HAL
open science

Using Posters to Recommend Anime and Mangas in a Cold-Start Scenario

Jill-Jênn Vie, Florian Yger, Ryan Lahfa, Basile Clement, Kévin Cocchi, Thomas Chalumeau, Hisashi Kashima

To cite this version:

Jill-Jênn Vie, Florian Yger, Ryan Lahfa, Basile Clement, Kévin Cocchi, et al.. Using Posters to Recommend Anime and Mangas in a Cold-Start Scenario. MANPU 2017 held in conjunction with ICDAR 2017, Nov 2017, Kyoto, Japan. 10.1109/ICDAR.2017.287 . hal-01740137

HAL Id: hal-01740137

https://hal.science/hal-01740137

Submitted on 21 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Using Posters to Recommend Anime and Mangas in a Cold-Start Scenario

Jill-Jênn Vie
Mangaki
RIKEN Center for AIP
Tokyo, Japan
jill-jenn.vie@riken.jp

Florian Yger
LAMSADE
Université Paris-Dauphine
Paris, France
florian.yger@dauphine.fr

Ryan Lahfa, Basile Clement,
Kévin Cocchi, Thomas Chalumeau
Mangaki
Mitry-Mory, France
{ryan, basile, kevin, thomas}@mangaki.fr

Hisashi Kashima
Kyoto University
RIKEN Center for AIP
Kyoto, Japan
kashima@i.kyoto-u.ac.jp

***Abstract*—Item cold-start is a classical issue in recommender systems that affects anime and manga recommendations as well. This problem can be framed as follows: how to predict whether a user will like a manga that received few ratings from the community? Content-based techniques can alleviate this issue but require extra information, that is usually expensive to gather. In this paper, we use a deep learning technique, Illustration2Vec, to easily extract tag information from the manga and anime posters (e.g., sword, or ponytail). We propose BALSE (Blended Alternate Least Squares with Explanation), a new model for collaborative filtering, that benefits from this extra information to recommend mangas. We show, using real data from an online manga recommender system called Mangaki, that our model improves substantially the quality of recommendations, especially for less-known manga, and is able to provide an interpretation of the taste of the users.**

***Index Terms*—recommender system, cold-start, collaborative filtering, LASSO, tag prediction**

I. Introduction

Recommender systems are useful to help users decide what to enjoy next. In the case of anime and mangas, users, easily overwhelmed by the ever-growing amount of works, end up barely scratching the surface of what Japanese animation has to offer. Collaborative filtering is a popular technique that relies on existing rating data from users on items in order to predict unseen ratings (Aggarwal, 2016). However, it is still hard to recommend items for which little information is available, e.g., items for which few or no ratings have been provided by the community. This problem has been referred to as the *item cold-start problem*.

In order to alleviate this problem, it is possible to rely on extra information about the items, such as metadata (e.g., for movies: directors, composers, release date). However, such information is not always available: new anime projects may only have a poster or a trailer, and a title. Such a poster is usually the first contact that a user has with an item and plays a large role in the user's decision to watch it or not. Especially in the manga and anime industry, posters contain a lot of information about the characters, in order to maximize the visual appeal for the consumers. Hence, it is natural to consider posters as a source for additional metadata. In recent years, convolutional neural networks (CNNs) have established themselves as the *de-facto* method for extracting semantic information from image content in a wide variety of tasks. We propose using a CNN for extracting meaningful tags directly from the item's poster. Such extracted tags can help draw links between items, which can be useful when few ratings are available. In this paper, we present BALSE[1] (Blended Alternate Least Squares with Explanation), a new method leveraging tag information extracted from the posters for tackling the item cold-start problem and improving the recommendation performance for little-known items. We show using real data that our method provides better rating predictions than existing techniques, and gives interpretable insight about the user's taste.

To the best of our knowledge, this is the first research work that uses tag prediction on item posters in order to improve the accuracy of a recommender system and explain to users why they are recommended little-known items.

This paper is organized as follows. We first present existing work related to this research. Then, we expose the context of collaborative filtering and item cold-start, together with a few common assumptions. We then describe our model, BALSE, and present some experimental results on a real dataset. We finish by discussing the results and future work.

II. Related Work

Using side information in order to improve recommendations has been the core of existing research (Kula, 2015) and several approaches have been developed to take into account extra data about users or items, whether coming from text (Alexandridis, Siolas, and Stafylopatis, 2017; Fang and Si, 2011), social networks (Delporte et al., 2013), images or other types of data (Nedelec, Smirnova, and Vasile, 2017; Kim and Choi, 2014; Xu, Jin, and Z.-H. Zhou, 2013). More recently, deep learning techniques have been used for this purpose. YouTube is extracting features from the videos browsed within a user history in order to improve their recommendations (Covington, Adams, and Sargin, 2016). Researchers have also analyzed music content as extra information (Van den Oord, Dieleman, and Schrauwen, 2013). They managed to recover explainable latent features, corresponding to certain types of music, without any human annotation. Such side information

[1] http://knowyourmeme.com/memes/events/balse

is particularly useful in order to mitigate the cold-start problem (Wei et al., 2017; Kula, 2015; Biswas, Lakshmanan, and Ray, 2017; Bobadilla et al., 2012). In the exact context of movies, Zhao et al. (2016) extract latent features from the posters using CNNs and improve the recommendations using those latent features. However, those extracted features do not have semantic meaning, therefore they cannot be used to explain to the user why extra works are recommended to them.

Several approaches have tried to bridge the gap between content-based approaches and collaborative filtering (Burke, 2002). The main idea behind those so-called hybrid methods is to combine different recommendation models in order to overcome their limitations and build more robust models with higher predictive power. Existing techniques can take on several names: blending or stacking (Roda, Costa, and Liberti, 2011; Jahrer, Töscher, and Legenstein, 2010), or the general ensemble methods for machine learning estimators. These techniques use the output of different models as features for a higher-level model. This higher-level model is usually a linear model (Sill et al., 2009). Such blended methods have played an important role in achieving top performance in challenges such as the Netflix Prize (Sill et al., 2009; Koren, 2009). The approach described in this paper builds upon these ideas, as we are presenting a blended model, but the combination we present is nonlinear. We complement a classical collaborative filtering recommender system with a fallback model that will compensate the prediction error on hard-to-predict data points, i.e. items with few ratings.

III. Context

We assume the typical setting for collaborative filtering: we have access to a $n \times m$ rating matrix R containing the ratings of n users on m items that can be either manga or anime: r_{ij} represents the rating that user i gave to item j. In practice, since users only rate the few items that they have interacted with, the rating matrix R tend to be very sparse: in the dataset considered in this paper, less than 1% of the entries are known; other popular datasets in the field (Aggarwal, 2016; Huazheng Wang, Wu, and Hongning Wang, 2016) report similar levels of sparsity. Therefore, it is challenging to infer the missing entries of R.

Another assumption is that the whole rating matrix can be explained by few latent profiles, i.e. each user rating vector can be decomposed as a combination of few latent vectors. Therefore, matrix completion is usually performed using matrix factorization: we try to devise a factorization $R \approx UV^T$ where a $n \times r$ matrix U represents the user feature vectors and a $m \times r$ matrix V represents the item feature vectors. Once this model is trained, that is, when the available entries of R match their counterparts in UV^T, computing a missing entry (i, j) of the rating matrix R is simply performed by looking at the corresponding (i, j) entry of UV^T, or, equivalently, computing the dot product $U_i^T V_j$ where U_i in the i-th row of U and V_j is the j-th row of V.

Finally, we also assume that we have access to the posters of some of the items. This is all the content we have.

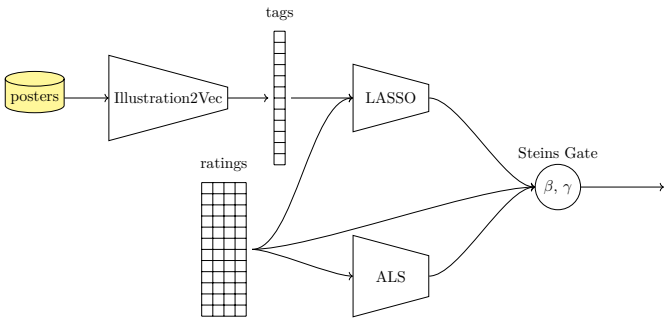

Fig. 1. The BALSE architecture.

IV. Our Model: BALSE

We now describe BALSE (Blended Alternate Least Squares with Explanation), our model for recommending anime and mangas. The main idea is to rely on the rating matrix when possible, and on the posters when rating information barely exists. We expect a nonlinear blending of two models considering these sources of information to achieve higher performance than any of the models. BALSE is composed of several blocks:

- an Illustration2Vec block, which is a convolutional neural network that takes a poster as input and outputs tag predictions;
- an ALS[2] block, that performs a matrix factorization of the rating matrix for collaborative filtering using alternate least squares with λ-weighted regularization;
- a LASSO[3] block, that performs a regularized linear regression of each row of the rating matrix, using tag predictions, in order to infer explainable user preferences;
- a Steins gate, that performs a blending of the outputs of ALS and LASSO models, in order to overcome their limitations and provide a final rating value.

The main architecture of our model is presented in Figure 1. Both posters and ratings are used for the predictions.

A. *Illustration2Vec*

This block extracts tag information from the posters, such as "1girl" or "weapon". Such tags are associated with confidence weights that represent how likely a certain tag appears in a certain poster. Formally, from the poster database, we want to extract a $m \times t$ matrix T where m is the number of items and t is the number of possible tags, such that $t_{jk} \in [0, 1]$ represents how likely a tag k appears in the poster of item j. T is computed using Illustration2Vec (Saito and Matsui, 2015), a VGG-16 neural network (Simonyan and Zisserman, 2014) that predicts a variety of tags based on illustrations, pre-trained on ImageNet and trained on manga illustrations labeled with tags from the community website Danbooru. We use the implementation provided by the authors, which is freely available. The output of the network is for each poster j, a vector $T_j = (t_{j1}, \ldots, t_{jt})$ where for tag $k = 1, \ldots, t$, component

[2]ALS stands for Alternate Least Squares.

[3]LASSO stands for Least Absolute Shrinkage and Selection Operator.

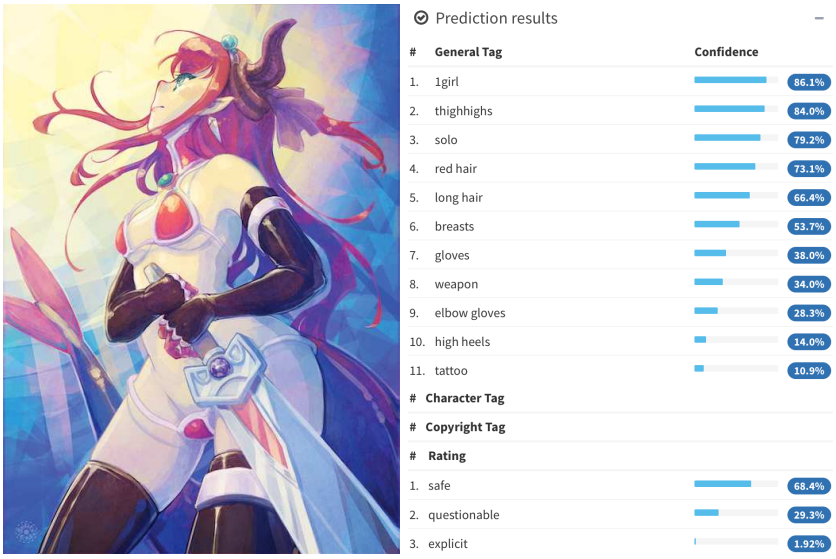

Fig. 2. Example of tag prediction on a manga illustration. ©Solène Pichereau, http://sedeto.fr.

$t_{jk} \in [0, 1]$ represents how likely tag k describes the poster of item j. In other words, the output of Illustration2Vec is a row of matrix T. We will call such a vector a tag prediction. See Fig. 2 for an example of an output of the Illustration2Vec model.

B. LASSO

The LASSO block approximates the rating matrix R with a regularized linear regression model called LASSO (Tibshirani, 1996), using the tag predictions as features for the items. We train a LASSO model for every user in the train set.

$$R \approx PT^T$$

where:

- P contains the parameters to learn, a $n \times t$ matrix of user preferences, of which the i-th row is denoted as P_i (likewise, R_i denotes the i-th row of R);
- T is the given $m \times t$ matrix of tag predictions for each item.

LASSO comes with an α parameter which induces a L1 regularization term to prevent overfitting, and to provide explanation of user preferences as we will show later. Therefore, for every user i of the train set, we estimate the parameters P_i that minimize:

$$\frac{1}{2\mathcal{N}_i}\|R_i - P_i T^T\|_2^2 + \alpha\|P_i\|_1$$

where $\mathcal{N}_i$ is the number of items rated by user i.

The output of the LASSO block is a rating prediction for each pair (i, j):

$$\hat{r}_{ij}^{LASSO} = \tau(P_i^T T_j).$$

where $\tau : x \mapsto \max(\min(x, 2), -2)$ is a function that shrinks its input to values between -2 and 2. Such a function prevents the regressor from providing invalid predictions that are outside the range of rating values.

C. ALS

The ALS block performs matrix factorization of the $n \times m$ sparse rating matrix R, in order to provide an estimate $\hat{r}_{ij}^{ALS}$ for the missing entries (i, j). Thus, we learn the parameters of the following factorization:

$$R \approx UV^T$$

where:

- U is the $n \times r$ matrix of user latent vectors;
- V is the $m \times r$ matrix of item latent vectors.

In order to avoid overfitting, we regularize the parameters to estimate. Therefore, as we want to minimize the squared error, the loss function to minimize has the following form:

$$\sum_{i,j|r_{ij}\neq 0} (r_{ij} - U_i^T V_j)^2 + \lambda\left(\|U_i\|_2^2 + \|V_j\|_2^2\right)$$

where U_i for every $i = 1, \ldots, n$ are the rows of U and V_j for every $j = 1, \ldots, m$ are the rows of V, and λ is a regularization parameter. This estimation is made by using alternate least squares with weighted λ-regularization (ALS-WR) (Y. Zhou et al., 2008).

Once the parameters have been learned, the prediction for rating of user i on item j is:

$$\hat{r}_{ij}^{ALS} = U_i^T V_j.$$

D. Steins Gate

At this step, we have predictions from two different blocks: ALS trained on the ratings and LASSO trained on the tag predictions of the posters. We want to improve the predictive power of the overall model, thus we learn a rule that would

automatically choose the best model according to the number of ratings of the item considered. Formally, we want to learn parameters β and γ such that:

$$\hat{r}_{ij}^{BALSE} = \sigma(\beta(\mathcal{R}_j - \gamma))\hat{r}_{ij}^{ALS} + (1 - \sigma(\beta(\mathcal{R}_j - \gamma)))\,\hat{r}_{ij}^{LASSO}$$

where:

- $\mathcal{R}_j$ is the number of ratings of the item j;
- $\hat{r}_{ij}^{ALS}$ is the rating prediction of ALS model for user i on item j;
- $\hat{r}_{ij}^{LASSO}$ is the rating prediction of LASSO model for user i on item j;
- $\sigma : x \mapsto 1/(1 + e^{-x})$ is the sigmoid function.

The intuition behind this formula is the following: we want to find a threshold γ such that when the number of ratings of item j verifies $R_j \gg \gamma$, BALSE mimics ALS, e.g., $\hat{r}_{ij}^{BALSE} \approx \hat{r}_{ij}^{ALS}$, while when $R_j \ll \gamma$, i.e. in a cold-start setting, BALSE mimics LASSO, e.g., $\hat{r}_{ij}^{BALSE} \approx \hat{r}_{ij}^{LASSO}$. β is just a scaling parameter that indicates how sharp the passage from LASSO to ALS will be. Formally, we want to estimate the parameters β and γ that minimize:

$$\sum_{i,j|r_{ij}\neq 0} \left(\hat{r}_{ij}^{BALSE} - r_{ij}\right)^2.$$

This formula is differentiable with respect to γ, thus it makes its optimization easier. It can be seen as a soft switch between the two possible predictions (ALS and LASSO), according to the number of ratings of the item. The parameters β and γ are learned using gradient descent.

V. Experiments and Results

A. Mangaki dataset

Mangaki[4] (Vie, Laïly, and Pichereau, 2015) is a website where people can rate items that represent either manga or anime, and receive recommendations based on their ratings. Mangaki can be seen as an open source version of Movielens (Harper and Konstan, 2015) for manga and anime. The Mangaki dataset is a 2079×9979 anonymized matrix of 334390 ratings from 2079 users on 9979 items. 80% of the items have a poster.

Users can either rate an item with {favorite, like, neutral, dislike} if they watched it, or {willsee, wontsee} if they did not watch it, i.e. testify whether they want to watch it or not, based on the content presented: poster, possibly synopsis, or some statistics.

B. Models

The models considered in this benchmark are:

- ALS: alternate least squares with weighted λ-regularization from (Y. Zhou et al., 2008), that ignores posters;

[4]https://mangaki.fr

Fig. 3. This is Steins gate's choice: $\gamma = 0.79040$. For items having at least one rating, it is better to rely more on the ratings predicted by ALS than by LASSO.

- LASSO: regularized linear regression using ratings and the tag predictions from Illustration2Vec, that is content-based;
- BALSE: the proposed method.

In practice, we use $\lambda = 0.1$ and rank $r = 20$ for every ALS model trained and $\alpha = 0.01$ for every LASSO model trained. Ratings are mapped into custom values: (favorite, like, neutral, dislike) = (4, 2, 0.1, -2) and (willsee, wontsee) = (0.5, -0.5). The Steins gate is optimized using gradient descent with exponential decay implemented in TensorFlow. The learning rate starts at 0.9 and decays every 20 steps with a base of 0.997. All the code is available on our GitHub repository[5].

C. 5-fold cross validation

We perform a 5-fold cross validation over the triplets (i, j, r_{ij}) of the database, keeping 30% of the train set as a validation set. Therefore, our data is split into a train set (56%), a validation set (24%) and a test set (20%).

The vanilla models ALS and LASSO are trained on both the train set and the validation set. For BALSE, the ALS and LASSO blocks are first trained using the train set only, and the Steins gate parameters β and γ are trained using the validation set[6], in order to prevent overfitting. For the final predictions of BALSE, blending is performed using the learned β and γ parameters, and the vanilla ALS and LASSO models. Finally, the root mean squared error (RMSE) is computed over the test set.

We distinguish the performance of all three models on three sets: the whole test set, a set of little-known items that received less than 3 ratings in the train and validation set (that represents 1000 ratings, therefore 3% of the test set), and cold-start items, i.e. items that were never seen in the train and validation sets.

D. Results

BALSE achieves a comparable performance than ALS overall, but substantially improves the recommendations on little-known items, see Table I.

The learned parameter γ of the Steins gate was less than 1, see Figure 3, which means that items having at least 1 rating can start to rely on ALS (their ratings) more than LASSO (their poster) for the predictions. However, BALSE provides

[5]https://github.com/mangaki/balse

[6]Please also note that in Steins gate, the number of ratings R_j of item j is computed over the train set.

TABLE I
RESULTS OF RMSE ON VARIOUS SUBSETS OF THE TEST SET.

RMSE	Whole test set	Little-known items (3% least rated)	Cold-start items (0 rating)
ALS	1.15681 ± 0.004	1.29269 ± 0.029	1.50047 ± 0.03500
LASSO	1.44444 ± 0.002	1.31331 ± 0.036	1.37909 ± 0.05600
BALSE	**1.14954 ± 0.004**	**1.22714 ± 0.036**	**1.34533 ± 0.04500**

better predictions than ALS for cold-start items, because ALS was not trained on them in the train set, therefore it outputs constant predictions.

ALS converges after 10 iterations. Steins gate takes 15k iterations to converge. LASSO is the bottleneck of the proposed approach because one LASSO model should be trained per user that appears in the train set.

E. Explanation of user taste

Using the tags, it is possible to provide an explanation of the taste of any user i using the preference matrix P learned by LASSO, because the columns of P_i are labeled with tags.

LASSO has been appreciated for its explainability (Tibshirani, 1996): the row preferences of each user are sparse, allowing to capture the tags that explain best the ratings of every user.

As an example, for a certain user among the authors, LASSO or BALSE report that his six most preferred tags are: *kneehighs*, *cat*, *serafuku*[7], *twin braids*, *japanese clothes* and *angry* whereas his six most disliked tags are: *pleated skirt*, *standing*, *silver hair*, *window*, *torn clothes* and *skirt*. Using this information, LASSO or BALSE can explain a recommendation: "We recommend to you the anime *Chivalry of a Failed Knight*, because there is a girl with *twin braids*, *serafuku* and *japanese clothes*" or a warning: "You might not like the anime *The Asterisk War: The Academy City on the Water* because there is a girl with a *pleated skirt*, even though there are *kneehighs* and *serafuku*."

VI. CONCLUSION AND FUTURE WORK

We proposed BALSE, a model for recommending anime and manga that makes use of information that is automatically extracted from posters. We showed that our model performs better than the baseline models, especially in the item cold-start scenario.

This paper is a proof a concept and the obtained results are very encouraging. Indeed, the blending Steins gate is such that any improvement made on any block would improve the overall performance of the approach. As future work, we plan to replace blocks in our architecture with more complex models: Illustration2Vec could be replaced with residual networks (He et al., 2016), ALS could be replaced with factorization machines (Rendle, 2010) or co-factorization (Fang and Si, 2011), LASSO could be replaced with Localized Lasso (Yamada et al., 2016), a variant that works well for few samples, many features. We also to integrate more side information, for instance the drawing style of the image, or tags coming from open databases such as AniDB[8] or AniList[9], in order to improve the explanation of the users' preferences.

For the sake of simplicity, we mapped the categorical ratings like, dislike, etc. to ad-hoc values, but we could instead use ordinal regression methods (Pedregosa, Bach, and Gramfort, 2017). However, they require more computation to be trained properly. Ensemble methods that blend more than two models could be considered (Sill et al., 2009), or that rely not also on the number of ratings provided for a certain item, but on the number of ratings provided by a certain user, or the number of works that contain a certain tag.

Here, we mitigated the problem of item cold-start recommendation through the use of extra information on the item side. Obviously, similar results could be obtained for the user cold-start problem, provided enough data is available to describe the users.

Using BALSE, recommender systems can automatically replenish their database, where new items go through the tag prediction track and the explainable model in order to justify the recommendations for their first users, and automatically go to the main track when sufficient ratings have been collected.

ACKNOWLEDGMENTS

This work was carried out while Florian Yger was a visiting researcher at RIKEN Center for AIP, and Kévin Cocchi and Thomas Chalumeau were interns at Mangaki. We would like to thank Nicolas Hug and Étienne Simon for their helpful comments and Solène Pichereau for kindly providing the example illustration of Figure 2.

REFERENCES

Aggarwal, Charu C (2016). *Recommender systems*. Springer (cit. on pp. 1, 2).

Alexandridis, Georgios, Georgios Siolas, and Andreas Stafylopatis (2017). "ParVecMF: A Paragraph Vector-based Matrix Factorization Recommender System". In: *arXiv preprint arXiv:1706.07513* (cit. on p. 1).

Biswas, Sampoorna, Laks VS Lakshmanan, and Senjuti Basu Ray (2017). "Combating the Cold Start User Problem in Model Based Collaborative Filtering". In: *arXiv preprint arXiv:1703.00397* (cit. on p. 2).

Bobadilla, JesúS, Fernando Ortega, Antonio Hernando, and JesúS Bernal (2012). "A collaborative filtering approach to mitigate the new user cold start problem". In: *Knowledge-Based Systems* 26, pp. 225–238 (cit. on p. 2).

[7] *Serafuku* means "Japanese school uniform".

[8] http://anidb.net

[9] https://anilist.co

Burke, Robin (2002). "Hybrid recommender systems: Survey and experiments". In: *User modeling and user-adapted interaction* 12.4, pp. 331–370 (cit. on p. 2).

Covington, Paul, Jay Adams, and Emre Sargin (2016). "Deep neural networks for youtube recommendations". In: *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, pp. 191–198 (cit. on p. 1).

Delporte, Julien, Alexandros Karatzoglou, Tomasz Matuszczyk, and Stéphane Canu (2013). "Socially enabled preference learning from implicit feedback data". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer Berlin Heidelberg, pp. 145–160 (cit. on p. 1).

Fang, Yi and Luo Si (2011). "Matrix co-factorization for recommendation with rich side information and implicit feedback". In: *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*. ACM, pp. 65–69 (cit. on pp. 1, 5).

Harper, F. Maxwell and Joseph A. Konstan (2015). "The MovieLens Datasets: History and Context". In: *ACM Trans. Interact. Intell. Syst.* 5.4, 19:1–19:19. ISSN: 2160-6455. DOI: 10.1145/2827872 (cit. on p. 4).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778 (cit. on p. 5).

Jahrer, Michael, Andreas Töscher, and Robert Legenstein (2010). "Combining predictions for accurate recommender systems". In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 693–702 (cit. on p. 2).

Kim, Yong-Deok and Seungjin Choi (2014). "Scalable variational bayesian matrix factorization with side information". In: *Artificial Intelligence and Statistics*, pp. 493–502 (cit. on p. 1).

Koren, Yehuda (2009). "The bellkor solution to the netflix grand prize". In: *Netflix prize documentation* 81, pp. 1–10 (cit. on p. 2).

Kula, Maciej (2015). "Metadata embeddings for user and item cold-start recommendations". In: *arXiv preprint arXiv:1507.08439* (cit. on pp. 1, 2).

Nedelec, Thomas, Elena Smirnova, and Flavian Vasile (2017). "Specializing Joint Representations for the task of Product Recommendation". In: *arXiv preprint arXiv:1706.07625* (cit. on p. 1).

Pedregosa, Fabian, Francis Bach, and Alexandre Gramfort (2017). "On the consistency of ordinal regression methods". In: *Journal of Machine Learning Research* 18.55, pp. 1–35 (cit. on p. 5).

Rendle, Steffen (2010). "Factorization machines". In: *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, pp. 995–1000 (cit. on p. 5).

Roda, Fabio, Alberto Costa, and Leo Liberti (2011). "Optimal recommender systems blending". In: *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*. ACM, p. 60 (cit. on p. 2).

Saito, Masaki and Yusuke Matsui (2015). "Illustration2Vec: A Semantic Vector Representation of Illustrations". In: *SIGGRAPH Asia Technical Briefs* (cit. on p. 2).

Sill, Joseph, Gábor Takács, Lester Mackey, and David Lin (2009). "Feature-weighted linear stacking". In: *arXiv preprint arXiv:0911.0460* (cit. on pp. 2, 5).

Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (cit. on p. 2).

Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288 (cit. on pp. 3, 5).

Van den Oord, Aaron, Sander Dieleman, and Benjamin Schrauwen (2013). "Deep content-based music recommendation". In: *Advances in Neural Information Processing Systems*, pp. 2643–2651 (cit. on p. 1).

Vie, Jill-Jênn, Camille Laïly, and Solène Pichereau (2015). *Mangaki: an Anime/Manga Recommender System with Fast Preference Elicitation*. Tech. rep. (cit. on p. 4).

Wang, Huazheng, Qingyun Wu, and Hongning Wang (2016). "Learning Hidden Features for Contextual Bandits". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, pp. 1633–1642 (cit. on p. 2).

Wei, Jian, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang (2017). "Collaborative filtering and deep learning based recommendation system for cold start items". In: *Expert Systems with Applications* 69, pp. 29–39 (cit. on p. 2).

Xu, Miao, Rong Jin, and Zhi-Hua Zhou (2013). "Speedup matrix completion with side information: Application to multi-label learning". In: *Advances in Neural Information Processing Systems*, pp. 2301–2309 (cit. on p. 1).

Yamada, Makoto, Koh Takeuchi, Tomoharu Iwata, John Shawe-Taylor, and Samuel Kaski (2016). "Localized Lasso for High-Dimensional Regression". In: *arXiv preprint arXiv:1603.06743* (cit. on p. 5).

Zhao, Lili, Zhongqi Lu, Sinno Jialin Plan, and Qiang Yang (2016). "Matrix factorization+ for movie recommendation". In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press, pp. 3945–3951 (cit. on p. 2).

Zhou, Yunhong, Dennis Wilkinson, Robert Schreiber, and Rong Pan (2008). "Large-scale parallel collaborative filtering for the netflix prize". In: *Lecture Notes in Computer Science* 5034, pp. 337–348 (cit. on pp. 3, 4).