Bayesian Recommendation Systems

Isaac C. Liao [1]

Abstract

Low-rank approximations are one of the primary techniques for large matrix completion problems, which underlie technologies such as recommendation systems. In this paper, we derive an equivalence between the alternating least squares (SVD-ALS) solution algorithm and the coordinate ascent variational inference (CAVI) algorithm, to reimagine SVD-ALS from within the Bayesian framework. We then expand the class of approximate posteriors to construct an extension of SVD-ALS which outperforms the original by over 2% in Netflix Prize Dataset movie rating prediction RMSE.

1. Introduction

Recommender systems are a powerful technology for product marketing, whereby a service can increaset profits by accurately recommending products to users which they are most likely to choose and/or rate highly. Oftentimes, the simplest data source to generate predictions from is the history of past purchases and ratings, with each purchase marked by the user ID, product ID, datetime, and rating. In this paper, we will focus on the application of movie recommendations, where the products are movies and a "purchase" is when a user watches a movie and rates it.

A common way to formulate this problem is as a large matrix completion problem. The simplest version of this problem assumes that no user watches the same movie twice, all users always rate a movie after watching, and the datetimes are irrelevant. All the data can then be completely described by a rating matrix $R \in (\mathbb{R} \cup \{\text{blank}\})^{n \times m}$ where there are n movies and m users. The element R_{ij} is the rating that user j gave movie i if they watched it, or is left blank if the user did not watch this movie. We try to estimate these blank values with as small as possible root-mean-squared-error (RMSE) value on held-out ratings, to recommend the highest predicted movies to each user.

[1]Reearch Lab for Electronics, MIT, Cambridge, MA 02139, USA. Correspondence to: Isaac C. Liao <iliao@mit.edu>.

A typical solution idea is to assume that each movie i is described by a vector $U_{i,\cdot} \in \mathbb{R}^r$ of r "characteristics" (eg. how much action it has, how much romance it has) and that each user j has some amount of affinity for each characteristic, represented by a vector $V_{\cdot,j} \in \mathbb{R}^r$. The estimated rating that user j would give movie i is then the dot product $U_{i,\cdot} \cdot V_{\cdot,j}$, so we can construct an estimated rating matrix $\tilde{R} = UV$. This low-rank approximation $\tilde{R} = UV$ to the rating matrix R is tuned to minimize the squared errors on the known ratings, a solution known as the "SVD" method in the literature.

One way to solve for the SVD solution is to use alternating least squares (ALS), whereby the the squared error is noted to be quadratic in U and is solved for while holding V fixed, and vice versa repeatedly (Takács & Tikk, 2012). L2 regularization is used on the component matrices U and V to reduce overfitting (Gower, 2014). Another way to solve for the squared error minimizer is to use gradient descent (Ma, 2008).

Additional modifications can be made to improve the fit, such as additive global rating biases for each movie and user. (Bokde et al., 2015) Other common solutions make rating matrix predictions $\tilde{R}$ using principal-component-analysis (PCA) (Vozalis & Margaritis, 2007). Many top performing solutions find some way to incorporate movie release and consumption date information into the predictions and also make use of restricted Boltzmann machines. Other techniques include the clustering of users via K-nearest-neighbors, and kernel ridge regression (Paterek, 2007).

The purpose of this paper is to rethink the SVD with ALS algorithm from a Bayesian modeling perspective and to study an extension to SVD-ALS which naturally arises as a result, with the goal of better achieving the data analysis goal of improving RMSE accuracy of rating estimation, in a similar manner to Mnih & Salakhutdinov (2007), Zhang & Liu (2014), and Salakhutdinov & Mnih (2008).

We use the Netflix Prize Dataset, which contains about 100M ratings of 18k movies by 400k users, forming a matrix with $\sim 1\%$ sparsity. This dataset was part of a contest from 2006 to 2009, where the best solution from Koren (2009) won a large monetary prize set out by Netflix. Netflix's own existing algorithm at the time predicted held out ratings with an RMSE of about 0.9514 (Koren et al., 2009). Simpler

SVD based solutions tended to predict held out ratings with an RMSE of about 0.91, and as more information from the dataset was included (such as datetime), this dropped to about 0.88. This winning solution was a large conglomerate of several systems, and achieved an RMSE of 0.8554. These RMSE values were measured with an undisclosed dataset split which was more difficult than a random split. Since we must use a random split which is easier, our RMSE values cannot be compared with those reported by other papers.

2. SVD Method

We begin by outlining the mathematics of the SVD solution.

Let there be a true rating matrix $R \in (\mathbb{R} \cup \{\text{blank}\})^{n \times m}$ and a mask $M \in \{0,1\}^{n \times m}$ indicating which ratings are not blank. We seek to construct some estimator $\tilde{R}$ of the rating matrix R; M masks out some of the data in R, which must then be estimated from only the visible values.

2.1. Optimization Problem

The SVD method's estimator is a low-rank decomposition $\tilde{R} = UV$ where $U \in \mathbb{R}^{n \times r}$ and $V \in \mathbb{R}^{r \times m}$. The matrices U and V are chosen to minimize the total squared error plus regularization,

$$\ell = \sum_{i,j} M_{ij}([UV]_{ij} - R_{ij})^2 + \lambda(||U||_F^2 + ||V||_F^2). \quad (1)$$

with hyperparameters rank $r \in \mathbb{N}$ and regularization strength $\lambda \in \mathbb{R}$.

2.2. Alternating Least Squares Solver

While standard stochastic gradient descent would suffice in many large matrix completion problems, this can be too slow for larger problems, so we choose to study the ALS algorithm because it is often faster. The main idea of ALS is to notice that the loss is quadratic and convex in both U and V independently, so one can alternately solve for U via least squares while holding V fixed, and vice versa. We solve for U by setting the derivative of the loss to zero,

$$0 = \frac{\mathrm{d}\ell}{\mathrm{d}U_{i\cdot}} = \frac{2}{N}(U_{i\cdot}V - R_{i\cdot})\text{diag}(M_{i\cdot})V^T + 2\lambda U_{i\cdot} \quad (2)$$

$$U_{i\cdot} = (R_{i\cdot} \odot M_{i\cdot})V^T(V\text{diag}(M_{i\cdot})V^T + N\lambda I)^{-1} \quad (3)$$

where $\odot$ denotes the elementwise product, and $U_{i\cdot}$ denotes the ith row of U. The same, with U and V swapped and transposed, is done to find the best V, and then the whole process is repeated many times. The time complexity is $O(r^3)$ due to the matrix inverse.

2.3. The Bayesian Version

We would now like to rethink this solution from the Bayesian modeling perspective. This can be done by finding a Bayesian model where variational inference (VI) gives us a KL divergence that lines up with ℓ, along with a VI algorithm to replicate the behavior of ALS. We have built up the details of this model through trial and error until it reproduced the SVD-ALS solution, but we save our past failed attempts and only present the final successfully reproducing model.

We begin with a graphical model with latent variables U and V and observables R. Our prior p presumes that U and V are drawn from iid normal distributions of variance α, and that the ratings are $R = UV + \epsilon$ for some iid normally distributed noise ϵ of variance β. The estimator $\tilde{R}$ we construct is then the expected value of R under the posterior over U and V. Let us use a mean-field posterior approximation q, with U and V both independent and normally distributed with means μ_U and μ_V but with fixed variance γ. Furthermore, let $\alpha, \beta, \gamma \in \mathbb{R}_+$ be some predetermined constants (ie. hyperparameters). The relevant KL divergence to be minimized, in terms of μ_U and μ_V, can then be shown to be

$$\begin{aligned}
\text{KL}(q||p) =& \mathbb{E}_{U,V\sim q}\left[\log \frac{q(U,V)}{p(U,V)}\right] && (4)\\
=& KL(\mathcal{N}(\mu_U,\gamma)||\mathcal{N}(0,\alpha)) \\
&+ KL(\mathcal{N}(\mu_V,\gamma)||\mathcal{N}(0,\alpha)) \\
&+ \frac{1}{2\beta}\sum_{i,j} M_{ij}\mathbb{E}_{U,V\sim q}\left[([UV]_{ij} - R_{ij})^2\right] \\
&+ \text{const} && (5)\\
=& \frac{1}{2\beta}\sum_{i,j} M_{ij}([\mu_U\mu_V]_{ij} - R_{ij})^2 \\
&+ \frac{1}{2\alpha}\left(||\mu_U||_F^2 + ||\mu_V||_F^2\right) \\
&+ O(\gamma\mu^2/\beta) + \text{const} && (6)
\end{aligned}$$

Notice the close resemblance of this KL divergence to the form of ℓ, with the main difference being the $O(\gamma)$ term. We can resolve this by choosing $\alpha = 1/2\lambda$ and $\beta = 1/2$, and very small $\gamma \to 0$, to finally obtain $\text{KL}(q||p) = \ell + \text{const}$. Then, minimizing the KL and constructing $\tilde{R} = \mathbb{E}_{U,V\sim q}[R] = \mu_U\mu_V$ is equivalent to standard SVD. We can use Coordinate-Ascent Variational Inference (CAVI) (Bishop & Nasrabadi, 2006) to perform the minimization; it optimizes each of μ_U and μ_V alternately. Solving for the minimum with respect to μ_U for the CAVI step, we get

$$\mu_{Ui\cdot} = (R_{i\cdot} \odot M_{i\cdot})\mu_V^T\left(\mu_V\text{diag}(M_{i\cdot})\mu_V^T + \frac{\beta I}{\alpha}\right)^{-1} \quad (7)$$

and correspondingly for μ_V. This is exactly the same as what ALS computes to optimize U and V. This concludes the successful reconstruction of the SVD-ALS solution from the Bayesian viewpoint.

3. Flexible Variances

From the Bayesian modeling perspective, it is strange and unnatural to make the choice that the approximate posterior q has infinitesimal variance γ. It would appear that this modeling choice is made to eliminate the $O(\gamma\mu^2/\beta)$ term in the KL loss such that its minimization simplifies a least squares problem, thus leading to the SVD-ALS algorithm. While this is well known to be tractable, the use of a constant variance limiting to zero is an unnecessary sacrifice, and the minimizer still has an easily computable solution when the variances are unrestricted. We will show that this extension to SVD-ALS has a positive effect on its performance. We will refer to this flexible-variance model as "Bayesian SVD-ALS".

Mathematically, we replace the constant variance γ with per-element variances $v_U \in \mathbb{R}^{n\times r}$ and $v_V \in \mathbb{R}^{r\times m}$ in the mean-field approximate posterior q. The KL loss can now be rederived in the same way as was done in Equations (4)-(6), resulting in the expression:

$$\begin{aligned}\mathrm{KL}(q||p) =& \frac{1}{2\beta}\sum_{i,j} M_{ij}([\mu_U\mu_V]_{ij} - R_{ij})^2 \\ &+ \frac{1}{2\alpha}\left(||\mu_U||_F^2 + ||\mu_V||_F^2\right) \\ &+ \sum_{ik}\frac{v_{Uik}/\alpha - \ln v_{Uik}}{2} \\ &+ \sum_{kj}\frac{v_{Vkj}/\alpha - \ln v_{Vkj}}{2} \\ &+ \frac{1}{2\beta}\sum_{ijk} M_{ij}\Big((\mu_{Uik}^2 + v_{Uik})(\mu_{Vkj}^2 + v_{Vkj}) \\ &\qquad - \mu_{Uik}^2\mu_{Vkj}^2\Big) + \text{const.} \qquad (8)\end{aligned}$$

Again, we seek to minimize the KL, but now we have four matrix parameters μ_U, v_U, μ_V, v_V to optimize instead of two. Solving for μ_U, a correction of $\mathrm{diag}(M_{i\cdot}v_V^T)$ appears,

$$\begin{aligned}\mu_{Ui\cdot} =& (R_{i\cdot}\odot M_{i\cdot})\mu_V^T\Big(\mu_V\mathrm{diag}(M_{i\cdot})\mu_V^T + \beta I/\alpha \\ &+ \mathrm{diag}(M_{i\cdot}v_V^T)\Big)^{-1}. \qquad (9)\end{aligned}$$

When we solve for the minimum with respect to v_U, we get

$$v_{Uik} = \left(1/\alpha + M_{i\cdot}\left(\mu_{Vk\cdot}\odot\mu_{Vk\cdot} + v_{Vk\cdot}\right)^T/\beta\right)^{-1} \qquad (10)$$

The formula for the constructed rating matrix $\tilde{R} = \mathbb{E}_{U,V\sim q}[R] = \mathbb{E}_{U,V\sim q}[UV] = \mu_U\mu_V$ remains the same since U and V are independent. Importantly, the time complexity has not increased from the original ALS update.

A good sanity check is to interpret what these solutions entail on an intuitive level. Firstly, we observe from (9) that variance in V acts to regularize/suppress the mean of U. This makes sense, because our best estimates of a movie's characteristics should be less extreme if we are uncertain about the characteristics of the users who watch the movie.

(10) shows that the variance of U is α by default, but it decreases with growing V. In other words, the user's characteristics begin with default uncertainty from the prior, but become more precise for users i who have given more ratings and whose ratings are more extreme and varied, which makes sense.

4. Experiments

In this section, we compare the empirical performance of SVD-ALS against Bayesian SVD-ALS, and try to figure out what makes Bayesian SVD-ALS different.

We split the data randomly 99 to 1 train to validation, and subtracted the mean rating off from all the ratings. We then optimized the SVD-ALS model for up to 25 iterations with early stopping at the lowest validation RMSE, and used the Covariance Matrix Adaptation Evolutionary Strategy (Hansen et al., 2003) to tune the hyperparameters to $\lambda = 6.047, r = 23$. For Bayesian SVD-ALS, we tuned only μ_U, μ_V for the first 16 iterations with v_U, v_V set uniformly to 10^{-8}, and then trained for up to 16 more iterations including v_U, v_V with early stopping, and tuned the hyperparameters to $\alpha = 0.108, \beta = 0.593, r = 77$. Table 1 shows that Bayesian SVD-ALS achieves over a 2% improvement in RMSE over traditional SVD-ALS.

Table 1. Performance of SVD-ALS against Bayesian SVD-ALS.

Method	Train RMSE	Validation RMSE
SVD-ALS	0.7562	0.8303
Bayesian SVD-ALS (ours)	0.7132	**0.8097**

Figure 1 shows that in SVD-ALS, the magnitude of effect of the kth movie/user characteristic on ratings entirely determines how much information the posterior has gathered about that characteristic, as all the $(||\mu||, KL)$ points fall on a thin curve. In contrast, Bayesian SVD-ALS is able to distinguish between the degree of effect from the amount of information learned about that characteristic, as the curve shows a spread. An intuitive example of how this can happen is the posterior allows the effect of the characteristic to be known as the same value but at different levels of accuracy, ie. with the same mean $\mu_U \approx 0$ but any variance v_U. Interestingly, Figure 1 also shows that Bayesian SVD-ALS models movies with more characteristics than SVD-ALS, but with each characteristic having a weaker effect on ratings. The effect of top user characteristics on ratings is also pronounced in Bayesian SVD-ALS.

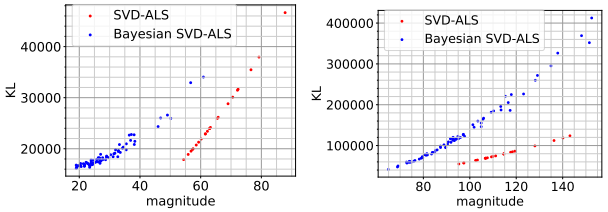

Figure 1. **Left:** Amount of information collected about which movies each characteristic k (denoted by one point), compared to how much this characteristic determines movie ratings. The amount of collected information is measured by the KL contribution $\mathrm{KL}(\mathcal{N}(\mu_{U_{\cdot,k}}, \mathrm{diag}(v_{U_{\cdot,k}})) || \mathcal{N}(0, \alpha))$ to Equation (5), which is affected by both mean and variance; and the degree of effect is measured by the magnitude $||\mu_{U_{\cdot,k}}||$. For the original SVD-ALS, the approximate posterior variance is fixed, $\mathrm{diag}(v_{U_{\cdot,k}}) = \gamma \mathbf{I}$, and the minimum possible contribution $\mathrm{KL}(\mathcal{N}(0, \gamma \mathbf{I}) || \mathcal{N}(0, \alpha))$ is subtracted off. **Right:** The equivalent of the left plot but for users having preferences for characteristic k.

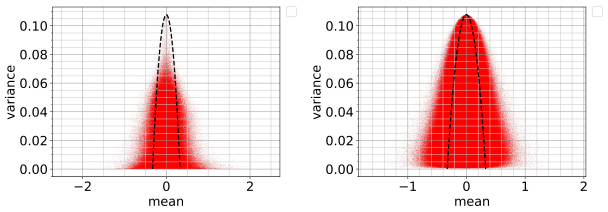

Figure 2. **Left:** Scatterplot of the approximate posterior components $q(U_{ik})$ for every i and k, in the Bayesian SVD-ALS solution. Every red dot represents a normal distribution parameterized by mean $\mu_{U_{ik}}$ and variance $v_{U_{ik}}$. The black dashed curve is $\mu_{U_{ik}}^2 + v_{U_{ik}} = \alpha$, where the prior and posterior both estimate U_{ik} to have the same squared value, $\mathbb{E}_p[U_{ik}^2] = \mathbb{E}_q[U_{ik}^2]$. The blue X marks the prior distribution $\mathcal{N}(0, \alpha)$; any approximate posterior component here contains no information, and components farther from the X contain more information about the user. **Right:** The equivalent of the left plot but for movies V_{kj} instead of users U_{ik}.

Figure 2 shows the posterior $q(U, V)$ through the statistics of its individual components $q(U_{ik})$ and $q(V_{kj})$ for every i, k, j, each of which is a normal distribution. The (μ, v) pairs for every component are shown in a scatterplot. None of the variances rise above α, indicating that latent variable precision only ever increases as a result of data. Furthermore, when v is lower, μ deviates more from zero. In other words, the less we know about the amount of a certain characteristic a movie has, the closer to average our best estimate of this amount must be. Notice that the hyperparameter α seems well tuned, since the dashed parabola has a horizontal width roughly comparable to the scatterplot's spread.

Figure 3 highlights that the original SVD-ALS and the Bayesian SVD-ALS have significantly different assignments of the KL information content of the data matrix R. Namely, according to Bayesian SVD-ALS, most of the information

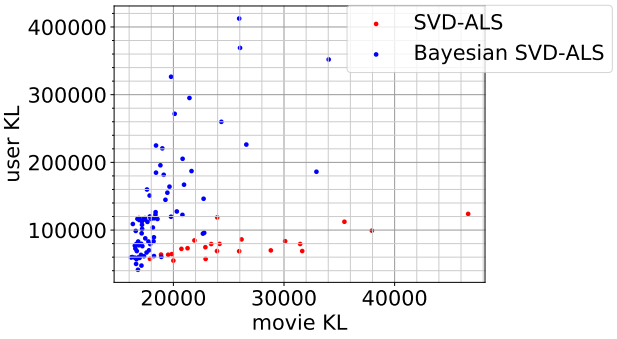

Figure 3. The amount of information about which users like a given characteristic k against the amount of information about which movies have characteristic k. Information measures are in the same way as in Figure 1.

in the dataset describes an understanding of user preferences, as opposed to the movie characteristics which remain more unknown, whereas to SVD-ALS, the comparitively more information is allocated to movie characteristics and less is about user preferences. This can be observed in the fact that the Bayesian SVD-ALS cluster has more user-related and less movie-related KL than the SVD-ALS cluster.

5. Discussion/Conclusion

In our analysis of SVD-ALS, we developed a Bayesian model for movie ratings, and a procedure for predicting estimated ratings, such that these predicted ratings can be used to recommend new movies to users. To do this, we reformulated standard SVD for large matrix completion as a variational inference problem with a particular Bayesian model, and we understood the ALS algorithm for SVD to be equivalent to the CAVI algorithm in this context.

We then widened the space of approximate posteriors, and rederived Bayesian SVD-ALS, the equivalent of the SVD-ALS algorithm which incorporates this extension.

Finally, we compared the empirical performance of SVD-ALS against Bayesian SVD-ALS, and found that Bayesian SVD-ALS reduces the validation RMSE by more than 2% on the Netflix Prize Dataset. We found that Bayesian SVD-ALS describes users and movies with more characteristics than SVD-ALS, and deduces comparatively more about the users and less about the movies. We also found that Bayesian SVD-ALS estimates the effect of user/movie characteristics on ratings to be larger for those effects which are more certain.

Overall, we have used the Bayesian framework to provide insight into the construction of large matrix completion algorithms for recommendation systems, resulting in an improved algorithm which achieves better performance.

References

Bishop, C. M. and Nasrabadi, N. M. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Bokde, D., Girase, S., and Mukhopadhyay, D. Matrix factorization model in collaborative filtering algorithms: A survey. *Procedia Computer Science*, 49:136–146, 2015. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2015.04.237. URL https://www.sciencedirect.com/science/article/pii/S1877050915007462. Proceedings of 4th International Conference on Advances in Computing, Communication and Control (ICAC3'15).

Gower, S. Netflix prize and svd. *University of Puget Sound*, 2014.

Hansen, N., Müller, S. D., and Koumoutsakos, P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18, 2003.

Koren, Y. The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81(2009):1–10, 2009.

Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37, 2009.

Ma, C.-C. A guide to singular value decomposition for collaborative filtering. *Computer (Long Beach, CA)*, 2008: 1–14, 2008.

Mnih, A. and Salakhutdinov, R. R. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20, 2007.

Paterek, A. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pp. 5–8, 2007.

Salakhutdinov, R. and Mnih, A. Bayesian probabilistic matrix factorization using mcmc. *ICML'08*, 2008.

Takács, G. and Tikk, D. Alternating least squares for personalized ranking. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pp. 83–90, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450312707. doi: 10.1145/2365952.2365972. URL https://doi.org/10.1145/2365952.2365972.

Vozalis, M. G. and Margaritis, K. G. A recommender system using principal component analysis. In *Published in 11th panhellenic conference in informatics*, pp. 271–283, 2007.

Zhang, Z. and Liu, H. Application and research of improved probability matrix factorization techniques in collaborative filtering. *International Journal of Control & Automation*, 7(8):79–92, 2014.