Collaborative Recurrent Neural Networks for Dynamic Recommender Systems

Park, Seung Yun

Contents

- Introduction
- Preliminaries
- Model (RNN, C-RNN, GRU)
- Experiment(Dataset, Comparison)
- Conclusion

Introduction

- A large number of goods and services online
- Because a lot of information can be overwhelming to the user, it will be ordered through the recommender system.
- The system can use the user's explicit or implicit feedback.

Problem

- User behavior is very contextual.

- Explicit information provided by a user typically occurs only once and does not provide much contextual information.

Solution

- **Feedback Implicit**

- The implicit information is richer and easier to collect.

- It it typically collected in the Activity log.

- This type of feedback is updated more frequently, so the dynamic model can benefit more.

- This method is widely applicable because activity logs are very common in many sites.

Solution

• **The temporal aspects of the activity log and the richness of the implicit feedback information can be used to approximate the user's contextual behavior.**

Preliminaries

- Set of users U, where u represents the user.
- Set of items I, where i represents the user.
- For each user u∈ U , They have a sequence like:

$$X^{(u)} = [X_1^{(u)}, \ldots, X_{Tu}^{(u)}] \text{ with elements } X_t^{(u)} \in I$$

- They are not interested in absolute time, but in time patterns, so they define

$$X_{<t}^{(u)} = [X_1^{(u)}, \ldots, X_{t-1}^{(u)}], X_{\geq t}^{(u)} = [X_t^{(u)}, \ldots, X_{Tu}^{(u)}]$$

- In this way, they should be able to quantify the probability that a user u will consume in t=k,k+1,k+2,… having access to $X_{<k}^{(u)}$

Preliminaries

- **The prediction tasks is to model the observed sequences in a probabilistic method.**

$$P_\theta(x^{(u)}) = \prod_{t=1}^{Tu} P_\theta(x_t^{(u)} \mid x_{<t}^{(u)})$$

$$P_\theta(x_t^{(u)} \mid x_{<t}^{(u)}) = p_t^{(u)}\left(\theta, x_{<t}^{(u)}\right)$$

$$\theta = parameters\ of\ the\ model$$

$$p_t^{(u)} = a\ vector\ of\ probabilities.$$

Preliminaries

- **Evaluation metric**

- Average negative log likelihood computed for a held-out part of the sequence : $x_{<r_u}^{(u)}$ $x_{\geq r_u}^{(u)}$, $1< r_u<T_u$

$$\mathrm{E}(x_{\geq r_1}^{(1)}, \ldots, x_{\geq r_U}^{(U)} \mid \theta) = -\frac{1}{U}\sum_{u\in U}\frac{1}{T_u - r_u + 1} \log P_\theta(x_{\geq r_u}^{(u)})$$

Model

1) Flexibility : Models that can get good results from other domains

2) Long-range Dependencies : Powerful model enough to represent dependencies in long sequences

3) Collaboration : For each user, there are components that learn from all users as well as their own components

Model

- **RNN**

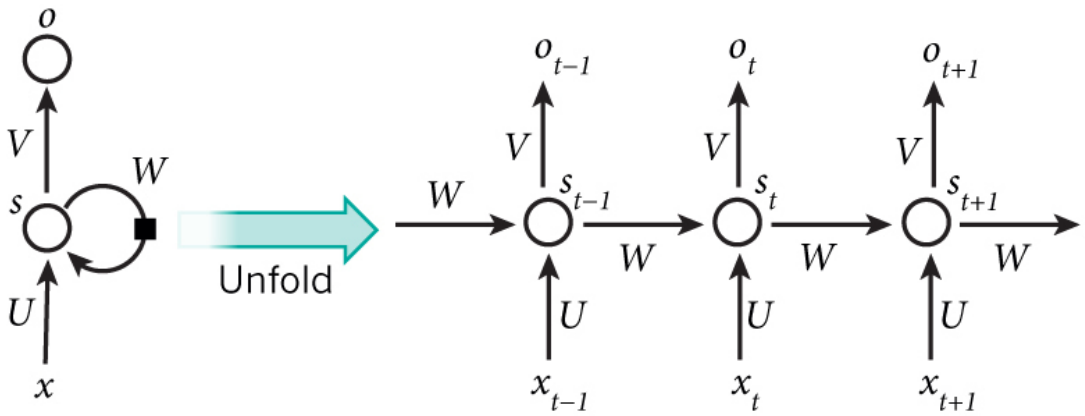
- Recursive Neural Networks (RNN) based models can leverage collaborative aspects to efficiently use multi-user sequencing of datasets.

RNN

- **An artificial neural network including multiple hidden layers between an input layer and an output layer**

$$h_t = f_W(h_{t-1}, x_t) \longrightarrow h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$
$$y_t = W_{hy}h_t$$

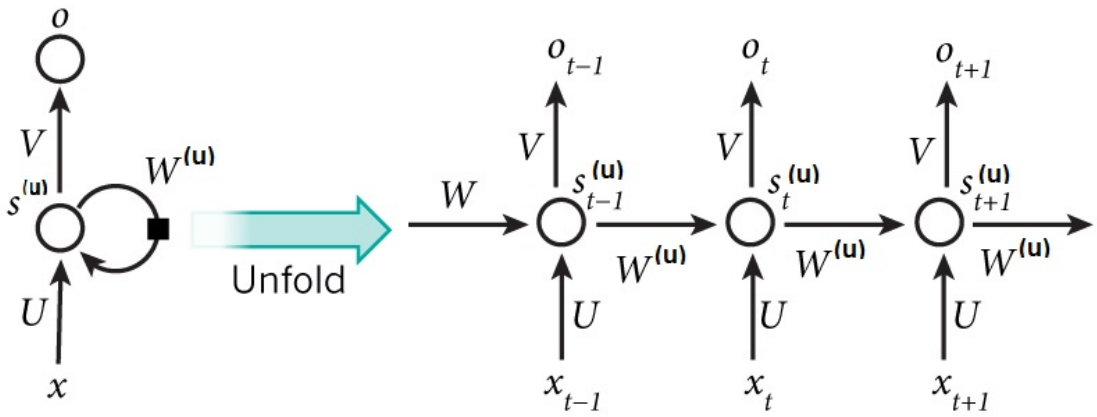
- **gradient vanishing problem**



Collaborative-RNN(C-RNN)

- The input weights U and output V can be considered as embeddings to the real values, so they are common for all users. In this way, the number of parameters is saved.

- The internal states must represent the context of each user, so each user will have their own matrix W



GRU(Gate Recurrent Units)

- like LSTM, GRU have gating mechanism :

Sigmoid function
Value : 0~1

Update Gate $z_t^{(u)} = \sigma\left(W_{zh}^{(u)} h_{t-1}^{(u)} + W_{zi} x_t\right),$

Reset Gate $r_t^{(u)} = \sigma\left(W_{rh}^{(u)} h_{t-1}^{(u)} + W_{ri} x_t\right),$

Tanh function

New memory $\tilde{h}_t^{(u)} = g\left(W_h(r_t^{(u)} \circ h_{t-1}^{(u)}) + W_{in} x_t\right),$

Final memory $h_t^{(u)} = (1 - z_t^{(u)}) \circ h_{t-1}^{(u)} + z_t^{(u)} \tilde{h}_t^{(u)}.$

- Authors choose GRUs over LSTMs for their simpler formulation and parameterization.

Learning Algorithm

Algorithm 1 `processSequence()`

Input: Sequence x, $\theta = \{W_{in}, W_h, W_{out}\}$, batch size B

Output: Updated parameters θ

1: $\mathcal{B} \leftarrow$ Split x into sub-sequences of length B
2: $h_0 \leftarrow \epsilon 1$
3: **for** $b \in \mathcal{B}$ **do**
4: $\quad h_1, \ldots, h_B \leftarrow \text{RNN}(b, h_0; \theta)$ (Forward pass using Eq. 15)
5: $\quad \nabla_\theta \log E \leftarrow \text{BPTT}(b, h_1, \ldots, h_B; \theta)$ (Backward pass, see Appendix A)
6: $\quad \theta \leftarrow \text{LearningRule}(\nabla_\theta, \theta)$
7: $\quad h_0 = h_B$
8: **end for**

Algorithm 2 Collaborative RNN Training

Input: Sequences $x^{(1)}, \ldots, x^{(U)}$, Batch size B

1: Init $\theta = \left\{W_{in}, W_{out}, W_h^{(1)}, \ldots, W_h^{(U)}\right\}$
2: **repeat**
3: $\quad \mathcal{R} = randperm(U)$ (process users in random order)
4: $\quad$ **for** $u \in \mathcal{R}$ **do**
5: $\quad\quad \theta_u \leftarrow \left\{W_{in}, W_{out}, W_h^{(u)}\right\}$
6: $\quad\quad W_{in}, W_{out}, W_h^{(u)} \leftarrow$ `processSequence` $(x^{(u)}, \theta_u, B)$ (Algorithm 1)
7: $\quad$ **end for**
8: **until** Early Stopping Criterion holds

Datasets

- **Brightkite**

  - used to be a location-based social network where users could actively announce (“check in”) their location and find their nearby friends.

  - user-id, location-id, check-in time

- **LastFM**

  - Sequences of songs heard by users. They consider the sequences of artists.

Datasets

The form of the datasets is as follows:

	Users	Items	Events	$E_{\text{Unif.}}$
BK	1 679	23 243	599 618	10.05
LFM	954	48 188	4 320 170	10.78

$E_{Unif.}$ is the negative log probability assigned to a sequences by the uniform baseline in 'Static Uniform'

Comparison of RNN Variants

- **Gated recurrent unit (GRU) is a model based on RNN that adds two vectors (update and reset) that act as gateway to solve the gradient vanishing problem, so the author decides to use an RNN GRU as for his model instead of a basic RNN (tanh RNN).**

- **The author also compares his Collaborative GRU(C-GRU) with the following models:**

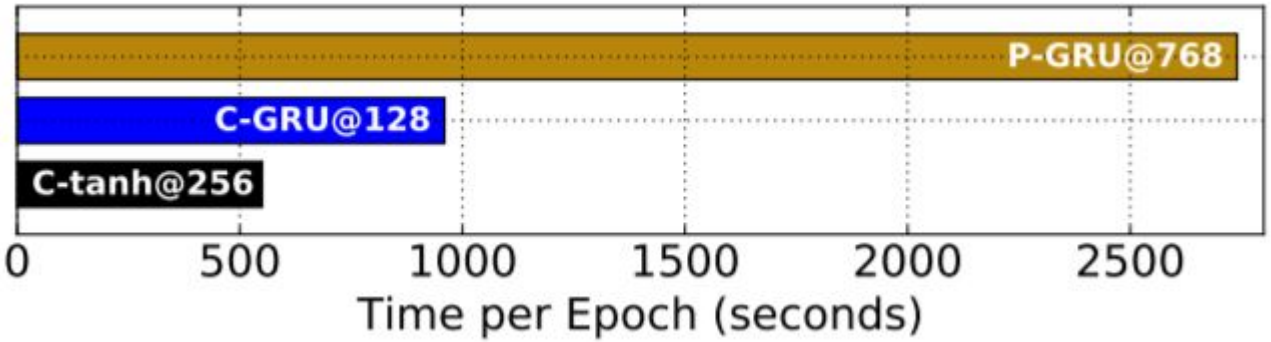
P-GRU

C-tanh

Comparison of RNN Variants

The results between the different RNNs are the following:

	P-GRU	C-tanh	C-GRU
BK	6.45	6.08	**6.02**
LFM	4.58	4.90	**4.51**



Baseline Comparison

The results between the different baselines are the following:

	1-gram	MF	HMM	2-gram	C-GRU
BK	9.53	9.40	8.81	6.73	**6.02**
LFM	8.60	8.86	7.66	5.87	**4.51**

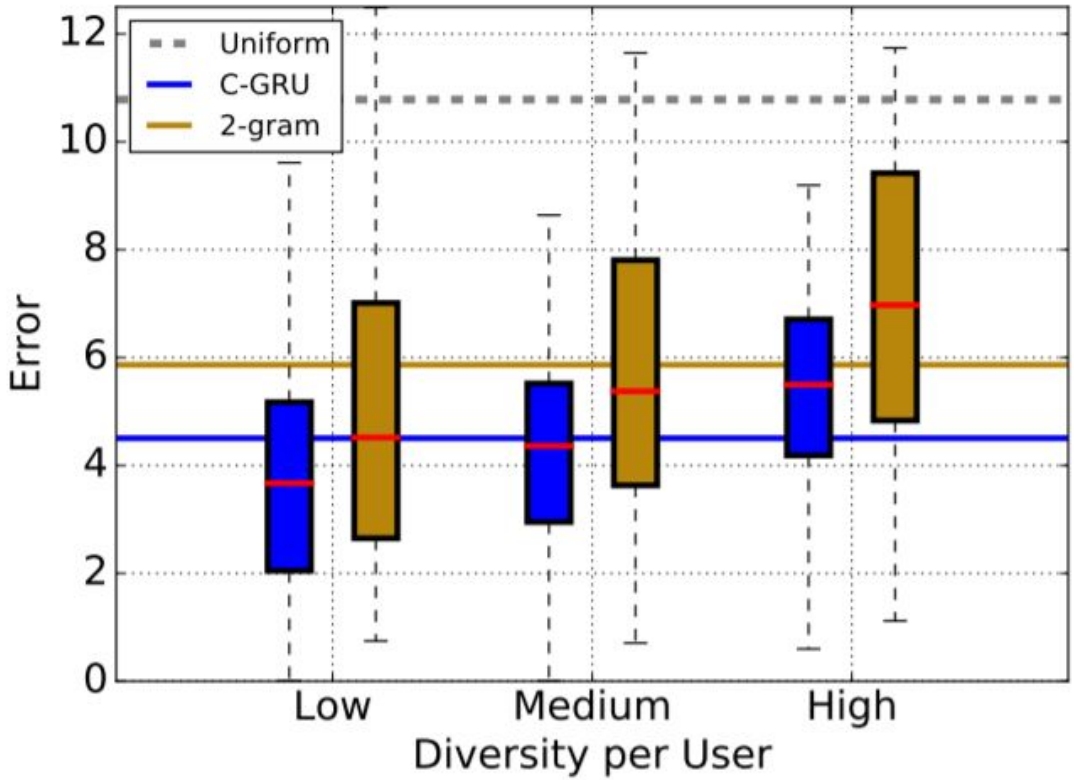
Characterization of Error

- **Error in relation to the difficulty of the sequence.**

- The users were separated into 3 bins according to their Shannon entropy calculated according to the distribution of items that the user consumes.

Characterization of Error

The result is as follows:



Conclusion

- The Collaborative RNN was able to analyze sequences and obtain better results than the baseline in two different datasets.

- The model is practical since it can be scaled to large datasets taking advantage of parallelism.