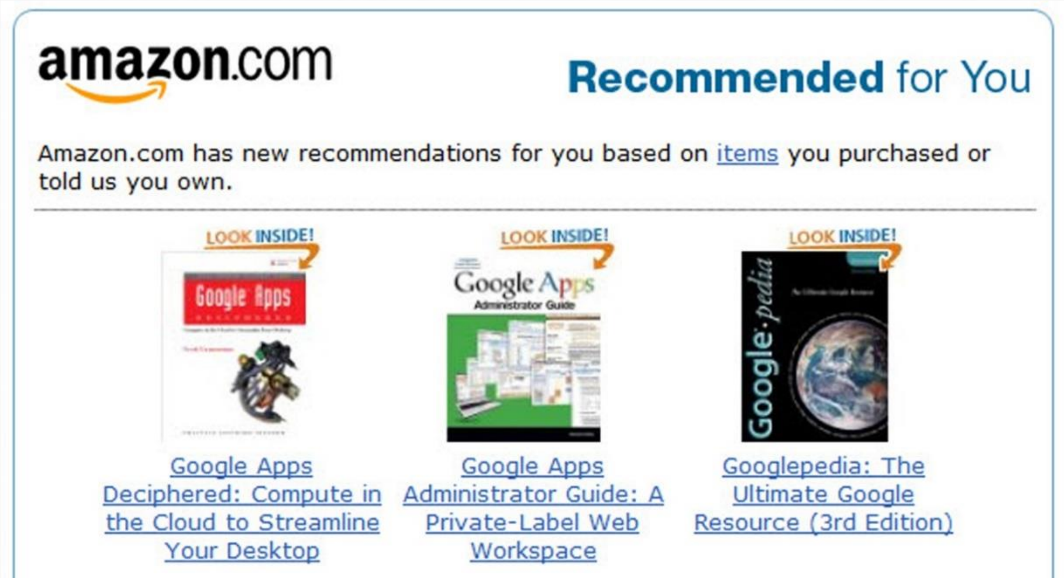
* Matrix Factorization and Recommendation Systems

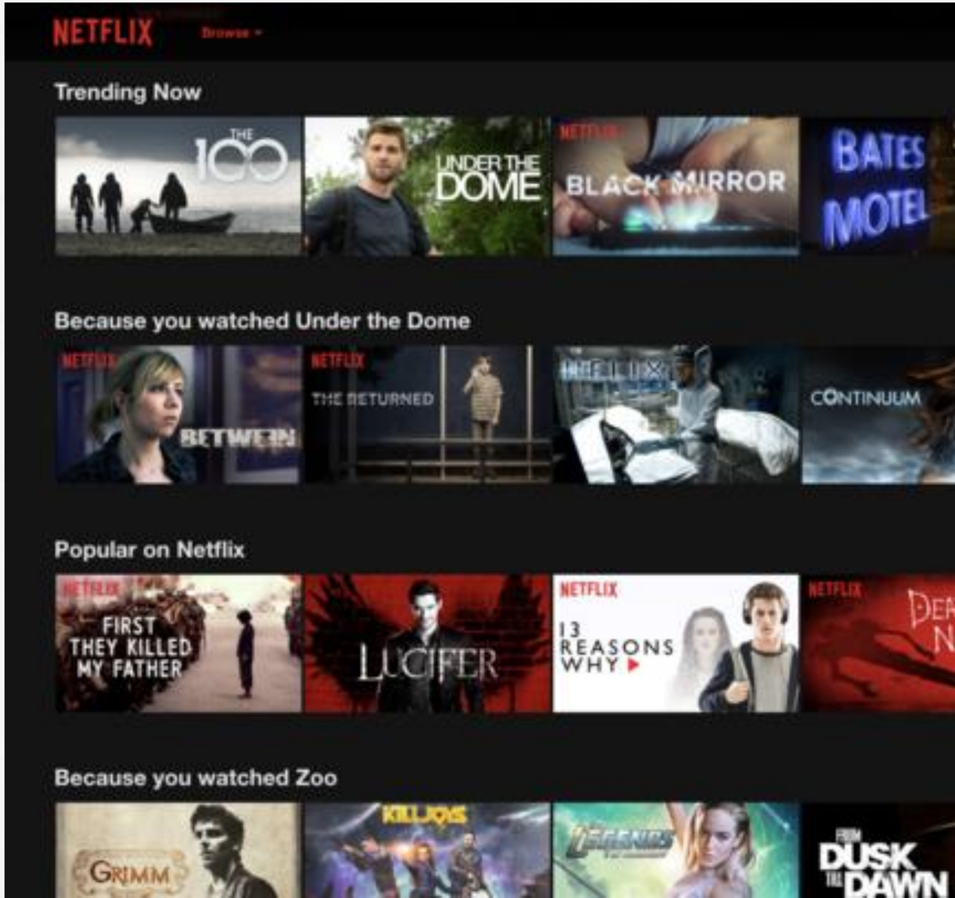
Originally presented at HLF Workshop on Matrix Factorization
with Loren Anderson (University of Minnesota Twin Cities)
on 25th September, 2017

15th March, 2018
Datalys

Recommendation is must have in e-commerce

- Movies, products, etc.
- Preventing information overload
- Explicit (rating) vs. implicit (click) feedback





Recommendation = Prediction of rating

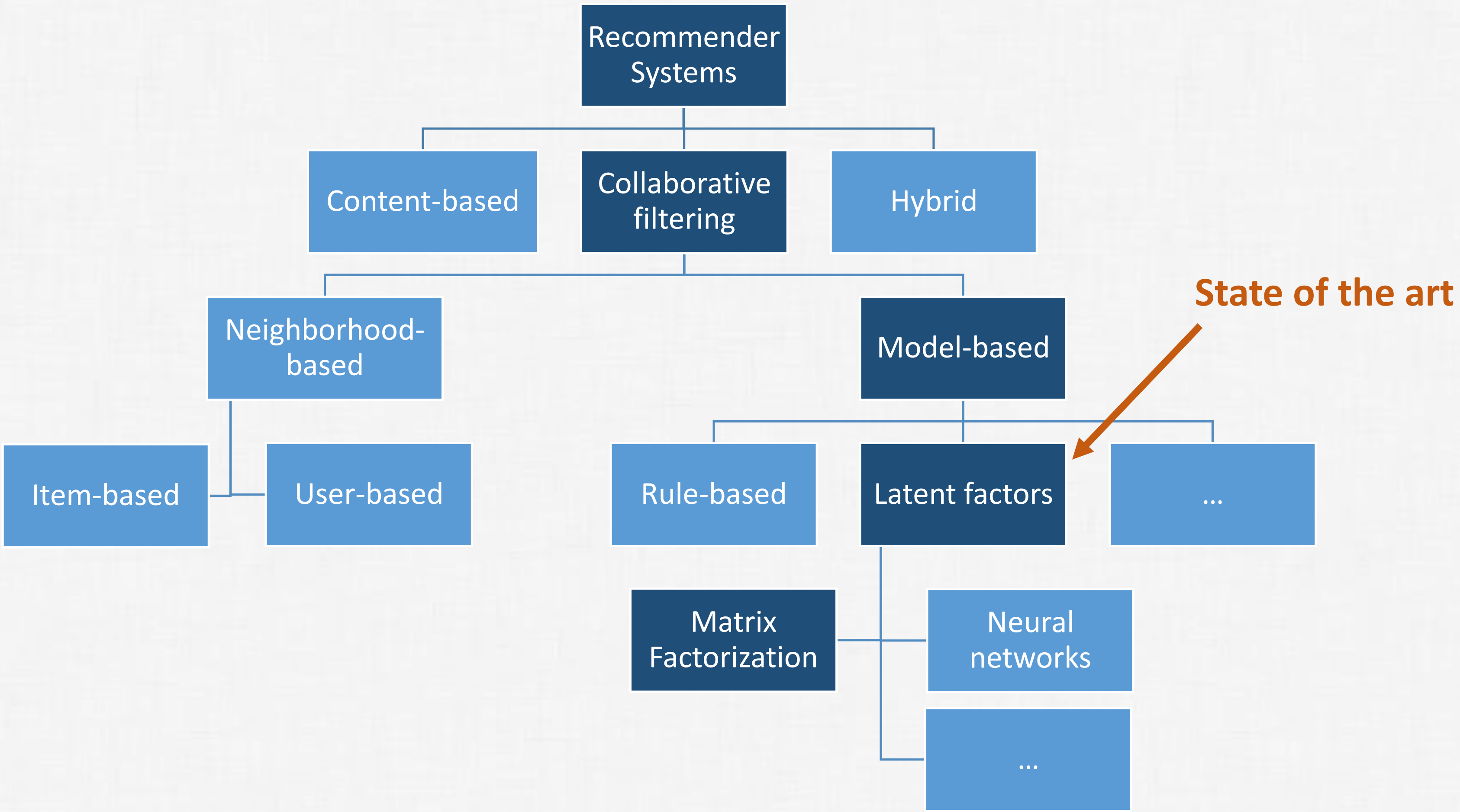
	Item 1	Item 2	Item 3	Item 4	Item 5
Bob	5	3	4	5	?
User 1	3	1	2	3	3
User 2	4	3	4	3	5
User 3	3	3	1	5	4
User 4	1	5	5	2	1

Recommendation = Prediction of rating

	Item 1	Item 2	Item 3	Item 4	Item 5
Bob	3		4		?
User 1	3				3
User 2		3		3	
User 3	3		4		4
User 4	1			2	

- In real systems, matrix is really big and sparse
  - Dataset from Netflix Price: 500,000 users × 17,000 movies
  - 100 million (1%) ratings are known

Recommender systems - Classification



Matrix factorization – Formalization

- Goal is to map users and items into latent factor f-dimensional space
  - $m \times n$ ratings matrix R is approximately factorized into an $m \times f$ matrix P and an $n \times f$ matrix Q

$$R \approx PQ^T$$

$$Users \overset{Items}{\begin{bmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mn} \end{bmatrix}} \approx Users \overset{Factors}{\begin{bmatrix} p_{11} & \cdots & p_{1f} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mf} \end{bmatrix}} \quad \text{Factors} \overset{Items}{\begin{bmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{f1} & \cdots & q_{fn} \end{bmatrix}}$$

Matrix factorization – Formalization

- Goal is to map users and items into latent factor f-dimensional space
  - $m \times n$ ratings matrix R is approximately factorized into an $m \times f$ matrix P and an $n \times f$ matrix Q

$$R \approx PQ^T$$

$$Users \overset{Items}{\begin{bmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mn} \end{bmatrix}} \approx Users \overset{Factors}{\begin{bmatrix} p_{11} & \cdots & p_{1f} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mf} \end{bmatrix}} \ \text{Factors} \overset{Items}{\begin{bmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{f1} & \cdots & q_{fn} \end{bmatrix}}$$

- Each item i is associated with a vector q_i (item factor)
  - q_i measures affinity of the item i towards those f factors
- Each user u is associated with a vector p_u (user factor)
  - p_u measures interest of the user in these f factors

Matrix factorization – Formalization

- Goal is to map users and items into latent factor f-dimensional space
  - $m \times n$ ratings matrix R is approximately factorized into an $m \times f$ matrix P and an $n \times f$ matrix Q

$$R \approx PQ^T$$

$$Users \overset{Items}{\begin{bmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mn} \end{bmatrix}} \approx Users \overset{Factors}{\begin{bmatrix} p_{11} & \cdots & p_{1f} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mf} \end{bmatrix}} \quad \text{Factors} \overset{Items}{\begin{bmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{f1} & \cdots & q_{fn} \end{bmatrix}}$$

- Each item i is associated with a vector q_i (item factor)
  - q_i measures affinity of the item i towards those f factors
- Each user u is associated with a vector p_u (user factor)
  - p_u measures interest of the user in these f factors
- Rating $\widehat{r_{ui}}$ of user u on item i can be approximated as a dot product $\widehat{r_{ui}} = q_i^T p_u$

Matrix factorization – Basic model

- The main challenge is computing of user and item factors
  - Incomplete and sparse rating matrix R (conventional SVD is not applicable)
  - Overfitting when only few ratings are available

Matrix factorization – Basic model

- The main challenge is computing of user and item factors
  - Incomplete and sparse rating matrix R (conventional SVD is not applicable)
  - Overfitting when only few ratings are available
- Solution 1: amputation of missing values
  - Memory inefficient
  - Introduce bias

Matrix factorization – Basic model

- The main challenge is computing of user and item factors
  - Incomplete and sparse rating matrix R (conventional SVD is not applicable)
  - Overfitting when only few ratings are available
- Solution 2: modeling only known ratings + minimize error $e_{ui} = (r_{ui} - \widehat{r_{ui}})$

$$\min_{q,p} \sum_{(u,i)\in R} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2)$$

  - Parameter λ controls the extent of regularization
- Optimization problem -> learning algorithms
  - Stochastic gradient descent
  - Alternating least squares

Matrix factorization – Differences in models

- Constraints imposed on P and Q
  - Unconstrained MF
  - Orthogonality of the latent vectors (SVD)
  - Non-negativity of the latent vectors (NMF)

* Matrix factorization – Differences in models

- Constraints imposed on P and Q
  - Unconstrained MF
  - Orthogonality of the latent vectors (SVD)
  - Non-negativity of the latent vectors (NMF)
- Objective function
  - Minimize squared (Frobenius) norm
  - Maximizing the likelihood estimation

Matrix factorization – Differences in models

- Constraints imposed on P and Q
  - Unconstrained MF
  - Orthogonality of the latent vectors (SVD)
  - Non-negativity of the latent vectors (NMF)
- Objective function
  - Minimize squared (Frobenius) norm
  - Maximizing the likelihood estimation
- Additional information used in modeling of ratings
  - User and item bias (SVD)
  - Implicit feedback (SVD++)
  - Temporal dynamics (TimeSVD++)

* Improvement of basic model for real systems

Incorporating user and item bias

- Real systems usually exhibit some systematical tendencies
  - Some users tend to give higher ratings
  - Some items are widely rated better

$$\widehat{r_{ui}} = \mu + b_i + b_u + q_i^T p_u$$

  - μ denotes overall average rating
  - b_i represents item bias
  - b_u represents user bias

Incorporating implicit feedback

- Real systems must tackle with cold-start problem
  - Users usually provide very few explicit ratings
  - Implicit feedback can provide some insights into user preferences

$$\widehat{r_{ui}} = \mu + b_i + b_u + q_i^T\left(p_u + |R(u)|^{-1/2} \sum_{j \epsilon R(u)} y_j\right)$$

  - $|R(u)|^{-1/2}$ stabilizes variance across the range of observed $|R(u)|$
  - y_j additional item factor reflecting implicit feedback

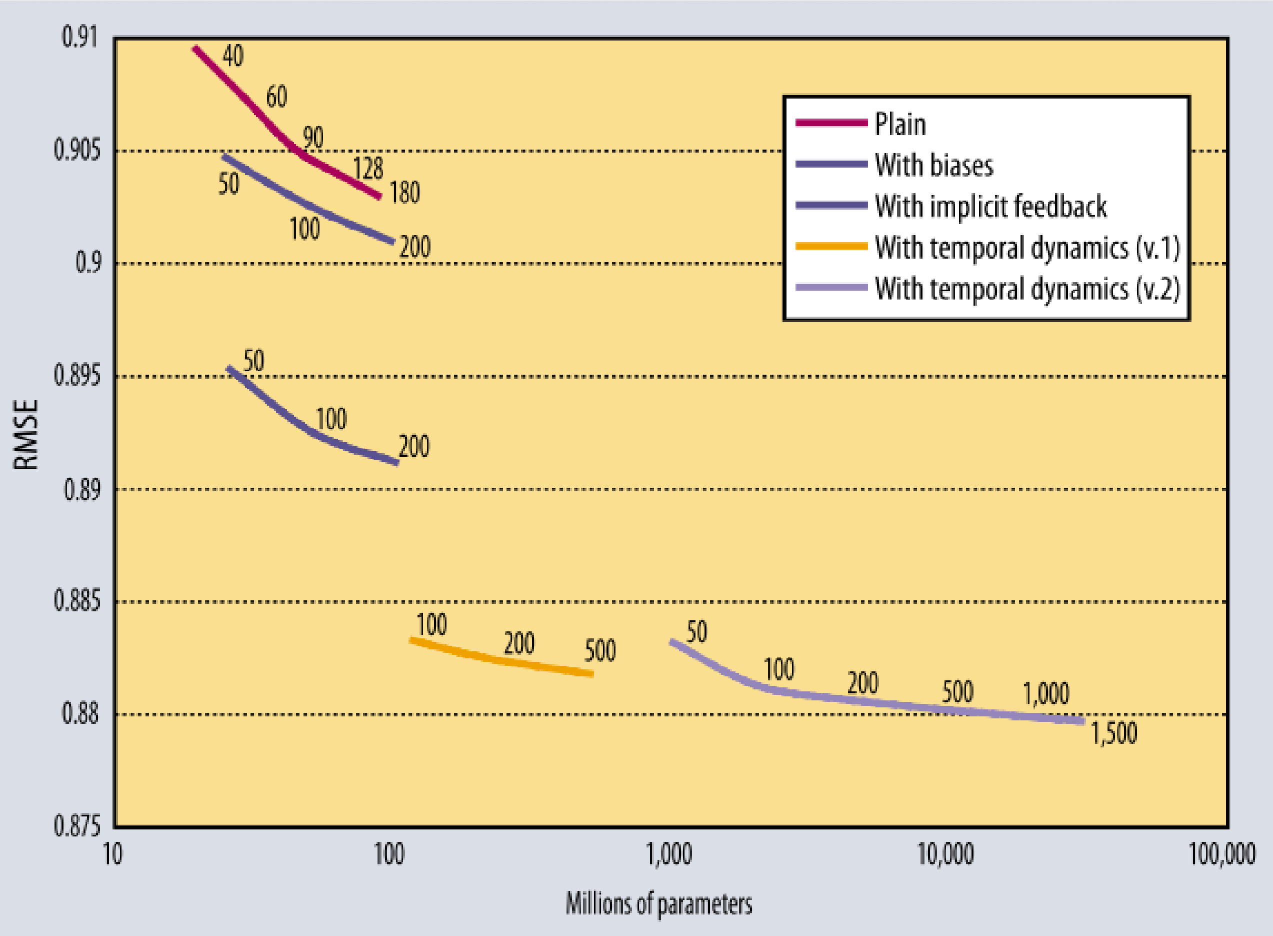
Incorporating temporal dynamics

- Real systems must tackle with temporal shifts
  - Item popularity changes over time
  - User baseline rating changes over time
  - User preferences change over time

$$\widehat{r_{ui}} = \mu + b_i(t_{ui}) + b_u(t_{ui}) + q_i^T \left(p_u(t_{ui}) + |R(u)|^{-1/2} \sum_{j \epsilon R(u)} y_j\right)$$

  - t_{ui} denotes the time of rating r_{ui}

Performance comparison on Netflix Price dataset



Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.

* Advanced models

Advanced models

- Probabilistic matrix factorization (PMF)
- Constrained PMF
  - Mnih, Andriy, and Ruslan R. Salakhutdinov. "Probabilistic matrix factorization." *Advances in neural information processing systems*. 2008.
- Bayesian PMF using Markov Chain Monte Carlo (MCMC)
  - Salakhutdinov, Ruslan, and Andriy Mnih. "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo." *Proceedings of the 25th international conference on Machine learning*. ACM, 2008.
- Local Low-Rank Matrix Approximation (LLORMA)
  - Lee, J., Kim, S., Lebanon, G., & Singer, Y. (2013). Local Low-Rank Matrix Approximation, *28*.

Conclusion

- Matrix factorization = state of the art within collaborative filtering recommenders
- High scalability
  - Can be applied also at large datasets
- Possibility to easily integrate various additional aspects of data
  - User and item biases
  - Implicit feedback
  - Temporal dynamics

Literature and sources

- Aggarwal, Charu C. 2016. Recommender Systems - The Textbook. Communications of the ACM. Vol. 40. Cham: Springer International Publishing.
- Ricci, Francesco, Lior Rokach, Bracha Shapira, and Paul B. Kantor, eds. 2015. Recommender Systems Handbook. 2nd Edition. Boston, MA: Springer US.
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.
- Lee, J., Kim, S., Lebanon, G., & Singer, Y. (2013). Local Low-Rank Matrix Approximation, *28*. Retrieved from http://theanalysisofdata.com/gl/lee_icml_2013.pdf
- Lei Guo. 2012. Matrix Factorization Techniques For Recommender Systems, Retrieved from https://www.slideshare.net/studentalei/matrix-factorization-techniques-for-recommender-systems
- Mnih, Andriy, and Ruslan R. Salakhutdinov. "Probabilistic matrix factorization." *Advances in neural information processing systems*. 2008.
- Salakhutdinov, Ruslan, and Andriy Mnih. "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo." *Proceedings of the 25th international conference on Machine learning*. ACM, 2008.