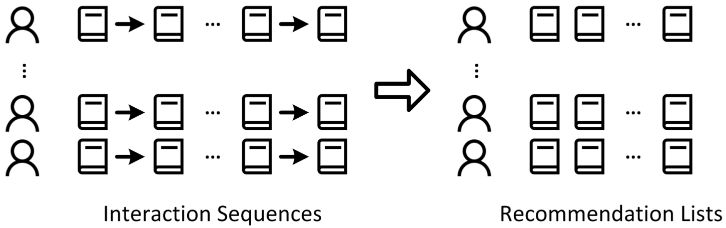
Translation-based Recommendation

Zhuoxin Zhan (revised by Liulan Zhong and Weike Pan)

College of Computer Science and Software Engineering
Shenzhen University

Reference: Translation-based Recommendation (RecSys 2017)
by Ruining He, Wang-Cheng Kang and Julian McAuley

Problem Definition



Next-Item Recommendation

- Input: $(u, \mathcal{S}_u)$, i.e., a sequence of items for each user u.
- Goal: Rank the un-interacted items in $\mathcal{I} \backslash \mathcal{S}_u$ and use the top-k items with the highest preference values to construct a recommendation list for each user u.

Notations (1/2)

Table: Some notations and explanations.

n	number of users
m	number of items
$\mathcal{U}, \lvert\mathcal{U}\rvert = n$	the whole set of users
$\mathcal{I}, \lvert\mathcal{I}\rvert = m$	the whole set of items
$u \in \{1, 2, \ldots, n\}$	user ID
$j \in \{1, 2, \ldots, m\}$	item ID
$\mathcal{S}_u = \{i_u^1, i_u^2, \ldots, i_u^{\lvert\mathcal{S}_u\rvert}\}$	a sequence of items
i_u^t	the tth item in $\mathcal{S}_u$
$\mathcal{P} = \{(u, i_u^t), i_u^t \in \mathcal{S}_u, u \in \mathcal{U}\}$	the whole set of observed (u, i_u^t) pairs

Notations (2/2)

Table: Some notations and explanations (cont.).

$V_{i_u^t\cdot} \in \mathbb{R}^{1\times d}$	the embedding vector of item i_u^t
$\mathbf{t} \in \mathbb{R}^{1\times d}$	the global translation vector
$U_{u\cdot} \in \mathbb{R}^{1\times d}$	the translation vector of user u
$b_i \in \mathbb{R}$	the bias of item i
Θ	the set of model parameters
$d(\boldsymbol{x}, \boldsymbol{y})$	distance between $\boldsymbol{x} \in \mathbb{R}^{1\times d}$ and $\boldsymbol{y} \in \mathbb{R}^{1\times d}$
$\lvert\lvert \boldsymbol{x} \rvert\rvert = \sqrt{\boldsymbol{x}^T \boldsymbol{x}}$	L_2 norm (i.e., length of $\boldsymbol{x}$)
γ	learning rate
$\beta_b, \alpha_v, \alpha_u, \alpha_t$	the regularization parameter
T	iteration number in the algorithm

Assumption

If a user u transitions from item i_u^t to item i_u^{t+1}, we assume,

$$V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t} \approx V_{i_u^{t+1}\cdot}, \quad (1)$$

which means that $V_{i_u^{t+1}\cdot}$ is a nearest neighbor of $V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t}$ according to some distance metric $d(\boldsymbol{x}, \boldsymbol{y})$, e.g., L_1 or L_2 distance.

Note that we use the L_2 distance for ease of gradient calculation, i.e., $d(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x} - \boldsymbol{y}||$, and $\frac{\partial d(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{x}} = ||\boldsymbol{x} - \boldsymbol{y}||^{-1}(\boldsymbol{x} - \boldsymbol{y})$.

Prediction Rule (1/2)

In the training phase, the probability that a user u transitions from item i_u^t to its next item i_u^{t+1},

$$\hat{p}_{u,i_u^t,i_u^{t+1}} = b_{i_u^{t+1}} - d(V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t}, V_{i_u^{t+1}\cdot}) \quad (2)$$

where $V_{i_u^t\cdot}$ and $V_{i_u^{t+1}\cdot}$ are in a unit ball, i.e., $||V_{i_u^t\cdot}|| \leq 1$ and $||V_{i_u^{t+1}\cdot}|| \leq 1$, and $d(V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t}, V_{i_u^{t+1}\cdot}) = ||V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t} - V_{i_u^{t+1}\cdot}||$.

Prediction Rule (2/2)

In the training phase, the probability that a user u transitions from item i_u^t to an item $j \in \mathcal{I} \backslash \mathcal{S}_u$,

$$\hat{p}_{u,i_u^t,j} = b_j - d(V_{i_u^t \cdot} + U_{u\cdot} + \mathbf{t}, V_{j\cdot}). \tag{3}$$

In the test phase, for an item $j \in \mathcal{I} \backslash \mathcal{S}_u$, we have the probability,

$$\hat{p}_{u,i_u^{|\mathcal{S}_u|},j} = b_j - d(V_{i_u^{|\mathcal{S}_u|} \cdot} + U_{u\cdot} + \mathbf{t}, V_{j\cdot}), \tag{4}$$

which can be rewritten in a slightly different form (see the page on “Nearest Neighbor Search”).

Objective Function

For a tuple $(u, i_u^t, i_u^{t+1}, j), j \in \mathcal{I} \backslash \mathcal{S}_u$, we have the following tentative objective function to be maximized,

$$\ln \sigma(\hat{p}_{u,i_u^t,i_u^{t+1}} - \hat{p}_{u,i_u^t,j}) - \mathcal{R}(\Theta), \tag{5}$$

where
$\mathcal{R}(\Theta) = \frac{\beta_b}{2} b_i^2 + \frac{\alpha_v}{2} ||V_{i_u^t \cdot}||^2 + \frac{\alpha_v}{2} ||V_{i_u^{t+1} \cdot}||^2 + \frac{\alpha_v}{2} ||V_{j \cdot}||^2 + \frac{\alpha_u}{2} ||U_{u \cdot}||^2 + \frac{\alpha_t}{2} ||\mathbf{t}||^2$
is a regularization term used to avoid overfitting.

Gradients (1/2)

For a tuple $(u, i_u^t, i_u^{t+1}, j), j \in \mathcal{I} \backslash \mathcal{S}_u$, we have the gradient of each parameter w.r.t. the tentative objective function,

$$
\begin{aligned}
\nabla b_{i_u^{t+1}} &= \sigma(-e_{u,i_u^t,i_u^{t+1},j}) - \beta_b b_{i_u^{t+1}}, && (6)\\
\nabla b_j &= -\sigma(-e_{u,i_u^t,i_u^{t+1},j}) - \beta_b b_j, && (7)\\
\nabla V_{i_u^t \cdot} &= \sigma(-e_{u,i_u^t,i_u^{t+1},j})[||V_{i_u^t \cdot} + U_{u\cdot} + \mathbf{t} - V_{i_u^{t+1} \cdot}||^{-1}(V_{i_u^t \cdot} + U_{u\cdot} + \mathbf{t} - V_{i_u^{t+1} \cdot}) \\
&\quad -||V_{i_u^t \cdot} + U_{u\cdot} + \mathbf{t} - V_{j\cdot}||^{-1}(V_{i_u^t \cdot} + U_{u\cdot} + \mathbf{t} - V_{j\cdot})] - \alpha_v V_{i_u^t \cdot}, && (8)
\end{aligned}
$$

Gradients (2/2)

$$
\begin{aligned}
\nabla V_{i_u^{t+1}\cdot} &= \sigma(-e_{u,i_u^t,i_u^{t+1},j})||V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t} - V_{i_u^{t+1}\cdot}||^{-1} \\
&\quad (V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t} - V_{i_u^{t+1}})(-1) - \alpha_v V_{i_u^{t+1}\cdot}, && (9)\\
\nabla V_{j\cdot} &= \sigma(-e_{u,i_u^t,i_u^{t+1},j})(-1)||V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t} - V_{j\cdot}||^{-1} \\
&\quad (V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t} - V_{j\cdot})(-1) - \alpha_v V_{j\cdot}, && (10)\\
\nabla U_{u\cdot} &= \sigma(-e_{u,i_u^t,i_u^{t+1},j})[||V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t} - V_{i_u^{t+1}\cdot}||^{-1}(V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t} - V_{i_u^{t+1}\cdot}) \\
&- ||V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t} - V_{j\cdot}||^{-1}(V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t} - V_{j\cdot})] - \alpha_u U_{u\cdot}, && (11)\\
\nabla \mathbf{t} &= \sigma(-e_{u,i_u^t,i_u^{t+1},j})[||V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t} - V_{i_u^{t+1}\cdot}||^{-1}(V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t} - V_{i_u^{t+1}\cdot}) \\
&- ||V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t} - V_{j\cdot}||^{-1}(V_{i_u^t\cdot} + U_{u\cdot} + \mathbf{t} - V_{j\cdot})] - \alpha_t \mathbf{t}, && (12)
\end{aligned}
$$

where $e_{i_u^{t+1}j} = \hat{p}_{u,i_u^t,i_u^{t+1}} - \hat{p}_{u,i_u^t,j}$.

Update Rule

We have the update rule in the stochastic gradient ascent algorithm for each parameter $\theta \in \Theta$,

$$\theta = \theta + \gamma \nabla \theta, \tag{13}$$

where $\gamma > 0$ is the learning rate.

Initialization and Normalization

- **Initialization.** $V_{i_u^t \cdot}$ and $\mathbf{t}$ are randomly initialized to be unit vectors, i.e., $||V_{i_u^t \cdot}||^2 = 1$ and $||\mathbf{t}||^2 = 1$, and b_i and $U_{u \cdot}$ are initialized as $b_i = 0$ and $U_{u \cdot} = \mathbf{0}$.
- **Normalization.** $V_{i_u^t \cdot}$, $V_{i_u^{t+1} \cdot}$ and $V_{j \cdot}$ are re-normalized to be vectors in a unit ball via $x = \frac{x}{\max(1, ||x||)}$ in the learning algorithm.

Algorithm

Algorithm 1 The algorithm of TransRec.

Initialize the model parameters Θ
for *iter* $= 1, ..., T$ **do**
  for *iter2* $= 1, ..., |\mathcal{P}|$ **do**
    Randomly pick up a pair $(u, i_u^t) \in \mathcal{P} \backslash \{i_u^{|\mathcal{S}_u|}\}$
    Take the item i_u^{t+1}
    Randomly pick up an item $j \in \mathcal{I} \backslash \mathcal{S}_u$
    Calculate the gradients via Eqs.(6-12)
    Update the model parameters via Eq.(13)
    Re-normalize $V_{i_u^t \cdot}$, $V_{i_u^{t+1} \cdot}$ and $V_{j \cdot}$
  end for
end for

Nearest Neighbor Search

In the test phase (i.e., recommendation)

1. We replace b_j with $b'_j = b_j - \max_{k \in \mathcal{I}} b_k$ for $j \in \mathcal{I} \backslash \mathcal{S}_u$.
   Note that shifting the bias terms does not change the ranking of the items.
2. For $j \in \mathcal{I} \backslash \mathcal{S}_u$, we absorb b'_j into $V_{j\cdot}$ and have
   - $V'_{j\cdot} = [V_{j\cdot}, \sqrt{-b'_j}] \in \mathbb{R}^{1 \times (d+1)}$ for $\mathcal{L}_2$ distance
   - $V'_{j\cdot} = [V_{j\cdot}, b'_j] \in \mathbb{R}^{1 \times (d+1)}$ for $\mathcal{L}_1$ distance
3. Finally, we use $[V_{i_u^{|\mathcal{S}_u|}\cdot} + U_{u\cdot} + \mathbf{t}, 0] \in \mathbb{R}^{1 \times (d+1)}$ to retrieve some nearest neighbor $V'_{j\cdot}$, $j \in \mathcal{I} \backslash \mathcal{S}_u$ for recommendation.

Dataset

We adopt the commonly used dataset in the experiments, i.e., MovieLens 100K. We treat all the observed behaviors as positive feedback and preprocess the dataset as follows.

- We remove the records of the users who rate fewer than five times.
- We remove the records of the items that are rated fewer than five times.
- We sort all the records according to the timestamps and split each user's sequence into three parts, i.e., the item(s) at the last step for test, the item(s) at the penultimate step for validation, and the remaining items for training.

Baseline

- Bayesian personalized ranking (BPR) [Rendle et al., 2009]
- Factorizing personalized Markov chains (FPMC) [Rendle et al., 2010]

Parameter Configurations

- We fix the number of dimensions $d = 20$, the learning rate $\gamma = 0.01$, and adopt stochastic gradient descent (SGD) or stochastic gradient ascent (SGA) algorithm to train the factorization-based methods.
- We choose the tradeoff parameter of the regularization terms $\beta_b = \alpha_v = \alpha_u = \alpha_t$ from $\{0.1, 0.01, 0.001\}$ and the iteration number T from $\{100, 500, 1000\}$ via the NDCG@20 performance on the validation data.
- We use the same sampling strategy, i.e., randomly selecting one negative sample each time, for fair comparison.
- For each validation data, we select the optimal parameters according to the averaged performance of NDCG@20 of three runs. With the optimal parameter values, the final results on the test data are also the averaged values of three runs.

Evaluation Metrics

- Precision@20
- Recall@20
- NDCG@20

Results

Method	Pre@20	Rec@20	NDCG@20
BPR	0.0282±0.0004	0.1974±0.0049	0.1032±0.0012
FPMC	0.0273±0.0003	**0.2292**±0.0071	**0.1147**±0.0020
TransRec	**0.0288**±0.0003	0.2258±0.0012	0.1142±0.0017

Conclusion

- The sequence modeling approach in TransRec is effective and have almost equal performance with FPMC.

He, R., Kang, W.-C., and McAuley, J. (2017).
Translation-based recommendation.
In *Proceedings of the 11th ACM Conference on Recommender Systems*, RecSys'17, pages 161–169.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009).
BPR: Bayesian personalized ranking from implicit feedback.
In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461.

Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. (2010).
Factorizing personalized Markov chains for next-basket recommendation.
In *Proceedings of the 19th International Conference on World Wide Web*, WWW'10, pages 811–820.