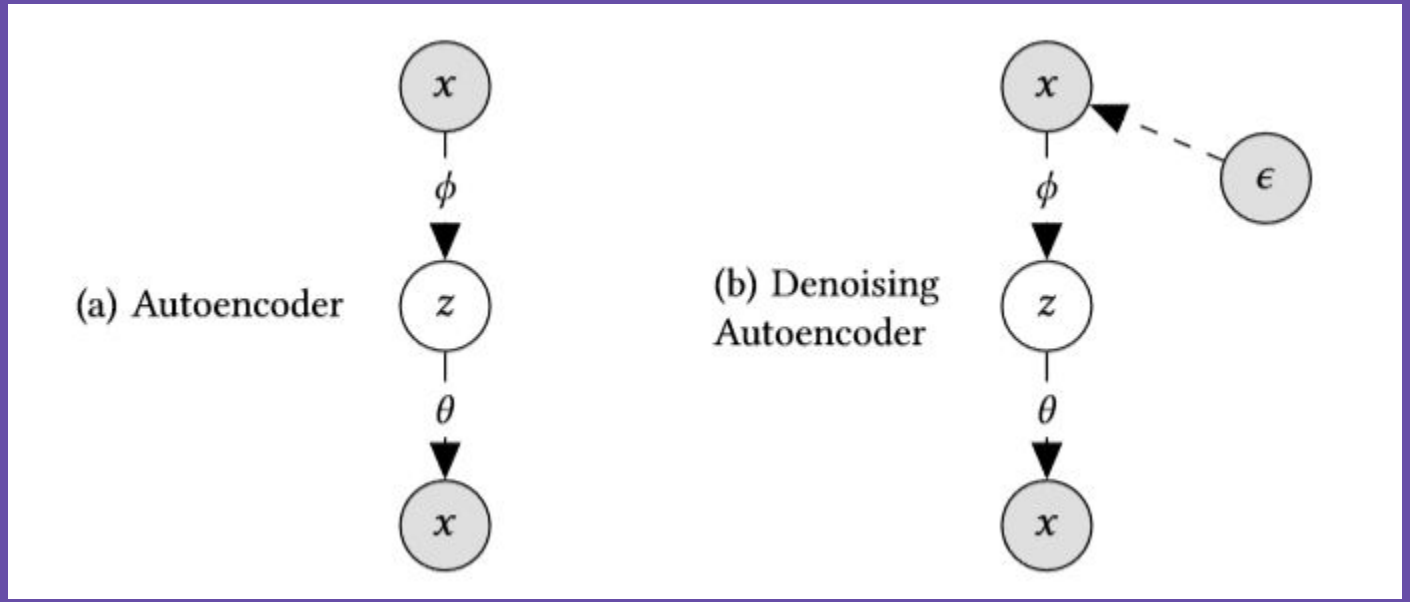
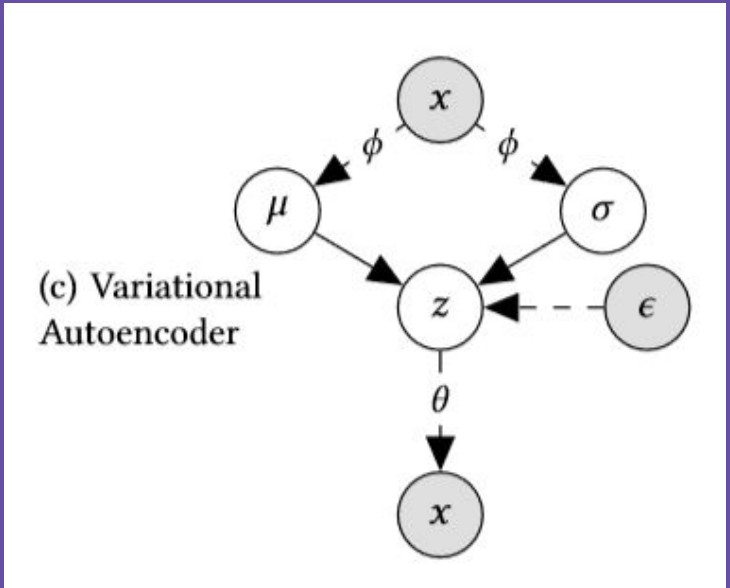
Variational Autoencoders for Collaborative Filtering

Liang, Dawen; Krishnan, Rahul; Hoffman, Matthew; Jebara, Tony.
2018

Autoencoder



Variational Autoencoder



(c) Variational Autoencoder

Modelo probabilístico no lineal

Auto-encoding Variational Bayes

Posterior intratable de la marginal likelihood. Se busca aproximar con distribución obtenida de manera variacional.

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$$

Marginal likelihood **KL-divergencia de la posterior real** **Lower bound**

ELBO: maximizarlo equivale a minimizar KL-div

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z})\right]$$

Auto-encoding Variational Bayes

Truco de re-parametrización permite optimizar fácilmente la ecuación mediante el uso de SGD → Neural Networks.

$$\widetilde{\mathcal{L}}^{B}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \frac{1}{L}\sum_{l=1}^{L}(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)}))$$

$$\text{where} \quad \mathbf{z}^{(i,l)} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{x}^{(i)}) \quad \text{and} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$$

VAE en Collaborative Filtering

X → Matriz de preferencias de usuarios (clicks)

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

f → Multilayer perceptron.

x_u → bag-of-words vector con las preferencias de un usuario.

VAE en Collaborative Filtering

Con ello, la log likelihood de un usuario estará dada por:

$$\log p_\theta(\mathbf{x}_u \mid \mathbf{z}_u) \stackrel{c}{=} \sum_i x_{ui} \log \pi_i(\mathbf{z}_u).$$

Construir el modelo presentado requiere resolver la posterior intratable $p(z_u|x_u)$ → Aproximar con inferencia Bayesiana.

Al asumir $q(z_u)$ normal, deseamos encontrar los parámetros que la definen.

VAE en Collaborative Filtering

El objetivo del encoding será encontrar los parámetros de

$$q_\phi(\mathbf{z}_u \mid \mathbf{x}_u) = \mathcal{N}(\mu_\phi(\mathbf{x}_u), \mathrm{diag}\{\sigma_\phi^2(\mathbf{x}_u)\}).$$

Por su parte, el objetivo del decoder será reconstruir X, mediante la optimización del lower bound → disminución de la KL-div.

$$\mathcal{L}(\mathbf{x}_u; \theta, \phi) \equiv \mathbb{E}_{q_\phi(\mathbf{z}_u \mid \mathbf{x}_u)}[\log p_\theta(\mathbf{x}_u \mid \mathbf{z}_u)] - \mathrm{KL}(q_\phi(\mathbf{z}_u \mid \mathbf{x}_u) \| p(\mathbf{z}_u))$$

Interpretación alternativa del ELBO

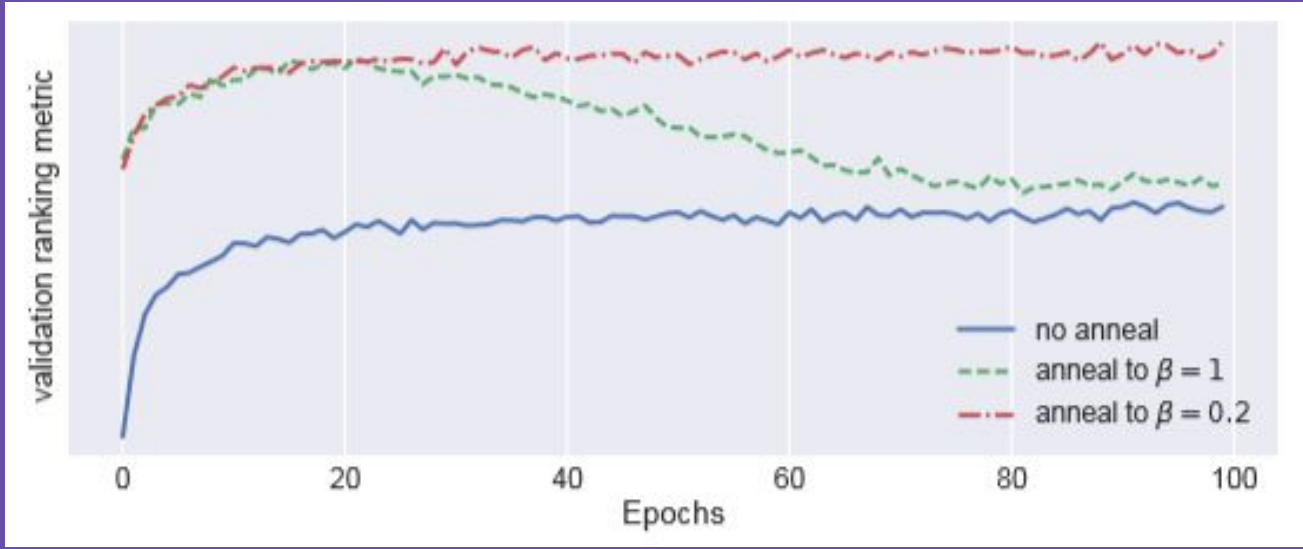
$$\mathcal{L}(\mathbf{x}_u;\theta,\phi) \equiv \mathbb{E}_{q_\phi(\mathbf{z}_u \mid \mathbf{x}_u)}\left[\log p_\theta(\mathbf{x}_u \mid \mathbf{z}_u)\right] - \mathrm{KL}(q_\phi(\mathbf{z}_u \mid \mathbf{x}_u)\|p(\mathbf{z}_u))$$

Error de Reconstrucción **Término de Regularización**

Podemos introducir un **tradeoff** entre la habilidad de generar **ancestral sampling** vs el **desempeño** de ajuste que el modelo alcanza.

$$\mathcal{L}_\beta(\mathbf{x}_u;\theta,\phi) \equiv \mathbb{E}_{q_\phi(\mathbf{z}_u \mid \mathbf{x}_u)}\left[\log p_\theta(\mathbf{x}_u \mid \mathbf{z}_u)\right] - \beta \cdot \mathrm{KL}(q_\phi(\mathbf{z}_u \mid \mathbf{x}_u)\|p(\mathbf{z}_u)).$$

Efecto de Annealing



Autores sugieren mantener $\beta < 1$

Casos de estudio

Tres dataset de estudio: se binariza los datos.

a) **Movie-Lens:** user-movie ratings. Usuarios con al menos 5 películas.
b) **Netflix Prize:** user-movie ratings. Usuarios con al menos 5 películas.
c) **Million Songs Dataset:** user-song play counts. Límite en canciones y usuario.

	ML-20M	Netflix	MSD
# of users	136,677	463,435	571,355
# of items	20,108	17,769	41,140
# of interactions	10.0M	56.9M	33.6M
% of interactions	0.36%	0.69%	0.14%
# of held-out users	10,000	40,000	50,000

Métricas

Para estudiar el potencial del método variacional se compara el desempeño de Mult-VAE con la versión más simple: Denoising Autoencoder (Mult-DAE).

Bajo el objetivo de rankear se evalúan dos métricas:

- **Recall**
- **nDCG**

Se espera comprobar:

- Es óptimo asumir un **likelihood multinomial.**
- En qué casos **Mult-DAE** puede mostrar mejor **desempeño** que **Mult-VAE**

Baselines a comparar.

Trabajo relevantes, incluyendo dos en NN:

- **Weighted Matrix Factorization:** Métodos de MF con pesos.
- **Sparse Linear Methods:** Aprende matrices de similaridad entre ítemes.
- **Collaborative denoising autoencoder:** Modelo en Redes Neuronales. Extiende denoising al agregar un factor a cada usuario.
- **Neural collaborative filtering:** Explora interacciones no lineales mediante Redes Neuronales.

Resultados

(a) ML-20M

	Recall@20	Recall@50	NDCG@100
Mult-VAE^{PR}	**0.395**	**0.537**	**0.426**
Mult-DAE	0.387	0.524	0.419
WMF	0.360	0.498	0.386
SLIM	0.370	0.495	0.401
CDAE	0.391	0.523	0.418

(b) Netflix

	Recall@20	Recall@50	NDCG@100
Mult-VAE^{PR}	**0.351**	**0.444**	**0.386**
Mult-DAE	0.344	0.438	0.380
WMF	0.316	0.404	0.351
SLIM	0.347	0.428	0.379
CDAE	0.343	0.428	0.376

Desempeño **superior a** métodos de **“estado del arte”** en ambas aproximaciones de estudio: Mult-VAE y Mult-DAE.

Resultados

(c) MSD

	Recall@20	Recall@50	NDCG@100
Mult-VAE^{PR}	**0.266**	**0.364**	**0.316**
Mult-DAE	**0.266**	**0.363**	0.313
WMF	0.211	0.312	0.257
SLIM	—	—	—
CDAE	0.188	0.283	0.237

Método SLIM no compiló en 2 semanas. **Desempeño similar** en dataset muy grande.

Resultados

NCF **no logra resultado competitivo** en dataset estudiados.

Se compara su desempeño en dataset más pequeños, los usados originalmente en su publicación.

Mult-DAE **supera** al método incluso sobre un modelo pre-entrenado.

(a) ML-1M

	NCF	NCF (pre-train)	Mult-DAE
Recall@10	0.705	**0.730**	0.722
NDCG@10	0.426	**0.447**	0.446

(b) Pinterest

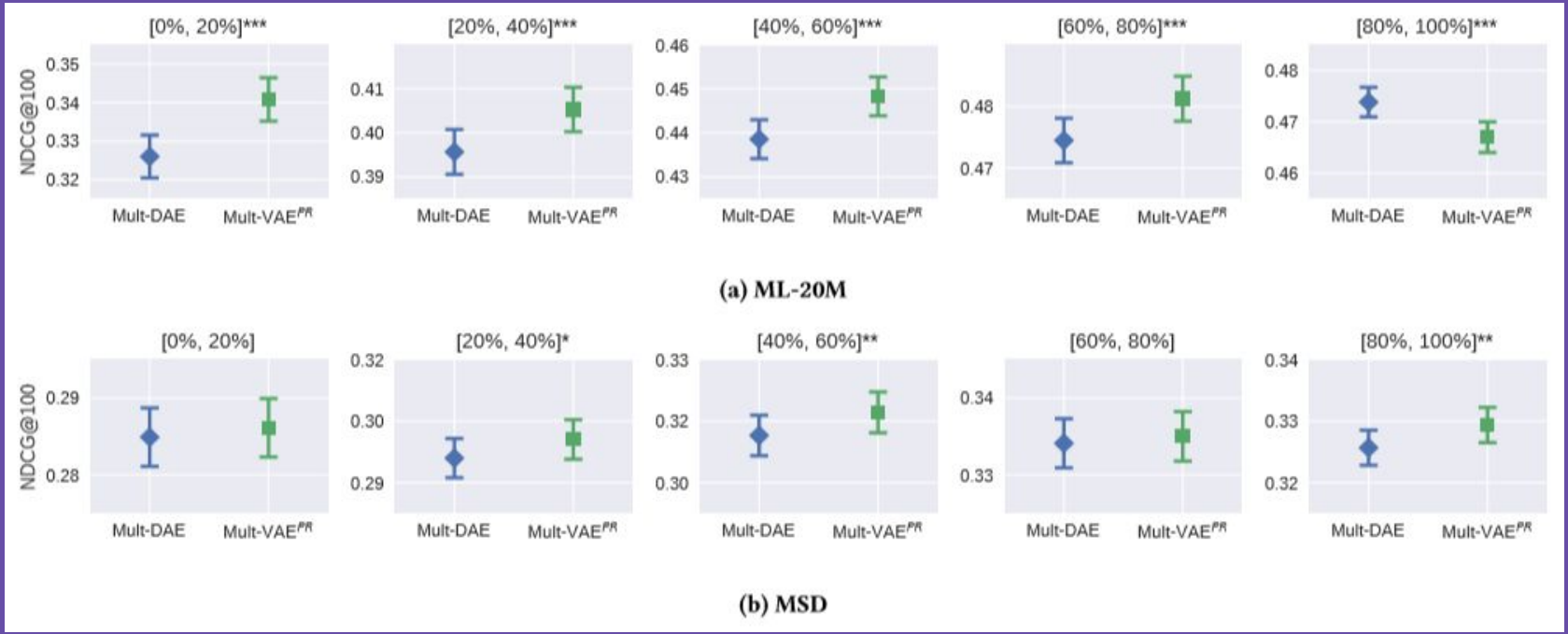
	NCF	NCF (pre-train)	Mult-DAE
Recall@10	0.872	0.880	**0.886**
NDCG@10	0.551	0.558	**0.580**

Comparando diversos likelihoods.

	Recall@20	Recall@50	NDCG@100
Mult-VAE^{PR}	**0.395**	**0.537**	**0.426**
Gaussian-VAE^{PR}	0.383	0.523	0.415
Logistic-VAE^{PR}	0.388	0.523	0.419
Mult-DAE	**0.387**	**0.524**	**0.419**
Gaussian-DAE	0.376	0.515	0.409
Logistic-DAE	0.381	0.516	0.414

Multinomial likelihood parece ser óptimo para modelar **implicit feedback** data, seguido de Logistic likelihood**.**

¿Cuándo Mult-VAE es mejor que Mult-DAE?



Conclusiones

- Multinomial likelihood bien situada para representar datos de implicit feedback.
- Una nueva interpretación del objetivo de VAE permite la inclusión de una regularización adicional.
- Mult-VAE y Mult-DAE tienen mejores desempeños que los baseline. Aproximación bayesiana más robusta.

Referencias

1. Liang, D. et al. (2018). Variational Autoencoders for Collaborative Filtering
2. Kingma, D. et al. (2014). Auto-Encoding Variational Bayes
3. Ning, X. and Karypis, G. (2011). Slim: Sparse linear methods for top-n recommender systems
4. He, X. et al. (2017). Neural collaborative filtering
5. Wu, Y. et al. (.2016). Collaborative denoising auto-encoders for top-n recommender systems.