Low-rank Linear Cold-Start Recommendation from Social Data

Suvash Sedhain[†*], Aditya Krishna Menon[*†], Scott Sanner[‡†], Lexing Xie[†*], Darius Braziunas[S]
{[†]Australian National University, [*]Data61}, Canberra, ACT, Australia
[‡] University of Toronto, Toronto, Canada
[S] Rakuten Kobo Inc., Toronto, Canada
suvash.sedhain@anu.edu.au, aditya.menon@data61.csiro.au, ssanner@mie.utoronto.ca, lexing.xie@anu.edu.au, dbraziunas@kobo.com

Abstract

The cold-start problem involves recommendation of content to new users of a system, for whom there is no historical preference information available. This proves a challenge for collaborative filtering algorithms that inherently rely on such information. Recent work has shown that *social* metadata, such as users' friend groups and page likes, can strongly mitigate the problem. However, such approaches either lack an interpretation as optimising some principled objective, involve iterative non-convex optimisation with limited scalability, or require tuning several hyperparameters. In this paper, we first show how three popular cold-start models are special cases of a *linear content-based model*, with implicit constraints on the weights. Leveraging this insight, we propose *LoCo*, a new model for cold-start recommendation based on three ingredients: (a) *linear regression* to learn an optimal weighting of social signals for preferences, (b) a *low-rank parametrisation* of the weights to overcome the high dimensionality common in social data, and (c) scalable learning of such low-rank weights using *randomised SVD*. Experiments on four real-world datasets show that LoCo yields significant improvements over state-of-the-art cold-start recommenders that exploit high-dimensional social network metadata.

Introduction

Collaborative filtering has emerged as the gold-standard approach to personalised recommendation of content to users (Leavitt 2013). The central idea of collaborative filtering is to infer a user's preferences based on their past interactions with a system, as well as the preferences of other like-minded users (Goldberg et al. 1992; Resnick et al. 1994). While this approach has seen considerable success, it has an obvious failure mode: how do we recommend content to new users without any historical preference information? This is known as the *user cold-start* problem (Schein et al. 2002), and is pervasive in real-world recommendation applications.

Cold-start problems may be addressed by exploiting exogenous *side-information* about users, such as demographic attributes. This can be done using content-based rather than collaborative filtering, where the central idea of the former is to infer a user's preferences by explaining their past interactions based on the side-information (Pazzani and Billsus 2007). The cold-start problem does not plague such approaches, and has thus been addressed both by vanilla content-based filtering recommenders (Billsus and Pazzani 1999; Mooney and Roy 2000) as well as hybrid collaborative and content-based filtering recommenders (Schein et al. 2002; Gunawardana and Meek 2009).

The precise form of side-information available has an impact on the accuracy of cold-start predictions (Gantner et al. 2010; Sedhain et al. 2014). Recent work has shown that *social* information, such as users' friend groups and page likes, is sufficiently rich to strongly mitigate the cold-start problem. Various means of incorporating this information into neighbourhood (Zhang et al. 2010; Sahebi and Cohen 2011; Sedhain et al. 2014; Rosli et al. 2014; Rohani et al. 2014) and matrix factorisation (or latent feature) approaches (Ma et al. 2008; Cao, Liu, and Yang 2010; Jamali and Ester 2010; Noel et al. 2012; Krohn-Grimberghe et al. 2012) have been studied, with encouraging results. However, both strands of work have limitations. Neighbourhood methods lack an interpretation as minimising some principled objective, potentially resulting in sub-optimal solutions. Matrix factorisation methods, on the other hand, involve time-consuming iterative optimisation of a non-convex objective and require tuning of a potentially large number of hyperparameters.

This paper proposes an efficient, accurate, learning-based approach for the cold-start problem that leverages social data. Our first contribution is to show how three popular cold-start models (Sedhain et al. 2014; Gantner et al. 2010; Krohn-Grimberghe et al. 2012) can be seen as a special case of a *linear content-based model*, which explains some of their drawbacks. Leveraging this insight, our second contribution is a new model, *LoCo*, that overcomes these limitations by employing three ingredients:

(a) *multivariate linear regression* to learn an optimal weighting of social signals for preferences,
(b) a *low-rank parametrisation* of the regression weights to address the high dimensionality common in social data,
(c) highly scalable learning of such low-rank weights via *randomised SVD* (Halko, Martinsson, and Tropp 2011).

While each of these ideas is simple, using them in conjunction is powerful: experiments on four real-world datasets demonstrate that LoCo yields substantial improvements over state-of-the-art cold-start recommenders leveraging high-dimensional side-information from a social network.

Background and notation

Suppose we have a database of U users and I items. Let $\mathbf{R} \in \{0,1\}^{U \times I}$ denote a purchase[1] matrix, where $\mathbf{R}[u,i] = 1$ means that user u purchased item i. Let $\mathbf{R}[:,i] \in \{0,1\}^{U}$ denote the vector of item purchases. In many applications, we additionally have a side-information (or metadata) matrix $\mathbf{X} \in \mathbb{R}^{U \times P}$. We will think of $\mathbf{X}_{up}$ being whether or not user u "likes" a webpage p, though $\mathbf{X}$ could equally reflect *e.g.* users' group memberships, friend circles, *et cetera*.

Collaborative and content-based filtering

The three high-level approaches to personalised recommendation may be summarised as follows:

(1) *Content-based filtering*: exploit correlations between user side-information $\mathbf{X}$ and item preferences $\mathbf{R}$, *e.g.* by deriving metadata-based user similarities (Billsus and Pazzani 1999), or learning metadata-to-preference classifiers (Mooney and Roy 2000);

(2) *Collaborative filtering*: exploit correlations amongst the preferences $\mathbf{R}$ of all users, *e.g.* by k-nearest neighbour recommendation (Herlocker et al. 1999) or matrix factorisation (Koren, Bell, and Volinsky 2009);

(3) *Hybrid filtering*: exploit both forms of correlation (Basu, Hirsh, and Cohen 1998; Melville, Mooney, and Nagarajan 2002; Basilico and Hofmann 2004).

As k-nearest neighbour and matrix factorisation methods feature heavily in the sequel, we briefly summarise them here. In (user) k-nearest neighbour approaches, one models

$$\mathbf{R} \approx \mathbf{S}\mathbf{R}_{\mathrm{tr}} \tag{1}$$

for some pre-defined similarity matrix $\mathbf{S}$, typically based on the cosine similarity of $\mathbf{R}$ with itself. In matrix factorisation approaches, one models

$$\mathbf{R} \approx \mathbf{U}\mathbf{V} \tag{2}$$

for some latent representations $\mathbf{U} \in \mathbb{R}^{U \times K}, \mathbf{V} \in \mathbb{R}^{K \times I}$ with *latent dimensionality* $K \ll \min(U, I)$. The parameters $\mathbf{U}, \mathbf{V}$ are typically estimated by solving

$$\min_{\mathbf{U},\mathbf{V}} ||\mathbf{R} - \mathbf{U}\mathbf{V}||_F^2 + \frac{\lambda_U}{2}||\mathbf{U}||_F^2 + \frac{\lambda_V}{2}||\mathbf{V}||_F^2. \tag{3}$$

The user cold-start problem

The recommendation problem we consider is the *cold-start* scenario, where a user has no prior purchases. We split the set of users into the (training) U_{tr} *warm-start* users with at least one purchase, and the rest as the (test) U_{te} *cold-start* users. We denote the corresponding slices of the purchase matrix by $\mathbf{R}_{\mathrm{tr}} \in \mathbb{R}^{U_{\mathrm{tr}} \times I}, \mathbf{R}_{\mathrm{te}} \in \mathbb{R}^{U_{\mathrm{te}} \times I}$, where by definition $\mathbf{R}_{\mathrm{te}} = \mathbf{0}$. Our interest will be in producing $\hat{\mathbf{R}}_{\mathrm{te}} \in \mathbb{R}^{U_{\mathrm{te}} \times I}$, a recommendation matrix for the cold-start users.

Personalised recommendation for cold-start users is intuitively impossible from $\mathbf{R}$ alone. But suppose we have a metadata matrix $\mathbf{X}$, with $\mathbf{X}_{\mathrm{tr}}, \mathbf{X}_{\mathrm{te}}$ being the metadata for the warm- and cold-start users respectively. Then, we might hope to leverage correlations between $\mathbf{X}_{\mathrm{tr}}$ and $\mathbf{R}_{\mathrm{tr}}$ to make meaningful predictions for the cold-start users.

[1]More generally, *purchases* can be substituted by any positive interactions between users and items.

Approaches to (social) cold-start recommendation

We summarise the various approaches to exploiting (social) side-information to ameliorate the cold-start problem.

Neighbourhood + metadata similarity In cold-start scenarios, one cannot use a neighbourhood model (Equation 1) with a similarity $\mathbf{S}$ computed from $\mathbf{R}_{\mathrm{te}}$ since, by definition, $\mathbf{R}_{\mathrm{te}} = \mathbf{0}$. One can however compute new similarity metrics based on metadata (Billsus and Pazzani 1999).

Recently, several works have designed $\mathbf{S}$ based on social information (Zhang et al. 2010; Sahebi and Cohen 2011; Sedhain et al. 2014; Rosli et al. 2014; Rohani et al. 2014). For example, (Sedhain et al. 2014) proposed

$$\hat{\mathbf{R}}_{\mathrm{te}} = \mathbf{X}_{\mathrm{te}} \odot \mathbf{X}_{\mathrm{tr}}^T \star \mathbf{R}_{\mathrm{tr}} \tag{4}$$

where $\odot, \star$ refer to generalised matrix operations. When $\star$ is the standard inner product, this is a neighbourhood method with metadata-derived similarity $\mathbf{S} = \mathbf{X}_{\mathrm{te}} \odot \mathbf{X}_{\mathrm{tr}}^T$.

Matrix factorisation with regularisation In cold-start scenarios, one cannot use a matrix factorisation model (Equation 2) with $\mathbf{U}$ estimated as per Equation 3 since it is optimal for $\mathbf{U}_{\mathrm{te}} = \mathbf{0}$. One can however regularise the latent features based on metadata similarity (Ma et al. 2008; Agarwal and Chen 2009; Cao, Liu, and Yang 2010; Jamali and Ester 2010; Yang et al. 2011; Noel et al. 2012; Krohn-Grimberghe et al. 2012). For example, (Krohn-Grimberghe et al. 2012) proposed an objective based on *collective matrix factorisation* (CMF) (Singh and Gordon 2008):

$$\begin{gathered}\min_{\mathbf{U},\mathbf{V},\mathbf{Z}} ||\mathbf{R} - \mathbf{U}\mathbf{V}||_F^2 + \mu||\mathbf{X} - \mathbf{U}\mathbf{Z}||_F^2 + \Omega(\mathbf{U},\mathbf{V},\mathbf{Z}), \\ \Omega(\mathbf{U},\mathbf{V},\mathbf{Z}) = \frac{\lambda_U}{2}||\mathbf{U}||_F^2 + \frac{\lambda_V}{2}||\mathbf{V}||_F^2 + \frac{\lambda_Z}{2}||\mathbf{Z}||_F^2.\end{gathered} \tag{5}$$

where $\mathbf{U} \in \mathbb{R}^{U \times K}, \mathbf{V} \in \mathbb{R}^{K \times I}$, and $\mathbf{Z} \in \mathbb{R}^{K \times P}$ for some latent dimensionality $K \ll \min(U, I)$. Intuitively, we find a latent subspace $\mathbf{U}$ for users that is jointly predictive of both their preferences and social characteristics. We then predict

$$\hat{\mathbf{R}}_{\mathrm{te}} = \mathbf{U}_{\mathrm{te}}\mathbf{V}. \tag{6}$$

While this prediction has the same form as Equation 2, the estimation of $\mathbf{U}_{\mathrm{te}}$ here will be non-trivial owing to the additional regularisation derived from requiring it to model $\mathbf{X}_{\mathrm{te}}$.

An alternate but less generic approach is the fLDA model (Agarwal and Chen 2010), which combines matrix factorisation and LDA when the metadata comprises textual features.

Matrix factorisation with feature mapping Matrix factorisation approaches may also be adapted to the cold-start regime via a two-step model. Here, the first step is to model the warm-start users by $\mathbf{R}_{\mathrm{tr}} \approx \hat{\mathbf{R}}_{\mathrm{tr}} = \mathbf{U}_{\mathrm{tr}}\mathbf{V}$, with latent features $\mathbf{U}_{\mathrm{tr}}, \mathbf{V}$ as before. The second step is to learn a mapping between the side-information $\mathbf{X}_{\mathrm{tr}}$ and latent features $\mathbf{U}_{\mathrm{tr}}$. This mapping is used to estimate $\mathbf{U}_{\mathrm{te}}$ from $\mathbf{X}_{\mathrm{te}}$, with predictions then made as per Equation 6.

A canonical example of this approach is *BPR-LinMap* (Gantner et al. 2010), where in the second step the mapping may be done via linear regression, so that one estimates

$$\mathbf{U}_{\mathrm{te}} = \mathbf{X}_{\mathrm{te}}\mathbf{T} \tag{7}$$

where $\mathbf{T} \in \mathbb{R}^{P \times K}$ is chosen so that $\mathbf{U}_{\mathrm{tr}} \approx \mathbf{X}_{\mathrm{tr}}\mathbf{T}$ via

$$\min_{\mathbf{T}} \|\mathbf{U}_{\mathrm{tr}} - \mathbf{X}_{\mathrm{tr}}\mathbf{T}\|_F^2 + \frac{\lambda_T}{2}\|\mathbf{T}\|_F^2. \tag{8}$$

Method	Weight $\mathbf{W}$
Social neighbourhood	$\mathbf{X}_{\text{tr}}^T\mathbf{R}_{\text{tr}}$
CMF	$\mathbf{Z}_*^T(\mathbf{V}_*\mathbf{V}_*^T + \mathbf{Z}_*\mathbf{Z}_*^T)^{-1}\mathbf{V}_*$
BPR-LinMap	$(\mathbf{X}_{\text{tr}}^T\mathbf{X}_{\text{tr}})^{-1}\mathbf{X}_{\text{tr}}^T\hat{\mathbf{R}}_{\text{tr}}$
Linear regression	$(\mathbf{X}_{\text{tr}}^T\mathbf{X}_{\text{tr}})^{-1}\mathbf{X}_{\text{tr}}^T\mathbf{R}_{\text{tr}}$
LoCo	$\mathbf{V}_K(\mathbf{V}_K^T\mathbf{X}_{\text{tr}}^T\mathbf{X}_{\text{tr}}\mathbf{V}_K)^{-1}\mathbf{V}_K^T\mathbf{X}_{\text{tr}}^T\mathbf{R}_{\text{tr}}$

Table 1: Summary of choice of weight matrix $\mathbf{W}$ in linear model $\hat{\mathbf{R}}_{\text{te}} = \mathbf{X}_{\text{te}}\mathbf{W}$ that yields various cold-start recommenders. See text for definitions of $\mathbf{Z}_*, \mathbf{V}_*, \mathbf{V}_K$.

Classification approaches Another approach to cold-start recommendation is to learn a classifier from metadata to preferences, with examples of classifier choices being rule induction (Basu, Hirsh, and Cohen 1998), naïve Bayes (Mooney and Roy 2000) and bilinear regression (Park and Chu 2009). Such approaches involve a prediction of the form

$$\hat{\mathbf{R}}_{\text{te}} = f(\mathbf{X}_{\text{te}}), \tag{9}$$

for some learned function $f\colon \mathbb{R}^P \to \mathbb{R}$ applied row-wise.

A unified view of existing methods

We now provide a unified view of popular examples of each approach to cold-start recommendation discussed in the previous section. Specifically, we will relate them to perhaps the most natural form of cold-start recommendation matrix, the *linear content-based model*

$$\hat{\mathbf{R}}_{\text{te}} = \mathbf{X}_{\text{te}}\mathbf{W} \tag{10}$$

where $\mathbf{W} \in \mathbb{R}^{P\times I}$. Evidently, this is a special case of the classification approach of the previous section, with $f\colon \mathbf{x} \mapsto \mathbf{W}^T\mathbf{x}$. We now show that popular examples of each of the other three cold-start approaches can be viewed as special cases of Equation 10. We summarise our findings in Table 1.

Social neighbourhood model

Consider the social neighbourhood model of (Sedhain et al. 2014) (Equation 4). When $\odot, \star$ are standard inner product operations, the prediction for cold-start users is

$$\hat{\mathbf{R}}_{\text{te}} = \mathbf{X}_{\text{te}}(\mathbf{X}_{\text{tr}}^T\mathbf{R}_{\text{tr}}), \tag{11}$$

corresponding exactly to the linear model of Equation 10 with a weight matrix $\mathbf{W} = \mathbf{X}^T\mathbf{R}$. Recalling that $\mathbf{R}_{\text{te}} \equiv \mathbf{0}$, the predicted rating for cold-start user u and item i is thus

$$\hat{\mathbf{R}}[u,i] = \frac{1}{2}\mathbf{X}[u,:]\cdot\left(\sum_{\mathbf{R}[u',i]=1}\mathbf{X}[u',:] - \sum_{\mathbf{R}[u',i]=0}\mathbf{X}[u',:] + \mathbf{1}^T\mathbf{X}\right)^T.$$

The third term above is independent of i, and thus plays the role of a per-user bias that does not affect ranking. The first two terms correspond to a *nearest unnormalised centroid* classifier: we measure whether the social metadata $\mathbf{X}_{u:}$ for the given user is more similar to that of the unnormalised metadata centroid of the users that like item i, or those that dislike item i. The normalised centroid classifier is also known as the Rocchio classifier in information retrieval (Manning, Raghavan, and Schütze 2008), and is attained if we normalise the columns of $\mathbf{R}$ to sum to 1.

CMF model

The parameters of the CMF model (Equation 5) can be learned by iteratively optimising with respect to each individual parameter, keeping all others fixed. Each such individual optimisation is a regression problem, and thus admits a closed form solution. Suppose $\mathbf{Z}_*, \mathbf{V}_*$ are the optimal choices of $\mathbf{Z}, \mathbf{V}$, which will depend in some non-trivial way on $\mathbf{R}, \mathbf{X}$. Then, the unregularised optimal solution for $\mathbf{U}$ is

$$\mathbf{U} = (\mathbf{R}\mathbf{V}_*^T + \mathbf{X}\mathbf{Z}_*^T)(\mathbf{V}_*\mathbf{V}_*^T + \mathbf{Z}_*\mathbf{Z}_*^T)^{-1} \tag{12}$$

Thus, the cold-start recommendation matrix is

$$\hat{\mathbf{R}}_{\text{te}} = \mathbf{U}_{\text{te}}\mathbf{V}_* = \mathbf{X}_{\text{te}}\mathbf{Z}_*^T(\mathbf{V}_*\mathbf{V}_*^T + \mathbf{Z}_*\mathbf{Z}_*^T)^{-1}\mathbf{V}_*,$$

recalling $\mathbf{R}_{\text{te}} = \mathbf{0}$. This is a special case of Equation 10 for low-rank weight matrix $\mathbf{W} = \mathbf{Z}_*^T(\mathbf{V}_*\mathbf{V}_*^T + \mathbf{Z}_*\mathbf{Z}_*^T)^{-1}\mathbf{V}_*$.

BPR-LinMap model

For the BPR-LinMap model (Equations 6, 7), for optimal latent features $\mathbf{V}_*$ and regression weights $\mathbf{T}_*$, we have

$$\hat{\mathbf{R}}_{\text{te}} = \mathbf{U}_{\text{te}}\mathbf{V}_* = \mathbf{X}_{\text{te}}\mathbf{T}_*\mathbf{V}_*,$$

which is a special case of Equation 10 for $\mathbf{W} = \mathbf{T}_*\mathbf{V}_*$. One can further explicitly compute the optimal (λ_T-regularised) linear regression weights $\mathbf{T}_*$ from Equation 8,

$$\mathbf{T}_* = (\mathbf{X}_{\text{tr}}^T\mathbf{X}_{\text{tr}} + \lambda_T\mathbf{I})^{-1}\mathbf{X}_{\text{tr}}^T\mathbf{U}_{\text{tr}}, \tag{13}$$

from which we conclude the weight matrix is equivalently

$$\mathbf{W} = (\mathbf{X}_{\text{tr}}^T\mathbf{X}_{\text{tr}} + \lambda_T\mathbf{I})^{-1}\mathbf{X}_{\text{tr}}^T\hat{\mathbf{R}}_{\text{tr}}. \tag{14}$$

Pros and cons of existing approaches

We now assess the pros and cons of each of the above approaches, using their interpretation as instances of a linear content-based model to compare them on an equal footing.

The social neighbourhood model is simple to compute and intuitive, and has been shown to perform well (Sedhain et al. 2014). However, its choice of weights $\mathbf{W}$ is not explicitly based on optimising some principled objective. Further, it does not account for correlations amongst features in any way. The model thus risks *underfitting*, as correlations amongst page likes are intuitively useful to exploit.

The BPR-LinMap model has the opposite problem: while it accounts for feature correlations, it risks *overfitting* for high-dimensional $\mathbf{X}$, as the estimate of $\mathbf{T}$ (Equation 13) from solving a high-dimensional regression problem might be unreliable. This was indeed observed in the high-dimensional experiments in (Gantner et al. 2010). As high dimensional $\mathbf{X}$ is the expected scenario for social side-information, this is an important limitation.

The CMF model implicitly seeks a low-rank factorisation of the metadata matrix $\mathbf{X}$, and thus is suitable for high-dimensional social side-information. However, CMF has limited scalability by virtue of requiring tuning of at least five hyperparameters $(\mu, \lambda_U, \lambda_V, \lambda_Z, K)$.

To summarise, none of these existing methods simultaneously satisfies the desiderata of having an interpretation as optimising some principled objective; accounting for feature correlations; being suitable for high-dimensional data; and being highly scalable to train and tune. We seek to address these issues in the next section.

Our model: LoCo

We present a preliminary attempt at directly learning suitable weights $\mathbf{W}$, which leads to our LoCo model.

Warm-up: multivariate linear regression

The previous section established the cold-start predictions of existing methods as special cases of a linear content-based model (Equation 10). This viewpoint suggests a natural alternate approach to producing cold-start predictions: directly optimising the weights $\mathbf{W}$ in this linear model to minimise some principled objective. The most apparent approach to learn $\mathbf{W}$ is by performing multivariate linear regression:

$$\min_{\mathbf{W}} ||\mathbf{R}_{\text{tr}} - \mathbf{X}_{\text{tr}}\mathbf{W}||_F^2 + \frac{\lambda}{2}||\mathbf{W}||_F^2, \tag{15}$$

for which we have the closed form solution

$$\mathbf{W} = (\mathbf{X}_{\text{tr}}^T\mathbf{X}_{\text{tr}} + \lambda\mathbf{I})^{-1}\mathbf{X}_{\text{tr}}^T\mathbf{R}_{\text{tr}}. \tag{16}$$

This linear regression model satisfies several desiderata: it has a principled training objective, it captures feature correlations (through the weighting by the inverse of the covariance matrix $\mathbf{X}_{\text{tr}}^T\mathbf{X}_{\text{tr}}$), it has only a single hyperparameter (λ) to tune, and it is efficient to train for modest P.

On the other hand, for high-dimensional metadata where $P \gg U$, such a model may be prohibitive to train, and possibly unreliable to estimate (though the use of ℓ_2 regularisation mitigates this somewhat). We resolve these issues next.

LoCo: linear low-rank regression

We now propose our model for cold-start recommendation, which rests upon three simple but effective insights.

The first insight is that to make the linear regression model suitable for high-dimensional metadata, we can enforce the weight matrix $\mathbf{W}$ to be *low rank*, so that spurious correlations are avoided. We thus modify Equation 15 to be:

$$\min_{\mathbf{W}:\ \text{rank}(\mathbf{W}) \leq K} ||\mathbf{R}_{\text{tr}} - \mathbf{X}_{\text{tr}}\mathbf{W}||_F^2 + \frac{\lambda}{2}||\mathbf{W}||_F^2 \tag{17}$$

with $K \ll \min(U, I)$ some latent dimensionality.

Employing a rank constraint comes at a price: the objective of Equation 17 is now non-convex. Our second insight is that we can efficiently optimise over a *parametrised subset* of low-rank matrices. Let $\mathbf{X}_{\text{tr}} \approx \mathbf{U}_K\mathbf{S}_K\mathbf{V}_K^T$ be the rank-K SVD approximation of $\mathbf{X}_{\text{tr}}$. Now suppose that we optimise over $\mathbf{W}$ of the form $\mathbf{W} = \mathbf{V}_K\mathbf{Z}$, so that $\text{rank}(\mathbf{W}) \leq K$ automatically. This may be understood as a type of principal component regression (Hastie, Tibshirani, and Friedman 2009). Since $\mathbf{V}_K$ is orthonormal, the resulting objective is

$$\min_{\mathbf{Z}} ||\mathbf{R}_{\text{tr}} - \mathbf{X}_{\text{tr}}\mathbf{V}_K\mathbf{Z}||_F^2 + \frac{\lambda}{2}||\mathbf{Z}||_F^2.$$

Importantly, this has explicit closed form solution

$$\mathbf{Z} = (\mathbf{V}_K^T\mathbf{X}_{\text{tr}}^T\mathbf{X}_{\text{tr}}\mathbf{V}_K + \lambda\mathbf{I})^{-1}\mathbf{V}_K^T\mathbf{X}_{\text{tr}}^T\mathbf{R}_{\text{tr}}. \tag{18}$$

While we have a closed form solution for the weights $\mathbf{W}$, this relies on being able to compute the truncated SVD of $\mathbf{X}_{\text{tr}}$ efficiently. Naïvely, this requires a superlinear dependence on U, P. Our final insight is that an *approximate* computation of the SVD can be found efficiently using *randomised SVD* algorithms (Halko, Martinsson, and Tropp 2011) which have a linear dependence on U, P.

To summarise, we propose:

(a) a low-rank parameterisation of linear model weights;
(b) randomised SVD to project the metadata;
(c) multivariate linear regression to learn the weights.

We call our resulting model of Equation 18 with $\mathbf{V}_K$ computed by randomised SVD *LoCo*, for LOw-rank COld-start recommendation. This model has all the desiderata of linear regression, while additionally being suitable for high dimensional $\mathbf{X}$. Thus, it meets all the desiderata we listed as lacking in previous methods.

We mention that one could make LoCo nonlinear by passing $\mathbf{X}_{\text{tr}}\mathbf{V}_K$ through some nonlinear activation function $f(\cdot)$, mimicking a single layer neural network.

Comparison to existing methods

Compared to the social neighbourhood method, the LoCo weights $\mathbf{W}$ explicitly account for correlations among the social page likes by virtue of using the (projected) covariance matrix $\mathbf{V}_K^T\mathbf{X}_{\text{tr}}^T\mathbf{X}_{\text{tr}}\mathbf{V}_K$ to scale the weights in Equation 18.

Interestingly, BPR-LinMap (Equation 14) is identical to the linear regression model of Equation 16, with one crucial difference: in the former, one uses $\hat{\mathbf{R}}_{\text{tr}}$ in place of $\mathbf{R}_{\text{tr}}$ i.e. one replaces the regression target matrix by a low-rank approximation. Compared to BPR-LinMap, then, LoCo performs a low-rank approximation on $\mathbf{X}$, rather than $\mathbf{R}$. This is sensible because $\mathbf{X}$ is high dimensional for most social metadata, and we expect this $\mathbf{X}$ to have low-rank structure.

Compared to CMF, LoCo is highly scalable, since computing the randomised SVD of $\mathbf{X}_{\text{tr}}$ is much more efficient than iterative optimisation of Equation 5. Further, LoCo requires tuning of only two hyperparameters (K, λ).

More broadly, while BPR-LinMap and CMF also implicitly consider low-rank $\mathbf{W}$, this is a side-effect of the particular objective considered by these methods. We however enforce the low-rank contraint explicitly, and obtain our $\mathbf{W}$ through the optimisation of an objective explicitly targetted to cold-start users. By virtue of directly focussing on predicting preferences from the metadata, we expect LoCo to have superior performance to these methods. (This reasoning holds even if we were to provide BPR-LinMap with a low-dimensional projection of $\mathbf{X}_{\text{tr}}$ as input.)

Computational complexity of LoCo

The training complexity of LoCo is determined by that of computing the randomised SVD of $\mathbf{X}_{\text{tr}}$, and computing $\mathbf{Z}$. The former requires $O(K^2 \cdot (U + P))$ operations (Halko, Martinsson, and Tropp 2011). For the latter, suppose $\mathbf{R}_{\text{tr}}$ ($\mathbf{X}_{\text{tr}}$) has r (x) nonzeros on average per column (row). Then, we can compute $\mathbf{X}_{\text{tr}}\mathbf{V}_K$ in time $O(KU_{\text{tr}}x)$, the term in the inverse in time $O(K^2U_{\text{tr}})$, the inverse itself in time $O(K^3)$, and the remaining matrix multiplication in time $O(K^2U_{\text{tr}} + KIr)$. This gives a total complexity of $O(K^3 + K \cdot (U_{\text{tr}}x + Ir) + K^2 \cdot (U + P))$. Since $K \ll \max(U_{\text{tr}}, I)$ in practice, the first term will typically not be prohibitive.

The prediction complexity of LoCo is that of computing $\mathbf{X}_{\text{te}}\mathbf{V}_K\mathbf{Z}$, which is simply $O(KU_{\text{te}}(x + I))$.

Experimental setup

We use four real-world datasets for cold-start evaluation:

(a) *Ebook*, a private anonymised dataset from a major online ebook retailer with more than 20 million readers. It contains ebook purchases and Facebook friends and page likes for a random subset of 30,000 users; these users purchased >80,000 books, and liked ∼6 million pages.

(b) *Flickr* (Tang and Liu 2009), which consists of 80,513 users, 195 groups joined by the users, and a social network with 5,899,882 friendship relationships.

(c) *Blogcatalog* (Tang and Liu 2009), which consists of 10,312 users, 39 groups joined by the users, and a social network with 333,983 friendship relationships.

(d) *Hetrec11-LastFM* (Cantador, Brusilovsky, and Kuflik 2011), which consists of 1,892 users, 17,632 artists the users listened to, and the tag assignments the user made to various artists out of a total 186,479 possibilities.

We create 10 temporal train-test splits for the *Ebook* dataset. We create 10 train-test folds on the other datasets by including random 10% of the users in the test set and remaining 90% users in the training set.

We compared **LoCo** to a number of baselines:

- **CBF-KNN-Low**, a neighborhood recommender where user-user similarites are computed from the low dimensional projection of user-attributes, *viz.* Equation 4 with $\odot$ as cosine similarity and $\star$ as inner product operators, as used in (Gantner et al. 2010). (We found using the full, unprojected $\mathbf{X}$ to yield significantly worse results.)
- **Cos-Cos**, *viz.* Equation 4 with $\odot, \star$ as cosine similarity operators, which was found to be the best choice on the *Ebook* dataset in (Sedhain et al. 2014).
- **BPR-LinMap-Low** of (Gantner et al. 2010), as in Equation 7, using the low dimensional projection of user-attributes. (We also experimented with k-nearest-neighbor rather than linear regression, but found this approach to be inferior. See Supplementary material.)
- **CMF**, as in Equation 5.

We used cosine similarity for the methods relying on similarity. As this implicitly normalises the entries of $\mathbf{R}$, we similarly normalised $\mathbf{R}$ as a pre-processing step for LoCo.

We report precision@k, recall@k and mean average precision (mAP@100) (KDD Cup 2012), and provide standard error bars corresponding to 95% confidence intervals. We used cross-validation with grid-search to tune all hyperparameters. For the latent factor methods, we tuned the latent dimension K from $\{5, 10, 50, 100, 500, 1000\}$. For the methods relying on ℓ_2 regularisation, we tuned all regularisation strengths from $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$.

Results and analysis

Main results. From Tables 2 – 5, we observe that:

- LoCo is always the best performer on mAP@100, with improvements over the next best method ranging from ∼5% (*Hetrec11-LastFM*) to ∼25% (*Ebook*). A Friedman-Iman-Davenport test (Demšar 2006) confirms the differences in average ranks is significant ($p = 0.05$). A post-hoc Holm test confirms the difference between LoCo and both CMF and BPR-LinMap-Low to be significant ($p = 0.05$), though we cannot reject the null hypothesis with CBF-KNN-Low or Cos-Cos.
- LoCo is always the best performer on Precision and Recall scores with thresholds up to 5, implying accurate recommendations at the head. At the threshold $K = 20$, LoCo is sometimes outperformed by a small margin.
- Among existing methods, the neighbourhood methods (CBF-KNN-Low and Cos-Cos) had the best overall performance. Surprisingly, they outperform the learning based BPR-LinMap-Low and CMF approaches, indicating that the objectives of the latter are perhaps not sufficiently attuned to cold-start recommendation.
- The high variances for all methods on the *Ebook* datasetindicate that with temporal splits, there is a potentially strong concept drift in the data that affects all methods since they do not leverage temporal information.

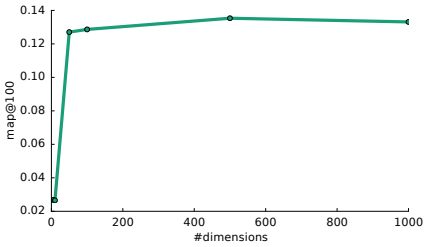

Figure 1: LoCo retrieval performance versus number of dimensions on the *Ebook* dataset.

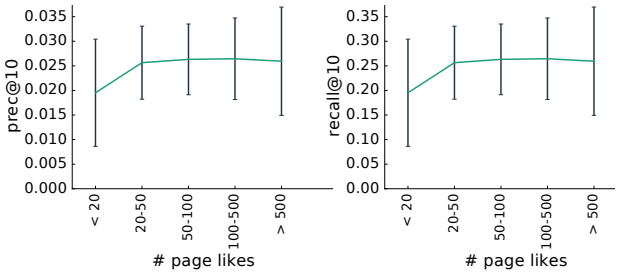

Figure 2: LoCo retrieval performance versus number of page likes on the *Ebook* dataset.

Sensitivity to latent dimension. In Figure 1, we evaluate the performance (mAP@100) with the latent dimension K on the *Ebook* dataset. We observe that performance increases sharply with the number of dimensions, but with diminishing returns beyond 100 dimensions. Thus, one can use a modest value of K to efficiently obtain accurate recommendations.

Sensitivity to page likes. To better understand how social information helps overcome the cold-start problem, in Figure 2, we analyse the performance of LoCo with the number of page likes on the *Ebook* dataset. We divide the test users into five categories based on the number of the pages they have liked. We observe that the performance increases with the number of page likes, but with diminishing return beyond 50 page likes. The high variance for few page likes

	BPR-LinMap-Low		CMF		CBF-KNN-Low		Cos-Cos		LoCo	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
@1	0.0088±0.0053	0.0088±0.0053	0.0492±0.0285	0.0492±0.0285	0.0376±0.0222	0.0376±0.0222	0.0380±0.0150	0.0380±0.0150	**0.0654±0.0305**	**0.0654±0.0305**
@3	0.0082±0.0038	0.0245±0.0113	0.0333±0.0187	0.1008±0.0560	0.0231±0.0102	0.0692±0.0305	0.0250±0.0100	0.0760±0.0290	**0.0433±0.0155**	**0.1300±0.0466**
@5	0.0118±0.0059	0.0588±0.0295	0.0272±0.0134	0.1400±0.0668	0.0215±0.0080	0.1077±0.0398	0.0230±0.0090	0.1170±0.0440	**0.0375±0.0114**	**0.1874±0.0572**
@10	0.0120±0.0065	0.1199±0.0647	0.0187±0.0079	0.1874±0.0796	0.0191±0.0059	0.1907±0.0592	0.0170±0.0060	0.1730±0.0590	**0.0259±0.0072**	**0.2589±0.0717**
@20	0.0100±0.0048	0.1991±0.0967	0.0108±0.0040	0.2206±0.0792	0.0135±0.0036	0.2696±0.0721	0.0100±0.0030	0.2050±0.0630	**0.0151±0.0038**	**0.3020±0.0752**
mAP	0.0409±0.0144		0.0926±0.0439		0.0790±0.0273		0.0750±0.0250		**0.1210±0.0406**	

Table 2: Comparison of cold-start recommenders on *Ebook* dataset.

	BPR-LinMap-Low		CMF		CBF-KNN-Low		Cos-Cos		LoCo	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
@1	0.1699±0.0035	0.1350±0.0032	0.1591±0.0029	0.1234±0.0029	0.2060±0.0024	0.1576±0.0019	0.2233±0.0046	0.1757±0.0045	**0.2864±0.0044**	**0.2252±0.0041**
@3	0.0961±0.0016	0.2224±0.0046	0.0987±0.0009	0.2170±0.0028	0.1319±0.0017	0.2924±0.0042	0.1363±0.0019	0.3042±0.0048	**0.1661±0.0014**	**0.3696±0.0040**
@5	0.0729±0.0011	0.2812±0.0054	0.0755±0.0006	0.2724±0.0036	0.1010±0.0011	0.3669±0.0044	0.1034±0.0013	0.3756±0.0052	**0.1199±0.0009**	**0.4347±0.0047**
@10	0.0485±0.0007	0.3718±0.0060	0.0512±0.0005	0.3657±0.0049	0.0658±0.0006	0.4713±0.0049	0.0671±0.0006	0.4784±0.0045	**0.0725±0.0006**	**0.5162±0.0050**
@20	0.0328±0.0003	0.5021±0.0052	0.0328±0.0003	0.4667±0.0049	0.0401±0.0002	0.5717±0.0041	**0.0412±0.0003**	**0.5824±0.0039**	**0.0412±0.0003**	**0.5831±0.0040**
mAP	0.2227±0.0029		0.2152±0.0025		0.2775±0.0023		0.2944±0.0042		**0.3453 ± 0.0040**	

Table 3: Comparison of cold-start recommenders on *Flickr* dataset.

	BPR-LinMap-Low		CMF		CBF-KNN-Low		Cos-Cos		LoCo	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
@1	0.1501±0.0210	0.1167±0.0178	0.1859±0.0086	0.1505±0.0078	0.2014±0.0120	0.1600±0.0095	0.3176±0.0094	0.2564±0.0078	**0.3765±0.0091**	**0.3035±0.0081**
@3	0.1106±0.0074	0.2548±0.0169	0.1085±0.0036	0.2557±0.0069	0.1463±0.0051	0.3373±0.0111	0.1768±0.0038	0.4099±0.0077	**0.2000±0.0041**	**0.4566±0.0088**
@5	0.1013±0.0014	0.3839±0.0077	0.0862±0.0020	0.3332±0.0079	0.1138±0.0019	0.4314±0.0081	0.1326±0.0019	0.4978±0.0074	**0.1426±0.0021**	**0.5324±0.0078**
@10	0.0777±0.0013	0.5763±0.0137	0.0656±0.0012	0.4915±0.0089	0.0841±0.0011	0.6047±0.0067	0.0886±0.0013	0.6474±0.0085	**0.0890±0.0013**	**0.6508±0.0071**
@20	0.0562±0.0009	0.8100±0.0107	0.0478±0.0007	0.6965±0.0051	**0.0588±0.0005**	0.8394±0.0046	0.0588±0.0006	**0.8427±0.0041**	0.0562±0.0007	0.8023±0.0100
mAP	0.2706±0.0142		0.2725±0.0051		0.3265±0.0083		0.4056±0.0064		**0.4470±0.0063**	

Table 4: Comparison of cold-start recommenders on *Blogcatalog* dataset.

	BPR-LinMap-Low		CMF		CBF-KNN-Low		Cos-Cos		LoCo	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
@1	0.3253±0.0261	0.0077±0.0012	0.3754±0.0139	0.0094±0.0009	0.5339±0.0294	0.0135±0.0021	0.3654±0.0284	0.0090±0.0012	**0.5811±0.0290**	**0.0142±0.0015**
@3	0.2809±0.0217	0.0191±0.0012	0.3302±0.0134	0.0232±0.0012	0.4925±0.0248	0.0344±0.0028	0.3367±0.0233	0.0255±0.0036	**0.5176±0.0242**	**0.0362±0.0026**
@5	0.2683±0.0181	0.0300±0.0017	0.3071±0.0093	0.0354±0.0014	0.4508±0.0218	0.0514±0.0031	0.3197±0.0200	0.0387±0.0043	**0.4809±0.0222**	**0.0551±0.0031**
@10	0.2399±0.0163	0.0531±0.0028	0.2564±0.0123	0.0595±0.0038	0.3965±0.0126	0.0890±0.0032	0.2807±0.0169	0.0656±0.0052	**0.4187±0.0172**	**0.0958±0.0060**
@20	0.2048±0.0133	0.0904±0.0050	0.2097±0.0110	0.0967±0.0062	0.3232±0.0113	0.1439±0.0049	0.2295±0.0142	0.1050±0.0076	**0.3398±0.0135**	**0.1542±0.0070**
mAP	0.0871±0.0054		0.0793±0.0047		0.1677±0.0079		0.1218±0.0098		**0.1780±0.0095**	

Table 5: Comparison of cold-start recommenders on *Hetrec11-LastFM* dataset.

Method	BPR-LinMap-Low	CMF	CBF-KNN-Low	Cos-Cos	LoCo
Validation time	20 hour 30 mins	2 hour 2 mins	5 mins 10 secs	2 secs	5 mins 3 secs

Table 6: Comparison of validation times on *Ebook* dataset.

indicate a lack of sufficient information, whereas the high variance for many page likes indicate a lack of selectiveness with page likes. Importantly, LoCo makes good item recommendations to users with moderate numbers of page likes.

Runtime comparison. Table 6 compares the hyperparameter validation time needed for all methods on *Ebook*. LoCo is seen to be orders of magnitude faster to tune than the learning-based methods CMF and BPR-KNN-Low. While the CBF and Cos-Cos methods are faster to tune than LoCo, the latter is more accurate. Hence, LoCo attains a suitable balance between accuracy and runtime.

Conclusion

We showed how three popular social cold-start models can be seen as special cases of a *linear content-based model*, with different constraints on the learned weights. We proposed a new model, LoCo, that directly learns a low-rank linear model efficiently via randomised SVD, and demonstrated substantial empirical improvements over state-of-the-art cold-start recommenders.

In future work we aim to explore nonlinear and streaming variants of our model, as well as means of further driving the runtime of LoCo down to that of Cos-Cos.

Acknowledgments

The authors thank the anonymous reviewers for their valuable feedback.

References

Agarwal, D., and Chen, B.-C. 2009. Regression-based latent factor models. In *KDD*.

Agarwal, D., and Chen, B.-C. 2010. flda: Matrix factorization through latent dirichlet allocation. In *WSDM*, 91–100. New York, NY, USA: ACM.

Basilico, J., and Hofmann, T. 2004. Unifying collaborative and content-based filtering. In *ICML*.

Basu, C.; Hirsh, H.; and Cohen, W. 1998. Recommendation as classification: Using social and content-based information in recommendation. In *AAAI*.

Billsus, D., and Pazzani, M. J. 1999. A hybrid user model for news story classification. In *UM*.

Cantador, I.; Brusilovsky, P.; and Kuflik, T. 2011. Hetrec '11. In *RecSys*.

Cao, B.; Liu, N. N.; and Yang, Q. 2010. Transfer learning for collective link prediction in multiple heterogenous domains. In *ICML*.

Demšar, J. 2006. Statistical comparisons of classifiers over multiple data sets. *JMLR* 7:1–30.

Gantner, Z.; Drumond, L.; Freudenthaler, C.; Rendle, S.; and Schmidt-Thieme, L. 2010. Learning attribute-to-feature mappings for cold-start recommendations. In *ICDM*.

Goldberg, D.; Nichols, D.; Oki, B. M.; and Terry, D. 1992. Using collaborative filtering to weave an information tapestry. *Commun. ACM* 35(12).

Gunawardana, A., and Meek, C. 2009. A unified approach to building hybrid recommender systems. In *RecSys*.

Halko, N.; Martinsson, P. G.; and Tropp, J. A. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.* 53(2).

Hastie, T.; Tibshirani, R.; and Friedman, J. 2009. *The Elements of Statistical Learning*. Springer New York Inc., 2nd edition.

Herlocker, J. L.; Konstan, J. A.; Borchers, A.; and Riedl, J. 1999. An algorithmic framework for performing collaborative filtering. In *SIGIR*.

Jamali, M., and Ester, M. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*.

KDD Cup. 2012. Evaluation. https://www.kddcup2012.org/c/kddcup2012-track1/details/Evaluation.

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

Krohn-Grimberghe, A.; Drumond, L.; Freudenthaler, C.; and Schmidt-Thieme, L. 2012. Multi-relational matrix factorization using Bayesian personalized ranking for social network data. In *WSDM*.

Leavitt, N. 2013. A technology that comes highly recommended. *Computer* 46(3).

Ma, H.; Yang, H.; Lyu, M. R.; and King, I. 2008. Sorec: Social recommendation using probabilistic matrix factorization. In *CIKM*.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Melville, P.; Mooney, R. J.; and Nagarajan, R. 2002. Content-boosted collaborative filtering for improved recommendations. In *AAAI*.

Mooney, R. J., and Roy, L. 2000. Content-based book recommending using learning for text categorization. In *DL*.

Noel, J.; Sanner, S.; Tran, K.-N.; Christen, P.; Xie, L.; Bonilla, E. V.; Abbasnejad, E.; and Della Penna, N. 2012. New objective functions for social collaborative filtering. In *WWW*.

Park, S.-T., and Chu, W. 2009. Pairwise preference regression for cold-start recommendation. In *RecSys*.

Pazzani, M. J., and Billsus, D. 2007. Content-based recommendation systems. In *The Adaptive Web*. Springer-Verlag. 325–341.

Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; and Riedl, J. 1994. Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW*.

Rohani, V. A.; Kasirun, Z. M.; Kumar, S.; and Shamshirband, S. 2014. An effective recommender algorithm for cold-start problem in academic social networks. *Mathematical Problems in Engineering*.

Rosli, A. N.; You, T.; Ha, I.; Chung, K.-Y.; and Jo, G.-S. 2014. Alleviating the cold-start problem by incorporating movies facebook pages. *Cluster Computing* 18(1).

Sahebi, S., and Cohen, W. W. 2011. Community-based recommendations: a solution to the cold start problem. In *RSWEB*.

Schein, A. I.; Popescul, A.; Ungar, L. H.; and Pennock, D. M. 2002. Methods and metrics for cold-start recommendations. In *SIGIR*.

Sedhain, S.; Sanner, S.; Braziunas, D.; Xie, L.; and Christensen, J. 2014. Social collaborative filtering for cold-start recommendations. In *RecSys*.

Singh, A. P., and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *KDD*.

Tang, L., and Liu, H. 2009. Relational learning via latent social dimensions. In *KDD*.

Yang, S.-H.; Long, B.; Smola, A.; Sadagopan, N.; Zheng, Z.; and Zha, H. 2011. Like like alike: joint friendship and interest propagation in social networks. In *WWW*.

Zhang, Z.-K.; Liu, C.; Zhang, Y.-C.; and Zhou, T. 2010. Solving the cold-start problem in recommender systems with social tags. *EPL* 92(2).