Don't Stop the Music: Playlist Continuation with Autoencoders

Kevin Wu, Vishal Anand, Kirill Sydykov

Columbia University
COMS 6998: Advanced Machine Learning
Final Project

04/30/2018

Introduction

Objective:

Compare the performance of classical collaborative filtering vs. deep autoencoder-based approaches on a real-world recommendation problem (playlist continuation).

Today's Presentation:

- Methods: Collaborative Filtering, Autoencoders
- Experiment: 2018 Recsys Challenge
- Findings and Discussion

User-Item Matrix Factorization

Given a set of N users and M items, and a binary click matrix $R \in \{0, 1\}^{NxM}$, we can approximate R as the product of separate low-rank user and item matrices $V \in \mathbb{R}^{NxK}$ and $W \in \mathbb{R}^{MxK}$. Given an observed set of clicks Ω, the loss function is written as:

$$\min_{V, W} \left\{ \sum_{(i,j) \in \Omega} [R_{i,j} - (VW^T)_{i,j}]^2 + \lambda(||V||_2 + ||W||_2) \right\}$$

Alternating least-squares technique: two-step iterative process that solves for V and W holding the other matrix constant.

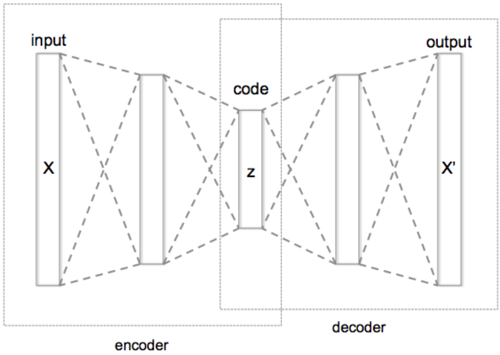
Algorithm:

$$v_i = (W^T W + \lambda I)^{-1} W^T M_i \quad \forall i$$
$$w_j = (V^T V + \lambda I)^{-1} V^T M_j \quad \forall j$$

Iterate until convergence.

Autoencoders: Overview

An autoencoder uses a neural network to learn a non-linear latent representation of the data.



Autoencoders, Part 1: Mult-DAE

(From Liang et al. 2018:)

In the general autoencoder and denoising autoencoder setting, we first calculate a K-dimensional latent representation for each user i, $z_i = g_\phi(x_i)$, where g_ϕ is a single or multi-layer perceptron with one or more nonlinearities (encoder).

Users i's clicks are drawn according to a multinomial distribution with probabilities $\pi(z_i) \propto \exp(f_\theta(z_i))$, where f_θ is another neural network (decoder).

To train this network, we seek to minimize the negative multinomial log-likelihood for user i:

$$\sum_{j \in items} x_{i,j} \log \pi(z_i)$$

Autoencoders, Part 2: Mult-VAE

Assume a generative model that for each user i, samples K-dimensional latent representation z_i with a Gaussian prior, $z_i \sim \mathcal{N}(0, I)$.

In this setting, calculating the encoder, $p_\theta(z_i|x_i)$ becomes intractable (cannot take the integral over z in the evidence $p_\theta(x)$).

Instead we approximate the posterior using $q_\phi(z_i|x_i)$ whose parameters are learned in the VAE.

New objective is to maximize the lower bound of the log likelihood for x_i.

$$\mathbb{E}_{q_\phi(z|x_i)}[\log p_\theta(x_i|z_i)] - KL(q_\phi(z|x_i)||p(z_i))]$$

Dataset: 2018 Recsys Challenge

Million Playlist Dataset (MPD) from Spotify:

- 1,000,000 user-generated playlists with titles
- Song metadata:
  - Artist name, track name, album name
  - Duration
  - In-playlist position
  - Spotify's URI to access more metadata via their API

Preprocessing:

- Compact dataset (sparsity: 0.143%):
  - Songs appearing in at least 100 playlists (69,675)
  - Playlists with at least 50 songs (373,740)
- Sparse dataset (sparsity: 0.034%):
  - Songs appearing in at least 25 playlists (190,897)
  - Playlists with at least 10 songs (919,695)

Neural Network Architecture and Training

Model architecture:

- Mult-DAE: 200-dimensional latent representation layer, dropout at input layer with, tanh nonlinearities, softmax activation for output layer.
- Mult-VAE: Hidden layer of size 600 for encoder and decoder; 200-dimensional latent representation.

Experimental setup:

- Train/test split: 10K playlists reserved for validation and testing for the autoencoder; 20% of click history in these playlists were omitted during training as holdout clicks.
- Evaluation metric: NDCG@100 on validation data.
- Hardware: Nvidia Tesla K80 GPUs.

Results: Million Playlists Dataset

Deep autoencoder methods consistently outperform a matrix factorization baseline.

	Recall@20	Recall@50	NDCG@100
Mult-DAE	**0.250**	**0.370**	**0.376**
Mult-VAE	0.232	0.327	0.346
CF	0.141	0.233	0.232

Table 1: Performance on compact subset of MPD.

Results: Million Playlists Dataset

Autoencoder performance was robust to an increase in the sparsity of the training data.

	Recall@20	Recall@50	NDCG@100
Mult-DAE (1)	0.250	0.370	0.376
Mult-DAE (2)	**0.376**	**0.577**	**0.406**

Table 2: Mult-DAE performance on compact (1) and sparse (2) MPD.

The improvement in evaluation metrics may be attributed to the larger N in the sparser dataset (920,000 vs. 370,000 playlists).

Example Recommendations

Deep latent representations of song preferences are able to represent playlists with seemingly diverse tastes.

```
PLAYLIST TRACKS
Song: Truffle Butter
Artist: Nicki Minaj
Song: Only
Artist: Nicki Minaj
Song: Antidote
Artist: Migos
Song: Wild for the Night
Artist: A$AP Rocky
Song: Everyday We Lit (feat. PnB Rock)
Artist: YFN Lucci
Song: Look At Me!
Artist: XXXTENTACION
Song: 0 To 100 / The Catch Up
Artist: Drake
Song: Super Trapper
Artist: Future
Song: Happy Working Song - From "Enchanted"/Soundtrack
Artist: Amy Adams
Song: Lovebug
Artist: Jonas Brothers
-----
```

```
TOP-RANKED NON-PLAYLIST TRACKS
*Song: Sliding Down The Pole - feat. Too Short
*Artist: E-40
Song: Want to Want Me
Artist: Jason Derulo
*Song: Da' Butt - From The "School Daze" Soundtrack
*Artist: E.U.
*Song: Leave It All To Me (Theme from iCarly)
*Artist: Miranda Cosgrove
*Song: Vacation
*Artist: Dirty Heads
*Song: In Time - Singularity Remix
*Artist: PeaceTreaty
*Song: Tied Up (Freestyle)
*Artist: Flash T.
Song: Kings of Summer - Single Version
Artist: ayokay
Song: Take You There
Artist: Sean Kingston
Song: Left Hand Free
Artist: alt-J
-----
```

(*) denotes a recommended track that also appears in the held-out data for the playlist.

References

Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. Variational Autoencoders for Collaborative Filtering. *arXiv:1802.05814 [cs, stat]*, February 2018.