Variational Autoencoders for Recommendation

Dawen Liang
Netflix Research

Rahul Krishnan  Matt Hoffman  Tony Jebara
Background

• Implicit feedback data (No more rating predictions with RMSE please :)

• In the form of user-item interaction matrix

• Both the observed and missing entries are taken into account for modeling

• Top-N recommender systems
Implicit feedback data (No more rating predictions with RMSE please :) 

- In the form of user-item interaction matrix

- Both the observed and missing entries are taken into account for modeling

- Top-N recommender systems

\[ x_u = [0, 0, 1, 0, \ldots, 1, 0] \]
Variational autoencoders:
Model & Inference

Kingma & Welling, Auto-encoding variational Bayes, 2013
Rezende et al., Stochastic backpropagation and approximation inference in deep generative models, 2014
Variational autoencoders: Model & Inference

• Model: multinomial non-linear factor analysis

For each user $u$

$$z_u \sim \mathcal{N}(0, I_K), \quad \pi(z_u) \propto \exp\{f_\theta(z_u)\},$$
$$x_u \sim \text{Mult}(N_u, \pi(z_u)).$$

Kingma & Welling, Auto-encoding variational Bayes, 2013
Rezende et al., Stochastic backpropagation and approximation inference in deep generative models, 2014
Variational autoencoders: Model & Inference

• Model: multinomial non-linear factor analysis

For each user $u$

\[
\begin{align*}
z_u &\sim \mathcal{N}(0, I_K), \\
\pi(z_u) &\propto \exp\{f_{\theta}(z_u)\}, \\
x_u &\sim \text{Mult}(N_u, \pi(z_u)).
\end{align*}
\]

Kingma & Welling, Auto-encoding variational Bayes, 2013
Rezende et al., Stochastic backpropagation and approximation inference in deep generative models, 2014
Variational autoencoders: Model & Inference

• Model: multinomial non-linear factor analysis

  For each user $u$

  $$z_u \sim \mathcal{N}(0, I_K), \quad \pi(z_u) \propto \exp\{f_\theta(z_u)\},$$

  $$x_u \sim \text{Mult}(N_u, \pi(z_u)).$$

• Inference: reason about the (intractable) posterior

  $$p(z_u \mid x_u) \approx q(z_u) = \mathcal{N}(\mu_u, \sigma_u^2)$$

Kingma & Welling, Auto-encoding variational Bayes, 2013
Rezende et al., Stochastic backpropagation and approximation inference in deep generative models, 2014
Variational autoencoders: Model & Inference

• Model: multinomial non-linear factor analysis

For each user $u$

$$z_u \sim \mathcal{N}(0, I_K), \quad \pi(z_u) \propto \exp\{f_\theta(z_u)\},$$

$$x_u \sim \text{Mult}(N_u, \pi(z_u)).$$

• Inference: reason about the (intractable) posterior

$$p(z_u | x_u) \approx q(z_u) = \mathcal{N}(\mu_u, \sigma^2_u)$$

Kingma & Welling, Auto-encoding variational Bayes, 2013
Rezende et al., Stochastic backpropagation and approximation inference in deep generative models, 2014
Variational autoencoders: Model & Inference

- Model: multinomial non-linear factor analysis

  For each user $u$

  \[ z_u \sim \mathcal{N}(0, I_K), \quad \pi(z_u) \propto \exp\{f_\theta(z_u)\}, \]

  \[ x_u \sim \text{Mult}(N_u, \pi(z_u)). \]

- Inference: data-dependent inference functions

Kingma & Welling, Auto-encoding variational Bayes, 2013
Rezende et al., Stochastic backpropagation and approximation inference in deep generative models, 2014
Variational autoencoders: Model & Inference

- **Model**: multinomial non-linear factor analysis

  For each user $u$

  \[
  z_u \sim \mathcal{N}(0, I_K), \quad \pi(z_u) \propto \exp\{f_\theta(z_u)\}, \\
  x_u \sim \text{Mult}(N_u, \pi(z_u)).
  \]

- **Inference**: data-dependent inference functions

  \[
  p(z_u | x_u) \approx q_\phi(z_u | x_u) = \mathcal{N}(\mu_\phi(x_u), \sigma^2_\phi(x_u))
  \]

Kingma & Welling, Auto-encoding variational Bayes, 2013
Rezende et al., Stochastic backpropagation and approximation inference in deep generative models, 2014
Variational autoencoders: Model & Inference

- Model: multinomial non-linear factor analysis

  For each user $u$

  $$z_u \sim \mathcal{N}(0, I_K), \quad \pi(z_u) \propto \exp \{f_\theta(z_u)\},$$
  $$x_u \sim \text{Mult}(N_u, \pi(z_u)).$$

- Inference: data-dependent inference functions

  $$p(z_u \mid x_u) \approx q_\phi(z_u \mid x_u) = \mathcal{N}(\mu_\phi(x_u), \sigma^2_\phi(x_u))$$

Kingma & Welling, Auto-encoding variational Bayes, 2013
Rezende et al., Stochastic backpropagation and approximate inference in deep generative models, 2014
Variational autoencoders: Model & Inference

- Model: multinomial non-linear factor analysis
  
  For each user $u$
  
  $z_u \sim N(0, I_K), \quad \pi(z_u) \propto \exp\{f_\theta(z_u)\}$,
  
  $x_u \sim \text{Mult}(N_u, \pi(z_u))$.  
  Non-linear function

- Inference: data-dependent inference functions
  
  $p(z_u \mid x_u) \approx q_\phi(z_u \mid x_u) = N(\mu_\phi(x_u), \sigma_\phi^2(x_u))$
  
  Non-linear function

---

Kingma & Welling, Auto-encoding variational Bayes, 2013
Rezende et al., Stochastic backpropagation and approximation inference in deep generative models, 2014
Why VAEs
(or rather, Bayesian?)
Why VAEs  
(or rather, Bayesian?)

• Generalize linear latent factor models

• Recover Gaussian matrix factorization as a special linear case

Salakhutdinov & Mnih, Probabilistic matrix factorization, 2008
Why VAEs
(or rather, Bayesian?)

- Generalize linear latent factor models
  - Recover Gaussian matrix factorization as a special linear case
- No iterative procedure required to rank all the items given a user’s watch history
  - Only need to evaluate inference and generative functions

Salakhutdinov & Mnih, Probabilistic matrix factorization, 2008
Why VAEs (or rather, Bayesian?)

- Generalize linear latent factor models
  - Recover Gaussian matrix factorization as a special linear case
- No iterative procedure required to rank all the items given a user’s watch history
  - Only need to evaluate inference and generative functions
- RecSys is more of a “small data” than a “big data” problem

Salakhutdinov & Mnih, Probabilistic matrix factorization, 2008
Training VAEs

\[ \mathbb{E}_{q(z \mid x)} \left[ \log p(x \mid z) \right] - \beta \cdot \text{KL}(q(z \mid x) \mid \mid p(z)) \]
Training VAEs

$$E_{q(z \mid x)} [\log p(x \mid z)] - \beta \cdot KL(q(z \mid x) || p(z))$$

(Negative) reconstruction error
Training VAEs

\[ \mathbb{E}_{q(z \mid x)} \left[ \log p(x \mid z) \right] - \beta \cdot \text{KL}(q(z \mid x) \| p(z)) \]

(Negative) reconstruction error  
“Regularization”
Training VAEs

\[ \mathbb{E}_{q(z \mid x)} \left[ \log p(x \mid z) \right] - \beta \cdot \text{KL}(q(z \mid x) \mid \mid p(z)) \]

(Negative) reconstruction error  “Regularization”
Training VAEs

\[ E_{q(z \mid x)} \left[ \log p(x \mid z) \right] - \beta \cdot \text{KL}(q(z \mid x) \| p(z)) \]

(Negative) reconstruction error “Regularization”

• Setting $\beta < 1$ relaxes the prior constraint

• For RecSys, we don’t necessarily need all the statistical property of a generative model

• Trading off the ability of performing ancestral sampling for better fitting the data
Selecting $\beta$

Bowman et al., Generating sentences from a continuous space, 2015
Training VAEs

\[ \mathbb{E}_{q(z \mid x)} [\log p(x \mid z)] - \beta \cdot \text{KL}(q(z \mid x) \parallel p(z)) \]

- Information-theoretic connections
- Maximum entropy discrimination & Information bottleneck principle
- Recent work on understanding the trade-offs in learning latent variable models with VAEs
- Variational lossy autoencoders, $\beta$-VAE, deep variational information bottleneck (hopefully many to come in ICLR)

Jaakkola et al., Maximum entropy discrimination, 2000
Tishby et al., The information bottleneck method, 2000
Chen et al., Variational lossy autoencoders, 2016
Higgins et al., $\beta$-VAE: Learning basic visual concepts with a constrained variational framework, 2016
Alemi et al., Deep variational information bottleneck, 2016
Evaluation:
Strong generalization

Marlin, Collaborative filtering: A machine learning perspective, 2004
Evaluation:
Strong generalization

Held-out user:
Not used in the training

Marlin, Collaborative filtering: A machine learning perspective, 2004
Evaluation:
Strong generalization

Held-out user:
Not used in the training

Marlin, Collaborative filtering: A machine learning perspective, 2004
Evaluation:
Strong generalization

Held-out user:
Not used in the training

“Fold-in” set:
• Learn necessary user-level representation
• Obtain predicted ranking

“Target” set:
Report ranking metrics (Recall@K, NDCG@K) on

Marlin, Collaborative filtering: A machine learning perspective, 2004
# Empirical studies

<table>
<thead>
<tr>
<th></th>
<th>ML-20M</th>
<th>Netflix</th>
<th>MSD</th>
</tr>
</thead>
<tbody>
<tr>
<td># of users</td>
<td>136,677</td>
<td>463,435</td>
<td>571,355</td>
</tr>
<tr>
<td># of items</td>
<td>20,108</td>
<td>17,769</td>
<td>41,140</td>
</tr>
<tr>
<td># of interactions</td>
<td>10.0M</td>
<td>56.9M</td>
<td>33.6M</td>
</tr>
<tr>
<td>% of interactions</td>
<td>0.36%</td>
<td>0.69%</td>
<td>0.14%</td>
</tr>
<tr>
<td># of held-out users</td>
<td>10,000</td>
<td>40,000</td>
<td>50,000</td>
</tr>
</tbody>
</table>
Quantitative results

- Multi-VAE\textsuperscript{PR}: Partially Regularized VAE with multinomial likelihood
- Multi-DAE: Denoising autoencoder with multinomial likelihood
- Baselines:
  - WMF & SLIM: linear collaborative filtering methods
  - CDAE: Non-linear neural network based method

<table>
<thead>
<tr>
<th></th>
<th>Recall@20</th>
<th>Recall@50</th>
<th>NDCG@100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-VAE\textsuperscript{PR}</td>
<td>0.395</td>
<td>0.537</td>
<td>0.426</td>
</tr>
<tr>
<td>Multi-DAE</td>
<td>0.387</td>
<td>0.524</td>
<td>0.419</td>
</tr>
<tr>
<td>WMF</td>
<td>0.360</td>
<td>0.498</td>
<td>0.386</td>
</tr>
<tr>
<td>SLIM</td>
<td>0.370</td>
<td>0.495</td>
<td>0.401</td>
</tr>
<tr>
<td>CDAE</td>
<td>0.391</td>
<td>0.523</td>
<td>0.418</td>
</tr>
<tr>
<td><strong>ML20M</strong></td>
<td><strong>0.351</strong></td>
<td><strong>0.444</strong></td>
<td><strong>0.386</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Recall@20</th>
<th>Recall@50</th>
<th>NDCG@100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-VAE\textsuperscript{PR}</td>
<td>0.351</td>
<td>0.444</td>
<td>0.386</td>
</tr>
<tr>
<td>Multi-DAE</td>
<td>0.344</td>
<td>0.438</td>
<td>0.380</td>
</tr>
<tr>
<td>WMF</td>
<td>0.316</td>
<td>0.404</td>
<td>0.351</td>
</tr>
<tr>
<td>SLIM</td>
<td>0.347</td>
<td>0.428</td>
<td>0.379</td>
</tr>
<tr>
<td>CDAE</td>
<td>0.343</td>
<td>0.428</td>
<td>0.376</td>
</tr>
<tr>
<td><strong>Netflix Prize</strong></td>
<td><strong>0.351</strong></td>
<td><strong>0.444</strong></td>
<td><strong>0.386</strong></td>
</tr>
</tbody>
</table>

Hu et al., Collaborative filtering with implicit feedback datasets, 2008
Wu et al., Collaborative denoising auto-encoders for top-N recommender systems, 2016
Why Bayesian? (cont.)

**ML20M**: each user has watched at least 5 movies

**MSD**: each user has listened to at least 20 songs

User activity:  
Low  
High
Conclusion

- We extend VAEs to collaborative filtering for implicit feedback

- We introduce a regularization parameter for the learning objective to trade-off the generative power for better predictive recommendation performance

- Besides competitive empirical performance, we also identify when and why a principled Bayesian approach performs better
Thanks!

• We extend VAEs to collaborative filtering for implicit feedback

• We introduce a regularization parameter for the learning objective to trade-off the generative power for better predictive recommendation performance

• Besides competitive empirical performance, we also identify when and why a principled Bayesian approach performs better

Netflix tech blog: https://medium.com/netflix-techblog