Amortized Inference of Variational Bounds for Learning Noisy-OR Supplementary Materials

Yiming Yan
University of Southern California

Melissa Ailem
University of Southern California

Fei Sha
Google Research

1 Derivation of variational posterior

In this section we provide detailed derivation for variational posterior.

$$q(\mathbf{z}|\mathbf{x},\boldsymbol{\psi}) = \frac{1}{Z}\prod_{i:x_i=1}^{D}\tilde{p}(x_i=1|\mathbf{z},\psi_i) \prod_{i:x_i=0}^{D}p(x_i=0|\mathbf{z})\prod_{k=0}^{K}p(z_k) \tag{1}$$

where Z is the normalization term and

$$Z = \sum_{\mathbf{z}}\prod_{i:x_i=1}^{D}\tilde{p}(x_i=1|\mathbf{z},\psi_i) \prod_{i:x_i=0}^{D}p(x_i=0|\mathbf{z})\prod_{k=0}^{K}p(z_k) \tag{2}$$

The approximate joint probability $\tilde{p}(\mathbf{x},\mathbf{z},\boldsymbol{\psi})$ is

$$\begin{aligned}\tilde{p}(\mathbf{x},\mathbf{z},\boldsymbol{\psi}) &= \prod_{i:x_i=1}^{D}\tilde{p}(x_i=1|\mathbf{z},\psi_i)\prod_{i:x_i=0}^{D}p(x_i=0|\mathbf{z})\prod_{k=0}^{K}p(z_k)\\ &= \exp\Big(\sum_{i=1}^{D}x_i(\psi_i\boldsymbol{\theta}_i^T\mathbf{z}-g(\phi_i))-(1-x_i)\boldsymbol{\theta}_i^T\mathbf{z}\Big)p(\mathbf{z})\\ &= \exp\Big(C+\sum_{i=1}^{D}(x_i\psi_i-(1-x_i))\sum_{k=0}^{K}\theta_{ik}z_k\Big)p(\mathbf{z})\end{aligned} \tag{3}$$

where $C=-\sum_{i=1}^{D}x_i g(\psi_i)$.

The normalized term Z is the marginal likelihood $\tilde{p}(\mathbf{x},\boldsymbol{\psi})$, which can be computed as

$$\begin{aligned}Z &= \exp(C)\mathbb{E}_{p(\mathbf{z})}\Big[\prod_{k=0}^{K}\exp\Big(\sum_{i=1}^{D}(x_i\psi_i-(1-x_i))\theta_{ik}z_k\Big)\Big]\\ &= \exp(C)\prod_{k=0}^{K}\mathbb{E}_{p(z_k)}\Big[\Big(\sum_{i=1}^{D}(x_i\psi_i-(1-x_i))\theta_{ik}z_k\Big)\Big]\\ &= \exp(C)\prod_{k=0}^{K}\Big[\mu_k\sum_{i=1}^{D}(x_i\psi_i-(1-x_i))\theta_{ik}+(1-\mu_k)\Big]\end{aligned} \tag{4}$$

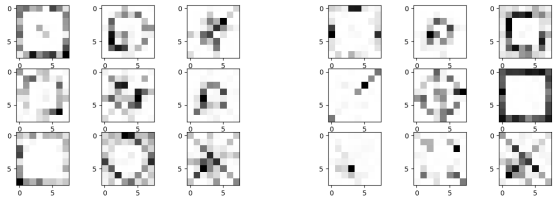
We substitute eq. (3) and (4) to eq. (1), and obtain the variational posterior

$$\begin{aligned}q(z_k=1|\mathbf{x},\boldsymbol{\psi}) &= \frac{\mu_k\exp\Big(\sum_{i=1}^{D}(x_i\psi_i-(1-x_i))\theta_{ik}\Big)}{\mu_k\exp\Big(\sum_{i=1}^{D}(x_i\psi_i-(1-x_i))\theta_{ik}\Big)+(1-\mu_k)}\\ &= \sigma\left(\sum_{i:x_i=1}\psi_i\theta_{ik}-\sum_{i:x_i=0}\theta_{ik}+\log\frac{\mu_k}{1-\mu_k}\right)\end{aligned} \tag{5}$$

2 Pseudocode for parameter estimation using ACP

Algorithm 1: Parameter estimation of ACP

Input: training set of binary vectors $\mathbf{X}=\{\mathbf{x}^{(n)},n=1,2,\ldots,N\}$;
Initialization: initialize the NOISY-OR model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$, and parameters $\boldsymbol{\phi}$ of MLP;
while *not converge* **do**
  Randomly get a batch $\mathbf{x}$ from $\mathbf{X}$;
  Obtain variational parameters $\boldsymbol{\psi}=\text{MLP}(\mathbf{x};\boldsymbol{\phi})$;
  for *each latent variable* k **do**
    Compute the parameter of approximate posterior $q_k=q(z_k=1|\mathbf{x},\boldsymbol{\psi})=\sigma\left(\sum_{i:x_i=1}\psi_i\theta_{ik}-\sum_{i:x_i=0}\theta_{ik}+\log\frac{\mu_k}{1-\mu_k}\right)$;
    for m *in* $1\ldots M$ **do**
      Sample latent variable z_{km} from $q(z_k|\mathbf{x},\boldsymbol{\psi})$ using gumbel softmax reparametrization trick;
    end
    Compute loss using eq. (16) $\mathcal{L}(\mathbf{x};\boldsymbol{\theta},\boldsymbol{\phi})=\frac{1}{M}\sum_{m=1}^{M}\sum_i x_i\log\big(1-\exp(-\theta_{i0}-\sum_k\theta_{ik}z_{km})\big)+\sum_i(1-x_i)(-\theta_{i0}-\sum_k\theta_{ik}q_k)-\big(\sum_k q_k\log\frac{q_k}{\mu_k}+(1-q_k)\log\frac{1-q_k}{1-\mu_k}\big)$;
    Update $\boldsymbol{\theta}$, $\boldsymbol{\mu}$ and $\boldsymbol{\phi}$ through back propagation
  end
end



(a) LB-CDI, $N_{train} = 1000$ (b) SVI, $N_{train} = 1000$

Figure 1: The recovered parameters after training with 1000 data points using LB-CDI and SVI.

3 Implementation details

All our experiments were performed using Adam optimizer [1] with a batch size of 128. During training, we set the number of Monte Carlo samples to $L = 10$ for each data point to compute the ELBO. We rely on Gumbel-softmax reparametrization trick [2] to approximate sampling latent variables $\mathbf{z}$ using continuous value to back-propagate gradients. Following [2], we schedule exponential temperature decay, with the initial temperature to be 0.5 and the minimum temperature to be 0.2. While during testing, we use the true discrete samples from the posterior and sample 100 times to compute ELBO. For ACP, the variational parameter ψ is the output of a neural network, which is constrained to be greater than 0. Thus we use a `softplus` layer as the last layer of the neural network. The architecture (number of hidden layers and hidden dimensions) of the inference model for both AVI and ACP, as well as other hyperparameters including learning rate, momentum, temperature decay rate and temperature decay step, are sampled randomly for 100 times. We only report the result with the best hyperparameters. All experiments results are averaged from 5 different random initializations.

4 Experiments

4.1 Parameter Estimation

Fig. 1 shows the recovered parameters using LB-CDI and SVI. Even with sufficient training data ($N_{train} = 1000$), both methods achieved bad estimation results. Both of them are able to learn the parameter patterns to some extend. However all the patterns are merged together. Hence we conclude ACP and AVI achieve better parameter estimation results comparing to the two non-amorized methods when we have sufficient training data.

Additionally, we did the parameter estimation experiments on MULTI-MNIST dataset. And the experiment results are depicted in Fig. 2. Here, since the training set of MULTI-MNIST is large, we did not do LB-CDI.

In Fig. 2, similar phenomenon has been observed. When we have large amount of training data, both AVI and ACP (Fig. 2a and 2b) recovered parameters well. Even though AVI did not capture pattern "1", it is indeed not trivial to separate pattern "1" and "7" in this dataset. However, SVI did not recover the parameters well.

When we reduce the amount of training data, the number of patterns detected by AVI decreased largely, as three weight patterns are recovered as "0", which also indicates worse latent representation learning. However for ACP, although it messed up pattern "4" and "5", it recovered all other patterns, even with small amount of training data.

5 Additional experiments

5.1 Document classification

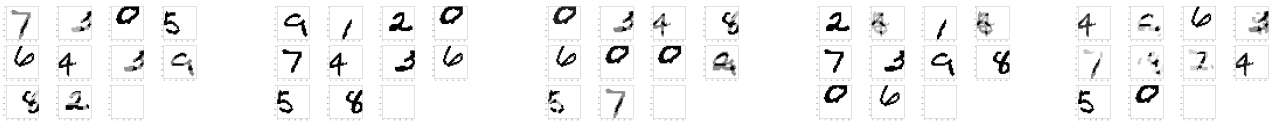
Herein, we aim to assess the impact of our inference method on NOISY-OR model's learned representations. In particular, we rely on document classification task to evaluate the quality of the features learned by our model. To this end, we use the Reuters corpus[1] from NLTK, which consists of 1.3 million words and 10,788 news articles organized into 90 categories. For this experiment, we retain the top 3 categories,[2] namely `acq`, `earn` and `money-fx`. Each document is represented by its headline. We lemmatize the words, remove stop words, and remove words with less than 5 occurrences. We obtain a final corpus of 839 unique words and 7030 documents, including 5048 for training and 1982 for test. Similar to topic modeling, each document is represented by a binary vector where each dimension indicates a word presence/absence.

After training AVI and ACP, we take the approximate posterior distribution $\left\{q(z_k^{(n)} = 1|\mathbf{x}^{(n)};\boldsymbol{\phi})\right\}_{k=1}^{K}$ as the latent representation of document $\mathbf{x}^{(n)}$. We evaluate the quality of learned representations on the test set. More specifically, we train a linear multilabel classifier, which takes the posterior distribution as input and predicts the document classes. We perform 5-fold cross-validation and report the average EM scores.

Fig. 3 shows the classification performance with different amount of training data and different dimensionality of latent variables. The black dashed line corresponds to the results obtained when performing classification on the original space $\mathbf{X}$. We notice that when using a training set of more than 1000 documents, AVI achieves higher classification accuracy owing to its

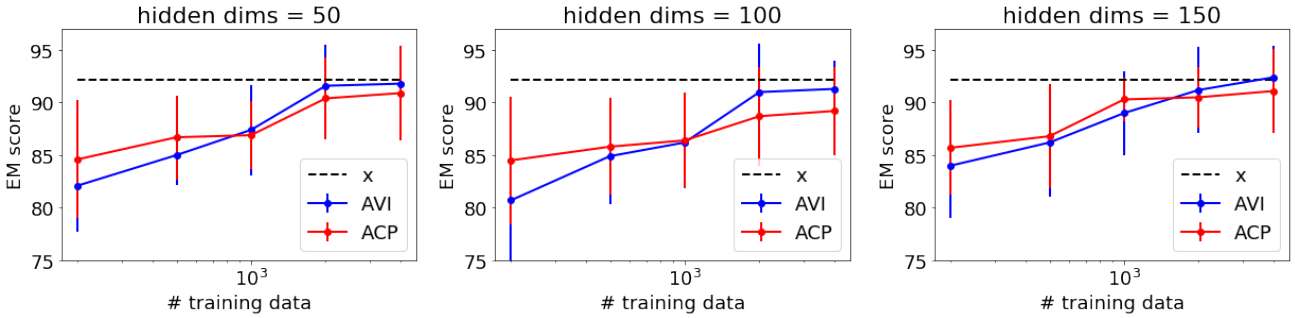
[1] https://www.nltk.org/book/ch02.html

[2] the 3 classes containing the most documents.



(a) AVI, $N_{train} = 50K$ (b) ACP, $N_{train} = 50K$ (c) AVI, $N_{train} = 8K$ (d) ACP, $N_{train} = 8K$ (e) SVI, $N_{train} = 50K$

Figure 2: The recovered parameters of MULTI-MNIST after training with $50K$ and $8K$ data points using AVI, ACP and SVI.



(a) (b) (c)

Figure 3: EM scores of AVI and ACP with different amount of training data and different hidden dimensions. The black dashed line indicates the classification performance with **x** in test set as input.

larger inference capacity and flexibility. However, its performance drops quickly as we reduce the size of the training set. In contrast, our ACP inference offers more stability w.r.t. to the amount of training examples, and reaches higher classification performance when using smaller training sets.

We present in Fig. 4, 5 and 6 the t-SNE visualizations of the approximate posterior distributions learned by each model using 50, 100 and 150 hidden dimensions respectively. We observe that when using a small training set (middle and right columns), the `acq` and `money-fx` features learned by AVI tend to fuse together, while with ACP, we can still distinguish the three categories. This observation confirms our previous results and claims about the effectiveness of our model when lacking training data.

References

[1] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[2] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

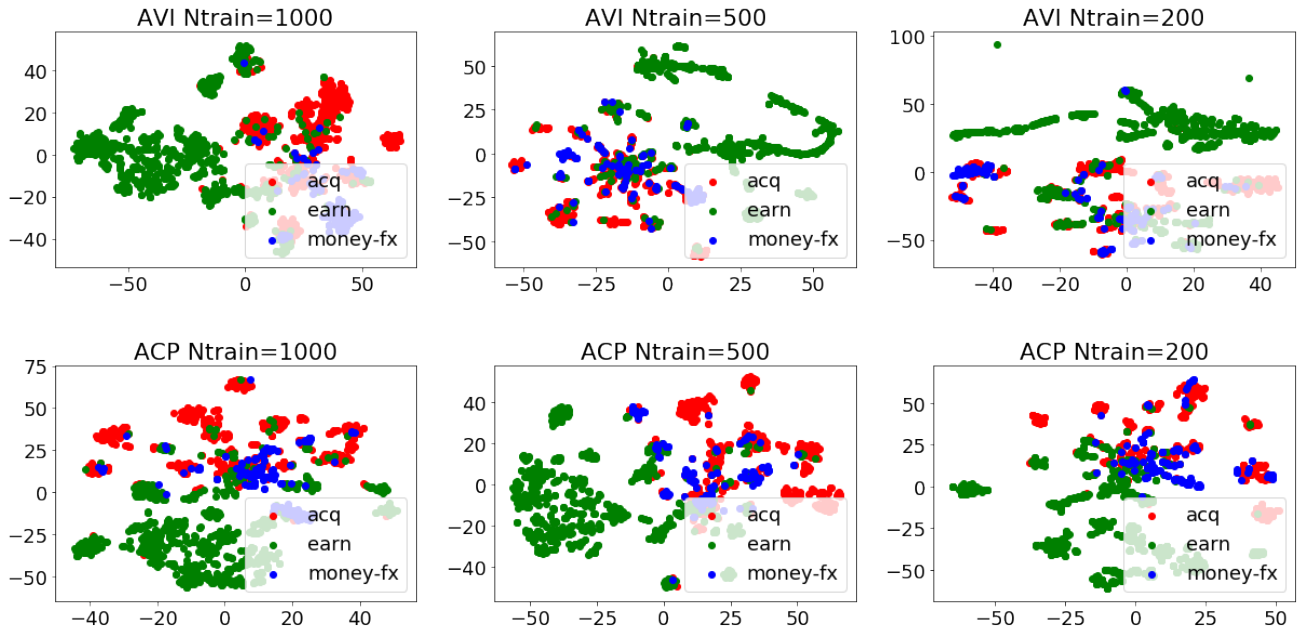


Figure 4: t-SNE visualization on latent representations on held out set when latent dimension is 50.

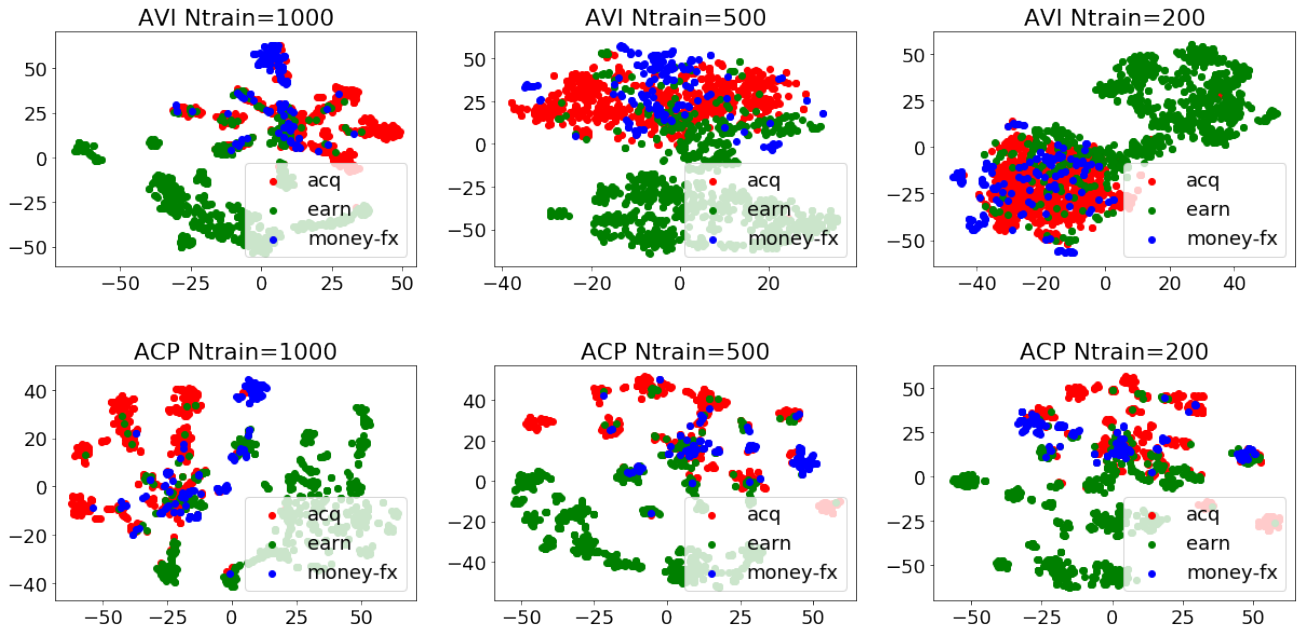


Figure 5: t-SNE visualization on latent representations on held out set when latent dimension is 100.

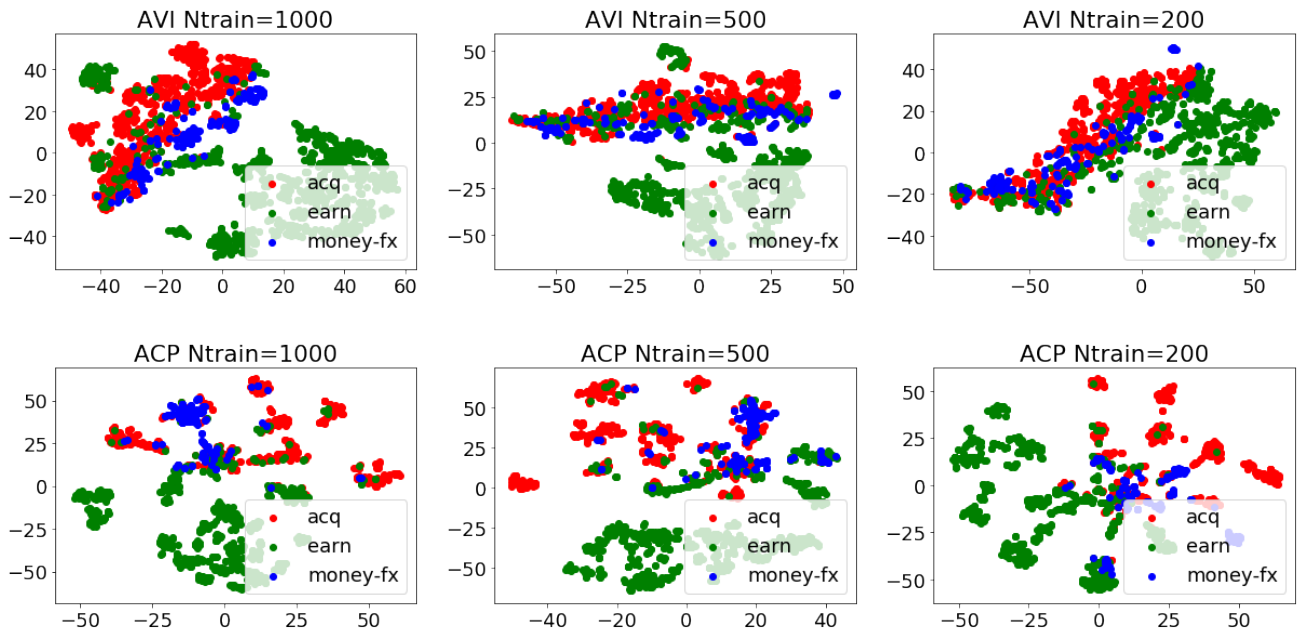


Figure 6: t-SNE visualization on latent representations on held out set when latent dimension is 150.