Markov Random Fields for Collaborative Filtering

Harald Steck (hsteck@netflix.com)

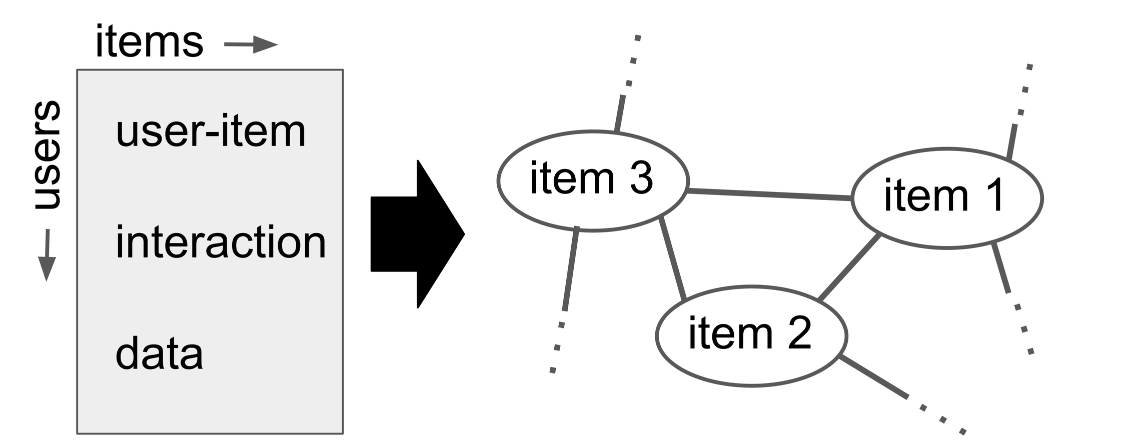
NETFLIX

Summary

- simple linear model
- competitive experimental results on three popular data-sets, compared to various baselines, including deep nonlinear models:
  - training-time reduced by factor 10 (dense model) to 400 (sparse model).
  - recommendation-accuracy increased by 20% on data-set with the largest number of items.
- tradeoff between accuracy and training time can be controlled by two hyper-parameters.

Model

Collaborative Filtering:



Markov Random Field (MRF): models the **similarities (dependencies) among items**.

- items ⥦ nodes in MRF ⥦ random variables $X = (X_1, ..., X_m)$
- users ⥦ samples drawn from $p(X)$

Besag's Approach (1975)

- instead of Hammersley-Clifford theorem
- for **computational efficiency**
- yields asymptotically consistent estimates
- Gaussian distribution $\mathcal{N}(0, \Sigma)$

1. auto-normal parameterization:

regress each item against its neighbors:

- conditional means:
$$\mathbb{E}[X_i | X_{\mathcal{I}\setminus\{i\}} = x_{\mathcal{I}\setminus\{i\}}] = \sum_{j\in\mathcal{I}\setminus\{i\}} \beta_{j,i} x_j = x \cdot \mathbf{B}_{\cdot,i}$$
- conditional variances: $\vec{\sigma}^2 := (\sigma_1^2, ..., \sigma_m^2)$
- symmetry of covariance matrix Σ imposes constraint: $\sigma_i^2 \beta_{i,j} = \sigma_j^2 \beta_{j,i}$

2. log pseudo-likelihood:

$L(\mathbf{X}|\mathbf{B}, \vec{\sigma}^2) = \sum_{i\in\mathcal{I}} L(\mathbf{X}_{\cdot,i} | \mathbf{X}_{\cdot,\mathcal{I}\setminus\{i\}}; \mathbf{B}_{\cdot,i}, \sigma_i^2)$

After dropping symmetry-constraint, likelihood decouples:

$$L(\mathbf{X}|\mathbf{B}) = -\sum_{i\in\mathcal{I}} ||\mathbf{X}_{\cdot,i} - \mathbf{X}\mathbf{B}_{\cdot,i}||_2^2 = -||\mathbf{X} - \mathbf{X}\mathbf{B}||_F^2$$

Paper & Code

Dense Model

Least-squares with L2-norm regularization:

$$\hat{\mathbf{B}} = \arg\min_B ||\mathbf{X} - \mathbf{X}\mathbf{B}||_F^2 + \lambda \cdot ||\mathbf{B}||_F^2 \quad \text{where} \quad \text{diag}(\mathbf{B}) = 0$$

Closed-form solution:

$\hat{\mathbf{B}}_{i,j} = -\frac{\hat{\mathbf{C}}_{i,j}}{\hat{\mathbf{C}}_{j,j}}$ for $i \neq j$

where: concentration matrix $\hat{\mathbf{C}} = \mathbf{S}_\lambda^{-1}$
$\mathbf{S}_\lambda = n^{-1}(\mathbf{X}^\top\mathbf{X} + \lambda \cdot \mathbf{I})$
$\mathbf{X}$ ... user-item interaction matrix

Derivation using vector of Lagrangian multipliers γ:

$$\begin{aligned}\hat{\mathbf{B}} &= (\mathbf{X}^\top\mathbf{X} + \lambda\cdot\mathbf{I})^{-1}(\mathbf{X}^\top\mathbf{X} - \text{dMat}(\gamma)) \\ &= n^{-1}\hat{\mathbf{C}}(n\hat{\mathbf{C}}^{-1} - \lambda\cdot\mathbf{I} - \text{dMat}(\gamma)) \\ &= \mathbf{I} - n^{-1}\hat{\mathbf{C}}\cdot\text{dMat}(\gamma + \lambda)\end{aligned}$$

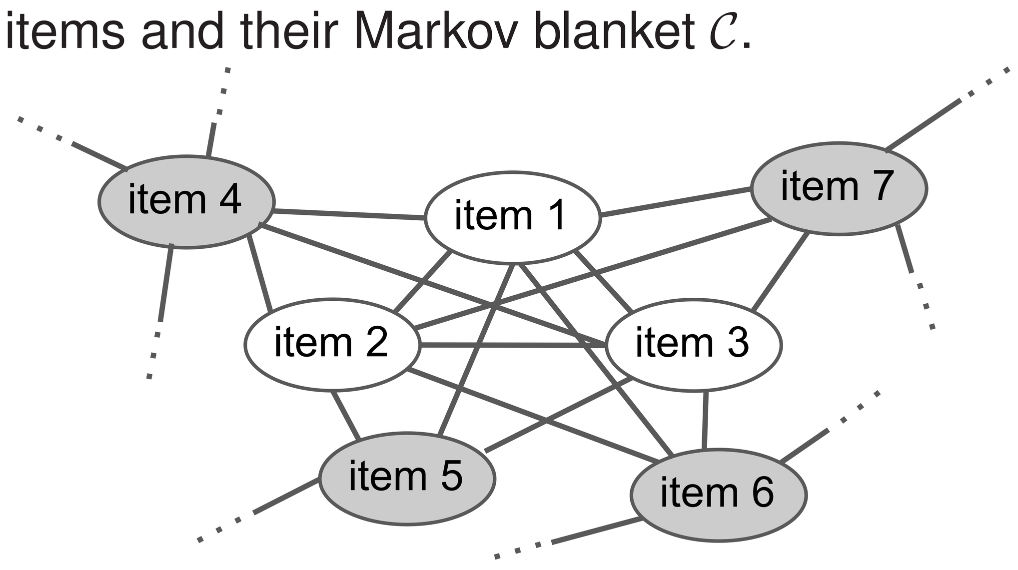
hence $\gamma + \lambda = n \oslash \text{diag}(\hat{\mathbf{C}})$

Sparse Approximation

Goal: reduced training-time (rather than increased accuracy)

High-level Idea:

invert small *submatrices* of the cov.-matrix Σ, each pertaining to a *subset* $\mathcal{D}$ of (completely) connected items and their Markov blanket $\mathcal{C}$.



Analogous to Besag's approach, this also yields asymptotically consistent estimates (cf. paper).

Approximation:

- determine sparsity pattern (MRF graph) by thresholding the *correlation* matrix.
- iterate through the items i in descending order of their number of neighbors $\mathcal{N}^{(i)}$ (skip i if $i \in \mathcal{D}^{(j)}$ of an earlier j):
  - compute closed-form solution for submatrix involving items $\{i\} \cup \mathcal{N}^{(i)}$. → exact solution for item i.
  - subset $\mathcal{D}^{(i)}$ contains $m^{(i)} := \text{round}(r \cdot |\mathcal{N}^{(i)}|)$ items with highest correlation to item i.
  - subset $\mathcal{C}^{(i)} := \mathcal{N}^{(i)} \setminus \mathcal{D}^{(i)}$.
  - we assume that $\mathcal{C}^{(i)}$ is the Markov blanket of subset $\mathcal{D}^{(i)}$ (and completely connected). Generally, this is an **approximation**, especially as parameter-value r is increased.
  - above closed-form solution provides **add'l** approximate estimates for all the items $j \in \mathcal{D}^{(i)}$. → Free Lunch!

Parameter r controls the sizes of the subsets, and hence the trade-off between accuracy and training-time.

Note: *correlation* matrix determines sparsity pattern, while *covariance* matrix determines non-zero values

Results of Dense Model

	Data Sets		
	ML-20M	*Netflix*	*MSD*
Models	**Recall@50**	(see paper for more)	
$\hat{\mathbf{B}}^{(dense)}$	0.522	**0.448**	**0.430**
reproduced from [Liang et al., 2018]:			
MULT-VAE	**0.537**	0.444	0.364
MULT-DAE	0.524	0.438	0.363
CDAE	0.523	0.428	0.283
SLIM	0.495	0.428	–dnf–
WMF	0.498	0.404	0.312
std. errors	0.002	0.001	0.001
Training Times			
$\hat{\mathbf{B}}^{(dense)}$	2m 0s	1m 30s	15m 45s
MULT-VAE	28m 10s	1h 26m	4h 30m
Data-Set Properties			
users	136,677	463,435	571,355
items	20,108	17,769	41,140
interactions	10 mil.	57 mil.	34 mil.

- 10× faster training (vs. MULT-VAE).
- about 20% increase in accuracy on the data-set with the largest number of items.

→ **preventing self-similarity of items (zero diagonal in B) is an effective alternative to low-rank embeddings.**

Results of Sparse Model

Models	**Recall@50**	**Training Times**
$\hat{\mathbf{B}}^{(dense)}$	0.430	15 min 45 sec
0.5% sparse approximation:		
$r = 0$	0.427	21 min 12 sec
$r = 0.1$	0.424	3 min 27 sec
$r = 0.5$	0.424	2 min 1 sec
0.1% sparse approximation:		
$r = 0$	0.421	3 min 7 sec
$r = 0.1$	0.417	1 min 10 sec
$r = 0.5$	0.417	39 sec
MULT-VAE	0.364	4 h 30 min

- sparsity and parameter r determine trade-off between accuracy and training time
- up to 400× faster training (vs. MULT-VAE)
- only slight drop in accuracy
- sparse MRF has much fewer parameters than MULT-VAE

→ **sparse full-rank modeling is more effective than dense low-rank modeling.**

Related Approaches