The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

Hybrid Learning Model with Barzilai-Borwein Optimization for Context-Aware Recommendations

Felipe Costa, Peter Dolog
Aalborg Uiversity
Selma Lagerløfs Vej 300, 9220
Aalborg
{fcosta,dolog}@cs.aau.dk

Abstract

We propose an improved learning model for non-negative matrix factorization in the context-aware recommendation. We extend the collective non-negative matrix factorization through hybrid regularization method by combining multiplicative update rules with Barzilai-Borwein optimization. This provides new improved way of learning factorized matrices. We combine ratings, content features, and contextual information in three different 2-dimensional matrices. We study the performance of the proposed method on recommending top-N items. The method was empirically tested on 4 datasets, including movies, music, and mobile apps, showing an improvement in comparison with other state-of-the-art for top-N recommendations, and time convergence to the stationary point for larger datasets.

Introduction

Recommender systems are traditionally focused on users, items, and their interactions to build a model to recommend a sorted list of N items, corresponding to the user's interests. However, it is important to incorporate context in some applications during the recommendation process, such as tourism (sights to be visited), movies (time and place), and so on. Researchers have identified the quality of recommendations increases when they use additional information, such as *time* and *location* (Adomavicius and Tuzhilin 2011).

There are two main challenges in recommendation process: 1) generating list of top-N recommendations and 2) time of convergence in learning the factorized matrices. Context-aware recommender models has shown significant improvement to cover the first challenge as presented by (Zheng, Burke, and Mobasher 2014; Codina, Ricci, and Ceccaroni 2016; Baltrunas et al. 2011), however, their models do not consider content features, which may influence the users' decision and improve recommendation accuracy. For the second challenge ALS has been applied in different matrix factorization models as presented by (Liu et al. 2013; He et al. 2014). Nonetheless, the convergence results for gradient descent methods assume the subproblems have unique solutions (Huang, Liu, and Zhou 2015).

As a solution for top-N recommendations and convergence of learning curve, we propose to extend the collective non-negative matrix factorization (CNMF) using the Barzilai-Borwein (BB) optimization method and multiplicative update rules, called the collective hybrid non-negative matrix factorization (CHNMF[1]). CHNMF factorizes ratings, content features and context in three non-negative low-rank matrices, represented in a common latent space. Our hypothesis is that using the same factor space to jointly decompose different matrices (e.g. attributes, context, and users' tastes) improves the prediction of top-N items. Further, BB improves convergence time for the learning model, by running the factorization tasks in parallel. Factorizing the rating, content features and contextual information collectively allows BB to perform better in larger dataset than ALS, due to higher density. Hence, BB only computes two projections and two gradients at even steps. Moreover, it determines the step length without using any line search.

We performed an experimental evaluation of the models on 4 datasets: LDOS-CoMoDa, InCarMusic, Frappe, and Movielens. The proposed model outperforms the state-of-art regarding convergence time in learning, as well as, in accuracy measured by metrics commonly used for evaluation of top-N recommendations quality.

This paper has the following contributions:

- An efficient hybrid learning model, based on multiplicative update rules and Barzilai-Borwein optimization;
- A collective model combining ratings, content features, and contextual information into a collective hybrid non-negative matrix factorization framework;
- Empirical experiments comparing the results between CHNMF and state-of-the-art methods for top-N recommendation.

Related Works

Related work can be divided in two areas: context-aware recommender systems, and collective matrix factorization (CMF).

Context-aware Recommender Systems. (Adomavicius and Tuzhilin 2011) categorized context-aware recommender

[1] Available at: https://github.com/felipeecosta

systems (CARS) into: pre-filtering, post-filtering, and contextual modeling. In this paper we have not considered post-filtering approach as baseline, because it has shown less efficiency in comparison with the others (Codina, Ricci, and Ceccaroni 2016).

Pre-filtering. (Zheng, Burke, and Mobasher 2014) introduces the UI-Splitting approach, which splits a given rating vector into two virtual vectors using a specific contextual factor. (Codina, Ricci, and Ceccaroni 2016) presents the distributional-semantics pre-filtering (DSPF), which proposes to build a matrix factorization using classified ratings with the most similar contextual situations.

Contextual modeling. (Baltrunas et al. 2011) proposes the context-aware matrix factorization (CAMF), which extends matrix factorization using context as baseline predictor to represent interaction of contextual information with items or users. (Zheng, Mobasher, and Burke 2015) discuss contextual SLIM (CSLIM) technique, which incorporates contextual factor to SLIM algorithm, through estimating the ranking score $\hat{S}_i, j, c$ for user u_i in item t_j in context c.

Collective Matrix Factorization. Multi-view clustering is a technique to split objects into clusters based on multiple representations of the object. (Liu et al. 2013; He et al. 2014) propose different methods using CMF. (Liu et al. 2013) proposed the MultiNMF, using a connection between NMF and PLSA. Comparing different views of factors in multi-view setting for clustering. (He et al. 2014) proposed a co-regularized NMF (CoNMF), where comment-based clustering is formalized as a multi-view problem using pair-wise and cluster-wise CoNMF.

Decoupled Target Specific Features Multi-Target Factorization (DMF), proposed by (Drumond et al. 2014), follows the same principle of CMF. DMF learns a set of single target models optimized for one relation, while downweighting the others. However, a number of parameters are used only for auxiliary relations and never for predicting the targets, what diverges from the proposed work in this paper.

Local collective embeddings (LCE) is a matrix factorization method proposed by (Saveski and Mantrach 2014), which exploits user-document and document-terms matrices, identifying a common latent space to both item features and rating matrix. LCE has shown effectiveness in cold-start problem for news recommendation, however, it has some limitations. The method does not perform well in our domain which covers movies, music and mobile apps, because it uses only two matrices as input and multiplicative update rules as learning model. In this paper, we extend the LCE approach proposing CHNMF to address these limitations. CHNMF decomposes a matrix as a product of three matrices: content features, rating, and context. Content features are data from each item's metadata, ratings represents user's preferences, while contextual information is the situation where the user rates an item. Furthermore, the hybrid technique is applied using multiplicative update rules and Barzilai-Borwein optimization to provide a faster convergence to stationary point during the learning model.

Problem Formulation

The research problem investigated in this paper is defined as follows: *Recommend a ranked list of items to each user, given by ratings, content features, and contextual information on user-item interactions*. Modeling the rating data from U users to I items under X_a types of content and X_c types of context as three 2-dimensional matrices, i.e., user-item matrix as $X_u \in \mathbb{R}^{u \times i}$; user-content feature matrix is formally defined as $X_a \in \mathbb{R}^{u \times a}$; and user-context matrix as $X_c \in \mathbb{R}^{u \times c}$. Where, u is the number of users, i is the number of items, a is the content size, and c is the context a user rated an item. The matrix $X_a \in \{0, 1\}$ represents whether a target preferable item belongs to a specific attribute or not. The rating matrix presents the user's preferences in a numerical scale as $X_u \in \{1, 2, 3, 4, 5\}$. Finally, $X_c \in [0, 1]$ presents how often a user rated an item in a specific context.

Factor models aims to decompose the original user-item interaction matrix into two low-rank approximation matrices. CHNMF is a generalization of the classical matrix factorization methods for content features and contextual information. The latent features are stored in three low-rank matrices: ratings as $W \times H_u$; categories as $W \times H_a$; and context $W \times H_c$. Where, H_u denotes a row vector, which represents the latent features for user u. Similarly, H_a represents the category's latent features a, and H_c represents the context's latent features c.

Collective Non-Negative Matrix Factorization

Considering the notation used in the problem formulation, it is factorized X_a into two lower dimensional matrices, obtaining the $factor \times attributes$ scores belonging to an item. Factorizing X_u leads to find $factor \times items$ scores, presenting the users' preferences. Likewise, the factorization of matrix X_c allow us to identify the hidden contextual factors related to the item. CHNMF represents ratings, content features, and contexts in a common latent space, collectively factorizing X_u, X_a, and X_c into a low-dimensional representation. The formal definition, is given as the following optimization problem:

$$\begin{aligned} min: f(W) = \frac{1}{2}[\alpha\|X_u - WH_u\|_2^2 + \beta\|X_a - WH_a\|_2^2 \\ +\gamma\|X_c - WH_c\|_2^2 \\ +\lambda(\|W\|_2 + \|H_u\|_2 + \|H_a\|_2 + \|H_c\|_2)] \\ s.t. W \geq 0, H_u \geq 0, H_a \geq 0, H_c \geq 0 \end{aligned} \quad (1)$$

where W represents the common latent space during the decomposition of X_u, X_a, and X_c. $\{\alpha, \beta, \gamma\} \in [0, 1]$ are hyper-parameters controlling the importance of each factorization. The remaining terms are Tikhonov regularization of W, H_u, H_a, and H_c controlled by the hyper-parameter $\lambda \geq 0$, used to enforce a smooth solution and avoid overfitting.

Optimization

The optimization performs as follows: (1) fix the value of W while minimizing $f(W)$ over H_u, H_a, H_c; then (2) fix

the value of H_u, H_a, H_c while minimizing $f(W)$ over W. Considering a matrix with x_i rows and y_j columns, with a relation defined by r_{ij}, we can define the correlation among n neighbors' data points. This results in a matrix A, used to measure the local closeness of two data points x_i and y_j.

Collective factorization reduces data points x_i from a matrix X, into a common-latent space W as w_i. The distance between two low dimensional data points is calculated using the Euclidean distance: $\|w_i - w_j\|^2$, and mapped into a matrix A. Based on the matrix A we can iterative run these two steps until the stationary point, or until the established number of max iterations as follow:

$$\begin{aligned} M &= \frac{1}{2}\sum_{i,j=1}^{n}\|w_i - w_j\|^2 A_i j \\ &= \sum_{i=1}^{n}(w_i^T - w_i)D_i i - \sum_{i,j=1}^{n}(w_i^T - w_i)D_i i \\ &= Tr(W^T DW) - Tr(W^T AW) = Tr(W^T LW), \end{aligned} \tag{2}$$

where $Tr(\bullet)$ denotes the trace function, and D is a diagonal matrix whose entries are row sums of A (or column, as A is symmetric), in other words, $D_i i = \sum_i A_i j$; $L = D - A$ is called the Laplacian matrix, we need to incorporate it to enforce the non-negative constraints.

The optimization problem of function $f(W)$ is written as:

$$\begin{aligned} min : f(W) = \frac{1}{2}[&\alpha|X_u - WH_u|_2^2 + \beta|X_a - WH_a|_2^2 \\ &+\gamma|X_c - WH_c|_2^2 + \varphi Tr(W^T LW) \\ &+\lambda(|W|_2 + |H_u|_2 + |H_a|_2 + |H_c|_2)] \\ &s.t. W \geq 0, H_u \geq 0, H_a \geq 0, H_c \geq 0 \end{aligned} \tag{3}$$

where L is the Laplacian matrix, and φ is a hyper-parameter which controls the objective function.

Hybrid Learning Model

CHNMF is a non-convex method, considering all parameters (W, H_u, H_a, H_c) together, it is unrealistic to expect the algorithm to find the global minimum. (Saveski and Mantrach 2014) propose an iterative algorithm based on multiplicative update rules (MUR) to achieve the stationary point. However, it has been observed that MUR converges relatively slowly (Huang, Liu, and Zhou 2015). In this paper, we present a hybrid learning model using MUR and Barzilai-Borwein (BB) method to solve the convergence problem.

Barzilai-Borwein

Since H_u, H_a, and H_c have the same behaviour, we represent them in this paper as H. We have to solve:

$$\min_{W \geq 0} : f(W, H) = \frac{1}{2}\|X - WH\|_F^2 \tag{4}$$

We map all the negative values into zero through $P(.)$. As H is a stationary point of Equation 4 for any $\alpha > 0$, then,

$$\|P[H - \alpha \nabla f(H)] - H\|_F = 0. \tag{5}$$

The gradient $\nabla f(W)$, of $f(H)$, is Lipschitz continuous with constant $L = \|W^T W\|_2$. Since $W^T W$ is a $k \times k$ and $k \ll min\{m, n\}$, the Lipschitz constant L is not expensive to obtain.

Algorithm 1 Barzilai-Borwein

procedure BB
  $\sigma \in \{0, 1\}, \alpha_m ax > \alpha_m in > 0;$
  $L \leftarrow \|W^T W\|_2\ H_0 \leftarrow H^k; \alpha_0 \leftarrow 1; t \leftarrow 0;$
  if H_t is a stationary point of (2) **then return** H_t
  loop:
  if $t/2 \neq 0$ **then return** $Z_t \leftarrow H_t;$
  else $Z_t \leftarrow P\left[W_t - \frac{1}{2}\nabla f(W)\right];$
  $D_t \leftarrow P\left[Z_t - \alpha_t \nabla Z_t\right] - Z_t$
  $\delta \leftarrow \langle D_t, W^T W D_t \rangle$
  if $\delta_t = 0$ **then return** $\lambda_t \leftarrow 1$
  else
    $\lambda_t \leftarrow min\tilde{\lambda_t}, 1$ where
    $\tilde{\lambda_t} = -\frac{(1-\sigma)\langle \nabla f(Z_t), D_t \rangle}{\delta_t}$
    $H_{t+1} \leftarrow Z_t + \lambda_t D_t$
  $S_t \leftarrow Ht + 1 - H_t$
  $Y_t \leftarrow \nabla f(H_{t+1}) - \nabla f(H_t)$
  if $\langle S_t, Y_t \rangle \leq 0$ **then** $\alpha_{t+1} \leftarrow \alpha_m ax$
  else
    if $t/2 \neq 0$ **then** $\alpha^B B_{t+1} \leftarrow \frac{\langle S_t, S_t \rangle}{\langle S_t, Y_t \rangle}$
    else $\alpha^B B_{t+1} \leftarrow \frac{\langle S_t, Y_t \rangle}{\langle Y_t, Y_t \rangle}$
    $\alpha_{t+1} \leftarrow min\{\alpha_m ax, max\{\alpha_m in, \alpha^B B_{t+1}\}\}$
  $t \leftarrow t + 1$
  goto *loop*.

We use $\|P[H - \alpha \nabla f(H)] - H\|_F \leq \epsilon_H$, where $\epsilon_H = max(10^{-3}, \epsilon)\|P[H - \alpha \nabla f(H)] - H\|_F$. If algorithm 1 solves Equation 4 without any iterations, we decrease the stopping tolerance by $\epsilon = 0.1\epsilon_H$. For a given $H_0 \geq 0$:

$$\mathcal{L}(H_0) = \{H | f(H) \leq f(H_0), H \geq 0\}. \tag{6}$$

By the definition of Equation 6 we have the stationary point of the Barzilai-Borwein method.

Multiplicative Update Rules

We combine multiple regularization methods, where H_u, H_a, H_c converge using the Barzilai-Borwein method, while W uses multiplicative update rules to achieve the stationary point. The partial derivatives of $f(W)$ is:

$$\begin{aligned} \nabla f(W) = &\alpha W H_u H_u^T - \alpha X_u H_u^T + \beta W H_a H_a^T \\ &-\beta X_a H_a^T + \gamma W H_c H_c^T - \gamma X_c H_c^T + \lambda I_k \end{aligned} \tag{7}$$

where I_k is the identity matrix with $k \times k$ dimensions. Applying the Karush-Kuhn-Tucker(KKT) first-order optimal conditions to $f(W)$, we derive:

$$W \geq 0, \nabla f(W) \geq 0, W \odot \nabla f(W) = 0, \tag{8}$$

where $\odot$ corresponds to the element-wise matrix multiplication operator.

Substituting the derivatives of $f(W)$ from Equation 7 in Equation 8 leads to following update rules:

$$W = \frac{[\alpha X_u H_u^T + \beta X_a H_a^T + \gamma X_c H_c^T]}{[\alpha H_u H_u^T + \beta H_a H_a^T + \gamma H_c H_c^T + \lambda I_k]}, \quad (9)$$

where $\frac{\bullet}{\bullet}$ corresponds to the element-wise matrix division.

Each iteration of CHNMF algorithm gives us a solution for the pair-wise division. As we map any negative values to zero, the W matrix becomes a non-negative after each update. Furthermore, the objective function and the delta decrease on each iteration of the above update rules, guaranteeing the convergence into a stationary point.

Complexity Analysis of CHNMF

(Saveski and Mantrach 2014) applied MUR using ALS as learning model due to its efficiency and simplicity. LCE updates matrix factors by multiplying each entry with a positive factor in every iteration round. However, MUR converges relatively slowly (Huang, Liu, and Zhou 2015).

CHNMF is non-convex and NP-hard problem, in relation to the variables W and H. However, ALS optimizes the subproblems W and H into convex problems. Despite the optimization, they might have more than one optimal solution because they are not strictly convex. The convergence gradient descent method assumes the subproblems have unique solutions (Huang, Liu, and Zhou 2015). Furthermore, most of the methods applying ALS are inefficient in finding a step length by using the line search, resulting in a slow convergence.

Regarding the computational complexity given by multiplicative update rules, $X_u(H_u^k)^T$, $X_a(H_a^k)^T$, $X_c(H_c^k)^T$, $(W^(k+1))^T X_u$ are $O(nmr)$ operations, where n and m are the matrix dimensions, and r is the stationary point. The former operations are $O(nmr)$, but the latter costs $O(max(m,n)r^2)$. When $r < min(m,n)$ the latter is better. In summary, the overall cost of MUR is:

$\#iterations \times O(nmr)$.

Monotone projected BB optimization model is used to solve CHNMF subproblems because it uses four stepsizes to improve the performance of the gradient methods (Barzilai and Borwein 1988). Finally, it determines the step length without using any line search.

CHNMF presents its highest complexity in conditional terms described between line 12 and 16 in Algorithm 1, besides the gradient computation itself. The complexity is shown as $O(nmr) + \#sub-iterations \times O(tmr^2)$, where k is the number of features. Consider H_u, H_a, and H_c are constant matrices. The overall cost is:

$\#iterations \times O(nmr)$
$+ \#sub-iterations \times O(kmr^2 + knr^2)$.

There are two $O(nmr)$ operations for each iteration: $X_u(H_u^k)^T$, $X_a(H_a^k)^T$, $X_c(H_c^k)^T$, $(W^(k+1))^T X_u$, as multiplicative update method. However, when k and $\#sub-iterations$ are small, this method is more efficient.

Big O notation aforementioned shows an improvement on the convergence when the factorization task is paralleled into two different learning processes, as small sub-threads. In this case W uses MUR method, and H_u, H_a, H_c uses Barzilai-Borwein optimization.

Recommendation Process

Barzilai-Borwein and multiplicative rules return the trained matrices W, H_u, H_a, and H_c containing the scores for prediction. Given the vector of unseen items v_i, we can predict the most preferable items according to the user's taste, represented as v_u. CHNMF projects the items vector v_i to the common latent space by solving the overdetermined system $v_i = wH_u$. The vector w, captures the factors to explain the preferable items v_i. Then, it uses low dimensional vector w to infer the missing part of the query: $v_u \leftarrow wH_t$. H_t is the concatenation of attribute and contextual matrices $H_t = H_a||H_c$. CHNMF ranks the items according to the predictions of the user's preference to unseen items stored in v_u.

Parameter Analysis

CHNMF has 5 essential parameters: k, the number of latent factors; α, β, and γ balance the factorization among ratings, content features, and contextual information; and λ, controlling the smoothness of the solution. The parameter k controls the quantity of factors considered by the system, consequently the complexity of the model. The small values of k underfit, while large values of k overfit the data and lead to poor performance.

Setting α, β, and γ with the same values give equal importance to all matrices, while parameters with different values give different levels of importance to each matrix. Setting the importance degree of ratings, content features and contextual information, for example, α β, and γ, ≈ 0.33 tends to achieve the best performance. Low values of α β, and γ tends to show lower performance in ranking quality.

The smoothness hyper-parameter $\lambda \geq 0$ is used to avoid overfitting. Lower values of λ oversimplify the model and decrease performance.

Experiments

Datasets. Four datasets are used to compare the methods: LDOS-CoMoDa (Košir et al. 2011), InCarMusic (Baltrunas et al. 2011), Frappe (Baltrunas et al. 2015), and Movielens (Harper and Konstan 2015). We performed a t-test to analyze the datasets' statistical significance of null hypothesis H_0: "if movie A and B share the same content-features and they are frequently viewed together, there should be some hidden relationships between them that raise the user's curiosity". In the case, the datasets do not reject the null hypothesis at the significance level $\alpha = 0.05$, presenting p-value as 0.0262 (LDOS-CoMoDa), 0.0393 (inCarMusic), 0.0365 (Frappe), and 0.0348 (Movielens).

Baselines for Comparison. *Pre-filtering*. UISplitting and DSPF techniques are trained on the ratings tagged with contextual similar situations to compute rating predictions for a specific target context.

[2]Due to memory limitation, it was not possible to reproduce the results on the larger datasets with the required setup

Table 1: Top-5 Recommendations

Algorithms	LDOS-CoMoDa			InCarMusic			Frappe			Movielens		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure	Precision	Recall	F-Measure	Precision	Recall	F-Measure
UISplitting	0.0006	0.0030	0.0010	0.0035	0.0175	0.0058	0.0094	0.1560	0.0177	NA[2]	NA[2]	NA[2]
DSPF	0.0138	0.0690	0.0230	0.1008	0.0504	0.0672	0.00261	0.0984	0.0412	NA[2]	NA[2]	NA[2]
CAMF-C	0.0008	0.0043	0.0013	0.0045	0.0229	0.0075	0.1384	0.5582	0.2218	0.0003	0.0008	0.0004
CSLIM-ICS	0.0026	0.0131	0.0043	0.0031	0.0159	0.0051	NA[2]	NA[2]	NA[2]	0.0943	0.0257	0.0403
CSLIM-LCS	0.0031	0.0157	0.0051	0.0049	0.0247	0.0081	NA[2]	NA[2]	NA[2]	0.0018	0.0005	0.0009
CLSIM-MCS	0.0023	0.0117	0.0038	0.0028	0.0143	0.0046	NA[2]	NA[2]	NA[2]	0.0017	0.0004	0.0006
LCE	0.1268	0.1368	0.1316	0.2111	0.1874	0.1985	0.5952	0.5749	0.5848	0.2000	0.1900	0.1948
CoNMF	0.1254	0.1467	0.1352	0.1943	0.1563	0.1732	0.5888	0.5735	0.5810	0.1988	0.1805	0.1892
MultiNMF	0.1305	0.1775	0.1504	0.1867	0.1743	0.1803	0.5830	0.5731	0.5780	0.1901	0.1800	0.1849
CHNMF	**0.1373**	**0.2033**	**0.1639**	**0.2222**	**0.1996**	**0.2103**	**0.5986**	**0.5763**	**0.5872**	**0.2032**	**0.1989**	**0.2010**

Table 2: Top-10 Recommendations

Algorithms	LDOS-CoMoDa			InCarMusic			Frappe			Movielens		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure	Precision	Recall	F-Measure	Precision	Recall	F-Measure
UISplitting	0.0011	0.0111	0.0020	0.0062	0.0628	0.0112	0.0004	0.0032	0.0007	NA[2]	NA[2]	NA[2]
DSPF	0.0005	0.0055	0.0009	0.0070	0.0707	0.0127	0.0003	0.0023	0.0005	NA[2]	NA[2]	NA[2]
CAMF-C	0.0009	0.0094	0.0016	0.0073	0.0735	0.0132	0.0006	0.0046	0.0010	0.0017	0.0008	0.0010
CSLIM-ICS	0.0024	0.0094	0.0038	0.0038	0.0380	0.0069	NA[2]	NA[2]	NA[2]	0.0471	0.0257	0.0332
CSLIM-LCS	0.0025	0.0255	0.0045	0.0037	0.0373	0.0067	NA[2]	NA[2]	NA[2]	0.0017	0.0009	0.0017
CLSIM-MCS	0.0024	0.0240	0.0043	0.0028	0.0287	0.0051	NA[2]	NA[2]	NA[2]	0.0017	0.0009	0.0011
LCE	0.1389	0.1189	0.1281	0.1999	0.1190	0.1491	0.2997	0.1599	0.2085	0.2100	0.1974	0.2035
CoNMF	0.1188	0.1366	0.1270	0.1983	0.1189	0.1486	0.2992	0.1444	0.1947	0.2005	0.1901	0.1951
MultiNMF	0.1364	0.1183	0.1267	0.1970	0.1183	0.1478	0.2981	0.1451	0.1951	0.2009	0.1807	0.1902
CHNMF	**0.1399**	**0.1191**	**0.1286**	**0.2091**	**0.1194**	**0.1520**	**0.3020**	**0.1639**	**0.2124**	**0.2181**	**0.2000**	**0.2086**

Table 3: NDCG Performance

Algorithms	LDOS-CoMoDa	InCarMusic	Frappe	Movielens
UISplitting	0.0032	0.0295	0.0004	NA[2]
DSPF	0.0050	0.0428	0.0012	NA[2]
CAMF-C	0.0034	0.0232	0.5716	0.0008
CSLIM-ICS	0.0122	0.0181	NA[2]	0.0034
CSLIM-LCS	0.0122	0.0254	NA[2]	0.0107
CSLIM-MCS	0.0116	0.0134	NA[2]	0.0011
LCE	0.3232	0.1013	0.8019	0.1119
CoNMF	0.3201	0.0947	0.6179	0.1001
MultiNMF	0.3227	0.0962	0.6111	0.1099
CHNMF	**0.3366**	**0.1080**	**0.8048**	**0.1221**

Contextual-modeling. CAMF-C, CSLIM-ICS, CSLIM-LCS, and CSLIM-MCS, had their setup defined as recommended by (Baltrunas et al. 2011; Zheng, Mobasher, and Burke 2015).

LCE. It was defined $\alpha = 0.5$ and $\lambda \in [0, 1]$ as recommended by (Saveski and Mantrach 2014), which had a better performance in their experiments.

CoNMF. It follows the authors' suggested settings (He et al. 2014), where they propose the regularization parameters set to 1 for all ratings and datasets. This model was applied before the recommender process to compare the technical performance.

MultiNMF. The authors suggested to set the regularization parameters uniformly to 0.01 (Liu et al. 2013). Initially, MultiNMF normalizes the data matrix using L1-norm, however, to become consistent with the technique presented in this paper it was decided to test it using L2-norm.

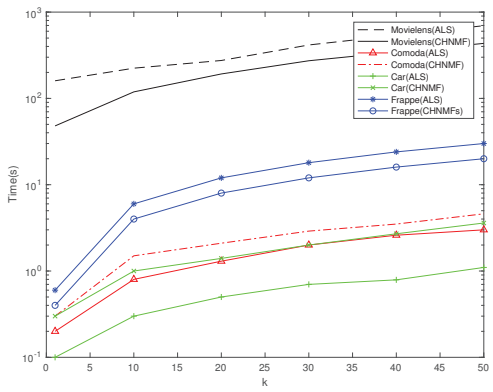

Figure 1: Convergence Time

Evaluation Metrics. NDCG, precision, recall and f-measure are used to test the ranking quality based on user's preferences scores generated by CHNMF. We set $N = 5$ and $N = 10$ because this value retrieves a smaller list of items, considering the user's taste. Large values of N would result in the extra work for the user to filter among a long list of relevant items.

To avoid overfitting we perform the experiments using 5-fold cross validation.

Results. The experiments were performed on Unix server with 32GB of RAM and 8 core CPU Intel Xeon with

2.80GHz, under the same parameters settings: *learning rate* = 0.001; k = 50; $iterations$ = 50; and λ = 0.5. For this experiment, Movielens dataset had its contextual information (timestamp), decoded into hours, representing all hours from a day. The input matrices X_u, X_c, and X_a are rating matrix, contextual matrix, and content-feature matrix, respectively. Tables 1, 2, e 3 show the performance of CHNMF and state-of-art for top-5 and top-10 recommendations.

CHNMF has achieved a comparable performance as LCE, with a slight improvement, due to the combination of three matrices: rating, content features and contextual information. The context plays an important role in achieving better precision score, hence it shows in which conditions a target user u prefers to play a specific media. Furthermore, CoNMF and MultiNMF has shown approximate values of ranking quality and effectiveness, however under-performed CHNMF.

Pre-filtering and contextual modeling techniques presented poor performance, hence it does not incorporate content feature information. CSLIM method did not present significant results for Frappe dataset during the experiments, due to the broad range ratings. While, CAMF-C showed a good NDCG value due the number of items combined with high contextual information. However, it did not overcome the result produced by the collective approaches.

Furthermore, Figure 1 presents the computational complexity analysis between ALS and CHNMF, comparing the convergence time (in logarithmic scale) against number of factors k. CHNMF had a better performance of 33% for Frappe and 34% for Movielens datasets compared to ALS. However, ALS had a better performance of 25% for LDOS-CoMoDa and 66% for InCarMusic datasets in comparison with CHNMF. Moreover, in both methods it was observed time increases linearly when compared with the number of factors. CHNMF performs better than ALS in larger datasets because Frappe and Movielens have denser matrices than LDOS-CoMoDa and InCarMusic.

Conclusions

We proposed CHNMF for a context-aware recommender system aggregating ratings, content features and contextual information in a common latent space. Furthermore, we introduced Barzilai-Borwein optimization into recommender systems combined with multiplicative update rules. Finally, we have experimentally shown the proposed methods, and generally outperform the state-of-the-art approaches considering the 4 datasets, LDOS-CoMoDa, InCarMusic, Frappe, and Movielens.

The top-N were addressed using three different matrices as input for CHNMF. We observed the content features, contexts, and ratings, when combined, play an important role for the user engagement: *users who rated an item i from an attribute a, and context c, tend to preferentially engage with each other about the same item in a specific context*.

We would like to extend CHNMF to offer explainable recommendations in natural language, presenting why the user receives a certain recommendation. Furthermore, optimizing the learning model may benefit CHNMF to perform better in scalable systems.

Acknowledgments. The authors wish to acknowledge the financial support and the fellow scholarship given to this research from the Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (grant# 206065/2014-0)

References

Adomavicius, G., and Tuzhilin, A. 2011. Context-aware recommender systems. In *Recc Sys. handbook*. Springer. 217–253.

Baltrunas, L.; Kaminskas, M.; Ludwig, B.; Moling, O.; Ricci, F.; Aydin, A.; Lüke, K.-H.; and Schwaiger, R. 2011. Incarmusic: Context-aware music recommendations in a car. In *E-Commerce and Web Technologies*. Springer. 89–100.

Baltrunas, L.; Church, K.; Karatzoglou, A.; and Oliver, N. 2015. Frappe: Understanding the usage and perception of mobile app recommendations in-the-wild. *CoRR* abs/1505.03014.

Barzilai, J., and Borwein, J. M. 1988. Two-point step size gradient methods. *IMA journal of numerical analysis* 8(1):141–148.

Codina, V.; Ricci, F.; and Ceccaroni, L. 2016. Distributional semantic pre-filtering in context-aware recommender systems. *User Modeling and User-Adapted Interaction* 26(1):1–32.

Drumond, L. R.; Diaz-Aviles, E.; Schmidt-Thieme, L.; and Nejdl, W. 2014. Optimizing multi-relational factorization models for multiple target relations. CIKM '14, 191–200. New York, NY, USA: ACM.

Harper, F. M., and Konstan, J. A. 2015. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.* 5(4):19:1–19:19.

He, X.; Kan, M.-Y.; Xie, P.; and Chen, X. 2014. Comment-based multi-view clustering of web 2.0 items. WWW '14, 771–782. New York, NY, USA: ACM.

Huang, Y.; Liu, H.; and Zhou, S. 2015. An efficient monotone projected barzilai–borwein method for nonnegative matrix factorization. *Applied Mathematics Letters* 45:12–17.

Košir, A.; Odic, A.; Kunaver, M.; Tkalcic, M.; and Tasic, J. F. 2011. Database for contextual personalization. *Elektrotehniški vestnik* 78(5):270–274.

Liu, J.; Wang, C.; Gao, J.; and Han, J. 2013. Multi-view clustering via joint nonnegative matrix factorization. In *Proc. of Intl. Conf. on Data Mining*, 252–260. SIAM.

Saveski, M., and Mantrach, A. 2014. Item cold-start recommendations: Learning local collective embeddings. RecSys '14, 89–96. New York, NY, USA: ACM.

Zheng, Y.; Burke, R.; and Mobasher, B. 2014. Splitting approaches for context-aware recommendation: An empirical study. In *Proc. of the 29th Annual ACM Symp.on Applied Computing*, SAC '14, 274–279. New York, NY, USA: ACM.

Zheng, Y.; Mobasher, B.; and Burke, R. 2015. Similarity-based context-aware recommendation. In *Intl. Conf. on Web Inf. Systems Eng.*, 431–447. Springer.