Collaborative Filtering via Group-Structured Dictionary Learning*

Zoltán Szabó[1], Barnabás Póczos[2], and András Lőrincz[1]

[1] Faculty of Informatics, Eötvös Loránd University,
Pázmány Péter sétány 1/C, H-1117 Budapest, Hungary
[2] Carnegie Mellon University, Robotics Institute,
5000 Forbes Ave, Pittsburgh, PA 15213
szzoli@cs.elte.hu,bapoczos@cs.cmu.edu,andras.lorincz@elte.hu

Abstract. Structured sparse coding and the related structured dictionary learning problems are novel research areas in machine learning. In this paper we present a new application of structured dictionary learning for collaborative filtering based recommender systems. Our extensive numerical experiments demonstrate that the presented method outperforms its state-of-the-art competitors and has several advantages over approaches that do not put structured constraints on the dictionary elements.

Keywords: collaborative filtering, structured dictionary learning

1 Introduction

The proliferation of online services and the thriving electronic commerce overwhelms us with alternatives in our daily lives. To handle this information overload and to help users in efficient decision making, recommender systems (RS) have been designed. The goal of RSs is to recommend personalized items for online users when they need to choose among several items. Typical problems include recommendations for which movie to watch, which jokes/books/news to read, which hotel to stay at, or which songs to listen to.

One of the most popular approaches in the field of recommender systems is *collaborative filtering* (CF). The underlying idea of CF is very simple: Users generally express their tastes in an explicit way by rating the items. CF tries to estimate the users' preferences based on the ratings they have already made on items and based on the ratings of other, similar users. For a recent review on recommender systems and collaborative filtering, see e.g., [1].

Novel advances on CF show that *dictionary learning* based approaches can be efficient for making predictions about users' preferences [2]. The dictionary learning based approach assumes that (i) there is a latent, unstructured feature space (hidden representation/code) behind the users' ratings, and (ii) a rating

* F. Theis et al. (Eds.): LVA/ICA 2012, LNCS 7191, pp. 247-254, 2012. The original publication is available at http://dx.doi.org/10.1007/978-3-642-28551-6_31.

of an item is equal to the product of the item and the user's feature. To increase the generalization capability, usually ℓ_2 regularization is introduced both for the dictionary and for the users' representation.

Recently it has been shown both theoretically and via numerous applications (e.g., automatic image annotation, feature selection for microarray data, multi-task learning, multiple kernel learning, face recognition, structure learning in graphical models) that it can be advantageous to force different kind of structures (e.g., disjunct groups, trees) on the hidden representation. This regularization approach is called *structured sparsity* [3]. The structured sparse coding problem assumes that the dictionary is already given. A more interesting (and challenging) problem is the combination of these tasks, i.e., learning the best structured dictionary and structured representation. This is the *structured dictionary learning* (SDL) problem. SDL is more difficult than structured sparse coding; one can only find few results in the literature [4–8]. This novel field is appealing for (i) transformation invariant feature extraction [8], (ii) image denoising/inpainting [4, 6], (iii) background subtraction [6], (iv) analysis of text corpora [4], and (v) face recognition [5].

Several successful applications show the importance of the SDL problem family. Interestingly, however, to the best of our knowledge, it has not been used for the collaborative filtering problem yet. The *goal of our paper* is to extend the application domain of SDL to CF. In CF further constraints appear for SDL since (i) online learning is desired, and (ii) missing information is typical. There are good reasons for them: novel items/users may appear and user preferences may change over time. Adaptation to users also motivate online methods. Online methods have the additional advantage with respect to offline ones that they can process more instances in the same amount of time, and in many cases this can lead to increased performance. For a theoretical proof of this claim, see [9]. Usually users can evaluate only a small portion of the available items, which leads to incomplete observations, missing rating values. In order to cope with these constraints of the collaborative filtering problem, we will use a novel extension of the structured dictionary learning problem, the so-called online group-structured dictionary learning (OSDL) [10]. OSDL allows (i) overlapping group structures with (ii) non-convex sparsity inducing regularization, (iii) partial observation (iv) in an online framework.

Our paper is structured as follows: We briefly review the OSDL technique in Section 2. We cast the CF problem as an OSDL task in Section 3. Numerical results are presented in Section 4. Conclusions are drawn in Section 5.

Notations. Vectors have bold faces ($\mathbf{a}$), matrices are written by capital letters ($\mathbf{A}$). For a set, $|\cdot|$ denotes the number of elements in the set. For set $O \subseteq \{1, \ldots, d\}$, $\mathbf{a}_O \in \mathbb{R}^{|O|}$ ($\mathbf{A}_O \in \mathbb{R}^{|O| \times D}$) denotes the coordinates (columns) of vector $\mathbf{a} \in \mathbb{R}^d$ (matrix $\mathbf{A} \in \mathbb{R}^{d \times D}$) in O. The ℓ_p (quasi-) norm of vector $\mathbf{a} \in \mathbb{R}^d$ is $\|\mathbf{a}\|_p = (\sum_{i=1}^d |a_i|^p)^{\frac{1}{p}}$ ($p > 0$). $S_p^d = \{\mathbf{a} \in \mathbb{R}^d : \|\mathbf{a}\|_p \leq 1\}$ denotes the ℓ_p unit sphere in $\mathbb{R}^d$. The point-wise product of $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ is $\mathbf{a} \circ \mathbf{b} = [a_1 b_1; \ldots; a_d b_d]$. For a set system[3] $\mathcal{G}$, the coordinates of vector $\mathbf{a} \in \mathbb{R}^{|\mathcal{G}|}$ are denoted by a^G ($G \in \mathcal{G}$), that is, $\mathbf{a} = (a^G)_{G\in\mathcal{G}}$.

2 The OSDL Problem

In this section we formally define the online group-structured dictionary learning problem (OSDL). Let the dimension of the observations be denoted by d_x. Assume that in each time instant ($i = 1, 2, \ldots$) a set $O_i \subseteq \{1, \ldots, d_x\}$ is given, that is, we know which coordinates are observable at time i, and the observation is $\mathbf{x}_{O_i}$. Our goal is to find a dictionary $\mathbf{D} \in \mathbb{R}^{d_x \times d_\alpha}$ that can accurately approximate the observations $\mathbf{x}_{O_i}$ from the linear combinations of the columns of $\mathbf{D}$. These column vectors are assumed to belong to a closed, convex, and bounded set $\boldsymbol{\mathfrak{D}} = \times_{i=1}^{d_\alpha} \boldsymbol{\mathfrak{D}}_i$. To formulate the cost of dictionary $\mathbf{D}$, first a *fixed* time instant i, observation $\mathbf{x}_{O_i}$, and dictionary $\mathbf{D}$ are considered, and the hidden representation $\boldsymbol{\alpha}_i$ associated to this $(\mathbf{x}_{O_i}, \mathbf{D}, O_i)$ triple is defined. Representation $\boldsymbol{\alpha}_i$ is allowed to belong to a closed, convex set $\boldsymbol{\mathcal{A}} \subseteq \mathbb{R}^{d_\alpha}$ ($\boldsymbol{\alpha}_i \in \boldsymbol{\mathcal{A}}$) with certain structural constraints. The structural constraints on $\boldsymbol{\alpha}_i$ are expressed by making use of a given $\mathcal{G}$ group structure, which is a set system on $\{1, \ldots, d_\alpha\}$. Representation $\boldsymbol{\alpha}$ belonging to a triple $(\mathbf{x}_O, \mathbf{D}, O)$ is defined as the solution of the structured sparse coding task

$$l(\mathbf{x}_O, \mathbf{D}_O) = \min_{\boldsymbol{\alpha}\in\boldsymbol{\mathcal{A}}} \left[\frac{1}{2} \|\mathbf{x}_O - \mathbf{D}_O\boldsymbol{\alpha}\|_2^2 + \kappa\Omega(\boldsymbol{\alpha}) \right], \tag{1}$$

where $l(\mathbf{x}_O, \mathbf{D}_O)$ denotes the loss, $\kappa > 0$, and $\Omega(\mathbf{y}) = \|(\|\mathbf{y}_G\|_2)_{G\in\mathcal{G}}\|_\eta$ is the structured regularizer associated to $\mathcal{G}$ and $\eta \in (0, 1]$. Here, the first term of (1) is responsible for the quality of approximation on the observed coordinates. The second term constrains the solution according to the group structure $\mathcal{G}$ similarly to the sparsity inducing regularizer Ω in [5], i.e., it eliminates the terms $\|\mathbf{y}_G\|_2$ ($G \in \mathcal{G}$) by means of $\|\cdot\|_\eta$. The OSDL problem is defined as the minimization of the cost function:

$$\min_{\mathbf{D}\in\boldsymbol{\mathfrak{D}}} f_t(\mathbf{D}) := \frac{1}{\sum_{j=1}^t (j/t)^\rho} \sum_{i=1}^t \left(\frac{i}{t}\right)^\rho l(\mathbf{x}_{O_i}, \mathbf{D}_{O_i}). \tag{2}$$

Here the goal is to minimize the average loss belonging to the dictionary, where ρ is a non-negative forgetting factor. If $\rho = 0$, we get the classical average.

As an example, let $\boldsymbol{\mathfrak{D}}_i = S_2^{d_x}$ ($\forall i$), $\boldsymbol{\mathcal{A}} = \mathbb{R}^{d_\alpha}$. In this case, columns of $\mathbf{D}$ are restricted to the Euclidean unit sphere and we have no constraints for $\boldsymbol{\alpha}$. Now, let $|\mathcal{G}| = d_\alpha$ and $\mathcal{G} = \{desc_1, \ldots, desc_{d_\alpha}\}$, where $desc_i$ represents the i^{th} node and its children in a fixed tree. Then the coordinates $\{\alpha_i\}$ are searched in a hierarchical tree structure and the hierarchical dictionary $\mathbf{D}$ is optimized accordingly.

[3] A set system is also called hypergraph or a family of sets.

Optimization of cost function (2) is equivalent to the joint optimization:

$$\operatorname*{arg\,min}_{\mathbf{D}\in\boldsymbol{\mathfrak{D}},\{\boldsymbol{\alpha}_i\in\boldsymbol{\mathcal{A}}\}_{i=1}^t} f_t(\mathbf{D},\{\boldsymbol{\alpha}_i\}_{i=1}^t)=\frac{1}{\sum_{j=1}^t(j/t)^\rho}\sum_{i=1}^t\left(\frac{i}{t}\right)^\rho\left[\frac{1}{2}\left\|\mathbf{x}_{O_i}-\mathbf{D}_{O_i}\boldsymbol{\alpha}_i\right\|_2^2+\kappa\Omega(\boldsymbol{\alpha}_i)\right].$$

By using the sequential observations $\mathbf{x}_{O_i}$, one can optimize $\mathbf{D}$ online in an alternating manner: The actual dictionary estimation $\mathbf{D}_{t-1}$ and sample $\mathbf{x}_{O_t}$ are used to optimize (1) for representation $\boldsymbol{\alpha}_t$. After this step, when the estimated representations $\{\boldsymbol{\alpha}_i\}_{i=1}^t$ are given, the dictionary estimation $\mathbf{D}_t$ is derived from the quadratic optimization problem

$$\hat{f}_t(\mathbf{D}_t)=\min_{\mathbf{D}\in\boldsymbol{\mathfrak{D}}} f_t(\mathbf{D},\{\boldsymbol{\alpha}_i\}_{i=1}^t). \tag{3}$$

These optimization problems can be tackled by making use of the variational property [5] of norm η and using the block-coordinate descent method, which leads to matrix recursions [10].[4]

3 OSDL Based Collaborative Filtering

Below, we transform the CF task into an OSDL problem. Consider the t^{th} user's known ratings as OSDL observations $\mathbf{x}_{O_t}$. Let the optimized group-structured dictionary on these observations be $\mathbf{D}$. Now, assume that we have a test user and his/her ratings, i.e., $\mathbf{x}_O\in\mathbb{R}^{|O|}$. The task is to estimate $\mathbf{x}_{\{1,\ldots,d_x\}\backslash O}$, that is, the missing coordinates of $\mathbf{x}$ (the missing ratings of the user). This can be accomplished by the following steps (Table 1).

Table 1: Solving CF with OSDL

1. Remove the rows of the non-observed $\{1,\ldots,d_x\}\backslash O$ coordinates from $\mathbf{D}$. The obtained $|O|\times d_\alpha$ sized matrix $\mathbf{D}_O$ and $\mathbf{x}_O$ can be used to estimate $\boldsymbol{\alpha}$ by solving the structured sparse coding problem (1).
2. Using the estimated representation $\boldsymbol{\alpha}$, estimate $\mathbf{x}$ as $\hat{\mathbf{x}}=\mathbf{D}\boldsymbol{\alpha}$.

According to the CF literature, *neighbor based correction* schemes may further improve the quality of the estimations [1]. This neighbor correction approach relies on the assumption that similar items (e.g., jokes/movies) are rated similarly. As we will show below, these schemes can be adapted to OSDL-based CF estimation too. Assume that the similarities $s_{ij}\in\mathbb{R}$ $(i,j\in\{1,\ldots,d_x\})$ between individual items are given. We shall provide similarity forms in Section 4. Let $\mathbf{d}_k\boldsymbol{\alpha}_t\in\mathbb{R}$ be the OSDL estimation for the rating of the k^{th} non-observed item of the t^{th} user ($k\notin O_t$), where $\mathbf{d}_k\in\mathbb{R}^{1\times d_\alpha}$ is the k^{th} row of matrix $\mathbf{D}\in\mathbb{R}^{d_x\times d_\alpha}$, and $\boldsymbol{\alpha}_t\in\mathbb{R}^{d_\alpha}$ is computed as described in Table 1. Let the prediction error on

[4] The Matlab code of the method is available at `http://nipg.inf.elte.hu/szzoli`.

the observable item neighbors (j) of the k^{th} item of the t^{th} user ($j \in O_t \backslash \{k\}$) be $\mathbf{d}_j\boldsymbol{\alpha}_t - x_{jt} \in \mathbb{R}$. These prediction errors can be used for the correction of the OSDL estimation ($\mathbf{d}_k\boldsymbol{\alpha}_t$) by taking into account the s_{kj} similarities:

$$\hat{x}_{kt} = \gamma_0(\mathbf{d}_k\boldsymbol{\alpha}_t) + \gamma_1 \left[\frac{\sum_{j \in O_t \backslash \{k\}} s_{kj}(\mathbf{d}_j\boldsymbol{\alpha}_t - x_{jt})}{\sum_{j \in O_t \backslash \{k\}} s_{kj}} \right], \tag{4}$$

where $\gamma_0, \gamma_1 \in \mathbb{R}$ are weight parameters, and $k \notin O_t$. Equation (4) is a simple modification of the corresponding expression in [2]. It modulates the first term with a separate γ_0 weight, which we found beneficial in our experiments.

4 Numerical Results

We have chosen the Jester dataset [11] for the illustration of the OSDL based CF approach. It is a standard benchmark dataset for CF. It contains $4,136,360$ ratings from $73,421$ users on 100 jokes. The ratings are in the continuous $[-10, 10]$ range. The worst and best possible grades are -10 and $+10$, respectively. A fixed 10 element subset of the jokes is called gauge set, and it was evaluated by all users. Two third of the users have rated at least 36 jokes, and the remaining ones have rated between 15 and 35 jokes. The average number of user ratings per joke is 46.

In the neighbor correction step (4), we need the s_{ij} values, which represent the similarities of the i^{th} and j^{th} items. We define this $s_{ij} = s_{ij}(\mathbf{d}_i, \mathbf{d}_j)$ value as the similarity between the i^{th} and j^{th} rows of the optimized OSDL dictionary $\mathbf{D}$. We made experiments with the following two similarities (S_1, S_2):

$$S_1: \quad s_{ij} = \left(\frac{\max(0, \mathbf{d}_i\mathbf{d}_j^T)}{\|\mathbf{d}_i\|_2 \, \|\mathbf{d}_j\|_2} \right)^{\beta}, \text{ and } S_2: \quad s_{ij} = \left(\frac{\|\mathbf{d}_i - \mathbf{d}_j\|_2^2}{\|\mathbf{d}_i\|_2 \, \|\mathbf{d}_j\|_2} \right)^{-\beta}. \tag{5}$$

Here $\beta > 0$ is the parameter of the similarity measure [2]. Quantities s_{ij} are non-negative. If the value of s_{ij} is close to zero (large), then the i^{th} and j^{th} items are very different (very similar).

In our numerical experiments we used the RMSE (root mean square error) measure for the evaluation of the quality of the estimation, since this is the most popular measure in the CF literature. The RMSE is the average squared difference of the true and the estimated rating values:

$$RMSE = \sqrt{\frac{1}{|\mathcal{S}|} \sum_{(i,t) \in \mathcal{S}} (x_{it} - \hat{x}_{it})^2}, \tag{6}$$

where $\mathcal{S}$ denotes either the validation or the test set. We also performed experiments using the mean absolute error (MAE) and got very similar results.

4.1 Evaluation

We illustrate the efficiency of the OSDL-based CF estimation on the Jester dataset using the RMSE performance measure. To the best of our knowledge, the top results on this database are RMSE $= 4.1123$ [12] and RMSE $= 4.1229$ [2]. The method in the first paper is called *item neighbor*, and it makes use of neighbor information only. In [2], the authors used a bridge regression based unstructured dictionary learning model with a neighbor correction scheme. They optimized the dictionary by gradient descent and set d_α to 100.

To study the capability of the OSDL approach in CF, we focused on the following questions:

- Is structured dictionary $\mathbf{D}$ beneficial for prediction purposes, and how does it compare to the dictionary of classical (unstructured) sparse dictionary?
- How does the OSDL parameters and the similarity applied affect the efficiency of the prediction?
- How do different group structures $\mathcal{G}$ fit to the CF task?

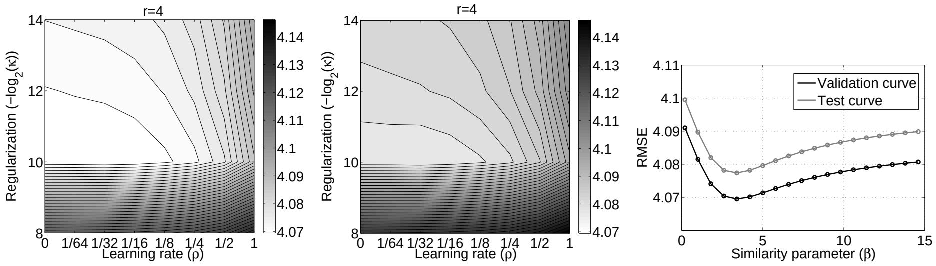
In our numerical studies we chose the Euclidean unit sphere for $\boldsymbol{\mathfrak{D}}_i = S_2^{d_x}$ $(\forall i)$ and $\boldsymbol{\mathcal{A}} = \mathbb{R}^{d_\alpha}$. We set η of the structure inducing regularizer $\mathit{\Omega}$ to 0.5. Group structure $\mathcal{G}$ was realized (i) either on a $\sqrt{d_\alpha} \times \sqrt{d_\alpha}$ toroid with $|\mathcal{G}| = d_\alpha$ applying $r \geq 0$ neighbors to define $\mathcal{G}$,[5] or (ii) on a hierarchy with a complete binary tree structure parameterized by the number of levels l ($|\mathcal{G}| = d_\alpha$, $d_\alpha = 2^l - 1$). The forgetting factor (ρ), the weight of $\mathit{\Omega}$ (κ), the size of the mini-batches in $\mathbf{D}$ optimization (R), and the parameter of the S_i similarities (β) were chosen from the sets $\{0, \frac{1}{64}, \frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1\}$, $\{\frac{1}{2^{-1}}, \frac{1}{2^0}, \frac{1}{2^1}, \frac{1}{2^2}, \frac{1}{2^4}, \frac{1}{2^6}, \ldots, \frac{1}{2^{14}}\}$, $\{8, 16\}$, and $\{0.2, 1, 1.8, \ldots, 14.6\}$, respectively. We used a $90\% - 10\%$ (80% training, 10% validation, 10% test) random split for the observable ratings in our experiments, similarly to [2].

First, we provide results using **toroid** group structure. The size of the toroid was 10×10 ($d_\alpha = 100$). In the first experiment we study how the size of neighborhood (r) affects the results. To this end, we set the neighborhood size to $r = 0$ (no structure), and then increased it to 1, 2, 3, 4, and 5. For each (κ, ρ, β), the minimum of the validation/test surface w.r.t. β is illustrated in Fig. 1(a)-(b). According to our experiences, the validation and test surfaces are very similar for a fixed neighborhood parameter r. It implies that the validation surfaces are good indicators for the test errors. For the best r, κ and ρ parameters, we can also observe that the validation and test curves (as functions of β) are very similar [Fig. 1(c)]. Note that (i) both curves have only one local minimum, and (ii) these minimum points are close to each other. The quality of the estimation depends mostly on the κ regularization parameter. The estimation is robust to the different choices of forgetting factor ρ (see Fig. 1(a)-(b)), and this parameter can only help in fine-tuning the results.

From our results (Table 2), we can see that structured dictionaries ($r > 0$) are advantageous over those methods that do not impose structure on the dictionary

[5] For $r = 0$ ($\mathcal{G} = \{\{1\}, \ldots, \{d_\alpha\}\}$) one gets the classical sparse code based dictionary.

elements ($r = 0$). Based on this table we can also conclude that the estimation is robust to the selection of the similarity (S) and the mini-batch size (R). We got the best results using similarity S_1 and $R = 8$. Similarly to the role of parameter ρ, adjusting S and R can only be used for fine-tuning. When we increase r up to $r = 4$, the results improve. However, for $r = 5$, the RMSE values do not improve anymore; they are about the same when using $r = 4$. The smallest RMSE we could achieve was 4.0774, and the best known result so far was RMSE = 4.1123 [12]. This proves the efficiency of our OSDL based collaborative filtering algorithm. We note that our RMSE result seems to be significantly better than that of the competitors: we repeated this experiment 5 more times with different randomly selected training, test, and validation sets, and our RMSE results have never been worse than 4.08.



(a) (b) (c)

Fig. 1: (a)-(b): validation and test surface as a function of forgetting factor (ρ) and regularization (κ). For a fixed (κ, ρ) parameter pair, the surfaces show the best RMSE values optimized in the β similarity parameter. (c): validation and test curves for the optimal parameters ($\kappa = \frac{1}{2^{10}}$, $\rho = \frac{1}{2^5}$, mini-batch size $R = 8$). (a)-(c): neighbor size: $r = 4$, group structure ($\mathcal{G}$): toroid, similarity: S_1.

Table 2: Performance of the OSDL prediction using toroid group structure ($\mathcal{G}$) with different neighbor sizes r ($r = 0$: unstructured case). Left: mini-batch size $R = 8$, right: $R = 16$. First row: S_1, second row: S_2 similarity. For fixed R, the best performance is highlighted with boldface typesetting.

	$R = 8$					$R = 16$				
	$r = 0$	$r = 1$	$r = 2$	$r = 3$	$r = 4$	$r = 0$	$r = 1$	$r = 2$	$r = 3$	$r = 4$
S_1	4.1594	4.1326	4.1274	4.0792	**4.0774**	4.1611	4.1321	4.1255	4.0804	**4.0777**
S_2	4.1765	4.1496	4.1374	4.0815	4.0802	4.1797	4.1487	4.1367	4.0826	4.0802

In our second experiment, we studied how the **hierarchical** group structure $\mathcal{G}$ affects the results. Our obtained results are similar to that of the toroid structure. We experimented with hierarchy level $l = 3, 4, 5, 6$ (i.e, $d_\alpha = 7, 15, 31, 63$), and achieved the best result for $l = 4$. The RMSE values decrease until $l = 4$, and then increase for $l > 4$. Our best obtained RMSE value is 4.1220, and it was achieved for dimension $d_\alpha = 15$. We note that this small dimensional, hierarchical group structure based result is also better than that of [2], which makes use of unstructured dictionaries with $d_\alpha = 100$ and has RMSE $= 4.1229$. Our result is also competitive with the RMSE $= 4.1123$ value of [12].

To sum up, in the studied CF problem on the Jester dataset we found that (i) the application of group structured dictionaries has several advantages and the proposed algorithm can outperform its state-of-the-art competitors. (ii) The toroid structure provides better results than the hierarchical structure, (iii) the quality of the estimation mostly depends on the structure inducing Ω regularization (κ, $\mathcal{G}$, r or l), and (iv) it is robust to the other parameters (ρ forgetting factor, S_i similarity, R mini-batch size).

5 Conclusions

We have proposed an online group-structured dictionary learning (OSDL) approach to solve the collaborative filtering (CF) problem. We casted the CF estimation task as an OSDL problem, and demonstrated the applicability of our novel approach on joke recommendations. Our extensive numerical experiments show that structured dictionaries have several advantages over the state-of-the-art CF methods: more precise estimation can be obtained, and smaller dimensional feature representation can be sufficient by applying group structured dictionaries.

Acknowledgments. The Project is supported by the European Union and co-financed by the European Social Fund (grant agreements no. TÁMOP 4.2.1/B-09/1/KMR-2010-0003 and KMOP-1.1.2-08/1-2008-0002). The research was partly supported by the Department of Energy (grant number DESC0002607).

References

1. Ricci, F., Rokach, L., Shapira, B., Kantor, P.: Recommender Systems Handbook. Springer (2011)
2. Takács, G., Pilászy, I., Németh, B., Tikk, D.: Scalable collaborative filtering approaches for large recommender systems. J. Mach. Learn. Res. **10** (2009) 623–656
3. Bach, F., Jenatton, R., Marial, J., Obozinski, G.: Convex optimization with sparsity-inducing norms. In: Optimization for Machine Learning. MIT Press (2011)
4. Jenatton, R., Mairal, J., Obozinski, G., Bach, F.: Proximal methods for sparse hierarchical dictionary learning. (In: ICML 2010) 487–494
5. Jenatton, R., Obozinski, G., Bach, F.: Structured sparse principal component analysis. AISTATS, J. Mach. Learn. Res.:W&CP **9** (2010) 366–373
6. Mairal, J., Jenatton, R., Obozinski, G., Bach, F.: Network flow algorithms for structured sparsity. (In: NIPS 2010) 1558–1566
7. Rosenblum, K., Zelnik-Manor, L., Eldar, Y.: Dictionary optimization for block-sparse representations. (In: AAAI Fall 2010 Symposium on Manifold Learning)

8. Kavukcuoglu, K., Ranzato, M., Fergus, R., LeCun, Y.: Learning invariant features through topographic filter maps. (In: CVPR 2009) 1605–1612
9. Bottou, L., Cun, Y.L.: On-line learning for very large data sets. Appl. Stoch. Model. Bus. - Stat. Learn. **21** (2005) 137–151
10. Szabó, Z., Póczos, B., Lőrincz, A.: Online group-structured dictionary learning. (In: CVPR 2011) 2865–2872
11. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A constant time collaborative filtering algorithm. Inform. Retrieval **4** (2001) 133–151
12. Takács, G., Pilászy, I., Németh, B., Tikk, D.: Matrix factorization and neighbor based algorithms for the Netflix prize problem. (In: RecSys 2008) 267–274