Music Recommendation

A synopsis by Tre and Kyle

What is Music Recommendation?

Music recommendation is the process of analyzing a user's taste in music, and then selecting similar musical items from a music list to suggest to the user based on their tastes.

Overview

-Music Recommendation is a highly important topic in both consumer products and research domains.

-We will cover the history of this problem, why recommendation systems are important, and how these systems are implemented.

History

Songza and Expert Recommendations

-Streaming company founded in 2007.

-"Music Experts" put together what they deemed to be interesting/coherent playlists.

-This approach does not take into account user's individual preferences.

[1] Sophia Ciocca

Pandora and a Tagging Approach

-Pandora used a tagging system to recommend music to its listeners.

-Songs could be tagged by listeners to describe the various features of the song.

-Pandora could create stations for listeners based on their favorite tags.

[1] Sophia Ciocca

YouTube

2005 - Present

Algorithmic Evolution

-The Echo Nest was a project in MIT's Media Lab.

-Became a funded company, providing services such as music identification and recommendation, playlist creation, audio fingerprinting, and analysis.

-The Echo Nest creates taste profiles based on the listening patterns they notice about a user using various cutting edge methods.

[3] Brian Whitman

Importance/Bigger Picture

Modern Day

The Echo Nest was acquired on March 6, 2014 by Spotify

€49.7 Million

Several other streaming services have adopted academic versions of music recommendation algorithms. These techniques involve Collaborative Filtering, Natural Language Processing, and Raw Audio Modeling.

[1] Sophia Ciocca

Outside of Music

Bigger Picture

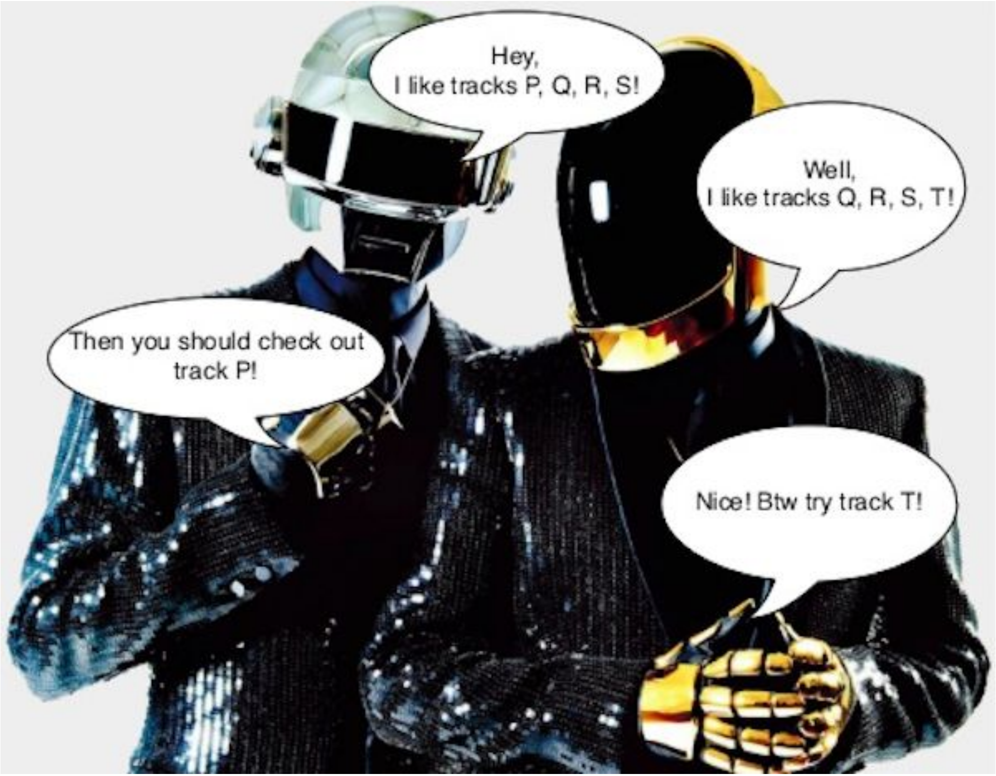
-Allow new content to be consumed in a much more efficient and effective manner.

-Help users find relevant musical artists and composers they weren’t aware of.

This is the future of how you will consume content!

Method 1: Collaborative Filtering

High level process, in a simple conversation



[2] Erik Bernhardsson

Nitty Gritty: Probabilistic Matrix Factorization

$$\min_{x_\star, y_\star} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u ||x_u||^2 + \sum_i ||y_i||^2 \right)$$

$$\begin{aligned} p_{ui} &= I(r_{ui} > 0), \\ c_{ui} &= 1 + \alpha \log(1 + \epsilon^{-1} r_{ui}) \end{aligned}$$

r_ui = the play count for user *u* and song *i*

p_ui = the preference variable defined for each user-song pair

c_ui = a confidence variable; α and ε are hyperparameters

x_u = the latent factor vector for user *u*

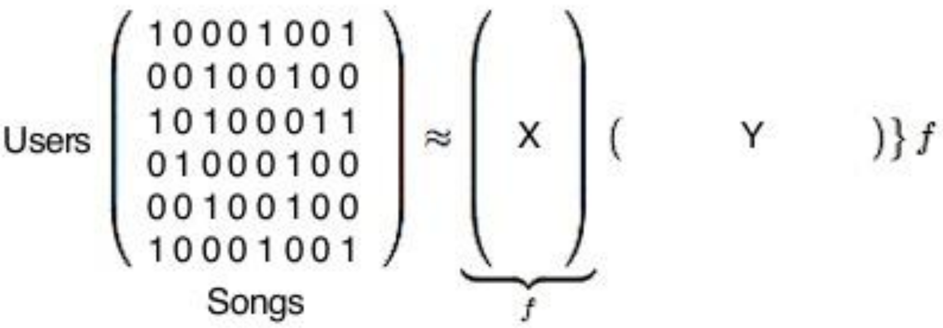
y_i = the latent factor vector for song *i*

λ = is a regularization parameter

This Probabilistic Matrix Factorization Formula is used by Spotify, and run using Python.

[4] Van den Oord et. al.

General Algorithm (Matrix Form)



-**X is a *user vector***, representing one single user's media library, and

-**Y is a *song vector***, representing similarity between songs

-Either of these can be used to recommend an item to another user

[2] Erik Bernhardsson

Naive Idea

-You can simply evaluate two user's user vectors for their similarity (dot product), then scan through the first user's song vector to determine a recommendation for the second user, and vice versa.

-We can determine the user(s) who are most similar to the listener via multiple ways, including distance algorithms and K-Nearest Neighbors

Demonstration

-Tinyurl.com/TreKyleSurvey
-Use First Name Only
-Use Three Distinct Colors

Method 2: Natural Language Processing

Natural Language Processing (NLP) Models

-Services like Spotify constantly search the web for blog posts, articles, and media about songs and artists.

-Echo Nest is the only sub-service to offer this mining feature, utilizing founder Brian Whitman's PhD thesis in data mining [3].

[3] Brian Whitman

NLP Approach

Echo Nest: creates “cultural vectors”. These vectors organize thousands of descriptions, per artist-song combo, daily.

n2 Term	Score	np Term	Score	adj Term	Score
dancing queen	0.0707	dancing queen	0.0875	perky	0.8157
mamma mia	0.0622	mamma mia	0.0553	nonviolent	0.7178
disco era	0.0346	benny	0.0399	swedish	0.2991
winner takes	0.0307	chess	0.0390	international	0.2010
chance on	0.0297	its chorus	0.0389	inner	0.1776
swedish pop	0.0296	vous	0.0382	consistent	0.1508
my my	0.0290	the invitations	0.0377	bitter	0.0871
s enduring	0.0287	voulez	0.0377	classified	0.0735
and gimme	0.0280	something's	0.0374	junior	0.0664
enduring appeal	0.0280	priscilla	0.0369	produced	0.0616

[2] Erik Bernhardsson

NLP Approach

-Once these terms have been categorized with their weight, the remaining process for music recommendation is similar to Collaborative Filtering:

-The terms and weights create a matrix representation of a song, that can be compared with data of other songs to determine if two pieces of music are similar.

Pros and Cons

Pro: This component of music recommendation is a “scale with care” application, real people feeding the algorithm [2]

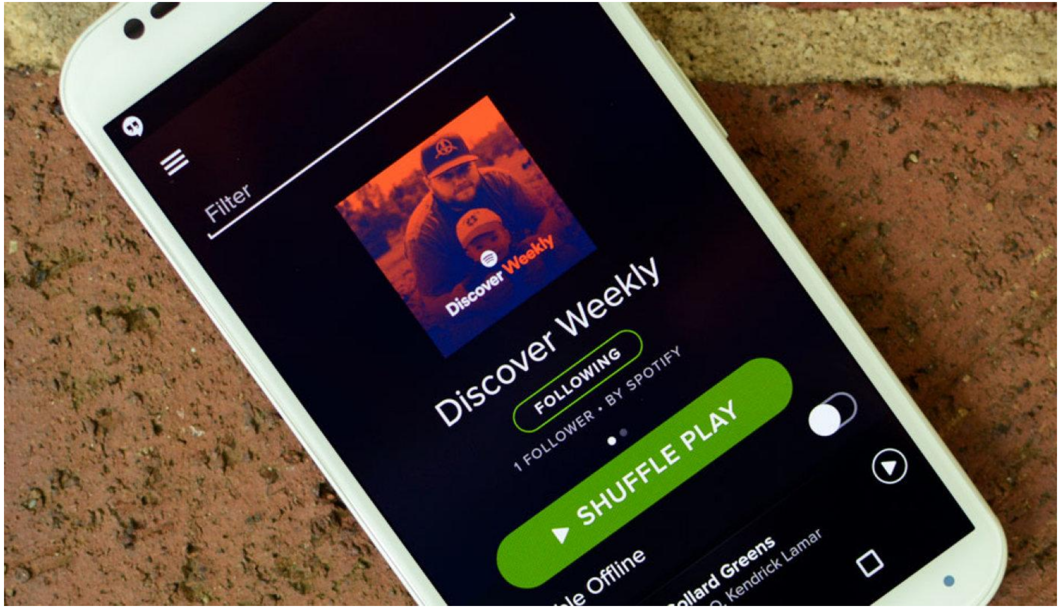
Con: Hard to find smaller artists and songs that might have a stronger culture vector for recommendation [people talk about popular music much more frequently online, that’s why it’s popular]

Method 3:
Audio Modeling

Motivation

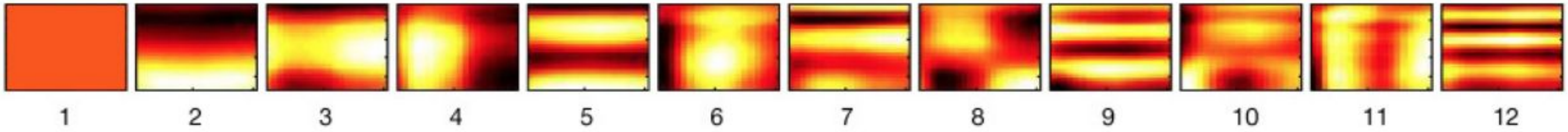
-Recommended playlists need consistency to be fully cohesive and entertaining

-Better single song recommendations above and beyond similarity measures



Feature Extraction

-According to The Echo Nest's documentation, features such as rhythm, pitch, loudness, and timbre are extracted, with much focus given to the timbre domain, as shown below:



12 basis functions for the timbre vector: x = time, y = frequency, z = amplitude

[9] Tristan Jehan

Research Implementations

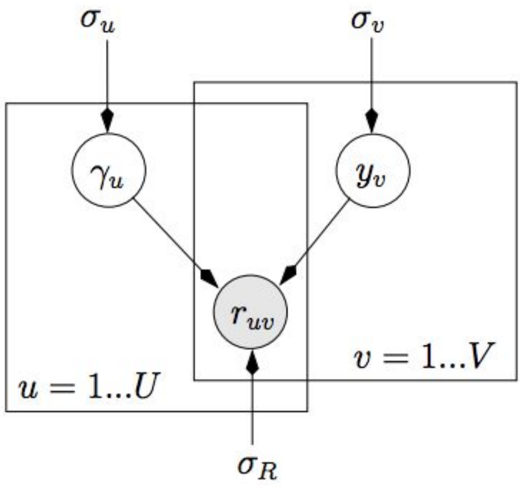
Wang Et. Al [6]

Motivation

-Combine acoustic modeling and collaborative filtering to achieve a good hybrid result

-The hybrid methodology will combine the methods “naive” collaborative filtering and of acoustic feature analysis to create a better recommendation

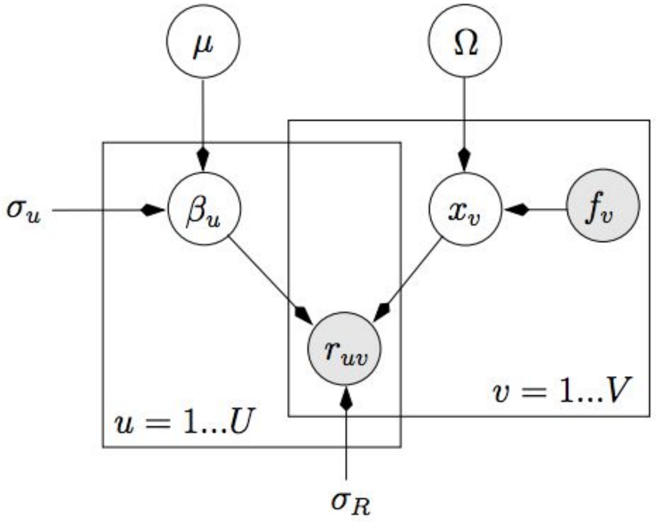
Deep Learning Methodology



Symbol	Description
u	User u
v	Song v
r_{uv}	The rating that user u gives to song v
γ_u	The latent features for u estimated by MF
y_v	The latent features for v estimated by MF
β_u	User u's preference of content features
μ	All users' common preference
x_v	The learnt content features for song v
Ω	The parameters of DBN
$\mathcal{U}, \mathcal{V}$	User and song sets, respectively
U, V	The number of users and songs, respectively
I	All user, song pairs in the training dataset

Table 1: Frequently used symbols

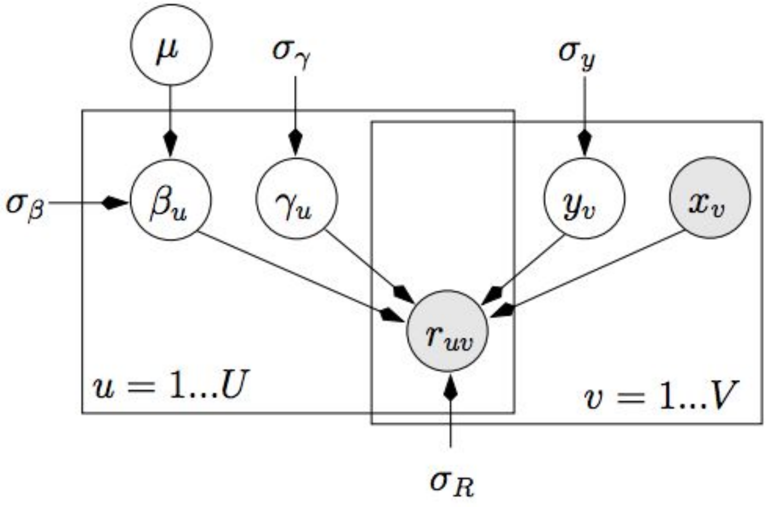
Deep Learning Methodology



Symbol	Description
u	User u
v	Song v
r_{uv}	The rating that user u gives to song v
γ_u	The latent features for u estimated by MF
y_v	The latent features for v estimated by MF
β_u	User u's preference of content features
μ	All users' common preference
x_v	The learnt content features for song v
Ω	The parameters of DBN
$\mathcal{U}$, $\mathcal{V}$	User and song sets, respectively
U, V	The number of users and songs, respectively
I	All user, song pairs in the training dataset

Table 1: Frequently used symbols

Deep Learning Methodology



Symbol	Description
u	User u
v	Song v
r_{uv}	The rating that user u gives to song v
γ_u	The latent features for u estimated by MF
y_v	The latent features for v estimated by MF
β_u	User u's preference of content features
μ	All users' common preference
x_v	The learnt content features for song v
Ω	The parameters of DBN
$\mathcal{U}$, $\mathcal{V}$	User and song sets, respectively
U, V	The number of users and songs, respectively
I	All user, song pairs in the training dataset

Table 1: Frequently used symbols

Xing Et. Al. [7]

Motivation

-Enhance music recommendation systems by use of “explorative” techniques to balance out the “exploitive/greedy” techniques of classical recommendation systems

-Use of reinforcement learning to model user rating:

$$U_{CF} = \boldsymbol{\theta}^T \mathbf{v}$$

Theta: User’s preferences for each feature (row) of V
V: Singular suggested songs feature vector

As can be guessed, this will have to be an iterative process

Iterative Processes

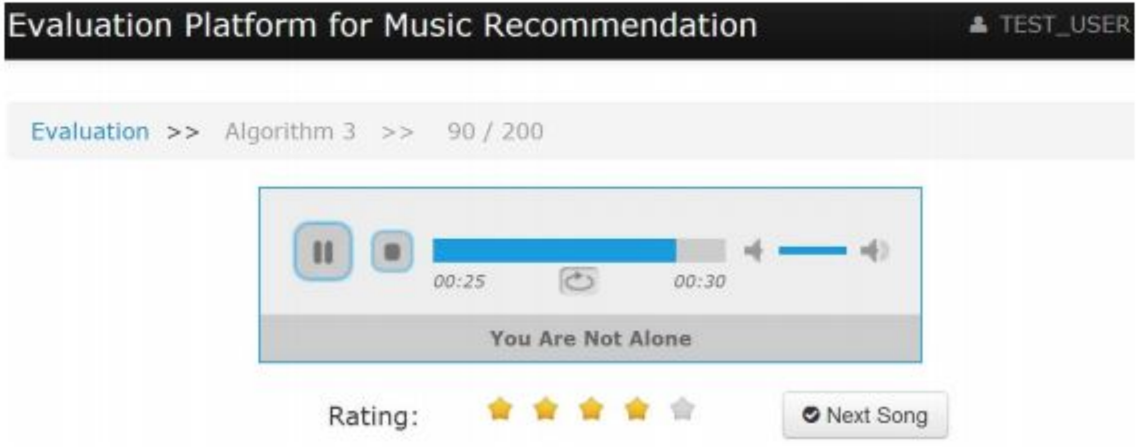
Algorithm 1 Exploration-Exploitation Balanced Music Recommendation

for $h = 1 \to N$ **do**
  if $h == 1$ **then**
    Recommend a song randomly;
  else
    Draw samples of $\boldsymbol{\theta}$ and s based on $p(\boldsymbol{\Omega} \mid \mathcal{D}_{h-1})$;
    for song $j = 1 \to |S|$ **do**
      Obtain $\mathbf{v}_j$ and t_j of song j and compute samples of U_j using Eq. (6);
      Estimate $p(U_j|\mathcal{D}_{h-1})$ using histogram of the samples of U_j;
      Compute quantile $q_j^h = Q\left(1 - \frac{1}{h}, p(U_j|\mathcal{D}_{h-1})\right)$;
    end for
    Recommend song $j^* = \arg\max_{j=1,\ldots,|S|} q_j^h$;
    Collect user rating r_h and update $p(\boldsymbol{\Omega} \mid \mathcal{D}_h)$;
  end if
end for

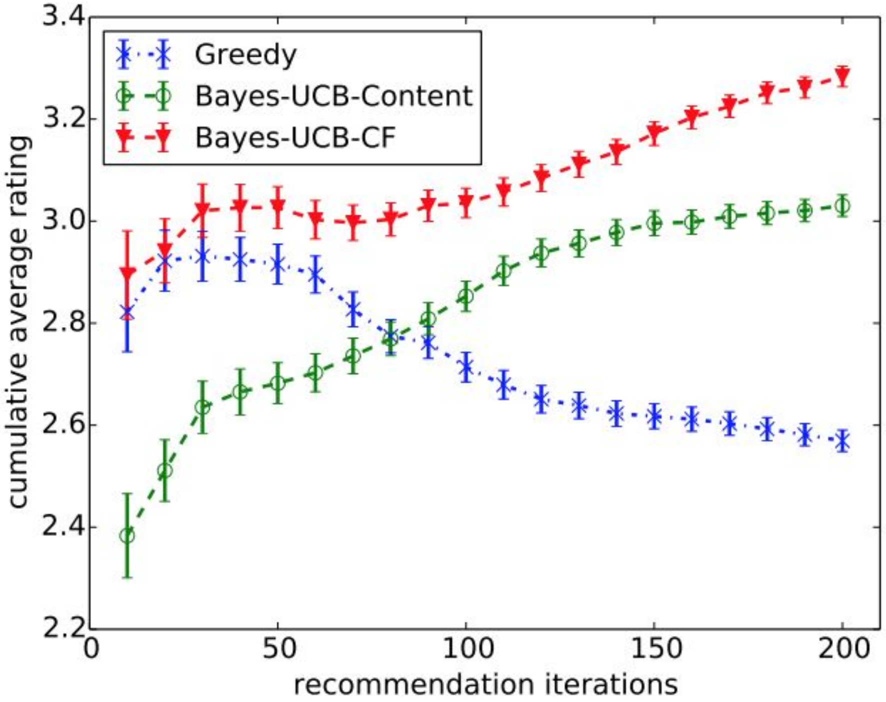
Algorithm 2 Gibbs Sampling for Bayesian Inference

Initialize $\boldsymbol{\theta}, s, \tau$;
for $t = 1 \to MaxIteration$ **do**
  Sample $\boldsymbol{\theta}^{(t+1)} \sim p(\boldsymbol{\theta}|\mathcal{D}, \tau^{(t)}, s^{(t)})$;
  Sample $\tau^{(t+1)} \sim p(\tau|\mathcal{D}, \boldsymbol{\theta}^{(t+1)}, s^{(t)})$;
  $s_{tmp} = s^{(t)}$;
  for $i = 1 \to K$ **do** # MH Step
    Draw $y \sim \mathcal{N}(s_{tmp}, 1)$;
    $\alpha = min\left(\frac{p(y|\mathcal{D}, \boldsymbol{\theta}^{(t+1)}, \tau^{(t+1)})}{p(s_{tmp}|\mathcal{D}, \boldsymbol{\theta}^{(t+1)}, \tau^{(t+1)})}, 1\right)$;
    Draw $u \sim Uniform(0, 1)$;
    if $u < \alpha$ **then**
      $s_{tmp} = y$;
    end if
  end for
  $s^{(t+1)} = s_{tmp}$;
end for

Subjective Results



Subjective Results



Non-Academic References

[1] Ciocca, Sophia. “How Does Spotify Know You So Well? – Member Feature Stories – Medium.” *Medium.com*, Medium, 21 June 2018, medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe.

[2] Bernhardsson, Erik. “Collaborative Filtering at Spotify.” *LinkedIn SlideShare*, 25 Jan. 2013, www.slideshare.net/erikbern/collaborative-filtering-at-spotify-16182818.

[3] Whitman, Brian. “How Music Recommendation Works - and Doesn't Work.” *Variogram by Brian Whitman*, 11 Dec. 2012, notes.variogr.am/2012/12/11/how-music-recommendation-works-and-doesnt-work/.

Academic References

[4] Van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. In *Advances in neural information processing systems* (pp. 2643-2651).

[5] Hu, Y. (2014). A model-based music recommendation system for individual users and implicit user groups.

[6] Wang, X., & Wang, Y. (2014, November). Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 627-636). ACM.

[7] Xing, Z., Wang, X., & Wang, Y. (2014, October). Enhancing Collaborative Filtering Music Recommendation by Balancing Exploration and Exploitation. In *Ismir* (pp. 445-450).

Process References

[8] Dieleman, Sander. “Recommending Music on Spotify with Deep Learning.” *Github*, 5 Aug. 2014, benanne.github.io/2014/08/05/spotify-cnns.html.

[9] Jehan, Tristan. *Analyzer Documentation: The Echo Nest*. 7 Jan. 2014, docs.echonest.com.s3-website-us-east-1.amazonaws.com/_static/AnalyzeDocumentation.pdf.

Questions?

Results

	# of users	# of songs	# of ratings
Total	100,000	282,508	28,258,926
Train	100,000	262,508	18,382,954
WS Valid	100,000	262,454	3,939,204
WS Test	100,000	262,457	3,939,206
CS Valid	99,963	10,000	1,025,654
CS Test	99,933	10,000	971,908

Table 2: Dataset statistics

	WS Valid	WS Test
Hybrid w/ HLDBN	**0.255**	**0.255**
Hybrid w/ CB2	0.270	0.270

Table 4: Predictive performance of our hybrid method with the features learnt by our HLDBN model and the baseline CB2 model (Root Mean Squared Error). WS stands for warm-start.

	WS Valid	WS Test
PMF	0.0109	0.0110
Hybrid w/ HLDBN	**0.0132**	**0.0131**
AM w/ traditional features	0.0108	0.0108
AM w/ features from HLDBN	0.0123	0.0120

Table 5: Comparison between hybrid methods using features learnt from HLDBN and traditional features (mean Average Precision).