

# A Variational Recurrent Neural Network for Session-Based Recommendations using Bayesian Personalized Ranking

Panayiotis Christodoulou

Cyprus University of Technology

*paa.christodoulou@edu.cut.ac.cy*

07/09/2017

# Overview

- 1 Motivation
- 2 Proposed Approach
- 3 Experiments

# Motivation

- Recommender Systems (RS) constitute a key part of modern e-commerce websites.
- Recent study on RS has mainly focused on matrix factorization (MF) methods and neighborhood search-type models.
- Session-based recommendation cannot be properly addressed by such conventional methodologies.
- In session-based RS, recommendation is based only on the actions of a user during a specific browsing session.
- This gives rise to hard challenges that stem from the unavailability of rich user profiles (data sparsity).
- On the other hand, user session data constitute action sequences potentially entailing rich, complex, and subtle temporal dynamics.

# Motivation

- We posit that, effective extraction of these underlying dynamics may facilitate addressing the problems stemming from data sparsity.
- In the past, Markov Decision Process (MDP) models have been used to this end, with only limited success.
- Recently, the Deep Learning breakthrough, especially, advances in Recurrent Neural Network (RNN) models, has inspired new, immensely promising lines of research in the field.
- Characteristically, Hidasi et al. (2016) employed gated recurrent unit (GRU) RNNs that are presented with data regarding the items a user selects and clicks on during a given session.

# Proposed Approach: Main Rationale

- We seek to increase the capability of RNN-driven session-based RS to ameliorate the negative effects of data sparsity.
- To this end, we attempt to account for uncertainty, by means of Bayesian inference.
- Our proposed approach is founded upon the fundamental assumption that the hidden units of the postulated RNNs constitute latent variables imposed some appropriate prior distribution.
- Then, we use the training data to infer the corresponding posteriors.
- For scalability, we employ amortized variational inference (AVI):
- (i) Parameterizes the inferred posteriors via conventional neural networks (inference networks); and (ii) casts inference as optimization.

## VRNN-BPR Formulation

- We frame the session-based recommendation problem as a sequence-based prediction problem.
- We denote as  $\{\mathbf{x}_i\}_{i=1}^n$  a user session; here,  $\mathbf{x}_i$  is the  $i$ th clicked item, which constitutes a selection among  $m$  alternatives, and is encoded in the form of an 1-hot representation.
- We formulate session-based recommendation as the problem of predicting the score vector  $\mathbf{y}_{i+1} = [y_{i+1,j}]_{j=1}^m$  of the available items with respect to the following user action, where  $y_{i+1,j} \in \mathbb{R}$  is the predicted score of the  $j$ th item.
- At each time point, we select the top- $k$  items (as ranked via  $\mathbf{y}$ ) to recommend to the user.

## VRNN-BPR Formulation

- Formally, the recurrent units activation vector,  $\mathbf{h}$ , of a GRU are updated at time  $i$  according to the following expression:

$$\mathbf{h}_i = (1 - \mathbf{z}_i) \cdot \mathbf{h}_{i-1} + \mathbf{z}_i \cdot \hat{\mathbf{h}}_i \quad (1)$$

where

$$\mathbf{z}_i = \tau(\mathbf{W}_z \mathbf{x}_i + \mathbf{U}_z \mathbf{h}_{i-1}) \quad (2)$$

$$\hat{\mathbf{h}}_i = \tanh(\mathbf{W} \mathbf{x}_i + \mathbf{U}(\mathbf{r}_i \cdot \mathbf{h}_{i-1})) \quad (3)$$

$$\mathbf{r}_i = \tau(\mathbf{W}_r \mathbf{x}_i + \mathbf{U}_r \mathbf{h}_{i-1}) \quad (4)$$

and  $\tau()$  is the logistic sigmoid function.

- VRNN-BPR extends upon these design principles by rendering the developed GRU model amenable to Bayesian inference.

## VRNN-BPR Formulation

- We consider that the component recurrent unit activations are stochastic latent variables; we impose a simple prior distribution:

$$p(\mathbf{h}_i) = \mathcal{N}(\mathbf{h}_i | \mathbf{0}, \mathbf{I}) \quad (5)$$

- Then, we seek an efficient means of inferring the corresponding posterior distributions, given the available training data.
- We draw inspiration from AVI; we postulate that the sought posteriors,  $q(\mathbf{h})$ , approximately take the form of Gaussians with means and isotropic covariance matrices parameterized via GRU networks.

$$q(\mathbf{h}_i; \theta) = \mathcal{N}(\mathbf{h}_i | \mu_{\theta}(\mathbf{x}_i), \sigma_{\theta}^2(\mathbf{x}_i) \mathbf{I}) \quad (6)$$



## VRNN-BPR Formulation

- Hence, we have:

$$[\boldsymbol{\mu}_\theta(\mathbf{x}_i), \log \sigma_\theta^2(\mathbf{x}_i)] = (1 - \mathbf{z}_i) \cdot [\boldsymbol{\mu}_\theta(\mathbf{x}_{i-1}), \log \sigma_\theta^2(\mathbf{x}_{i-1})] + \mathbf{z}_i \cdot \hat{\mathbf{h}}_i \quad (7)$$

where

$$\mathbf{z}_i = \tau(\mathbf{W}_z \mathbf{x}_i + \mathbf{U}_z [\boldsymbol{\mu}_\theta(\mathbf{x}_{i-1}), \log \sigma_\theta^2(\mathbf{x}_{i-1})]) \quad (8)$$

$$\hat{\mathbf{h}}_i = \tanh(\mathbf{W} \mathbf{x}_i + \mathbf{U}(\mathbf{r}_i \cdot [\boldsymbol{\mu}_\theta(\mathbf{x}_{i-1}), \log \sigma_\theta^2(\mathbf{x}_{i-1})])) \quad (9)$$

and

$$\mathbf{r}_i = \tau(\mathbf{W}_r \mathbf{x}_i + \mathbf{U}_r [\boldsymbol{\mu}_\theta(\mathbf{x}_{i-1}), \log \sigma_\theta^2(\mathbf{x}_{i-1})]) \quad (10)$$

while  $[\boldsymbol{\xi}, \boldsymbol{\zeta}]$  denotes the concatenation of vectors  $\boldsymbol{\xi}$  and  $\boldsymbol{\zeta}$ .

- Then, the values of the latent variables (stochastic unit activations)  $\mathbf{h}_i$  are drawn as samples from the inferred posterior density (6).

## VRNN-BPR Formulation

- We impose a suitable Multinoulli distribution over the generated output variables of our model,  $\mathbf{y}_{i+1}$ , conditional on the corresponding latent vectors,  $\mathbf{h}_i$ .

$$p(y_{i+1,j} = 1 | \mathbf{h}_i) \propto \tau(\mathbf{w}_y^j \cdot \mathbf{h}_i) \quad (11)$$

where  $\mathbf{W}_y = [\mathbf{w}_y^j]_{j=1}^m$  are trainable parameters of the output layer of the model.

- The proposed model uses the matrix factorization method Bayesian Personalized Ranking (BPR) that utilizes a pairwise ranking loss.
- BPR compares the score of the positive item with several sampled items and then utilizes their average as the loss.

## VRNN-BPR Training Algorithm

- Let us consider a training dataset  $\mathcal{D}$ , which comprises a number of click sequences (sessions), pertaining to a multitude of users.
- AVI for VRNN-BPR consists in maximization of the evidence lower-bound (ELBO) expression:

$$\log p(\mathcal{D}) \geq \sum_i \left\{ -\text{KL}[q(\mathbf{h}_i; \boldsymbol{\theta}) || p(\mathbf{h}_i)] - \mathbb{E}[L_s] \right\} \quad (12)$$

- Here,  $\text{KL}[q||p]$  is the KL divergence between  $q(\cdot)$  and  $p(\cdot)$ .
- Unfortunately, the posterior expectation  $\mathbb{E}[L_s]$  cannot be computed analytically; hence, its gradient becomes intractable.
- Approximating this expectation via Monte-Carlo (MC) samples would result in estimators with unacceptably high variance.

## VRNN-BPR Training Algorithm

- We ameliorate this problem by means of a smart reparameterization of the MC samples drawn from the Gaussian posterior density:

$$\mathbf{h}^\gamma = \boldsymbol{\mu}_\theta(\cdot) + \sigma_\theta(\cdot) \boldsymbol{\epsilon}^\gamma \quad (13)$$

where  $\boldsymbol{\epsilon}^\gamma$  is white random noise with unitary variance, i.e.  $\boldsymbol{\epsilon}^\gamma \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ .

- By adopting this reparameterization, the drawn posterior MC samples are expressed as differentiable functions of the sought parameter sets,  $\boldsymbol{\theta}$ , and some random noise variable, *with low variance*.
- Then, we employ the gradient of the reparameterized ELBO [via (13)], in the context of *Adagrad*, to train the model.

## Prediction Generation

- Recommendation generation in the context of a user session can be performed by computing the predicted ratings  $\mathbf{y}_{i+1}$ , and selecting the top- $k$  of them to recommend to the user.
- To this end, we sample the latent variables  $\mathbf{h}_i$  from the corresponding variational posterior distributions, under a standard MC-type rationale.

# Experiments

- We exploit the RecSys Challenge 2015 benchmark.
- This comprises click-stream data pertaining to user sessions with an e-commerce website.
- We split the available data into one set comprising 7,966,257 sessions (with a total of 31,637,239 click actions), and another one comprising the remainder 5,324 sessions (with a total of 71, 222 click actions); we use the former for training and the latter for testing.
- Both sets entail a total of 37,483 items that a user may select to click on. Thus, we are dealing with a very sparse dataset.
- Our source codes have been developed in Python/Theano.

## VRNN-BPR Model Configuration

- We perform Glorot Normal initialization of the model parameters.
- For regularization purposes, we apply Dropout on the component GRUs, with a rate equal to 0.5.
- To facilitate convergence, Nesterov momentum is additionally applied during training.
- We use learning rate 0.1 and momentum 0.3.
- We do not perform any tedious data augmentation procedure or model pretraining.

## Evaluation Metrics

- Recall@20 metric expresses the frequency at which the desired item in the test data appears in the 20 highest ranked items suggested by the evaluated approach.
- MRR@20 describes the average predicted score of the desired items in the test data, with the score values set to zero if the desired item does not make it to the top-20 list of ranked items.



# Experiments

Table: Best performance results of the evaluated methods.

Method	Model Size	Recall@20	MRR@20
BPR-MF	-	0.2574	0.0618
GRU w/ BPR Loss	1000	0.6322	0.2467
GRU w/ TOP1 Loss	1000	0.6206	0.2693
M2	100	0.7129	0.3091
M4	1000	0.6676	0.2847
VRNN-BPR	750	0.7971	0.7845

# Experiments

**Table:** Comparison of computational times (in seconds), for various selections of the employed loss function. Results pertain to network configurations yielding best accuracy.

Method	Network Size	Training time (total)	Prediction time per click event (average)
GRU w/ BPR Loss	1000 Units	48692.48	0.683
GRU w/ TOP1 Loss	1000 Units	44716.60	0.627
VRNN-BP	750 Units	43009.75	0.604

# Experiments

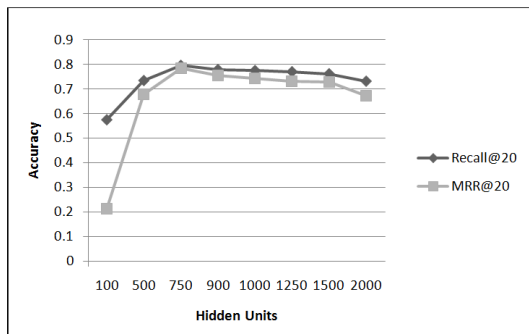


Figure: VRNN-BPR performance fluctuation with the number of latent variables.

# Questions...?



**DOSSIER-Cloud**  
**DevOpS-based Software**  
**engineERING for the cloud**



The project leading to this research has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 692251.