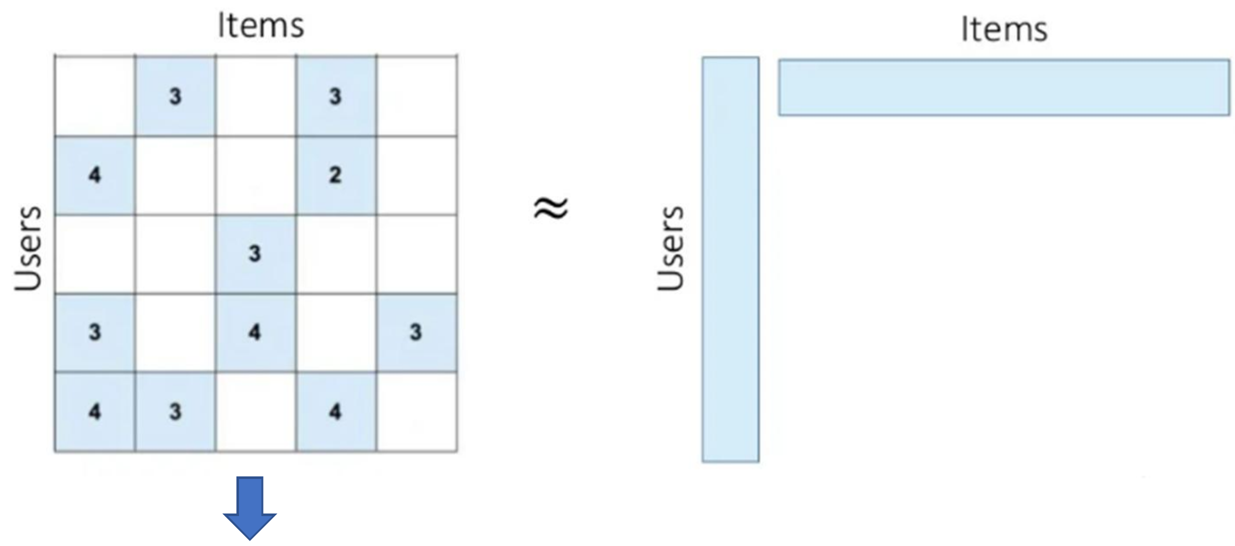
Collaborative Metric Learning

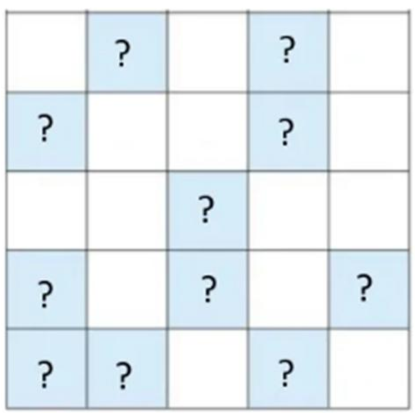
By Hsieh, Cheng-Kang, et al.

Presented by Kim Han

Contents

Introduction





- Only positive feedback is available

- Just estimating ratings
  → relationships between user-user or item-item pairs

Background

Metric Learning

$$d_E(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\sum_{i=1}^{d} (\mathbf{x}_{1,i} - \mathbf{x}_{2,i})^2}$$

Euclidean distance

$$d_M(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T S^{-1} (\mathbf{x}_1 - \mathbf{x}_2)} \Rightarrow d_A(x_i, x_j) = \sqrt{(x_i - x_j)^T A(x_i, -x_j)}$$

Mahalanobis distance

$$\mathcal{S} = \{(x_i, x_j) | x_i \text{ and } x_j \text{are considered similar}\}$$

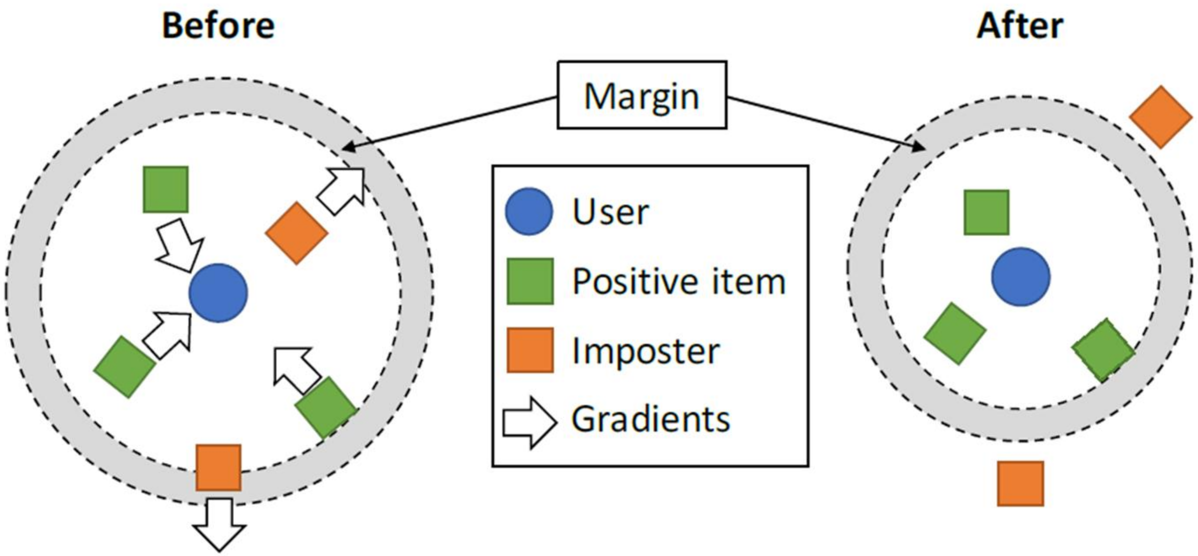
$$\mathcal{D} = \{(x_i, x_j) | x_i \text{ and } x_j \text{are considered dissimilar}\}$$

$$\min_A \sum_{(x_i, x_j) \in \mathcal{S}} d_A(x_i, x_j)^2$$

$$\text{s.t. } \sum_{(x_i, x_j) \in \mathcal{D}} d_A(x_i, x_j)^2 \geq 1 \text{ and } A \succeq 0.$$

Background

Metric Learning (Large Margin Nearest Neighbor)



$$\mathcal{L}_{pull}(d) = \sum_{j \rightsquigarrow i} d(x_i, x_j)^2$$

$$\mathcal{L}_{push}(d) = \sum_{i, j \rightsquigarrow i} \sum_{k} (1 - y_{ik})[1 + d(x_i, x_j)^2 - d(x_i, x_k)^2]_+$$

Model

Collaborative Metric Learning

1. $D(\vec{x}_i, \vec{x}_j) + D(\vec{x}_j, \vec{x}_k) \geq D(\vec{x}_i, \vec{x}_k)$ (triangular inequality).
2. $D(\vec{x}_i, \vec{x}_j) \geq 0$ (non-negativity).
3. $D(\vec{x}_i, \vec{x}_j) = D(\vec{x}_j, \vec{x}_i)$ (symmetry).
4. $D(\vec{x}_i, \vec{x}_j) = 0 \iff \vec{x}_i = \vec{x}_j$ (distinguishability).

- the items liked by this user previously
- the items liked by other users who share a similar taste with this user previously

Model

Collaborative Metric Learning

$u_i \in \mathcal{R}^r$: User vector

$v_j \in \mathcal{R}^r$: Item vector

$d(i,j) = \|\mathbf{u}_i - \mathbf{v}_j\|$: Distance (with euclidean distance)

$$w_{ij} = log(rank_d(i,j) + 1)$$

Approximated Ranking Weight

$$\lfloor \frac{J}{N} \rfloor \longrightarrow \lfloor \frac{J \times M}{U} \rfloor$$

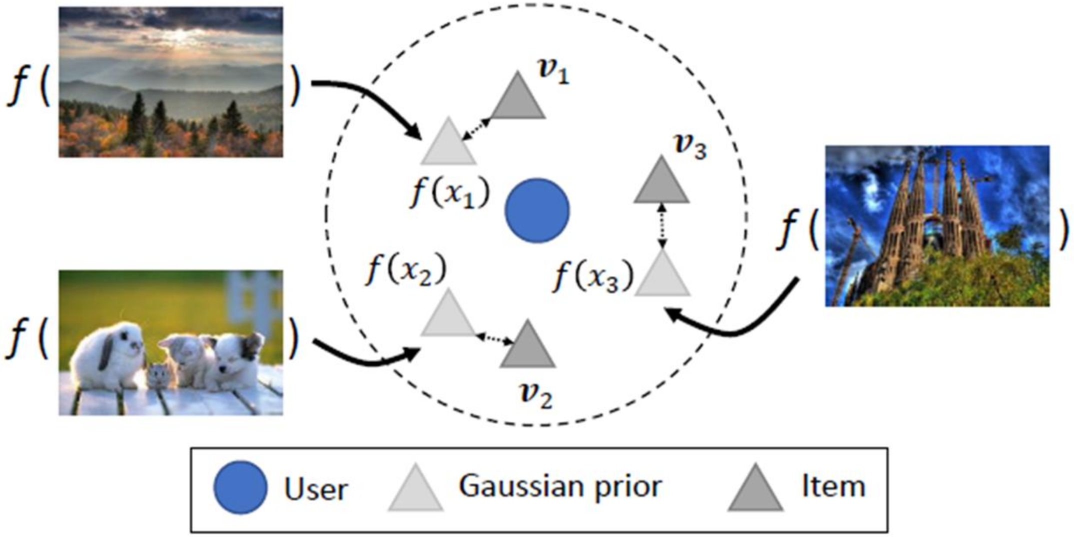
Hinge Loss

$$\mathcal{L}_m(d) = \sum_{(i,j)\in\mathcal{S}} \sum_{(i,k)\notin\mathcal{S}} w_{ij}[m + d(i,j)^2 - d(i,k)^2]_+, \quad (1)$$

Model

Integrating Item Features

$$\mathcal{L}_f(\theta, \mathbf{v}_*) = \sum_j \|f(\mathbf{x_j}, \theta) - \mathbf{v}_j\|^2$$



Model

Regularization

$$\|\mathbf{u}_*\|^2 \leq 1 \text{ and } \|\mathbf{v}_*\|^2 \leq 1$$

Normalize

$$C_{ij} = \frac{1}{N}\sum_n (y_i^n - \mu_i)(y_j^n - \mu_j)$$

$$\mu_i = \frac{1}{N}\sum_n y_i^n$$

$$\|C\|_f^2 = |a_{11}|^2 + |a_{12}|^2 + \cdots + |a_{mm}|^2$$

$$\|diag(C)\|_2^2 = |a_{11}|^2 + |a_{22}|^2 + |a_{33}|^2 + \cdots + |a_{mm}|^2$$

$$\mathcal{L}_c = \frac{1}{N}(\|C\|_f - \|diag(C)\|_2^2)$$

Model

Training

$$\min_{\theta, \mathbf{u}_*, \mathbf{v}_*} \mathcal{L}_m + \lambda_f \mathcal{L}_f + \lambda_c \mathcal{L}_c$$

$$\text{s.t.} \quad \|\mathbf{u}_*\|^2 \leq 1 \text{ and } \|\mathbf{v}_*\|^2 \leq 1$$

1. Sample N positive pairs from S.

2. For each pair, sample U negative items and approximate rank_d(i, j).

3. For each pair, keep the negative item k that maximizes the hinge loss and form a mini-batch of size N.

4. Compute gradients and update parameters with a learning rate controlled by AdaGrad.

5. Censor the norm of u* and v* by $\mathbf{y}' = \frac{\mathbf{y}}{\max(\|\mathbf{y}\|, 1)}$

6. Repeat this procedure until convergence.

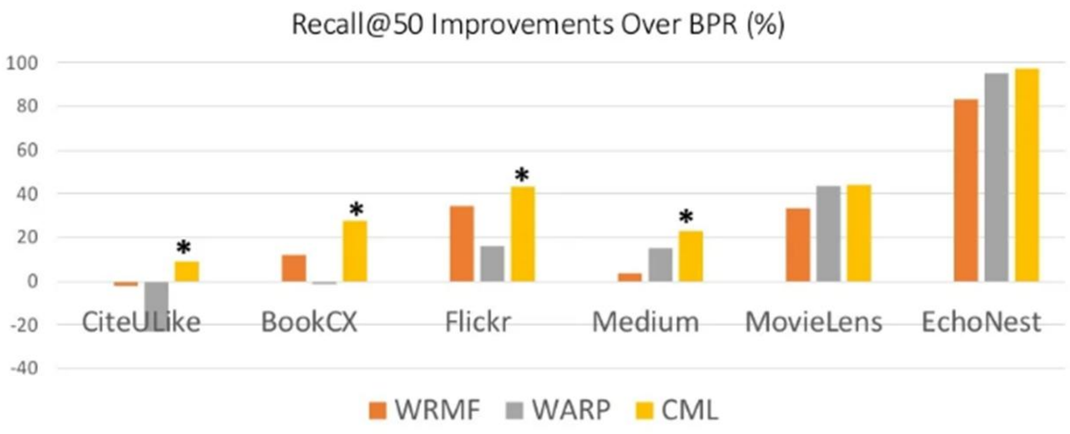
Experiment

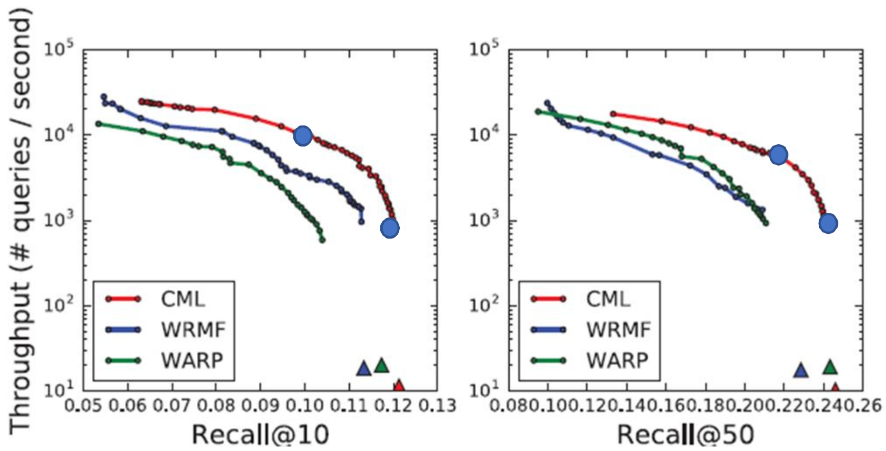
Table 1: Dataset Statistics.

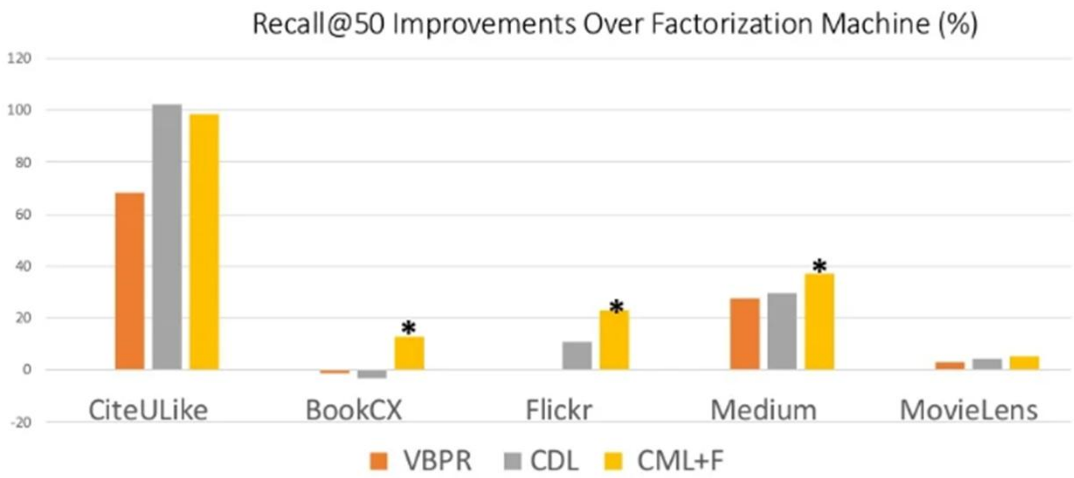
	CiteULike	BookCX	Flickr	Medium	MovieLens20M	EchoNest
Domain	Paper	Book	Photography	News	Movie	Song
# Users	7,947	22,816	43,758	61,909	129,797	766,882
# Items	25,975	43,765	100,000	80,234	20,709	260,417
# Ratings	142,794	623,405	1,372,621	2,047,908	9,939,873	7,261,443
Concentration[a]	33.47%	33.10%	13.48%	55.38%	72.52%	65.88%
Features Type	Tags	Subjects	Image Features	Tags	Genres, Keywords	NA
# Feature Dim.	10,399	7,923	2,048	2,313	10,399	NA

	WRMF	**BPR**	**WARP**	**CML**	*ours vs. best*	**FM**	**VBPR**	**CDL**	**CML+F**	*ours vs. best*
					Recall@50					
CiteULike	0.2437	0.2489	0.1916	**0.2714*****	9.03%	0.1668	0.2807	**0.3375****	0.3312	-1.86%
BookCX	0.0910	0.0812	0.0801	**0.1037*****	13.95%	0.1016	0.1004	0.0984	**0.1147*****	12.89%
Flickr	0.0667	0.0496	0.0576	**0.0711*****	6.59%	NA	0.0612	0.0679	**0.0753*****	10.89%
Medium	0.1457	0.1407	0.1619	**0.1730*****	6.41%	0.1298	0.1656	0.1682	**0.1780*****	5.82%
MovieLens	0.4317	0.3236	0.4649	**0.4665**	0.34%	0.4384	0.4521	0.4573	**0.4617***	0.96%
EchoNest	0.2285	0.1246	0.2433	**0.2460**	1.10%	NA	NA	NA	NA	NA
					Recall@100					
CiteULike	0.3112	0.3296	0.2526	**0.3411*****	3.37%	0.2166	0.3437	0.4173	**0.4255****	1.96%
BookCX	0.1286	0.1230	0.1227	**0.1436*****	11.66%	0.1440	0.1455	0.1428	**0.1712*****	17.66%
Flickr	0.0821	0.0790	0.0797	**0.0922*****	12.30%	NA	0.0880	0.0909	**0.1048*****	15.29%
Medium	0.2112	0.2078	0.2336	**0.2480*****	6.16%	0.1900	0.2349	0.2408	**0.2531*****	5.10%
MovieLens	0.5649	0.4455	0.5989	**0.6022**	0.55%	0.5561	0.5712	0.5943	**0.5976**	0.55%
EchoNest	0.2891	0.1655	0.3021	**0.3022**	0.00%	NA	NA	NA	NA	NA

Experiment



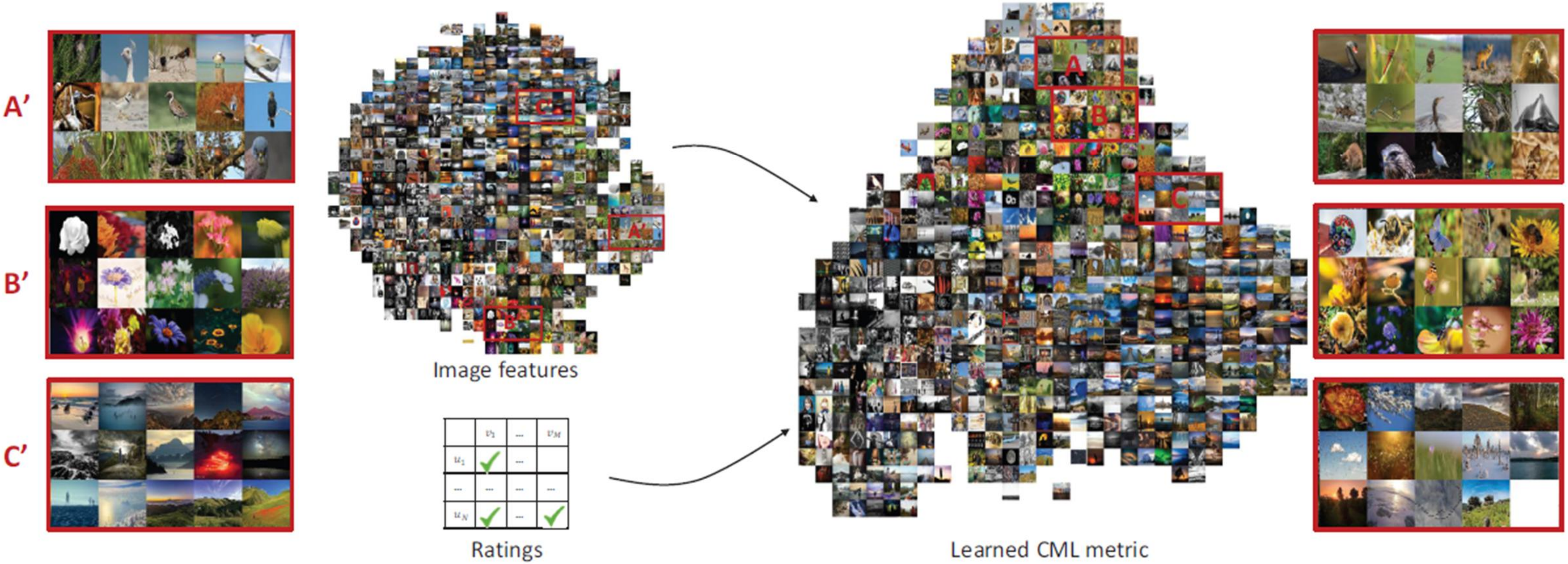




1. CML's throughput is improved by 106x with only 2% reduction in accuracy.

2. Over 8x, 9x faster than MF models given the same accuracy

Experiment



Conclusion

1. Implementing collaborative filtering in another way that is more efficient than the existing method

2. Shown to be very efficient and fast in the Top-K recommendation method

3. Not only user-item, but also user-user, item-item relationships were interpreted to make the entire system easier to understand

감사합니다.