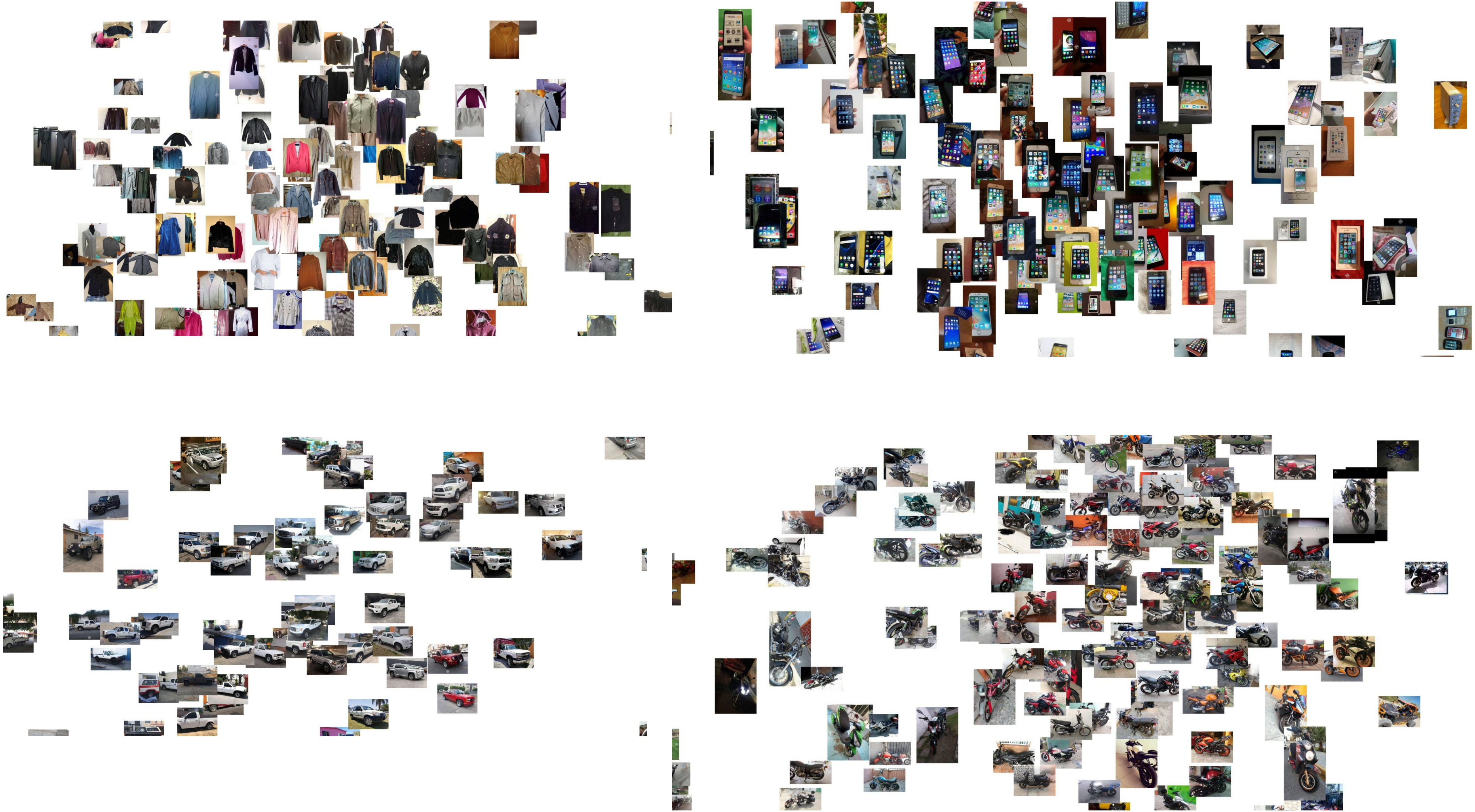


Other Research Supervised by Author

Image Embeddings

- Image-Based ItemKnn (Bc. Martin Pavlíček)
- Re-training convolutional networks from interactions (Bc. Petr Kasalický)





Other Research Supervised by Author

Hyper-Embeddings

- Ing. Ivan Povalyaev (Recombee)

Other Research Supervised by Author

Showmax Scene Embeddings for Recommendation

- Ing. Ivan Povalyaev (Recombee), Bc. Ondřej Bíža (ShowmaxLab)