Seeking Feature Sparsity in Collaborative Filtering with LassoNet

Will Armstrong
CS 229 final course project
Finance & Commerce category
wrmstrng@stanford.edu, SUID# 06606233

1 Introduction

The problem of matrix completion (or, equivalently, matrix imputation, or matrix reconstruction) can be summarized as follows: given a matrix $Z \in \mathbb{R}^{m\times n}$, and a set $\Omega \subset \{1,\ldots,m\} \times \{1,\ldots,n\}$ of indices of observed entries of Z, can we infer the values for the unobserved entries $Z_{\Omega^\perp}$? This has applications to the theory of control systems for system identification, to providing precise location estimates for sensors given distances to known markers, and to signal frequency and direction detection in signal processing [1].

Our interest is in the application of matrix completion to product recommendation systems through *collaborative filtering* (as in [4]); given a set of u individual customers, and a set of i items, let $z_{u,i}$ be given as a measure of the 'affinity' of individual $u \in \{1,\ldots,m\}$ for product $i \in \{1,\ldots,n\}$, whether measured directly (as through user reviews or star ratings) or indirectly (through, e.g., page views or purchase behaviour). The task is then to complete the matrix Z by inferring the unobserved affinities $Z_{\Omega^\perp}$ from user-item relationships for the observed affinities Z_Ω.

Collaborative filtering methods face a *cold start problem* problem that content-based recommendations do not; new items and new users both lack the observed affinities necessary from which to make meaninful inferences. One side of this problem can potentially be mitigated by actively capturing users' preferences for examples of items that are highly informative of their preferences early on. To this end, we investigate LassoNet [7], a regularization method for achieving feature sparsity in artifical neural networks.

2 Related work

There is a wealth of research into mitigating the cold-start problem in collaborative filtering. [5] proposes a merging of users explictly trusted (and specified) by a given user, while [11] proposes augmenting with demographic data and online social media activity. [2] proposes generating 'virtual, but plausible neighbors' to cold-start users. [10] proposes weighting schemes from cold-start, post cold-start, and power users separately. [13] is an example of one of the many avenues that seek to uncover more latent information with richer model architectures. Unsurprisingly, recommendation systems is a very large area with a great deal of application and an abundance of research.

3 Dataset and features

In the course of our investigation, we will use the somewhat famous MovieLens dataset[1], provided by GroupLens and the University of Minnesota. The MovieLens 1M dataset consists of 1,000,209 distinct ratings (on a scale from 1 to 5) of 3,706 movies by 6,040 users, each user having rated at least 20 movies. The data consist of the tuples (`userID`, `itemID`, `rating`, `timestamp`) for each rating. The MovieLens 100k dataset is similarly structured, and comprised of 100,000 distinct reviews by 943 users of 1,682 movies. For methods that rely on explicit feedback, the star rating was used directly; for implicit feedback, a positive case was taken to be a star rating of 4.0 or greater.

[1] https://grouplens.org/datasets/movielens/

4 Methods

[7] proposes a variant of the LassoNet algorithm (described in the same paper) for matrix completion, with a 'warm start' procedure that commences with row-mean imputation and then alternating rounds of unsupervised row reconstruction and re-imputation from the reconstructed elements. We attempt here a different warm-start scenario for LassoNet, where the machine learning task over which the regularization path is found is to reconstruct a user-item affinity matrix whose inferred completion is estimated by other means.

Two baseline methods, Item-based k-nearest-neighbor (ItemKNN) [3] and Matrix Factorization (MF) [6], were chosen, as well as two high benchmarking collaborative filtering methods: Bayesian Personalized Ranking (BPR) [9] and Bilateral Variational Autoencoder for Collaborative Filtering (BiVAECF) [12].

4.1 Collaborative filtering methods

4.1.1 Item-based k-nearest-neighbor (ItemKNN)

ItemKNN is an extension of k-nearest-neighbors which aims to provide recommendations of similar items. The distance measure here between item i and item j is simply $\cos(\theta)$, where θ is the angle between the zero-filled vectors z_i and z_j.

4.1.2 Matrix Factorization (MF)

Matrix factorization is a broad category of techniques; in its simplest form, we suppose the matrix $Z = WH$, with $W \in \mathbb{R}^{k\times n}, H \in \mathbb{R}^{u\times m}$, W a matrix of latent item factors, and R a matrix of latent user factors, with k the number of latent factors.

This decomposition can be found by minimizing $||\hat{Z} - Z_\Omega||_F + \lambda_W||W||_F + \lambda_H||H||_F$, where $||\cdot||_F$ is the Frobenius norm.

4.1.3 Bayesian Personalized Ranking (BPR)

Bayesian Personalized Ranking seeks to optimize the posterior probability of model parameters Θ,

$$p(\Theta| >_u) \sim p(>_u |\Theta)p(\Theta)$$

given a user's implicit preference relation $>_u$ among items, where the probability user u prefers item i over item j is defined as $p(i >_u j \mid \Theta) = \sigma(x_{uij}(\Theta))$, with σ the logistic function. The maximum a posteriori estimator

$$J = \sum_{(u,i,j)} \log \sigma(x_{uij}) - \lambda_\Theta||\Theta||^2$$

is differentable and solvable by stochastic gradient descent. x_{uij} is estimated by $\hat{x}_{uij} = \hat{x}_{ui} - \hat{x}_{uj}$, which in turn is estimated by matrix factorization:

$$\hat{x}_{ui} = \langle w_u, h_i \rangle = \sum_{f=1}^{k} w_{uf} \cdot h_{if}$$

4.1.4 Bilateral Variational Autoencoder for Collaborative Filtering (BiVAECF)

BiVAECF is a generative model that extends Variational Autoencoders for Collaborative Filtering [8]. One assumes z_ui to follow an exponential family of distributions, i.e. $z_{ui} \sim \text{EXPFAM}(r_{ui}; \eta(\theta_u; \beta_i; \omega))$, with β_i and θ_u representing latent item and user parameters, respectively, with $\mathbb{E}(z_{ui}|\theta_u, \beta_i) = g_\omega(\theta_u; \beta_i)$ for some nonlinear g_ω (in this case a neural network).

4.2 LassoNet

The motivation for LassoNet is to provide a means to achieve feature sparsity in arbitrary, potentially deep residual neural networks.

Taking f to be a residual feed-forward neural network, i.e.,

$$f(x) = \theta^T x + f_W(x),$$

we minimize the L_1 penalized loss

$$L(\theta, W) + \lambda||\theta||_1,$$

subject to the constraint

$$||W_j^{(0)}||_\infty \leq M|\theta_j|, j \in \{1, \ldots, d\},$$

where d is the dimension of θ. Regularization is achieved here in two ways; the L_1 penalty controls the complexity of the fitted model, and M, the hierarchy coefficient, controls the mixture of the linear (residual) and nonlinear components.

Minimizing the objective is solvable by a proximal gradient descent algorithm outlined in [7].

5 Results

The MovieLens 1M dataset was partitioned into a 70% / 15% / 15% train / validation / test split such that every item was reviewed at least once and every user contributed at least one review in each partition of the data. Hyperparamter tuning was done using grid search on k for ItemKNN, and random search for all others. For BPR, the hyperparameters that maximize the AUC of implicit preferences on the validation set were chosen, for all others, those maximizing RMSE of explicit ratings were chosen. The hyperparameters selected were: $k = 20$ for ItemKNN, $k = 150$[2] and learning rate $\alpha \approx 0.0084$ for MF, and $k = 120$ and $\alpha \approx 0.0099$ for BPR.

For BiVAECF, an architecture of one hidden layer of dimension 200 was chosen *a priori* with tanh activation, and the models were trained for 400 epochs. A batch size of 64, learning rate $\alpha \approx 0.0025$, and $k = 30$ hidden dimensions were the hyperparameters estimated.

Tables 1 and 2 show implicit preference validation metrics on the validation and test datasets, respectively.

	Evaluation metric				
	AUC	MAP	nDCG@10	Precision@10	Recall@10
ItemKNN	0.7436	0.0129	0.0059	0.0068	0.0046
MF	0.7882	0.0291	0.0383	0.0293	0.0382
BPR	**0.9355**	**0.0681**	**0.0791**	**0.0533**	**0.0915**
BiVAECF	0.9326	0.0641	0.0737	0.0497	0.0869

Table 1: Evaluation metrics on validation set (%15 sampled from MovieLens 1MM)

	Evaluation metric				
	AUC	MAP	nDCG@10	Precision@10	Recall@10
ItemKNN	0.7401	0.0166	0.0084	0.0097	0.0048
MF	0.7868	0.0338	0.0476	0.0399	0.0360
BPR	**0.9373**	**0.0819**	**0.0989**	**0.0769**	**0.0896**
BiVAECF	0.9354	0.0771	0.0909	0.0703	0.0841

Table 2: Evaluation metrics on test set (%15 sampled from MovieLens 1MM)

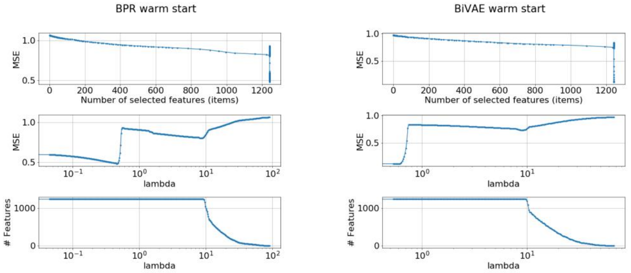
In addition to AUC on implicit preference, we show several information retrieval metrics: Normalized Discounted Cumulative Gain[3] as well as precision and recall, for the 10 highest inferred ranking items for each user.

The fitted BiVAECF and BPR models were then used to construct a matrix of inferred affinities for those users and items who also appear in the MovieLens100k dataset, and LassoNet was used to find a regularization path through movies over the task of reconstructing each row of standardized preferences through a residual feed-forward artifical neural network (once with a single hidden layer of 100 neurons, and once with 500), with the default hierarchy parameter of $M = 10$, with a 75% / 25% train / validation split. The regularization path over successive values of λ is shown in 5, as well as the feature importance, here defined as the value of λ at which the feature is removed.

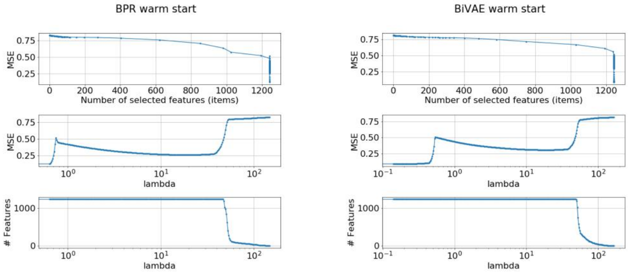
[2] For ItemKNN, k denotes the k nearest neighbors; for all others, k is the number of latent feature dimensions.

[3] nDCG@10 is given by $\sum_{i=1}^{10} \frac{2^{y_i} - 1}{\log_2(i+1)}$ where i ranges over the 10 items recommended to a given user, and y_i is the preference of that user for item i

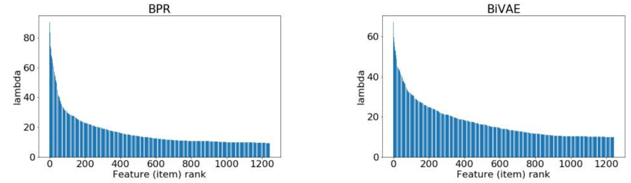
Regularization path, 100 hidden layers



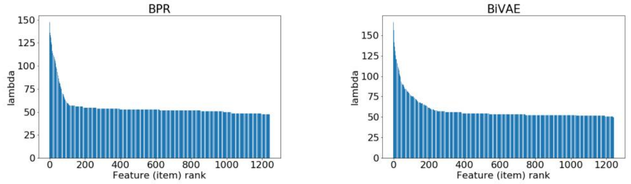
Regularization path, 500 hidden layers



Feature importance, 100 hidden layers



Feature importance, 500 hidden layers



Among the 100 most popular movies, the last to be removed on each regularization path were:

BPR warm start :

1. *Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb* (1964), [1201]
2. *The Fugitive* (1993) [1128]
3. *Terminator 2: Judgment Day* (1991) [1065]
4. *Fargo* (1995) [1032]
5. *The Shawshank Redemption* (1994) [1030]

BiVAECF warm start :

1. *Babe* (1995) [1242]
2. *Amadeus* (1984)
3. *The Silence of the Lambs* (1995) [1173]
4. *Casablanca* (1942) [1163]
5. *True Lies* (1997) [1155]

(Numbers in parentheses are the year the movie was released, numbers in square brackets are their feature importance ranking).

6 Conclusions and future work

It would be tempting to interpret the movies listed above as perhaps approaching a short list of movies that are at a saddle point between being most informative about a person's movie preferences and most abundant among peoples' preferences. If they are, the question remains whether it is such 'a' list, or 'the' list, since the order features are visited in the regularization path may be sensitive to small changes in the data.

Additionally, during regularization, a lot of predictive value seems to be given up for early values of λ. The feature importance curve is long tailed, and relatively high values of λ have to be arrived at before any features are effectively removed. There is a well-understood popularity bias in collaborative filtering. Unurprisingly, therefore, there is in fact a small number of 'important' movies that are predictive of preferences, but to maximize predictive power, as many preferences as possible are needed. Once only a few 'unimportant' movies are taken away, the error increases and doesn't increase much farther.

The default hierarchy parameter $M = 10$ was attested in [7] to work well for a large variety of datasets, but those authors also stated that it would be difficult to set without some expertise on the domain or task; the next, obvious step would be to try to tune this parameter.

Preferences that are informative of other preferences are at the core of collaborative filtering. There may be more direct, obvious ways to detect such highly informative items; this author is only beginning learning about recommendation systems and information retrieval. The question 'which preferences are most informative about a users' other preferences' is still, we believe, a reasonable one.

7 Acknowledgements & contributions

Thanks to Stanford Ph.D candidate Ismael Lemhadri, who sponsored this project, for being willing to share his insights into LassoNet and his research, and to the CS229 course staff for their patience and helpful advice. Will Armstrong was the sole CS229 student who contributed, with the advice and guidance of those thanked herein.

Code used can be found at `https://github.com/madwsa/cs229-project`.

References

[1] Emmanuel J. Candes and Terence Tao. The Power of Convex Relaxation: Near-Optimal Matrix Completion. *arXiv:0903.1476 [cs, math]*, March 2009. arXiv: 0903.1476.

[2] Dong-Kyu Chae, Jihoo Kim, Duen Horng Chau, and Sang-Wook Kim. Ar-cf: Augmenting virtual users and items in collaborative filtering for addressing cold-start problems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1251–1260, 2020.

[3] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[4] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, December 1992.

[5] Guibing Guo, Jie Zhang, and Daniel Thalmann. Merging trust in collaborative filtering to alleviate data sparsity and cold start. *Knowledge-Based Systems*, 57:57–68, 2014.

[6] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8):30–37, August 2009.

[7] Ismael Lemhadri, Feng Ruan, Louis Abraham, and Robert Tibshirani. LassoNet: A Neural Network with Feature Sparsity. *arXiv:1907.12207 [cs, stat]*, February 2021. arXiv: 1907.12207.

[8] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pages 689–698, 2018.

[9] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars BPR Schmidt-Thieme. Bayesian personalized ranking from implicit feedback. In *Proc. of Uncertainty in Artificial Intelligence*, pages 452–461, 2014.

[10] Alan Said, Brijnesh J Jain, and Sahin Albayrak. Analyzing weighting schemes in collaborative filtering: cold start, post cold start and power users. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 2035–2040, 2012.

[11] Suvash Sedhain, Scott Sanner, Darius Braziunas, Lexing Xie, and Jordan Christensen. Social collaborative filtering for cold-start recommendations. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 345–348, 2014.

[12] Quoc-Tuan Truong, Aghiles Salah, and Hady W Lauw. Bilateral variational autoencoder for collaborative filtering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 292–300, 2021.

[13] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, 69:29–39, 2017.