criteo AI Lab

NETFLIX

Variational Autoencoders for Recommender Systems:

A Critical Retrospective and a (Hopefully) Optimistic Prospective

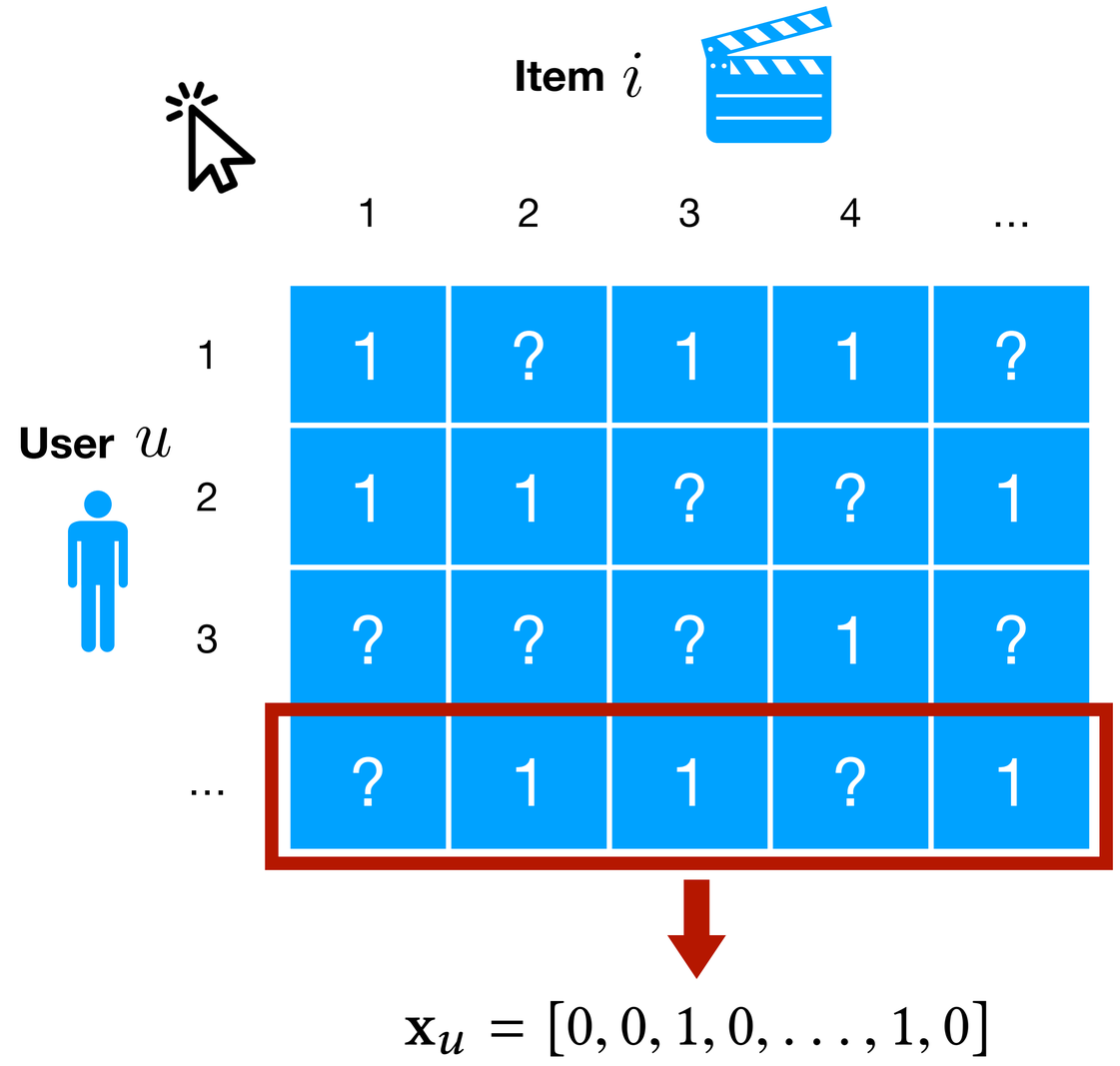
Dawen Liang
Netflix Research

Laplace's Demon Seminar Series

A Critical Retrospective

Background

- Implicit feedback data
  - In the form of user-item interaction matrix
  - Both the observed and missing entries are taken into account for modeling
  - Top-*N* recommender systems



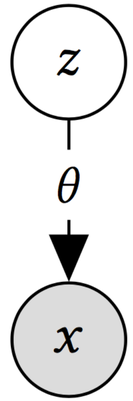
Variational autoencoders: Model & Inference

- Model: multinomial non-linear factor analysis

For each user u

$$\mathbf{z}_u \sim \mathcal{N}(0, \mathbf{I}_K), \quad \pi(\mathbf{z}_u) \propto \exp\{f_\theta(\mathbf{z}_u)\},$$
$$\mathbf{x}_u \sim \text{Mult}(N_u, \pi(\mathbf{z}_u)).$$

Non-linear function



- Inference: reason about the (intractable) posterior

$$p(\mathbf{z}_u \mid \mathbf{x}_u) \approx q(\mathbf{z}_u) = \mathcal{N}(\boldsymbol{\mu}_u, \boldsymbol{\sigma}_u^2)$$

Free parameters

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

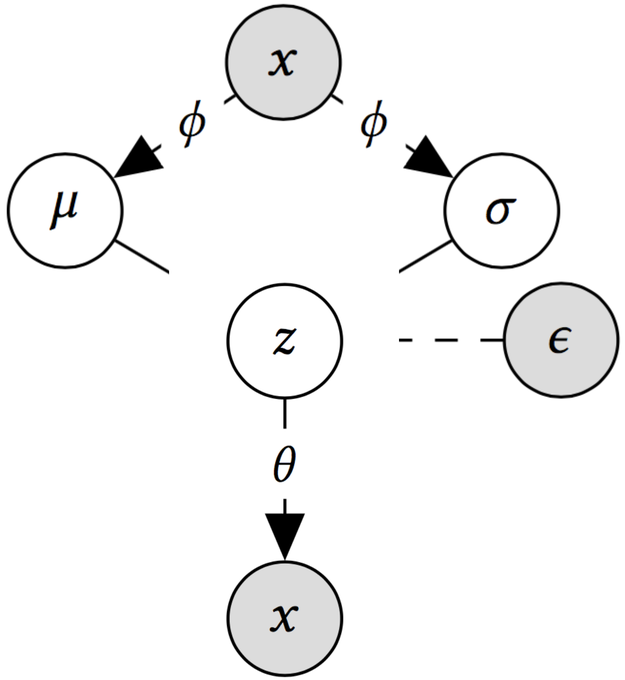
Variational autoencoders: Model & Inference

- Model: multinomial non-linear factor analysis

For each user u

$$\mathbf{z}_u \sim \mathcal{N}(0, \mathbf{I}_K), \quad \pi(\mathbf{z}_u) \propto \exp\{f_\theta(\mathbf{z}_u)\},$$
$$\mathbf{x}_u \sim \text{Mult}(N_u, \pi(\mathbf{z}_u)).$$

Non-linear function



- Inference: data-dependent inference functions

$$p(\mathbf{z}_u \mid \mathbf{x}_u) \approx q_\phi(\mathbf{z}_u \mid \mathbf{x}_u) = \mathcal{N}(\mu_\phi(\mathbf{x}_u), \sigma^2_\phi(\mathbf{x}_u))$$

Non-linear function

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

Why multinomial?

- Commonly used in language models (e.g., LDA) and economics (e.g., multinomial logit choice model)
- Close proxy to the top-*N* ranking loss relative to Gaussian and logistic
  - The likelihood rewards the model for putting probability mass on the non-zero entries in the click matrix
  - Since $\pi(\mathbf{z}_u)$ must sum to 1, the items have to compete for limited budget

→ Effectively ranking non-zero entries higher

Blei et al., Latent Dirichlet allocation, 2003
McFadden et al., Conditional logit analysis of qualitative choice behavior, 1973

Training VAEs

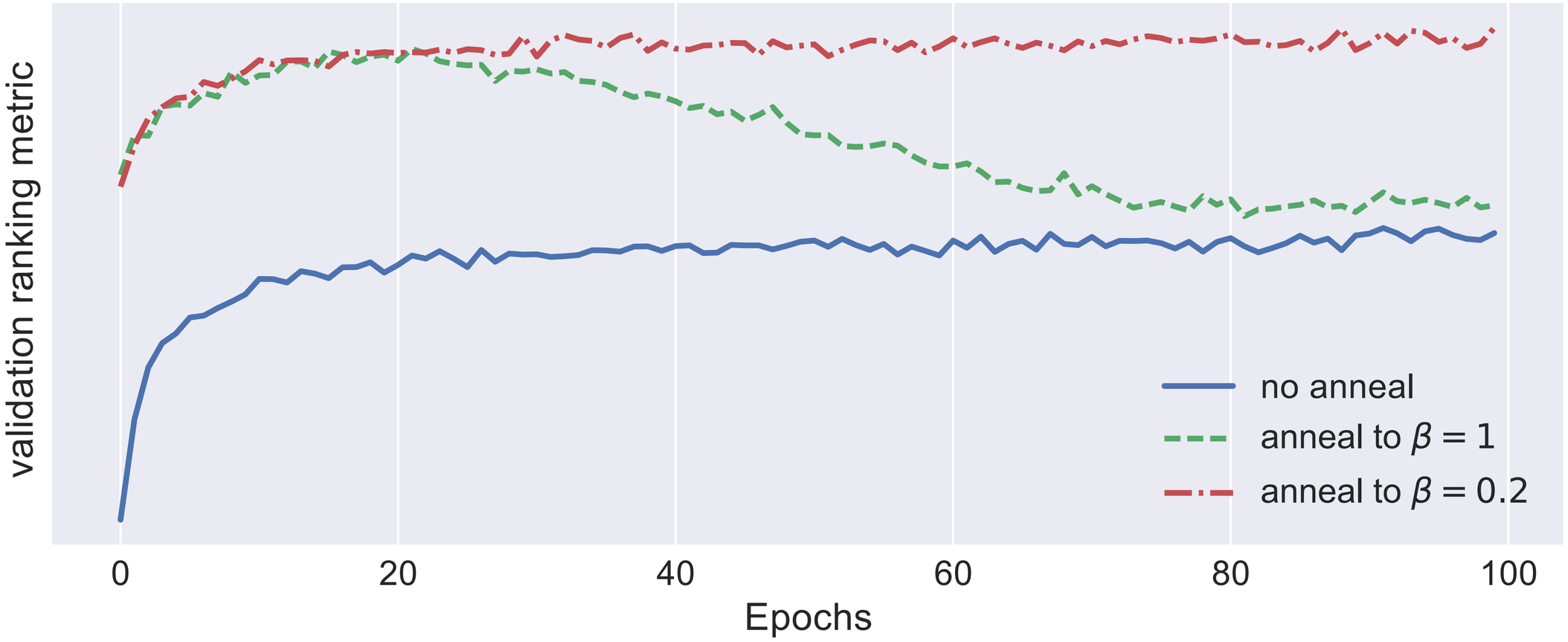
$$\mathbb{E}_{q(\mathbf{z}\mid\mathbf{x})}\left[\log p(\mathbf{x}\mid\mathbf{z})\right] - \beta \cdot \mathrm{KL}(q(\mathbf{z}\mid\mathbf{x})\|p(\mathbf{z}))$$

(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 β

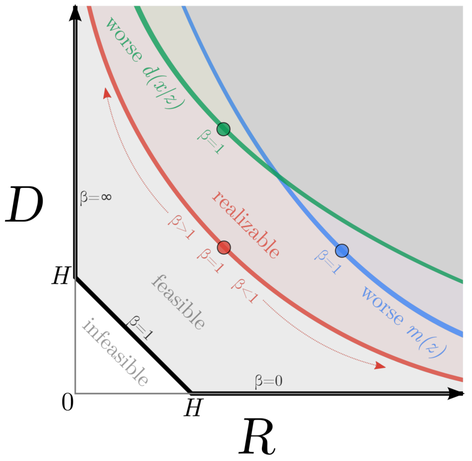


Bowman et al., Generating sentences from a continuous space, 2015

Training VAEs

$$\mathbb{E}_{q(\mathbf{z}\,|\,\mathbf{x})}\left[\log p(\mathbf{x}\,|\,\mathbf{z})\right] - \beta \cdot \mathrm{KL}(q(\mathbf{z}\,|\,\mathbf{x})\|p(\mathbf{z}))$$

- 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, β-VAE, rate-distortion analysis



Jaakkola et al., Maximum entropy discrimination, 2000
Chen et al., Variational lossy autoencoders, 2016
Higgins et al., β-VAE: Learning basic visual concepts with a constrained variational framework, 2016
Alemi et al., Deep variational information bottleneck, 2017
Alemi et al., Fixing a broken ELBO, 2018

Me (2018): Why VAEs (or rather, Bayesian?)

- Generalize linear latent factor models
  - Recover matrix factorization/LDA 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

Blei et al., Latent Dirichlet allocation, 2003
Salakhutdinov & Mnih, Probabilistic matrix factorization, 2008

Me (2020): What really makes it work?

- No iterative procedure required to rank all the items given a user's watch history
  - Only need to evaluate inference and generative functions

The importance of amortization is not VAE-specific

+ dropout

(We should also be aware of the amortization gap)

Me (2020): What really makes it work?

- Generalize linear latent factor models
  - Recover matrix factorization/LDA as a special linear case

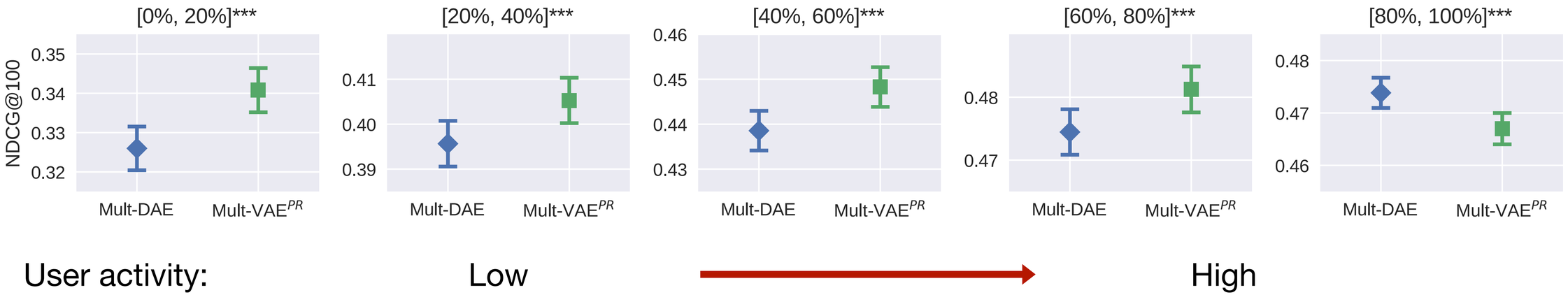
Trade off inference quality with
model complexity

Me (2020): What really makes it work?

- RecSys is more of a “small data” than a “big data” problem

…

Me (2020): Is Bayesian the answer?



- The prior regularization does prevent the model from crazy extrapolation with little data
  - Inadvertently hurt performance for more active users
- “Is RecSys a ‘big data’ or ‘small data’ problem?” is not exactly the right question

What do I mean by that?

- Thinking in terms of “small data” (v.s. “big data”), Bayesian approach is certainly among the answers
- But this question is missing the big picture: Recommender System is a system of interaction
  - Our goal is “interventional”, i.e., to help low-activity users discovery new content so that they become more active
- During interaction, taking uncertainty into account is important

Does it work well?

Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches

Maurizio Ferrari Dacrema
Politecnico di Milano, Italy
maurizio.ferrari@polimi.it

Paolo Cremonesi
Politecnico di Milano, Italy
paolo.cremonesi@polimi.it

Dietmar Jannach
University of Klagenfurt, Austria
dietmar.jannach@aau.at

ABSTRACT

Deep learning techniques have become the method of choice for researchers working on algorithmic aspects of recommender systems. With the strongly increased interest in machine learning in general, it has, as a result, become difficult to keep track of what represents the state-of-the-art at the moment, e.g., for top-n recommendation tasks. At the same time, several recent publications point out problems in today's research practice in applied machine learning, e.g., in terms of the reproducibility of the results or the choice of the baselines when proposing new models.

In this work, we report the results of a systematic analysis of algorithmic proposals for top-n recommendation tasks. Specifically, we considered 18 algorithms that were presented at top-level research conferences in the last years. Only 7 of them could be reproduced with reasonable effort. For these methods, it however turned out that 6 of them can often be outperformed with comparably simple heuristic methods, e.g., based on nearest-neighbor or graph-based techniques. The remaining one clearly outperformed the baselines but did not consistently outperform a well-tuned non-neural linear ranking method. Overall, our work sheds light on a number of potential problems in today's machine learning scholarship and calls for improved scientific practices in this area.

1 INTRODUCTION

Within only a few years, deep learning techniques have started to dominate the landscape of algorithmic research in recommender systems. Novel methods were proposed for a variety of settings and algorithmic tasks, including top-n recommendation based on long-term preference profiles or for session-based recommendation scenarios [36]. Given the increased interest in machine learning in general, the corresponding number of recent research publications, and the success of deep learning techniques in other fields like vision or language processing, one could expect that substantial progress resulted from these works also in the field of recommender systems. However, indications exist in other application areas of machine learning that the achieved progress—measured in terms of accuracy improvements over existing models—is not always as strong as expected.

Lin [25], for example, discusses two recent neural approaches in the field of information retrieval that were published at top-level conferences. His analysis reveals that the new methods do *not* significantly outperform existing baseline methods when these are carefully tuned. In the context of recommender systems, an in-depth analysis presented in [29] shows that even a very recent neural method for session-based recommendation can, in most cases,

Are we really making much progress?

- “Music in the 60s/70s/80s/etc is so much better…”
- Recommender Systems is a unique field
  - It is an interactive system but often treated as a static one in the literature
  - Good performance in the static setting *sometimes* can still translate to the “real-world” setting
- Counterfactual/off-policy evaluation

“I tried your code and it works.”

–A random stranger

A (Hopefully) Optimistic Prospective

The secular Bayesian

Over the years I came to terms with my Bayesian heritage, and I now live my life as a secular Bayesian. Certain elements of the Bayesian way are no doubt useful: Engineering inductuve biases explicitly into a prior distribution, using probabilities, divergences, information, variational bounds as tools for developing new algorithms. Posterior distributions can capture model uncertainty which can be exploited for active learning or exploration in interactive learning. Bayesian methods often - though not always - lead to increased robustness, better calibration, and so much more. At the same time, I can carry on living my life, use gradient descent to find local minima, use bootstrap to capture uncertainty. And first and foremost, I do not have to believe that my models really exist or perfectly describe reality anymore. I am free to think about model misspecification.

inFERENCe

posts on machine learning,
statistics, opinions on things
I'm reading in the space

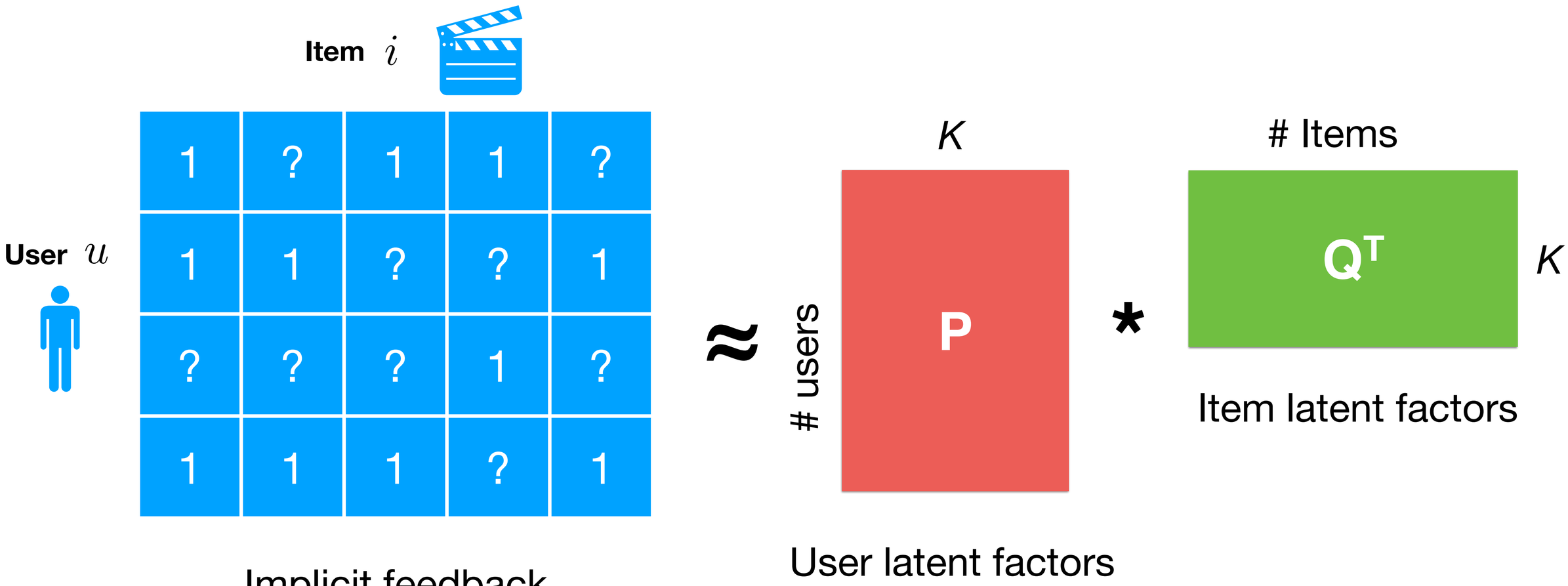
— The secular Bayesian: Using belief distributions without really believing

https://www.inference.vc/

A (Hopefully) Optimistic Prospective

A secular Bayesian's view on various RecSys models

Matrix factorization



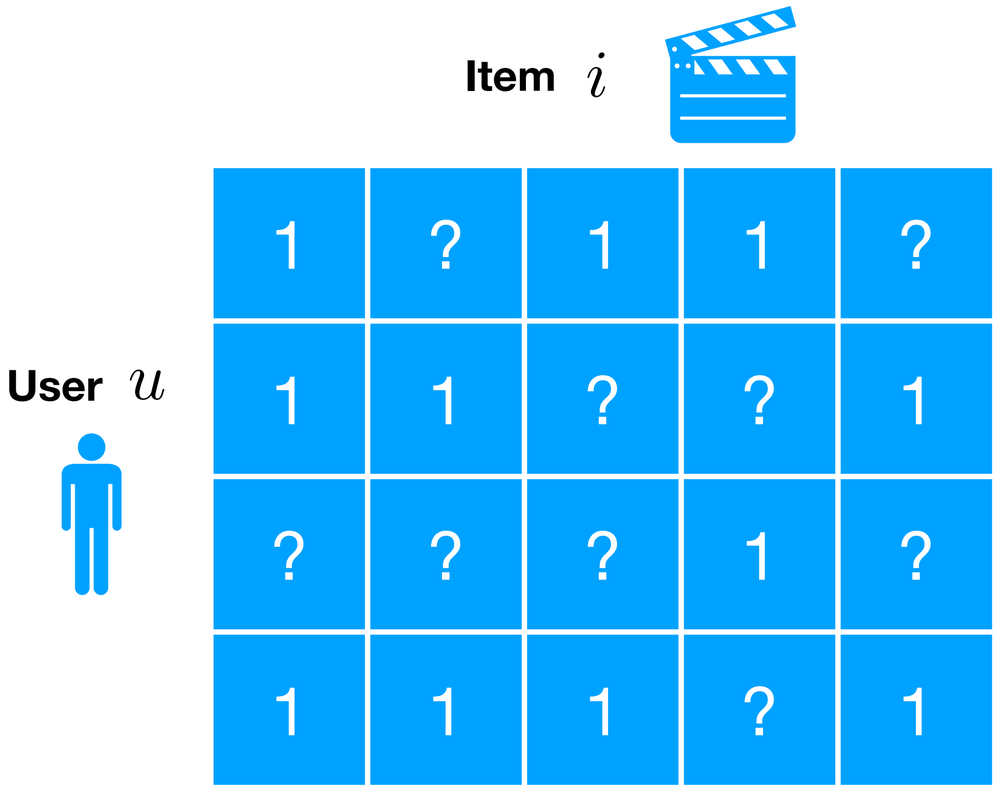
Matrix factorization

$$\mathbf{X} \approx \mathbf{P}\mathbf{Q}^\top$$

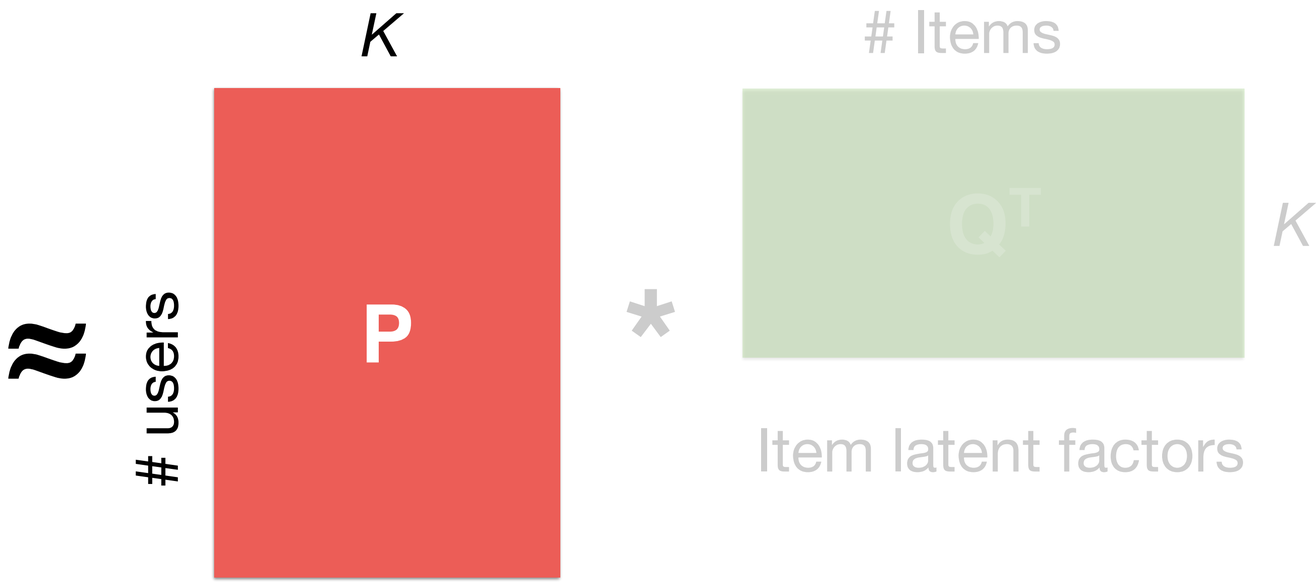
Loss function determines how to measure how “close” the reconstruction is to the original data

- Square loss: Gaussian likelihood
- Binary cross-entropy: Bernoulli likelihood
- Multi-class cross-entropy: Multinomial/Poisson likelihood

Matrix factorization



Implicit feedback



User latent factors

Item latent factors

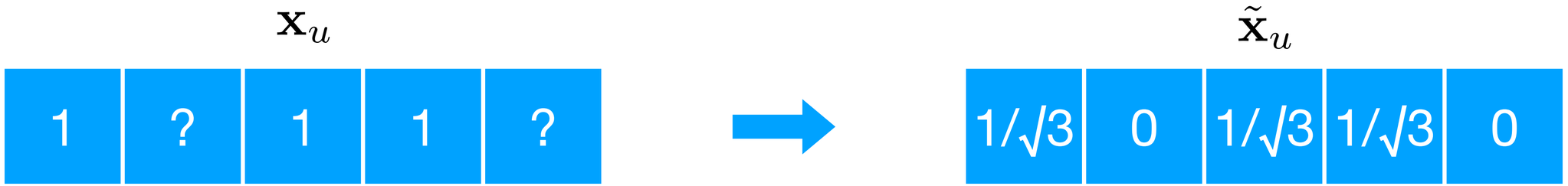
Asymmetric matrix factorization

- Introduce another set of item latent factors **V**
- Construct user factors as “average” embedding of the clicked items

$$\mathbf{p}_u = \frac{1}{\sqrt{|I_u|}} \sum_{i \in I_u} \mathbf{v}_i$$

The set of items user u clicked on

$$= \tilde{\mathbf{x}}_u \mathbf{V}$$



Paterek, Improving regularized singular value decomposition for collaborative filtering, 2007

AMF as a linear autoencoder

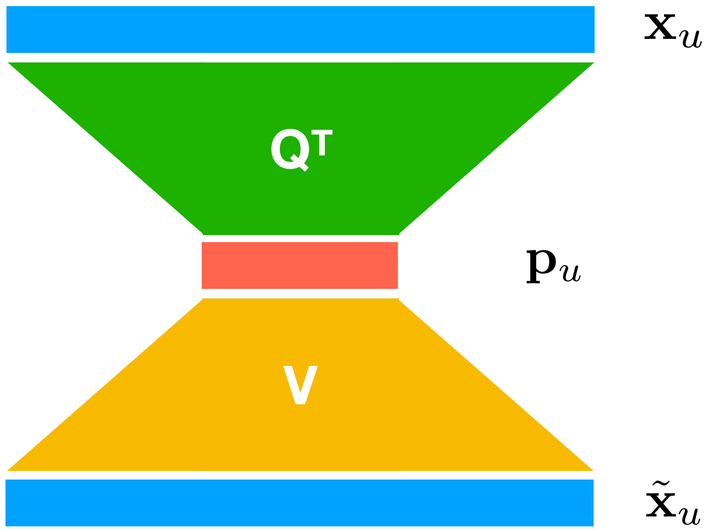
$$\mathbf{x}_u \approx \mathbf{p}_u \mathbf{Q}^\top$$

$$= \tilde{\mathbf{x}}_u \mathbf{V} \mathbf{Q}^\top$$

$$\mathbf{p}_u = g(\mathbf{x}_u)$$

$$\mathbf{x}_u = f(\mathbf{p}_u)$$

$$\dim(\mathbf{x}_u) \gg \dim(\mathbf{p}_u)$$



This is a form of amortization

Make it a non-linear autoencoder

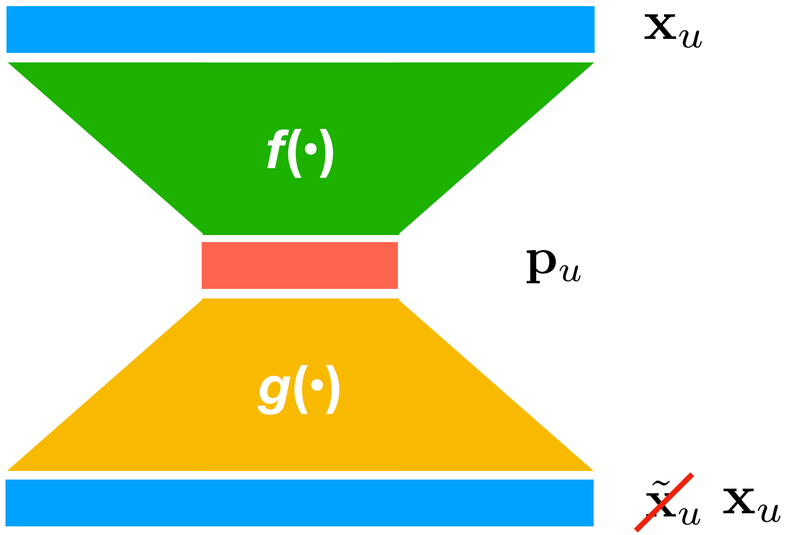
$$\mathbf{x}_u \approx \mathbf{p}_u \mathbf{Q}^\top$$

$$= \tilde{\mathbf{x}}_u \mathbf{V} \mathbf{Q}^\top$$

$$\mathbf{p}_u = g(\mathbf{x}_u)$$

$$\mathbf{x}_u = f(\mathbf{p}_u)$$

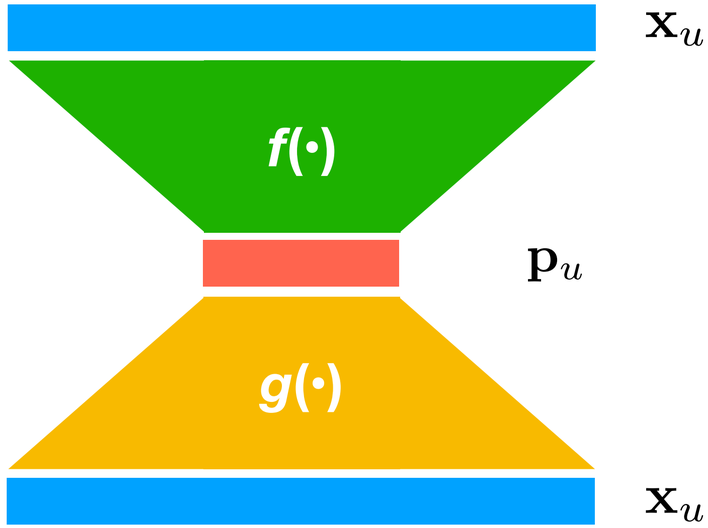
$$\dim(\mathbf{x}_u) \gg \dim(\mathbf{p}_u)$$



This is a form of amortization

Denoising autoencoders

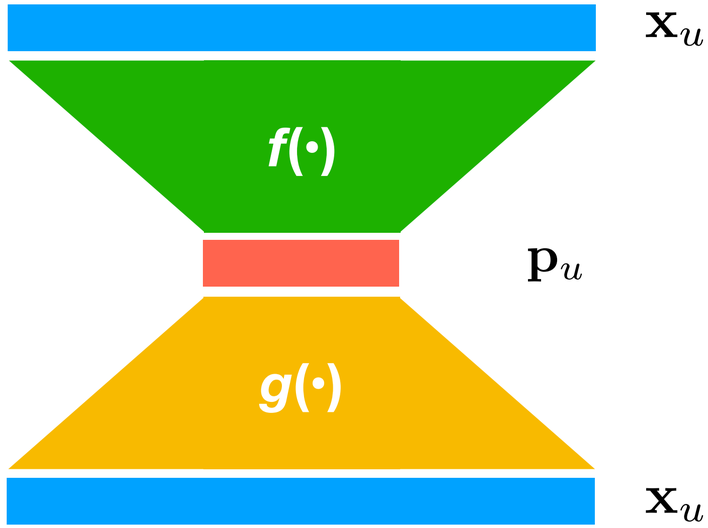
- With non-linear functions, certain loss functions can overfit easily
- Before fitting input $\mathbf{x}_u$ into the model, we corrupt it by adding some noise
  - e.g., randomly setting some of 1's to 0's (dropout)



Vincent et al., Extracting and Composing Robust Features with Denoising Autoencoders, 2008
Wu et al., Collaborative denoising auto-encoders for top-n recommender systems, 2016

Denoising autoencoders

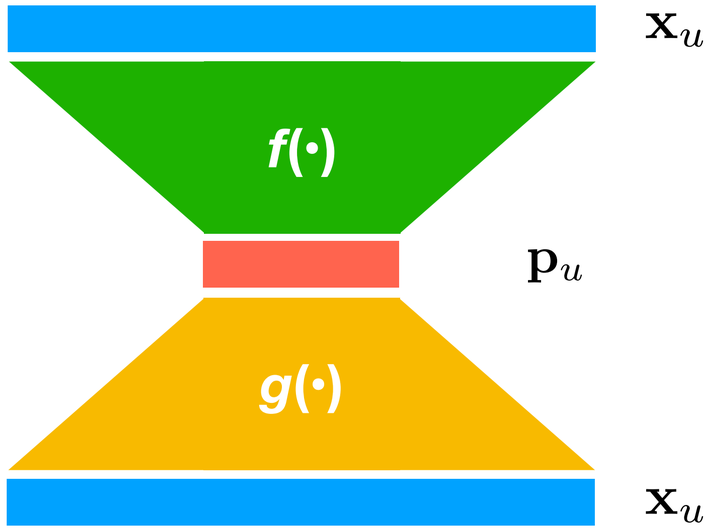
- Dropout is particularly well-suited for recommender systems
- Simulate the actual test scenario:
  - Give part of the click history, predict what else a user would like to click on



Vincent et al., Extracting and Composing Robust Features with Denoising Autoencoders, 2008
Wu et al., Collaborative denoising auto-encoders for top-n recommender systems, 2016

Denoising autoencoders

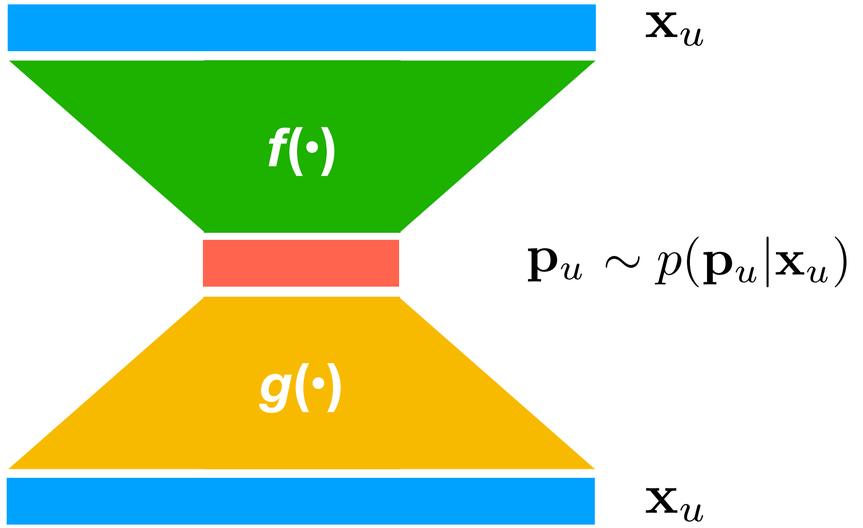
- One can be creative about designing specific dropout probability
  - e.g., to achieve inverse propensity weighting
  - e.g., to simulate other potential test scenarios
- See Steck (2020) for a deeper theoretical discussion



Alaa et al., Deep Counterfactual Networks with Propensity-Dropout, 2017
Steck, Autoencoders that don't overfit towards the identity, 2020

Variational autoencoders

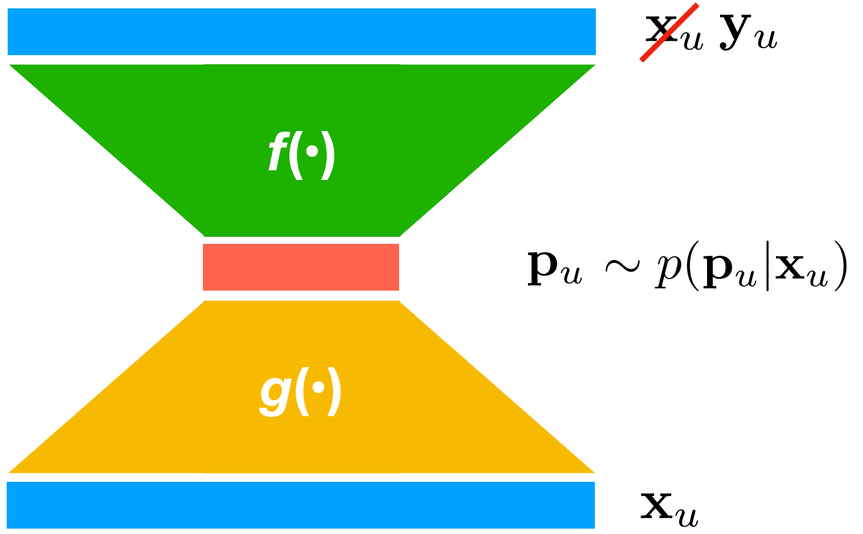
- What if rather than a single point estimate $\mathbf{p}_u$, we want a distribution $p(\mathbf{p}_u|\mathbf{x}_u)$?
- Alternative view to the “Model & Inference” from earlier



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

Next-click prediction model

- Rather than reconstructing the input, we ask the model to predict the next click
- More similar to supervised learning
- Neural language model
  - More sophisticated architectures, e.g., RNN



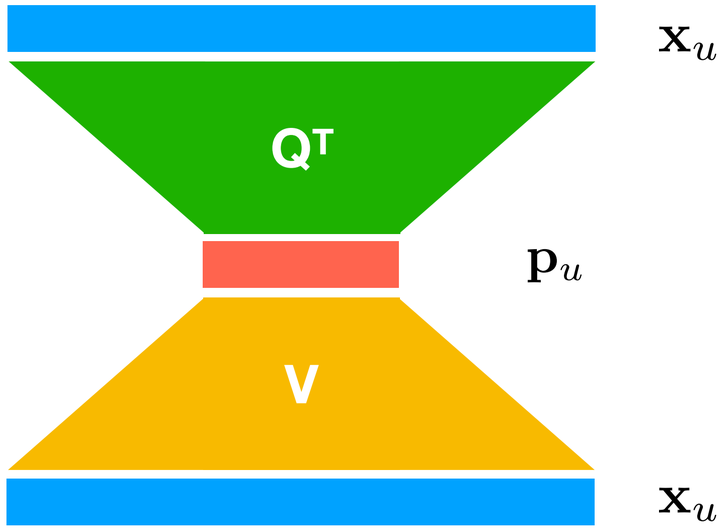
Bengio et al., A Neural Probabilistic Language Model, 2003
Alemi et al., Deep Variational Information Bottleneck, 2017

Back to linear autoencoder

$$\mathbf{x}_u \approx \mathbf{p}_u \mathbf{Q}^\top$$

$$= \tilde{\mathbf{x}}_u \mathbf{V}\mathbf{Q}^\top$$

A low-rank item-to-item similarity matrix $\mathbf{S}$



Back to linear autoencoder

$$\mathbf{x}_u \approx \mathbf{p}_u \mathbf{Q}^\top$$

$$= \tilde{\mathbf{x}}_u \mathbf{V}\mathbf{Q}^\top$$

A full-rank item-to-item similarity matrix $\mathbf{S}$ s.t., $\text{diag}(\mathbf{S}) = \mathbf{0}$

This is a (Gaussian) Markov random field

Ning & Karypis, SLIM: Sparse Linear Methods for Top-N Recommender Systems, 2011
Steck, Embarrassingly Shallow Autoencoders for Sparse Data, 2019

Why full-rank linear models?

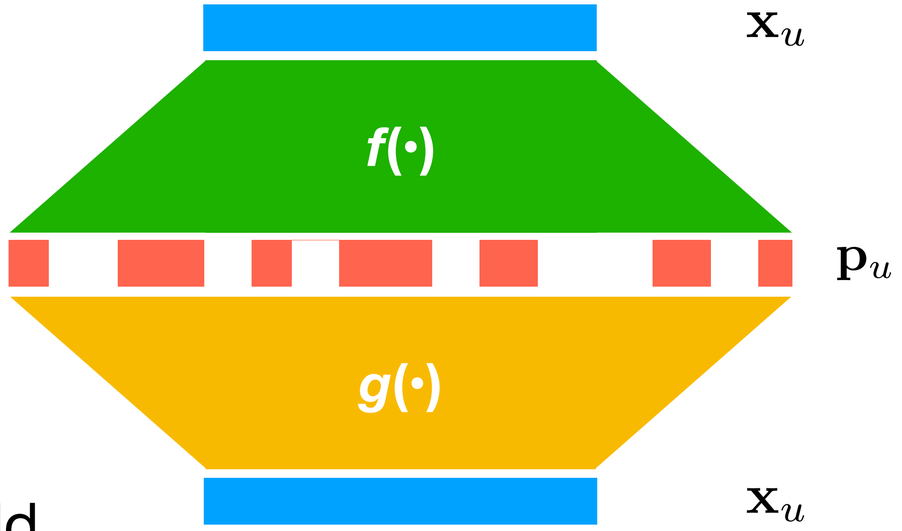
- Linear models are more “parameter efficient”
- Full-rank linear models tend to have much more parameters than a typical neural net used in RecSys
- No surprise that it outperforms many deep learning approaches
  - Especially for more personalized domain, e.g., music

Where can we go from here?

- Sparse full-rank linear model
- “Full-rank” autoencoders

$$\dim(\mathbf{p}_u) = \mathcal{O}(\dim(\mathbf{x}_u))$$

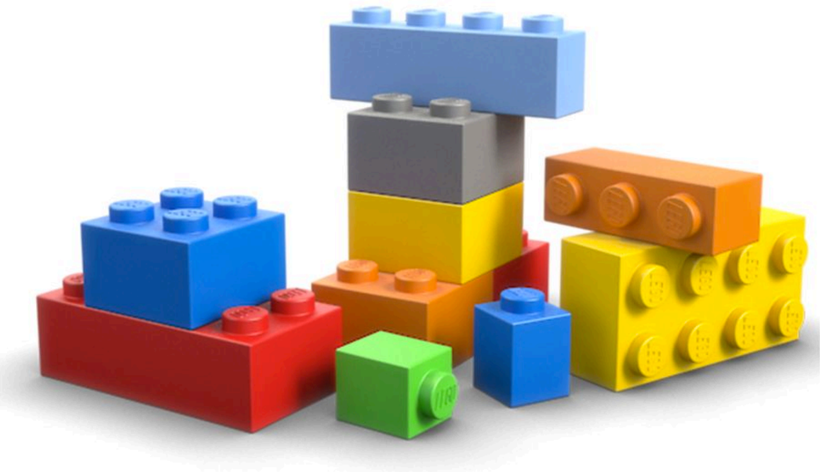
- (Sparse) higher-order Markov random field



Steck, Markov Random Fields for Collaborative Filtering, 2019
Khawar et al., Learning the Structure of Auto-Encoding Recommenders, 2020

Take away as a secular Bayesian

- I don’t think there is one superior model that rules them all
  - VAE is merely “one of them”
- Every model can be viewed through the lens of others. In practice, we can “plug and play”



Conclusion

- There might still be a place to be Bayesian in a largely pragmatic field like Recommender Systems
  - But it might not be how you would imagine coming in
- Through the lens of probabilistic models, we can have a holistic picture about various models, which opens up possibilities for future development

Thanks! Questions?

- There might still be a place to be Bayesian in a largely pragmatic field like Recommender Systems
  - But it might not be how you would imagine coming in
- Through the lens of probabilistic models, we can have a holistic picture about various models, which opens up possibilities for future development