Exploring the Capabilities of Classifier-Free Guidance in Recommendation Tasks

Master's Defense

Noah Buchanan

Dept. Of EECS, University of Arkansas

Thesis Advisor: ***Dr. Susan Gauch***

Committee Members: ***Dr. Brajendra Panda, Dr. Lu Zhang***

Dept. Of CSCE, University of Arkansas

Outline

- Motivation and Main Objectives
- Related Work
- Our Approach
- Experiments
- Results
- Conclusion
- Future Work

Motivation

- Generative Recommenders
  - Generative Recommenders infer user interaction probabilities for datasets involving:
    - Reviews of movies, shows, e-commerce
  - Primarily based on VAE and GAN implementations
  - Important task but limited by VAE and GAN limitations
  - Limited research into few-shot and zero-shot scenarios

Motivation

- DiffRec
  - DiffRec is a recent diffusion recommender system
    - Generative recommender
  - Shows promising performance as the first of a few
  - Denoising network architecture remains largely unexplored
  - Little to no zero-shot or few-shot capabilities reported
  - Pseudo-guidance only

Motivation

- Classifier-Free Guidance
  - Classifier-free guidance is a recent technique
  - We train two models:
    - One conditionally via guidance
    - Another unconditionally
  - These two models share parameters
  - Major benefit of this is we do not need a pre-trained classifier
  - Guidance = personalization

Motivation

- Zero Shot and Few Shot
  - An extremely common and ever-present problem for recommender systems
  - Diffusion recommender systems have no current work done for this scenario
  - Classifier-free guidance could provide us with this capability

Main Objectives

- Improve the performance of diffusion recommender systems using classifier-free guidance
- Contribute to the discussion of denoising network architecture for diffusion recommender systems
- Contribute to the few-shot zero-shot problems

Related Work

- VAE and GAN's introduced as a type of generative AI some years after AI recommendation is born. *Kingma, Welling, 2013. Goodfellow et al., 2014.*
- The combination of generative AI and AI recommendation was introduced shortly after using both VAE and GAN's to create some of the first generative recommenders. *Liang et al., 2018. Gao et al., 2020.*

Related Work

- First paper to introduce diffusion models in 2015, aimed to attack the weak points of VAE and GAN's. *Sohl-Dickstein et al., 2015.*
- First major adoption of diffusion following the release of this paper. *Ho et al., 2020.*
- Diffusion dominated image generation and became the de-facto state-of-the-art in class-conditional generation. *Ho et al., 2021. Dhariwal, Alex, 2021.*

Related Work

- As a result of this mass success and the ongoing research into generative AI, we find ourselves at the intersection following the first diffusion recommender system. *Wang Wenjie et al., 2023.*
- Other implementations followed after the introduction of diffusion-based recommender systems. *Wang Yu et al., 2023. Z. Li et al., 2023. Lin et al., 2023.*

Related Work

- Classifier guidance introduced previously, modifies diffusion models to guide them becoming class-conditional. *Dhariwal, Alex, 2021. Salimans et al., 2016.*
- Classifier-free guidance attempts to have the same effect as classifier guidance but without a dedicated classifier. *Ho et al., 2022.*

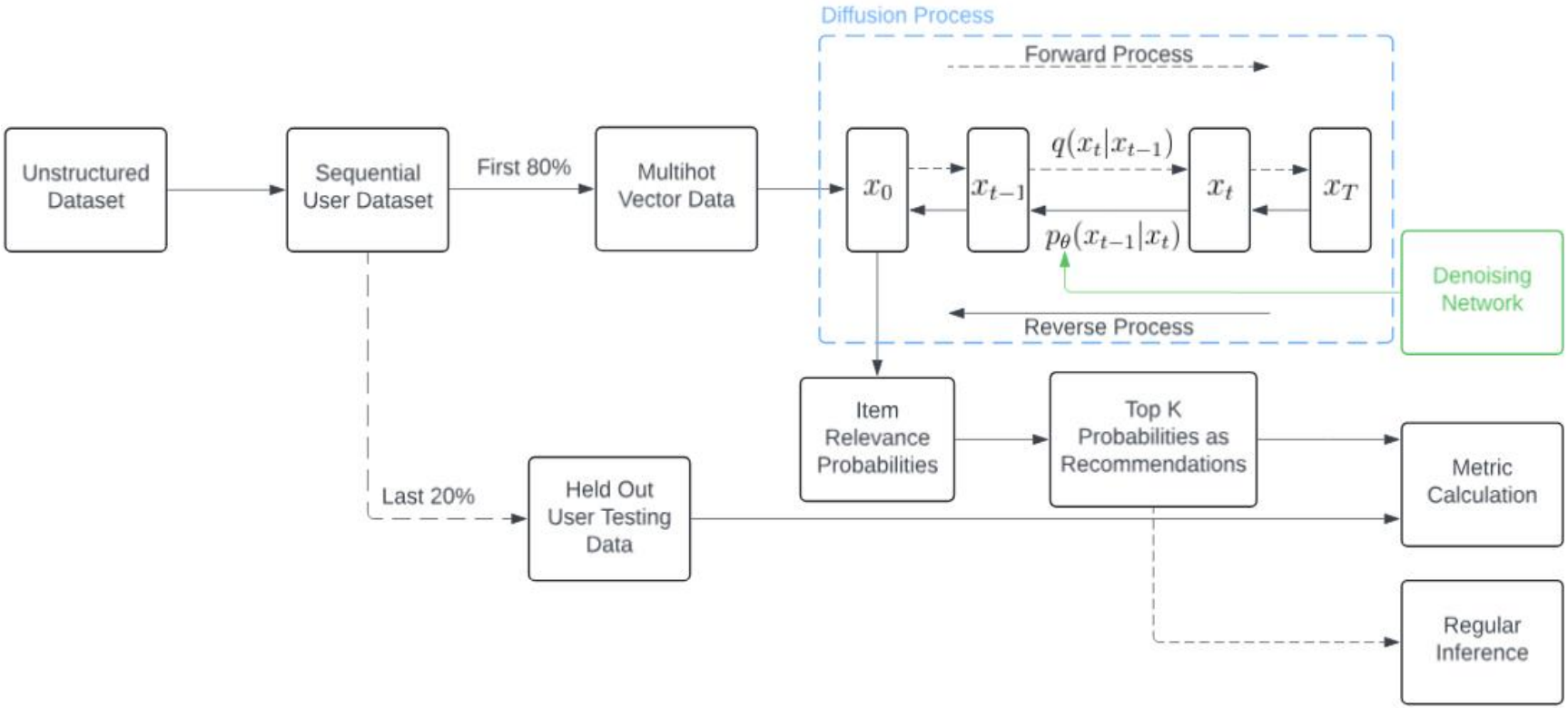
Observations from Related Work

- Diffusion models are a new type of generative AI dominating image generation
- VAE and GANs are capable of recommendation
- Diffusion recommender systems are introduced via the previous two observations
- None of the diffusion recommender systems implement classifier-free guidance

Our Approach

- Our approach consists of two major components
  - The first is the diffusion process
    - Remains relatively the same
    - Consists of forward and reverse diffusion
  - The second is the denoising network
    - Utilized within the reverse diffusion process
    - Core to the idea of diffusion

Our Approach



Our Approach

- Forward Diffusion Process
  - Forward process can be described as follows:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t, \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

  - Normal/Gaussian distribution
  - Both variables present in normal distribution are determined by the scheduler depending on the timestep
    - Mean
    - Variance

Our Approach

- Forward Diffusion Process
  - Entire amount of noise for the forward pass can be described as:

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|q_{t-1})$$

  - We can use the reparameterization trick to calculate the noise at any step:

$$\mathcal{N}(\mu, \sigma^2) = \mu + \sigma * \epsilon \qquad \alpha_t = 1 - \beta_t$$

$$q(x_t|x_0) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon = \mathcal{N}(x_t, \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I)$$

Our Approach

- Reverse Diffusion Process
  - Conversely, we can describe a step in the reverse process as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}, \mu_\theta(x_t, t), \sum_\theta(x_t, t))$$

  - Which translates to:

$$x_{t-1} = \mathcal{N}(x_{t-1}, \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_T}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)), \sqrt{\beta_t\epsilon}$$

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}}}\epsilon_\theta(x_t, t)) + \sqrt{\beta_t\epsilon}$$

UNIVERSITY OF ARKANSAS

Our Approach

- Adding Classifier-Free Guidance
  - The original implementation cannot use classifier-free guidance
    - Guidance through the semblance of the data left after noise is applied
    - Authors argued this mimicked the noisy nature of real user interactions
  - In order to implement classifier-free guidance we preserve data
    - The data after the forward pass and the un-noised version are both fed to the denoising network

Our Approach

- Adding Classifier-Free Guidance
  - Training the model to be capable of both conditional and un-conditional generation requires dropout
    - Zero out 20% of each batch of guidance
  - We have some options for applying this guidance
    - Guidance via concatenation
    - Guidance via cross-attention

UNIVERSITY OF ARKANSAS

Our Approach

- Adding Classifier-Free Guidance
  - This implementation used guidance via concatenation
  - At this point we can put it all together:
    - Fully noise batch in forward diffusion process
    - Keep un-noised iteration of batch (guidance)
    - Create zero-gradients for 20% of samples in guidance
    - Concatenate guidance and batch
    - Feed into denoising network to receive enhanced noise predictions

UNIVERSITY OF ARKANSAS

Our Approach

- Denoising Network Architecture
  - The architecture is relatively simple
    - Does not follow an autoencoder structure
    - Concatenate guidance and data batch → run through a custom embedding layer of varying size depending on dataset size → vector output
  - Hyperbolic tangent activation functions
  - Data batch normalization before concatenation

Experiments

- Datasets
  - A varying size of datasets were used
    - Some provided by the authors of the original work
    - Some provided by Amazon review data
  - MovieLens 1 Million Dataset
    - 1 million movie ratings
    - 6000 users
    - 4000 movies

Experiments

- Datasets
  - The datasets have clean and noisy configurations
    - Clean = reviews > 4 on a 5 point rating scale only
    - Noisy = keep all reviews regardless of rating
  - Noisy data is less ideal
  - The results for the noisy Yelp dataset will not be included due to hardware requirements

UNIVERSITY OF ARKANSAS

Experiments

- Datasets
  - MovieLens and Yelp both have numeric ratings
    - We know what constitutes a "hit"
    - More difficult if we don't have numeric ratings to work with
  - The Amazon datasets we used don't have numeric ratings
    - We instead make a naïve assumption that the data is clean
  - In practice we experienced no problems as a result of this naïve approach

Experiments

- Datasets

Dataset	Users	Items	Total Reviews
ML-1M Clean	5949	2810	403,277
ML-1M Noisy	5949	3494	429,993
Yelp Clean	54,574	34,395	1,014,486
Amazon Kitchen	11,566	7,722	100,464
Amazon Beauty	3,782	2,658	38,867
Amazon Office	1,719	901	21,466
Amazon Toys	2,676	2,474	26,850
Amazon VG	5,435	4,295	60,497

Experiments

- Pre-Processing
  - Interactions are sorted by user and then timestamp
    - All ratings 3 and below will already have been dropped
  - Index 1-k belong to user 1's k interactions at this point
  - The last 20% of each interaction sequence is removed and placed into the testing dataset
  - During runtime we transform this format into a sparse format
    - Multi-hot vectors to represent item interactions for a user
    - 3 and below AND missing reviews for items all represented by 0's

Experiments

- Experimental Configurations
  - Not all of the changes made were entirely beneficial
  - Rule of thumb was found for denoising network architecture
    - Less was more
  - Lots of changes did not remain in the final implementation
    - Cross-attentive conditioning as opposed to concatenation
    - Self-attention on the input sequence
    - More layers and more neurons
    - Custom embedding model for the guidance prior to concatenation

Experiments

- Experimental Configurations
  - Numerical rating aware system also resulted in a failure
    - Multi-hot vector input but with actual ratings
  - Did not perform was well as expected
  - As a result, settled on a seemingly simple implementation
  - Achieved far better results than any other configuration tested

Results

- Training Configuration
  - For all datasets we save the highest-performing model on testing data based on nDCG @ K = 10
  - All metrics described we calculated based on K @ [1,5,10,20]
  - Trained on each dataset until the performance on testing data stopped improving for nDCG @ 10
  - Hyperparameter configuration for both including embedding size are the same, *p_uncond* set @ 0.2

Results

➢ Training Results

Dataset	ML-1M_Clean Original	ML-1M_Clean Ours	ML-1m_Noisy Original	ML-1m_Noisy Ours	Yelp_Clean Original	Yelp_Clean Ours	Amazon Kitchen Original	Amazon Kitchen Ours
Preciscion @ [1,5,10,20]	0.0864	0.0925	0.0299	0.0463	0.028	0.0261	0.0035	0.0052
	0.0792	0.0861	0.0337	0.036	0.0222	0.208	0.0021	0.0024
	0.0716	0.0783	0.0346	0.0357	0.019	0.0187	0.0014	0.0016
	0.0654	0.0682	0.0345	0.0355	0.0161	0.0161	0.001	0.001
Recall @ [1,5,10,20]	0.0069	0.0092	0.0029	0.0061	0.0048	0.0048	0.0013	0.0039
	0.0318	0.0396	0.0151	0.0222	0.0185	0.0177	0.0042	0.0056
	0.0549	0.066	0.0304	0.037	0.0312	0.0311	0.0062	0.0068
	0.0972	0.1166	0.0555	0.0657	0.0514	0.0519	0.0089	0.0079
nDCG @ [1,5,10,20]	0.0864	0.0925	0.0299	0.0463	0.028	0.0261	0.0035	0.0052
	0.0833	0.09	0.0349	0.0418	0.0267	0.0253	0.0035	0.0055
	0.0843	0.0932	0.0397	0.0458	0.0294	0.0288	0.0041	0.0059
	0.0944	0.1056	0.0483	0.0557	0.0356	0.0353	0.005	0.0064
MRR @ [1,5,10,20]	0.0864	0.0925	0.0299	0.0463	0.028	0.0261	0.0035	0.0052
	0.1541	0.1639	0.0678	0.0827	0.0529	0.0494	0.0053	0.007
	0.1706	0.1827	0.0834	0.0973	0.0604	0.0577	0.0057	0.0074
	0.1815	0.1933	0.0938	0.109	0.0662	0.0635	0.0061	0.0077

Dataset	Amazon Beauty Original	Amazon Beauty Ours	Amazon Toys Original	Amazon Toys Ours	Amazon Office Original	Amazon Office Ours	Amazon VG Original	Amazon VG Ours
Preciscion @ [1,5,10,20]	0.0093	0.0106	0.0112	0.0075	0.0029	0.0058	0.0147	0.0129
	0.0087	0.009	0.0045	0.003	0.0035	0.0052	0.009	0.0109
	0.0083	0.0081	0.0032	0.0026	0.0026	0.0061	0.009	0.0084
	0.0062	0.0067	0.002	0.0023	0.0017	0.0044	0.007	0.0071
Recall @ [1,5,10,20]	0.0038	0.0042	0.0042	0.0047	0.0004	0.001	0.0082	0.0072
	0.0164	0.0171	0.0097	0.0064	0.0074	0.0065	0.0232	0.0263
	0.0344	0.0318	0.0115	0.0136	0.0108	0.0177	0.0463	0.0412
	0.0519	0.0528	0.0137	0.022	0.0134	0.0257	0.068	0.0699
nDCG @ [1,5,10,20]	0.0093	0.0106	0.0112	0.0075	0.0029	0.0058	0.0147	0.0129
	0.0137	0.0142	0.0094	0.0065	0.0061	0.0061	0.0194	0.0203
	0.0205	0.0192	0.0098	0.0089	0.0072	0.0106	0.0275	0.0254
	0.0258	0.0257	0.0104	0.0115	0.0082	0.0135	0.034	0.0349
MRR @ [1,5,10,20]	0.0093	0.0106	0.0112	0.0075	0.0029	0.0058	0.0147	0.0129
	0.0194	0.0176	0.0154	0.0087	0.0095	0.0109	0.0252	0.0261
	0.0222	0.0214	0.0159	0.0102	0.0105	0.0142	0.0307	0.0294
	0.0236	0.0232	0.0161	0.0114	0.011	0.0154	0.0337	0.0327

Results

- Training Results
  - Outperforms the original In 84 of the 128 categories
  - When our method is beaten it does not appear to be by a sizeable margin
    - The same cannot be said the other way around
  - On smaller sparser datasets, we achieve noticeably better results
    - Higher percentage increase from original to our method when testing only on sparse

UNIVERSITY OF ARKANSAS

Results

- Regular Datasets
  - Ours
    - 0.0349 Precision 36% ⬆
    - 0.0319 Recall 11% ⬆
    - 0.0368 nDCG 7% ⬆
    - 0.0881 MRR 19% ⬆
  - Original
    - 0.0256 Precision
    - 0.0288 Recall
    - 0.0344 nDCG
    - 0.0741 MRR

Results

- Sparse Datasets
  - Ours
    - 0.0047 Precision 104% ⬆
    - 0.0094 Recall 42% ⬆
    - 0.0074 nDCG 45% ⬆
    - 0.0092 MRR 35% ⬆
  - Original
    - 0.0023 Precision
    - 0.0066 Recall
    - 0.0051 nDCG
    - 0.0068 MRR

Conclusion

- In this work, we build upon the intersection of two existing domains using existing techniques that have yet to be adapted to this specific task and provide insight on a topic that has yet to be explored at all
  - Implementing classifier-free guidance which has not been adapted from regular diffusion models into diffusion recommender systems
  - Make one of the first contributions to the architecture of denoising networks for recommender systems specifically

Conclusion

- Our experiments were performed on MovieLens 1 Million, Yelp, and a variety of Amazon review datasets
- Our method showed a tangible improvement over the original in almost all metrics for most datasets
- We provide a doorway to few-shot and zero-shot recommendation for diffusion recommenders
- To the best of my knowledge, this is the first diffusion recommender system to implement classifier-free guidance

Future Work

- Some potential areas that could be expanded on in the future:
  - Numerical rating-aware diffusion recommender system
  - Transformer denoising network
  - Further exploration into few-shot and zero-shot

Thank you for listening!

UNIVERSITY OF ARKANSAS

Questions and Comments?