Supplementary Material: A Neural Autoregressive Approach to Collaborative Filtering

Yin Zheng YIN.ZHENG@HULU.COM
Bangsheng Tang BANGSHENG@HULU.COM
Wenkui Ding WENKUI.DING@HULU.COM
Hanning Zhou ERIC.ZHOU@HULU.COM
Hulu LLC., Beijing, 100084

Abstract

We provide here the derivation of the cost function associated with extending CF-NADE to a deep model.

1. Details about Extending CF-NADE to a Deep Model

As mentioned in our submission, training CF-NADE on stochastically sampled orderings corresponds, in expectation, to minimize the cost function over all possible orderings for each user. Thus, the cost function could be written as:

$$\mathcal{C} = \mathop{\mathbb{E}}_{o \in \mathcal{O}} \sum_{i=1}^{D} - \log p \left(r_{m_{o_i}} | \mathbf{r}_{m_{o_{<i}}}, o \right) \tag{1}$$

where $\mathcal{O}$ is the set of all possible orderings, $p(r_{m_{o_i}} | \mathbf{r}_{m_{o_{<i}}}, o)$ is the conditional over an arbitrary ordering o. Note that we here treat o as a random variable explicitly.

We can then move the expectation over ordering, $\mathop{\mathbb{E}}_{o \in \mathcal{O}}$, inside the summation over the conditionals, and split it into 3 parts: one over $o_{<i}$, standing for the first $i-1$ indices in the ordering o; one over o_i, which is the i^{th} index of the ordering o; and one over $o_{>i}$, standing for the remaining indices of the ordering. Thus, Equation 1 can be written as:

$$\mathcal{C} = \sum_{i=1}^{D} \mathop{\mathbb{E}}_{o_{<i}} \mathop{\mathbb{E}}_{o_i} \mathop{\mathbb{E}}_{o_{>i}} - \log p \left(r_{m_{o_i}} | \mathbf{r}_{m_{o_{<i}}}, o_{<i}, o_i, o_{>i} \right) \tag{2}$$

Note that the value of conditionals does not depend on $o_{>i}$,

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

Equation 2 can be simplified as

$$\mathcal{C} = \sum_{i=1}^{D} \mathop{\mathbb{E}}_{o_{<i}} \mathop{\mathbb{E}}_{o_i} - \log p \left(r_{m_{o_i}} | \mathbf{r}_{m_{o_{<i}}}, o_{<i}, o_i \right) \tag{3}$$

The cost in Equation 3 still needs to sum over a number of terms of too large to be performed in practice. Similar to DeepNADE (Uria et al., 2014) and DeepDocNADE (Zheng et al., 2015), we herein approximate the cost in Equation 3 by randomly sampling an i and $o_{<i}$, and approximate the cost function by

$$\hat{\mathcal{C}} = \frac{D}{D-i+1} \sum_{j \geq i} - \log p \left(r_{m_{o_j}} | \mathbf{r}_{m_{o_{<i}}} \right). \tag{4}$$

where D is the number of items that the user has rated and will vary between different users. In words, Equation 4 measures the ability of predicting, from a context of previous $i-1$ ratings $\mathbf{r}_{m_{o_{<i}}}$, the ratings of any remaining items in $\mathbf{r}_{m_{o_{\geq i}}}$. As mentioned in our submission, the factors $\frac{D}{D-i+1}$ in front of the sum comes from the fact that the complete number of elements in the sum will be D and that we use the empirical mean to approximate the expectation $\mathop{\mathbb{E}}_{o_i}$, which has $D-i+1$ possible choices for the item at position i.

Given a user who has rated D items, a training update with Equation 4 can be performed as follows:

1 Randomly draw an ordering from the set of permutations of $(1, 2, \ldots, D)$.

2 Sample a split position i from $\{1, 2, \ldots, D\}$ randomly.

3 Compute each of the conditionals in Equation 4, where $j \geq i$.

4 Compute and sum the gradients from each of the conditionals in Equation 4, and rescale it by $\frac{D}{D-i+1}$

Note that we repeat the above procedure for each update, hence, the ordering o and split position i will be different for different updates of the same training sample.

References

Uria, Benigno, Murray, Iain, and Larochelle, Hugo. A deep and tractable density estimator. *JMLR: W&CP*, 32(1): 467–475, 2014.

Zheng, Y., Zhang, Yu-Jin, and Larochelle, H. A deep and autoregressive approach for topic modeling of multimodal data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PP(99):1–1, 2015. ISSN 0162-8828. doi: 10.1109/TPAMI.2015.2476802.