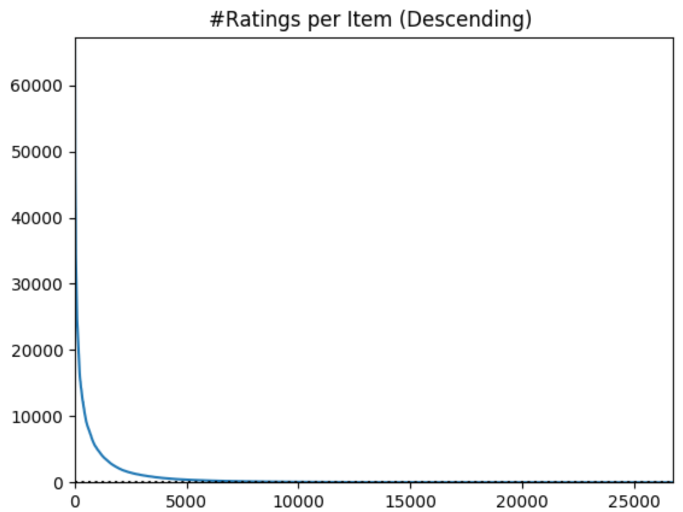
Item Popularity

Prediction

$$\hat{x}_{ui} = \frac{\#\text{interactions with item } i}{\#\text{interactions}}$$

Power-Law Nature of #Interactions



Frequency Distribution of Movielens-20M Ratings

Pros & Cons

- \+ simple & efficient
- \+ no parameters to train
- \+ solves cold-start problem
- \+ works well due to power-law nature of interactions
- − no tailoring of recommendations to users' specific preferences

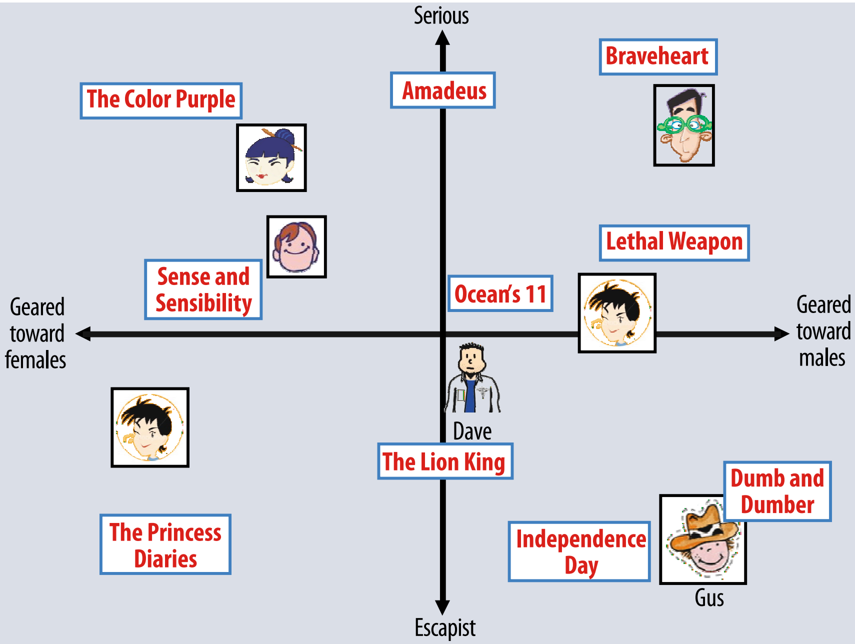
Matrix Factorization (MF)

Prediction

$$\hat{x}_{u,i} = \langle \mathbf{x}_u, \mathbf{x}_i \rangle + \mu_u + \mu_i + \mu$$

Training Objective

Squared Loss or BPR Loss



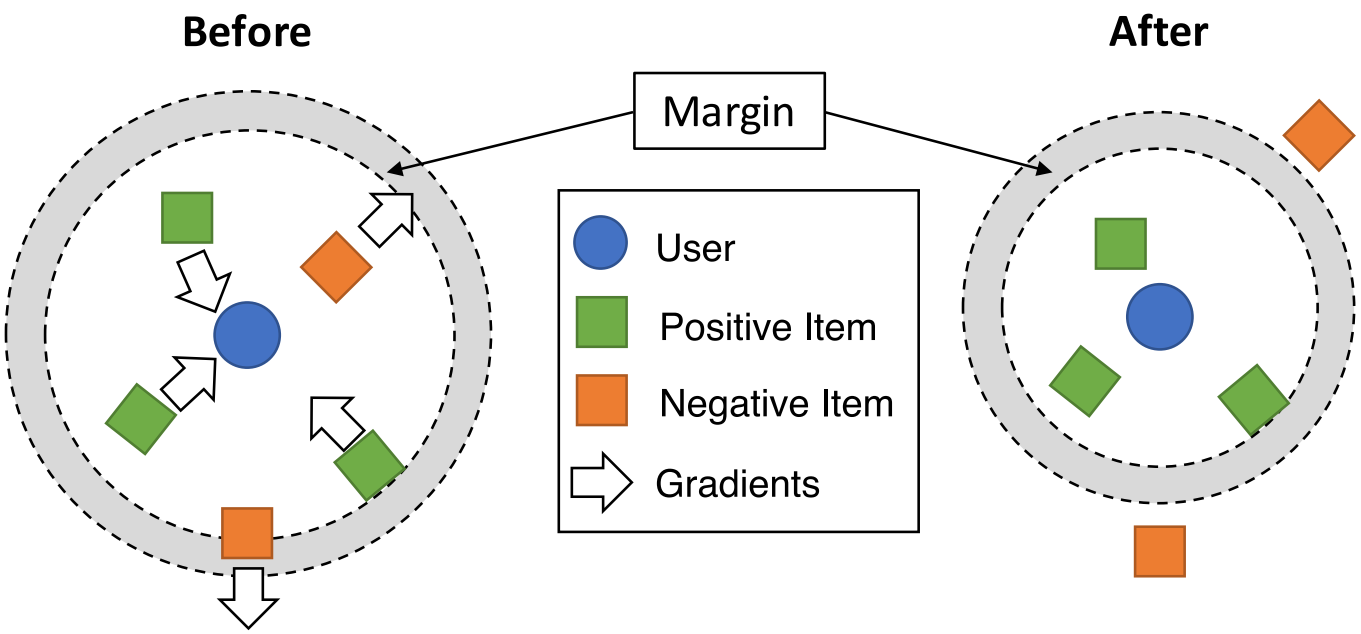
Pros & Cons

- $+$ Fast & efficient predictions tailored to user's tastes
- $+$ Well-established model with lots of extensions (ALS, time incorporation, online algs., Non-neg. MF, ...)
- $+$ Interpretability
- $+$ Usable with MIPS
- $-$ Fails to learn user-user and item-item similarities
- $-$ Prediction is only a bi-linear

Image Source: "Matrix factorization techniques for recommender systems" Y Koren, R Bell, C Volinsky

Collaborative Metric Learning (CML) (1/3)[4]

Embedding of users and items in a *metric space* $(U \uplus I, d)$ with distance metric $d(u, i)$.



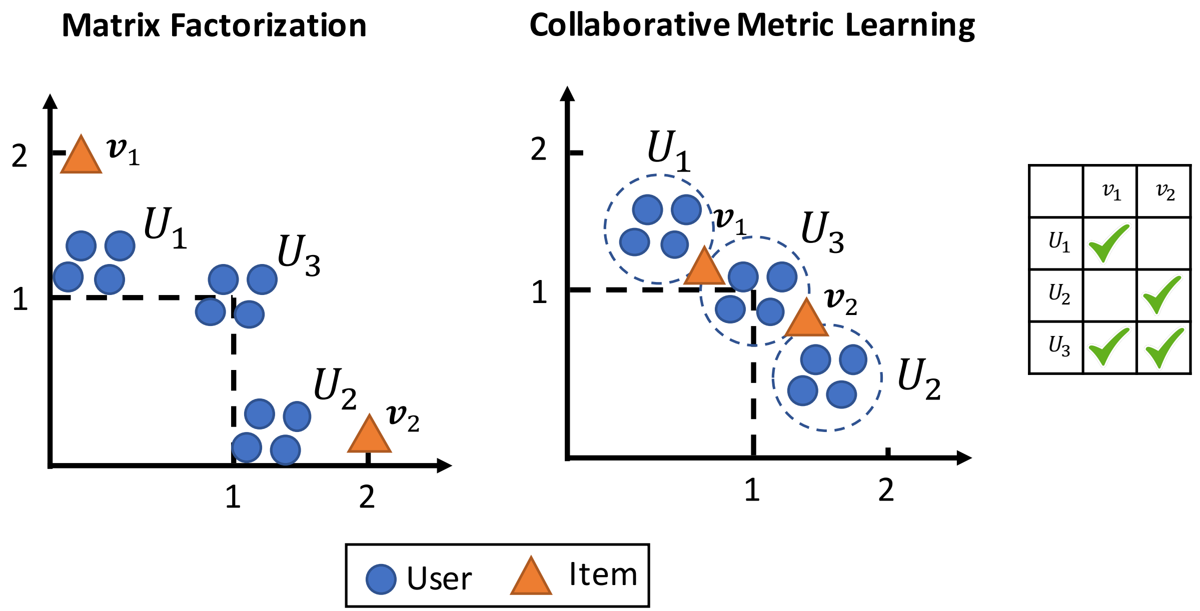
[4] "Collaborative Metric Learning", C K Hsieh, L Yang, Y Cui, T Y Lin, S Belongie, D Estrin

Collaborative Metric Learning (CML) (2/3)

Profits from the phenomenon of *similarity propagation*, because distance metric d respects the *triangle inequality*:

$$\forall u, i, j: \quad d(i, j) \leq d(u, i) + d(u, j)$$

$\rightarrow$ Similarity of (u, i) and (u, j) is propagated to (i, j).



$\rightarrow$ User-user and item-item similarities automatically learned.

Collaborative Metric Learning (CML) (3/3)

Prediction

$$\hat{x}_{ui} = -d(\mathbf{x}_u, \mathbf{x}_i)$$

Training Objective
WARP Loss with Cov. Reg.

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_m(\boldsymbol{\theta}) + \lambda\Omega(\boldsymbol{\theta}) \quad \text{s.t. } \|\mathbf{x}_*\| \leq 1.$$

$$\mathcal{L}_m(\boldsymbol{\theta}) = \sum_{(i,j)\in\mathcal{S}} \sum_{(u,k)\notin\mathcal{S}} w_{ij} \left[m + d(\mathbf{x}_u, \mathbf{x}_i)^2 - d(\mathbf{x}_u, \mathbf{x}_j)^2\right]_+,$$

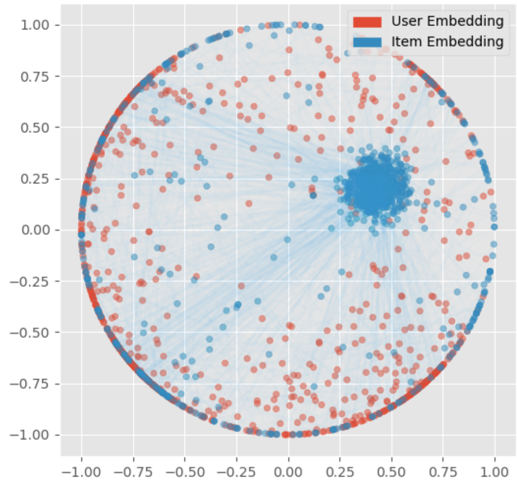
Pros & Cons

+ Benefits from *similarity propagation* → user-user & item-item similarities automatically learned
+ Interpretability
+ LSH possible
– Metric space geometry must suit the latent geometry

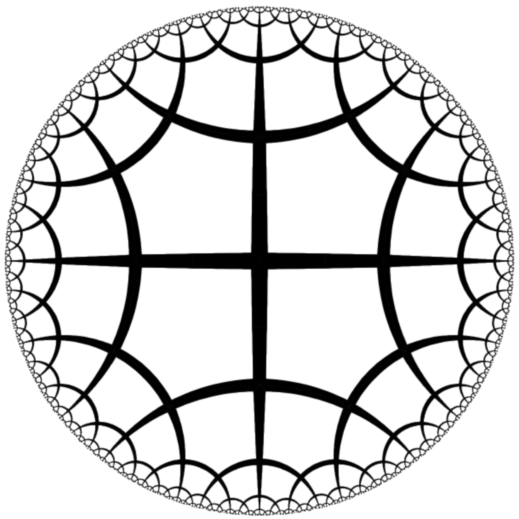
Hyperbolic Recommender Systems (1/2)[6]

Harness *hyperbolic* metric space to represent the relevances.

Amazon Sports Embeddings



Poincaré Ball



Motivation:[5]

Power-law nature of bi-partite interaction graph $\longleftrightarrow$ Complex Networks $\longleftrightarrow$ Hyperbolic Geometry

---

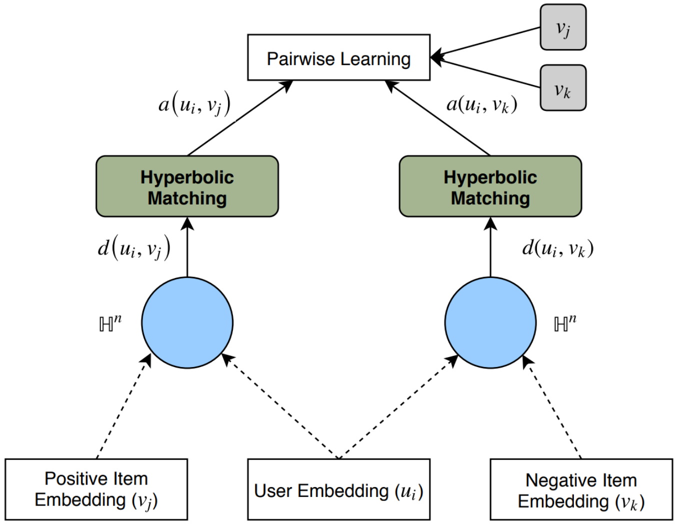
[5] "Hyperbolic geometry of complex networks", D Krioukov, F Papadopoulos, M Kitsak, A Vahdat, M Boguná

[6] "Hyperbolic recommender systems", T D Q Vinh, Y Tay, S Zhang, G Cong, X L Li
"Scalable Hyperbolic Recommender Systems" B. P. Chamberlain, S. R. Hardwick, D. R. Wardrope, F. Dzogang, F. Daolio, S. Vargas

Hyperbolic Recommender Systems (2/2)

Prediction

$$\hat{x}_{ui} = -\alpha d(\mathbf{x}_u, \mathbf{x}_i), \quad \alpha > 0.$$



Pros & Cons

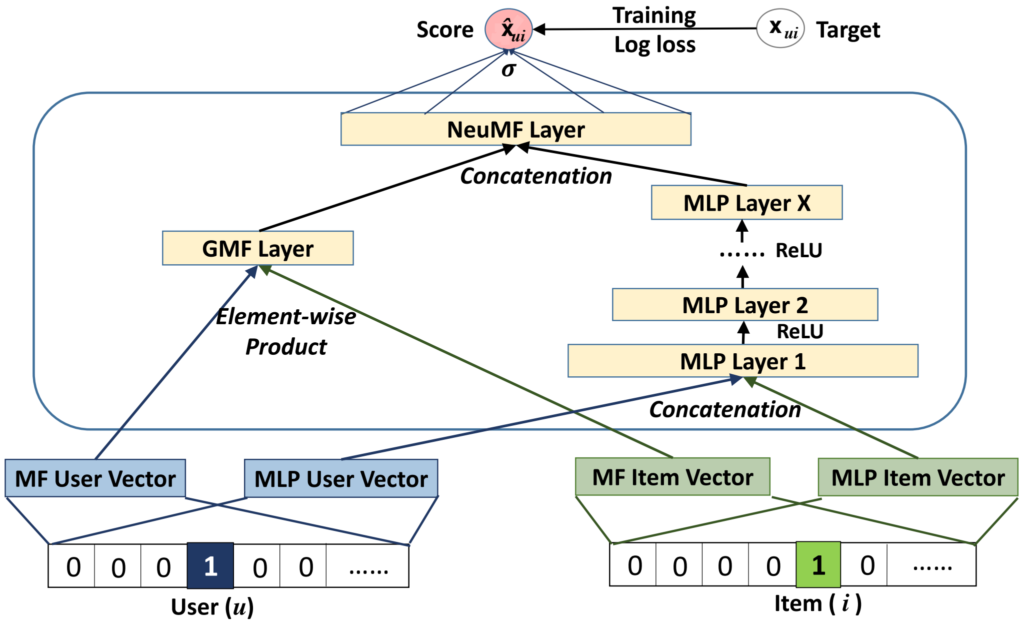
+ *Similarity propagation*
+ Interpretability
+ LSH possible
− Prediction power depends on suitability of geometry

Training Objective (BPR)

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{(u,i,j)\in\mathcal{D}} -\log\left(\sigma\left(\alpha\left(d(\mathbf{x}_u, \mathbf{x}_j) - d(\mathbf{x}_u, \mathbf{x}_i)\right)\right)\right),$$

Neural Collaborative Filtering (NCF)[3]

Prediction



Pros & Cons

+ Non-linear prediction
+ Robust joint-model of simple GMF and MLP
– Lacks interpretability
– Expensive prediction
– Cannot be used with LSH or MIPS

Training Objective

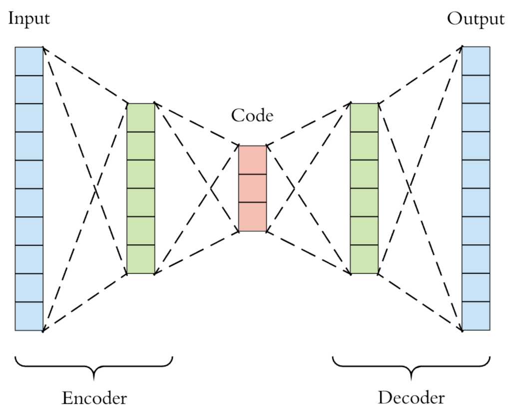
Binary Cross-Entropy with neg. sampling

$$\mathcal{L}(\boldsymbol{\theta}) = -\sum_{(u,i)\in\mathcal{D}} \left[x_{ui}\log(\hat{x}_{ui}) + (1 - x_{ui})\log(1 - \hat{x}_{ui})\right]$$

[3] "Neural collaborative filtering", X He, L Liao, H Zhang, L Nie, X Hu, T S Chua,

Autoencoders for Collaborative Filtering[2]

Prediction $\hat{x}_{ui} = h(\mathbf{x}_u, \boldsymbol{\theta})$



Training Objective
Reconstruction/Prediction
Error or ELBO

Pros & Cons

+ Non-Linear ranking
+ No negative sampling needed
− Lack of interpretability
− No automatic discovery of user-user item-item similarities
− Predictions over *all* users or items $\Theta(\min |U|, |I|)$
− LSH or MIPS not possible

Major Challenge: Sparsity of inputs/gradients

- Item-based RS
- Dense re-feeding: $h(h(\mathbf{x})) = h(\mathbf{x})$

[2] "Autorec: Autoencoders meet collaborative filtering." S Sedhain, A K M, S Sanner, L Xie
"Collaborative filtering with stacked denoising autoencoders and sparse inputs.", F Strub, M Jeremie
"Training deep autoencoders for collaborative filtering", O Kuchaiev, B Ginsburg
"Variational autoencoders for collaborative filtering", D Liang, R G Krishnan, M D Hoffman, T Jebara