Homework 2
Gaussian Matrix Factorization with Stochastic Gradient Descent
Griffin Adams (ga2530)
November 11, 2019

1 Problem 1

1.1 Problem Formulation

I implemented Gaussian Matrix Factorization on the MovieLens Dataset and trained it using Stochastic Gradient Descent (SGD).

Assuming Gaussian priors on both user preferences $theta_i$ and movie attributes β_j leads to a log posterior factorization, as discussed in class, of:

$$\log p(\theta, \beta | x) = -0.5 \sum_{i=1}^{n} \ell_2(\theta_i) - 0.5 \sum_{j=1}^{m} \ell_2(\beta_j) - 0.5 \sum_{i,j \in x} (x_{i,j} - \theta_i^T \beta_j)^2 + constant$$

where ℓ_2 is an L2-Regularizer of a K dimensional vector:

$$\ell_2(x) = \sum_{k=1}^{K} x_k^2$$

1.2 Optimization with SGD

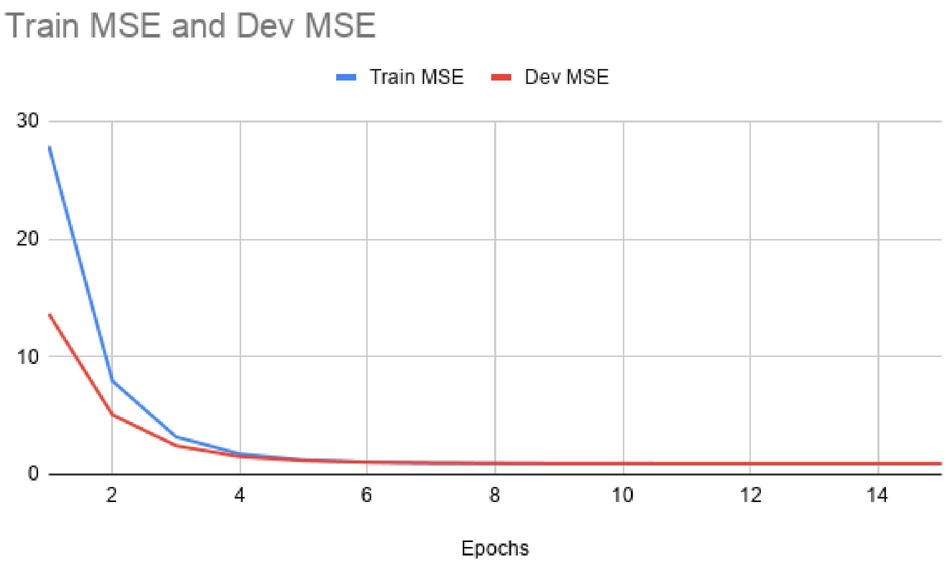
Conveniently, the log joint looks a lot like a negated L2-Regularized Mean-Squared Error loss function. Optimizing this with Stochastic Gradient Descent (SGD) is a very efficient, scalable inference procedure. To make training easier, I first translate the movie ratings (0.5-5) to have 0 mean.

I also added a bias term for both users and movies to allow for users and movies to have context-independent preferences and attributes. In other words, some movies are universally well liked and some users universally love movies, and vice versa.

I randomly sampled batches of size 128 and trained for 20 epochs. I used PyTorch for autodifferentiation and relied on the SGD Loss Function and set the weight decay parameter (L2 regularization) to be 1e-2. Aside from coefficients, this loss function is equivalent (in my understanding) to the log posterior up to an additive constant (the evidence). More regularization is needed due to the sparsity of the matrix. I did some minor hyperparameter tuning but found the held out MSE to be roughly invariant to minor changes in the hyperparameters.

1.3 Results

I plot the average mean squared error for the training set and the held out set (20% of the data). It is nice to see that the model does not overfit.



MSE and Dev MSE.png

I also qualitatively inspected movie attributes to see if they latently uncovered movie genre. I was pleasantly surprised to find that, despite some noise, they did. I sampled 5 movies at random and found the nearest 3 movies - using cosine similarity as a distance metric. I plot the movies and their nearest neighbors in attribute space, below.

Movie	Closest Movie	Second Closest	Third Closest
Fantasia	The Women	1408	Follow the Fleet
Racing Stripes	The Theory of Everything	Fun	The Extraordinary Adventures of Adele Blanc-Sec
The Girl Who Leapt Through Time	Theremin: An Electronic Odyssey	Late Night with Conan O'Brien	The Salt of the Earth
Love and Other Drugs	Oliver!	Father of the Bride	Outbreak
Jimi: All Is by My Side	The Unforgiven	Penn & Teller Get Killed	Holy Man

1.4 Future Work

Future work involves handling the cold start problem and potentially more principled normalization of the raw ratings data.

2 Problem 2

With partner Mert Ketenci

State of art word embedding models are good at learning the high dimensional representation of a word by mapping it into a vector-space using nearby words that are in the text. Yet, those models are not adequate for revealing the latent meaning of a word. For such models, the word "cell" is a word that can represent anything ranging from the organism (biology), prison (security) and phone (communication). Even though written the same, the word "cell" that is used in the context of biology differs from communication. In this study, we are going to propose a method to abbreviate this disambiguation in the context of clinical texts and distinguish homonyms. Our proposal relies on Gaussian word embeddings to uncover the contextualized senses of the words through the posterior Skip gram distribution. We extend the paper by Brazinskas et al* to incorporate multiple modes to directly capture acronyms expansion in clinical text. We train and evaluate our model on discharge note summaries (from Columbia and the publicly available MIMIC dataset), and find that it captures multiple senses better than any published model to date.

* Embedding Words as Distributions with a Bayesian Skip-gram Model