Human Behavior Discovery via Multidimensional Latent Factor Modeling

Massimo Guarascio, Francesco Sergio Pisani,
Ettore Ritacco and Pietro Sabatino

Institute for High Performance Computing and Networking
of the Italian National Research Council
ICAR - CNR

Abstract. Currently, the Latent Factor Modeling is one of the most used approaches in various research fields that aim at identifying interesting features that determine the evolution of a given phenomenon. Typically, the latent factors are used to relate individual atomic elements each other: for example, semantically similar words in documents of a textual corpus (text analysis), products to buy and users (recommendation) or news in a social network (information diffusion). In this paper, we define a new latent-factor-based approach aimed at discovering human behavioral profiles. The difference from the current literature is the relaxation of the atomicity constraint of the analyzed elements. We instantiate the proposed model within the context of Human Behavior Computing, where the elements in analysis are the human actions. The latter are characterized by multiple features defined over different domains, such as "what is being done", "where", "when", or "how". We performed a test on a real-life dataset to prove the validity of the proposed approach.

1 Introduction

The Behavior Computing [5] is a recent defined research field whose goal is to investigate mathematical models that could summarize the dynamics of a complex system or phenomenon.

In industry, the Behavior Computing is exploited for many applications, especially by Social Media Services, like Facebook, Twitter, Youtube or Tumblr. These services base a big portion of their business on methodologies and techniques able to discover user's behavioral profiles. Such profiles are exploited to understand the reasons why users perform specific actions, with the aim to recommend items that match their taste, suggest personalized information, share interesting content or propose social connections, see for instance [2].

One of the most used approach to address the user behavior computing problem is the Latent Factor Modeling [15], which consists in defining and estimating a set of unobserved variables that summarize the main characteristics of the underlying population. The literature is rich of latent factor techniques aimed to model complex behaviors (e.g. [3], [6], [10], [1], [12]), but most of them share a feature: they focus on atomic samples. In other words, the most part of the production of this research field assumes that the underlying phenomenon is characterized by elementary events govern by some distribution probability, for instance:

- Recommender systems deal with users and items (in many cases as *ID*);
- Link predictors in social networks deal with users and their atomic features;
- Community detectors deal with humans and links;
- Information diffusion predictors deal with humans and topics.

At the best of our knowledge, multidimensional (or multivariate) latent factor modeling has not been thoroughly investigated in literature, but see [16].

In this paper we propose a novel approach of Human Behavior Computing exploiting an extended version of Latent Dirichlet Allocation (LDA) [4] for multidimensional data. A human behavior is defined according to the set of actions performed by a person. Each action is an instantiation of her behavior and it is composed by a set of features: what is done, where and when, how, near what, and so on represent some types of these features. In other words, actions are not atomic.

The novelty of the approach consists in a more fine-grained perspective for the human behavior learning than the current one in the latent factor literature; each action is probable or not according to different angles of view (contexts or dimensions): for instance, a user entering in a bank during the day is a common event, while it's more rare that the user enters in the bank during the night.

Throughout this paper, we follow the notation defined in Table 1. The rest of the paper is organized as follows. Section 2 defines the proposed model and explains how to estimate its parameters; Section 3 describes the multidimensional nature of the data that feed the model; Section 4 shows how to exploit the model in a real case scenario, in order to prove its utility; finally, Section 5 concludes the paper.

Variable	**Description**
U	The set of users u
K	Number of topics
Z	Latent variable
D	The set of all dimensions, the index of a dimension is d, and $\lvert D\rvert = n$
A	The set of actions, A_u is the actions taken by user u
$\vec{\alpha}$	Hyperparameter mixture on users (a vector of K elements or a scalar if symmetric)
$\vec{\beta}^{(d)}$	Hyperparameter mixture on topics for each dimension d (a vector of V elements or a scalar if symmetric)
θ_u	Profile of user u, i.e. multinomial distribution over number of topics K
$\phi_k^{(d)}$	Mixture components over the elements given a topic k and a dimension d
$\left(i_a^{(1)}, \ldots, i_a^{(n)}\right)$	Vector representing action a where each $i_a^{(d)}$ is the value of dimension d within action a. Given Z, values in each dimension are conditionally independent from each other.

Table 1. Notation.

2 Multidimensional Latent Dirichlet Allocation

The proposed model is named Multidimensional Latent Dirichlet Allocation (MDLDA), that basically extends the LDA technique [4] to the multidimensional case. LDA considers a dyadic system and tries to define an association protocol between two related sets: for instance documents and words or users and item to purchase.

The intuition, that leads to the proposed extension, is that one of the underlying sets can contains composed elements, that are characterized by multiple values, defined on domains of different dimensions.

All the variables and quantities used for the model definition are shown in Table 1, a graphical form of the model is represented in Figure 1, and its generative process is given in Table 2.

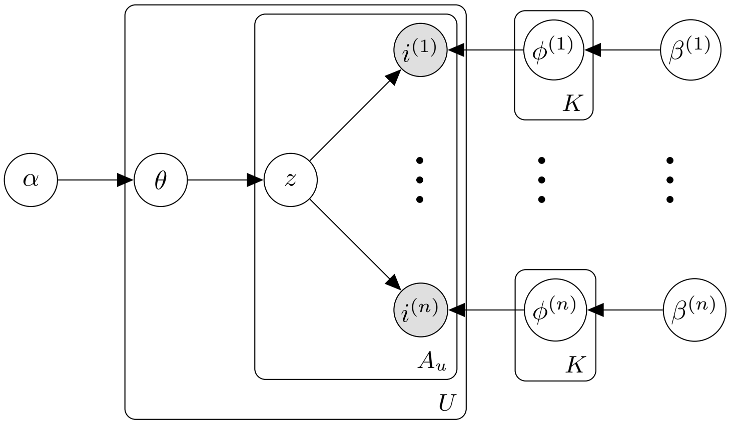

Fig. 1. Multidimensional Latent Dirichlet Model in plate notation.

2.1 Parameter inference

In order to infer parameters for the MDLDA model, let us start with the complete-data likelihood:

$$\Pr\left(\mathbf{A},\mathbf{Z},\boldsymbol{\Theta},\boldsymbol{\Phi}|\alpha,\beta\right) = \left\{\prod_{d=1}^{n}\prod_{k=1}^{K}\Pr\left(\phi_k^{(d)}|\beta^{(d)}\right)\right\}\prod_{u\in U}\Pr\left(\theta_u|\alpha\right)\prod_{a\in\mathbf{A}_u}\Pr\left(z_{u,a}|\theta_u\right)\prod_{d\in D}\Pr\left(i_{u,a}^{(d)}|\phi_{z_{u,a}}^{(d)}\right),$$

By the Law of Total Probability applied to θ and ϕ, we obtain a formula for the complete-data likelihood:

- For every user $u \in U$ choose $\theta_u \sim \mathit{Dirichlet}\,(\alpha)$
- For every dimension $d \in D$
  - Choose $\phi_k^{(d)} \sim \mathit{Dirichlet}\left(\beta^{(d)}\right)$ for $k \in \{1, \ldots, K\}$
- For every user $u \in U$ and every action $a \in A_u$
  - Choose the latent variable $z \sim \mathit{Discrete}\,(\theta_u)$
  - For every dimension $d \in D$
    - * Choose a value $i^{(d)} \sim \mathit{Discrete}\left(\phi_z^{(d)}\right)$

Table 2. Generative process for the proposed model

$$\Pr\left(\mathbf{A}, \mathbf{Z} | \alpha, \beta\right) = \int_\theta \int_{\phi^{(1)}} \ldots \int_{\phi^{(n)}} \Pr\left(\mathbf{A}, \mathbf{Z}, \boldsymbol{\Theta}, \boldsymbol{\Phi} | \alpha, \beta\right) d\theta d\phi^{(1)} \ldots d\phi^{(n)} \ ,$$

observe that integrals in the right hand side of the above equality are independent, consequently we can group them as follows:

$$\begin{aligned} &\left(\int_\theta \prod_{u \in U} \Pr\left(\theta_u | \alpha\right) \prod_{a \in \mathbf{A}_u} \Pr\left(z_{u,a} | \theta_u\right) d\theta\right) \cdot \\ &\left(\prod_{d=1}^{n} \int_{\phi^{(d)}} \left[\prod_{k=1}^{K} \Pr\left(\phi_k^{(d)} | \beta^{(d)}\right)\right] \prod_{u \in U} \prod_{a \in \mathbf{A}_u} \Pr\left(i_{u,a}^{(d)} | \phi_{z_{u,a}}^{(d)}\right) d\phi^{(d)}\right) \ . \end{aligned} \tag{1}$$

Let us consider the first integral in the above formula, and let us assume that n_u^k equals the number of times the user u is associated to the latent variable k, then:

$$\prod_{a \in \mathbf{A}_u} \Pr\left(z_{u,a} | \theta_u\right) = \prod_{k=1}^{K} \theta_{u,k}^{n_u^k} \ ,$$

moreover $\Pr\left(\theta_u | \alpha\right)$ is given by a Dirichlet distribution, and then

$$\Pr\left(\theta_u | \alpha\right) = \frac{1}{\mathit{\Delta}\left(\alpha\right)} \prod_{k=1}^{K} \theta_{u,k}^{\alpha_k - 1} \ ,$$

where for any vector $\mathbf{w} = \{w_1, \ldots, \ w_r\}$, $\mathit{\Delta}\left(\mathbf{w}\right)$ is defined by:

$$\mathit{\Delta}\left(\mathbf{w}\right) = \frac{\prod_{i=1}^{r} \mathit{\Gamma}\left(w_i\right)}{\mathit{\Gamma}\left(\sum_{i=1}^{r} w_i\right)} \ ,$$

$\Gamma(\cdot)$ denotes the Gamma function. By straightforward algebraic manipulations, we can rewrite the first integral in (1) as:

$$\int_\theta \prod_{u\in U} \Pr(\theta_u|\alpha) \prod_{a\in\mathbf{A}_u} \Pr(z_{u,a}|\theta_u) d\theta =$$

$$\prod_{u\in U} \int_{\theta_u} \Pr(\theta_u|\alpha) \prod_{a\in\mathbf{A}_u} \Pr(z_{u,a}|\theta_u) d\theta_u = \prod_{u\in U} \int_{\theta_u} \frac{1}{\Delta(\alpha)} \left(\prod_{k=1}^{K} \theta_{u,k}^{\alpha_k - 1} \right) \prod_{k=1}^{K} \theta_{u,k}^{n_u^k} d\theta_u =$$

$$\prod_{u\in U} \frac{\Delta(\alpha + \mathbf{n}_u)}{\Delta(\alpha)} \int_{\theta_u} \frac{1}{\Delta(\alpha + \mathbf{n}_u)} \prod_{k=1}^{K} \theta_{u,k}^{n_u^k + \alpha_k - 1} d\theta_u \ ,$$

where $\mathbf{n}_u = \left\{ n_u^k \right\}_{k=1}^{K}$. Observe that the above integral contains a Dirichlet distribution that integrated over the whole domain is equal to one, and then finally:

$$\int_\theta \prod_{u\in U} \Pr(\theta_u|\alpha) \prod_{a\in\mathbf{A}_u} \Pr(z_{u,a}|\theta_u) d\theta = \prod_{u\in U} \frac{\Delta(\alpha + \mathbf{n}_u)}{\Delta(\alpha)} \ .$$

By a similar argument, we can compute the second integral in (1). Indeed we have:

$$\Pr\left(\phi_k^{(d)}|\beta^{(d)}\right) = \frac{1}{\Delta(\beta)} \left(\prod_{i^{(d)} \in val(d)} \left(\phi_{k,i^{(d)}}^{(d)}\right)^{\beta_{i^{(d)}} - 1} \right) \ ,$$

and

$$\prod_{u\in U} \prod_{a\in\mathbf{A}_u} \Pr\left(i_{u,a}^{(d)}|\phi_{z_{u,a}}^{(d)}\right) = \prod_{k=1}^{K} \prod_{i^{(d)} \in val(d)} \left(\phi_{k,i^{(d)}}^{(d)}\right)^{n_{i^{(d)}}^k}$$

then

$$\prod_{d=1}^{n} \int_{\phi^{(d)}} \left[\prod_{k=1}^{K} \Pr\left(\phi_k^{(d)}|\beta^{(d)}\right) \right] \prod_{u\in U} \prod_{a\in\mathbf{A}_u} \Pr\left(i_{u,a}^{(d)}|\phi_{z_{u,a}}^{(d)}\right) d\phi^{(d)} =$$

$$\prod_{d=1}^{n} \int_{\phi^{(d)}} \left[\prod_{k=1}^{K} \frac{1}{\Delta(\beta)} \left(\prod_{i^{(d)} \in val(d)} \left(\phi_{k,i^{(d)}}^{(d)}\right)^{\beta_{i^{(d)}} - 1} \right) \right] \prod_{k=1}^{K} \prod_{i^{(d)} \in val(d)} \left(\phi_{k,i^{(d)}}^{(d)}\right)^{n_{i^{(d)}}^k} d\phi^{(d)} =$$

$$\prod_{d=1}^{n} \prod_{k=1}^{K} \frac{\Delta\left(\beta + \mathbf{n}_d^k\right)}{\Delta(\beta)} \int_{\phi_k^{(d)}} \frac{1}{\Delta\left(\beta + \mathbf{n}_d^k\right)} \prod_{i^{(d)} \in val(d)} \left(\phi_{k,i^{(d)}}^{(d)}\right)^{n_{i^{(d)}}^k + \beta_{i^{(d)}} - 1} d\phi_k^{(d)} \ ,$$

where $\mathbf{n}_d^k = \left\{ n_{i^{(d)}}^k \right\}_{i^{(d)} \in val(d)}$. Summing up:

$$\prod_{d=1}^{n} \int_{\phi^{(d)}} \left[\prod_{k=1}^{K} \Pr\left(\phi_k^{(d)}|\beta^{(d)}\right) \right] \prod_{u\in U} \prod_{a\in\mathbf{A}_u} \Pr\left(i_{u,a}^{(d)}|\phi_{z_{u,a}}^{(d)}\right) d\phi^{(d)} = \prod_{d=1}^{n} \prod_{k=1}^{K} \frac{\Delta\left(\beta + \mathbf{n}_d^k\right)}{\Delta(\beta)} \ ,$$

and finally we can write the complete-data likelihood:

$$\Pr(\mathbf{A}, \mathbf{Z}|\alpha, \beta) = \left(\prod_{u\in U} \frac{\Delta(\alpha + \mathbf{n}_u)}{\Delta(\alpha)} \right) \cdot \prod_{d=1}^{n} \prod_{k=1}^{K} \frac{\Delta\left(\beta + \mathbf{n}_d^k\right)}{\Delta(\beta)} \ . \qquad (2)$$

2.2 Final equations

The formulation of the likelihood contained in Equation (2), in order to be computationally tractable, must be simplified by means of some kind of heuristic argument and approximation. This can be accomplished in many ways, in this work we choose Gibbs Sampling (see [9], [8] and [14]) that is based on Monte Carlo methods. Finally, the MDLDA model employed in our experiments can be summarized as follows.

- Sampling of latent variables:

$$\Pr\left(z_{u,a}=k|Z_{-u,a},A;\alpha,\beta\right)\propto \left(n_u^k+\alpha_k-1\right)\prod_{d\in Dim(a)}\frac{n_{i^{(d)}}^k+\beta_{i^{(d)}}^{(d)}-1}{-1+\sum_{j^{(d)}}n_{j^{(d)}}^k+\beta_{j^{(d)}}^{(d)}}\ . \tag{3}$$

- Computation of behavioral profiles:

$$\theta_{u,k}=\frac{n_u^k+\alpha_k}{\sum_{h=1}^K n_u^h+\alpha_h}\ . \tag{4}$$

- Computation of similarity between domains and latent variables:

$$\phi_{k,i^{(d)}}^d=\frac{n_{i^{(d)}}^k+\beta_{i^{(d)}}^{(d)}}{\sum_{j^{(d)}}n_{j^{(d)}}^k+\beta_{j^{(d)}}^{(d)}}\ . \tag{5}$$

- Moreover we add a module that updates hyper parameters of behavioral profiles:

$$\alpha_k^{\text{new}}=\alpha_k\frac{\sum_{u\in U}\left[\Psi\left(n_u^k+\alpha_k\right)-\Psi\left(\alpha_k\right)\right]}{\sum_{u\in U}\left[\Psi\left(\sum_{k=1}^K n_u^k+\alpha_k\right)-\Psi\left(\sum_{k=1}^K\alpha_k\right)\right]}\ , \tag{6}$$

  where $\Psi\left(\cdot\right)$ represents the function Digamma, see Minka [13] for the above formulation.

2.3 Inference algorithm

The formulation described in previous Section is the base for an algorithm that infers behavioral profiles, the algorithm is illustrated in the following pseudocode contained in Algorithm 1. The starting point of the algorithm is a random initialization of latent variables, followed by a fixed point search procedure based on Gibbs Sampling. The fixed point search has two main steps. The first step, called *burn-in*, is intended to mitigate the influence of the random initialization, this is obtained by resampling various times latent variables. As soon as burn-in is finished, for each *sample-lag* iteration, user profiles are computed. Once the maximal number of iterations of fixed point search is reached, the algorithm returns users behavioral profiles as mean over all results of sample-lag, and associations between latent variables and contextual domains values.

```
input  : $nIterations$, maximal number of iterations; $burnIn$, number of iterations of the burn-in step
output: $\mathbf{\Theta}$ behavioral profiles matrix; $\mathbf{\Phi}^{(d)}$ latent variables – values of dimensions matrix
$\mathbf{\Theta}^* \leftarrow \varnothing$
for $d \in D$ do
    $\mathbf{\Phi}^{*(d)} \leftarrow \varnothing$
end
for $u \in U$ do
    for $a \in A_u$ do
        Choose latent factor $z_{u,a} \sim Uniform\,(K)$
    end
end
for $it \leftarrow 1$ to $nIterations$ do
    for $u \in U$ do
        for $a \in A_u$ do
            Update $z_{u,a} \sim \Pr\left(z_{u,a} | \mathbf{Z}_{-u,a}, \mathbf{A}; \alpha, \beta\right)$, (3)
        end
    end
    if $it > burnIn \wedge (it \equiv 0 \mod \mathit{sampleLag})$ then
        Compute $\mathbf{\Theta}$, (4)
        $\mathbf{\Theta}^* = \mathbf{\Theta}^* \cup \mathbf{\Theta}$
        for $d \in D$ do
            Compute $\mathbf{\Phi}^{(d)}$, (5)
            $\mathbf{\Phi}^{*(d)} = \mathbf{\Phi}^{*(d)} \cup \mathbf{\Phi}^{(d)}$
        end
    end
    Update $\alpha$, (6)
end
return $\mathit{mean}\,(\mathbf{\Theta}^*)$ and for each $d \in D$, $\mathit{mean}\left(\mathbf{\Phi}^{*(d)}\right)$
```

Algorithm 1: Pseudocode for MDLDA's parameter inference.

3 Data Model

The data, that feed the model for the user profile definition, consist of an observed-action log for each user. In our approach, a user is represented by a numeric identifier, while an action is a n-ary vector of discrete values, defined on domains that represent the different dimensions. A dimension is an arbitrary feature that characterize the action, for instance it could be the object of the action, the time when it is performed, where it takes place, the number and the type of devices used (e.g. computer, telephone, etc.), the services involved in the activity, the accessed files, and so on.

3.1 Data Manipulation

The number and the type of the dimensions are related to the scenario to analyze. In a business environment, some dimensions, related to security and permissions issues, are useful to describe a user behavior, but in a setting where all users have the same rights or have access to same places/documents such dimension types are not suitable. To overcome these issues, the model generalizes the action and the dimension definition. Each dimension is converted in a domain with a fixed discrete range of possible values. The discretization process converts all real values (the time for a time context, the places for a location type, etc.) in nominal values. Obviously, the so define data structure is a sparse three-dimensional tensor: the first dimension represents users, the second one actions, while the last one the action features. Since features are defined by finite sets of values, the mapping procedure is very simple: each value is indexed and replaced by the corresponding index. If the dimension values are defined in a continuous range (e.g. measures), then a discretize function must be applied to identify a finite set of intervals. For a date type field, the previous approaches are not suitable. In this case, a target time is considered as year zero and all values are expressed by the difference with the target time (and discretized to a finite set). For instance, a dimension that represents a date in a range of two years can be expressed as the number of days since the initial date. Each user action is defined as a tuple whose fields are the user unique identifier, the action unique identifier and a sequence of integers mapping all dimension values.

As aforesaid, data are stored in a cubic structure in which the users, the performed actions and the dimensions are stored in the first, second and third cube's dimension, respectively. A possible software implementation of the data cube is based on a sparse matrix structure: the missing values for each dimension are ignored, since they are empty value in the abstract matrix. Moreover, all users do not take the same number of actions and the corresponding row (the row of the a-th action with all dimension values) is completely missing. A sparse structure handles very well these data arrangement and the sparse matrix multiplication algorithms can improve the computation time.

4 Experimental Evaluation

In order to validate the utility of our approach in discovering user profiles, an experiment using a real dataset is conducted. The main goal of this section is to assess the capacity of the framework in profiling users exploiting simultaneously their exhibited behavior and environmental information. In the next subsection, more details are supplied concerning the dataset used.

4.1 Dataset

A real dataset, hereafter named *MovieLens1M*, have been employed to evaluate the proposed model. The data were collected in the context of "The GroupLens Research Project" developed within the Department of Computer Science and Engineering at the University of Minnesota [11]. This dataset contains 1,000,209 anonymous ratings on approximately 3,700 movies made by 6,040 users and it is widely used as a benchmark for several mining tasks.

MovieLens1M contains tuples $\langle u; i; r; t \rangle$, where u and i are, respectively, user and item identifiers, t is a timestamp, $r \in \{1, ..., 5\}$ is the rating value. Specifically, the dataset used for the evaluation contains the preferences exhibited by a group of users who have used the portal MovieLens between 1997 and 1998. In order to extract a useful sample for analyzing the model performances, the data were selected according to 2 criteria: *(i)* each user has at least 20 ratings, *(ii)* the selected users had to insert their demographics data into their portal account.

Side information is also available, both for users (*gender, age, occupation, Zip-code*) and movies (*title, genre*). In our case, these attributes represent the

Table 3. Discretization of the users' ages.

Age Label	**Real Age**
1	<18
18	18 - 24
25	25 - 34
35	35 - 44
45	45 - 49
50	50 - 55
56	>55

different dimensions that characterize user actions, in other words their preferences. The real age value was replaced with a label according to Table 3.

Occupation and genre are represented using codes. So, the fully-preprocessed dataset was obtained by combining user preferences, demographic information and information about movies. Following we report a table record as example:

$$u1; F; 45; 10; 2389; 0; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 4$$

the first four attributes are the user's code, the gender, the age and the employment code, respectively. The value 2389 is the code of the movie (in this case Psycho 1998). The next sequence of eighteen numbers refers to the categories: if a category field is valued 1 the movie belongs to category, 0 otherwise. In this case, the categories valued with 1 are *Crime, Horror and Thriller*. The last value is the rating provided by this user: in this case four stars.

4.2 User Profiles Discovery Analysis

MovieLens1M was used as case study to evaluate the proposed algorithm performance in discovering user profiles. In this application scenario, a behavioral profile identifies a group of users who share both a preference for a particular type of movie and additional features which can be exploited to predict their interests. The experiment consists of two steps. First, we apply the algorithm for the behavioral profiles discovery and a clustering algorithm for identifying the groups. Then, we analyzed the results by assigning semantics to the identified groups.

The algorithm MDLDA extracts a set of latent variables from the data, i.e. the links between the users and the provided preference. Latent variables allow us to associate the users to the movies which can meet their interests. In particular, the aim of this analysis is to understand the features that the users, belonging to a particular group, share. In order to cluster the users, we used the EM (Expectation-Maximization) clustering algorithm with a random initialization. We identified 10 user's groups which exhibit similar behaviors and exploiting the result of the MDLDA algorithm we have been able to discriminate the representative features for each cluster.

In Table 4, we report the parameters' setting for MDLDA algorithm.

Table 4. Parameter Setting.

Parameter	**Value**
symmetricAlpha	2
symmetricBeta	0.1
maxIterations	800
burnIn	300
sampleLag	25
seed	101
updateHyperParams	True
maxAlphaUpdateIterations	30

For the clustering step, the *EM* [7] algorithm was performed with the following parameters setting: 10 clusters, 100 trials, minimal standard deviation 1*10-6 and seed 100.

4.3 Evaluation Results

In this section we analyzed the clustering results. First, we selected the most influential latent variables for each cluster and then, for each latent variable we identified the user attributes and the predominant genres.

In Table 5, we show for each cluster the most characterizing latent variables. In some cases, we can note that a cluster is denoted by more latent variables.

Table 5. Latent Variables - Clusters

LV/Cluster	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**
LV#1	0.0167	0.0155	0.016	0.0171	0.017	0.0162	**0.1291**	0.0142	0.0164	0.0164
LV#2	0.0175	0.0155	0.0161	0.017	0.017	0.016	0.0787	0.0143	0.0165	0.0164
LV#3	0.0169	0.0155	0.0167	**0.6425**	0.017	0.0165	0.0165	0.0142	0.0164	0.0164
LV#4	0.0169	0.0155	0.016	0.017	0.017	0.0161	0.0162	**0.6963**	0.0164	0.0164
LV#5	0.0168	0.0158	0.0166	0.0177	0.0179	0.016	**0.1061**	0.0142	0.0165	0.0168
LV#6	0.0183	0.0164	0.016	0.017	0.0171	**0.209**	0.0164	0.0144	0.0174	0.0165
LV#7	0.0169	0.0155	0.0169	0.017	0.017	0.0159	**0.0947**	0.0142	0.0164	0.0166
LV#8	0.0168	**0.6478**	0.0162	0.0172	0.017	0.0159	0.0162	0.0142	0.0164	0.0165
LV#9	0.0168	0.0158	0.016	0.017	0.017	0.0165	0.0163	0.0142	**0.6191**	0.0166
LV#10	0.0179	0.0157	**0.6258**	0.017	0.017	0.016	0.0164	0.0142	0.0168	0.0164
LV#11	0.057	0.0703	0.0826	0.0495	0.0218	0.0755	0.0545	0.0446	0.0833	0.0644
LV#12	0.0167	0.0157	0.016	0.0174	0.0174	0.0161	0.0162	0.0142	0.0164	**0.6378**
LV#13	**0.6357**	0.0159	0.016	0.017	0.017	0.016	0.0162	0.0159	0.0164	0.0169
LV#14	0.0168	0.0155	0.016	0.017	**0.6701**	0.016	0.0163	0.0148	0.0164	0.0164
LV#15	0.0168	0.0155	0.0168	0.017	0.017	0.0159	0.0893	0.0142	0.017	0.0164
LV#16	0.0167	0.0156	0.0167	0.017	0.017	**0.2355**	0.0163	0.0142	0.0164	0.0164
LV#17	0.0181	0.0156	0.016	0.017	0.017	0.0159	**0.0954**	0.0153	0.0164	0.0165
LV#18	0.0167	0.0155	0.016	0.017	0.0172	0.0159	0.0874	0.0142	0.0164	0.0164
LV#19	0.0167	0.0155	0.016	0.017	0.0171	0.016	0.0853	0.0142	0.0164	0.0173
LV#20	0.0172	0.0157	0.016	0.0179	0.0174	**0.223**	0.0163	0.0142	0.0165	0.0167

In Table 6 we show the distribution of the instances for each cluster. A preliminary analysis allows to identify a larger cluster compared with the others, which vice versa exhibit similar sizes.

A more detailed analysis, based on the discriminative latent variables, permits to associate interesting knowledge to the clusters. The identified features don't allow to model all users, but are discriminative for the group. For example,

Table 6. Cluster Distributions

ClusterId	**Distribution**
C1	410 (7%)
C2	259 (4%)
C3	409 (7%)
C4	322 (5%)
C5	281 (5%)
C6	895 (15%)
C7	2443 (40%)
C8	414 (7%)
C9	319 (5%)
C10	288 (5%)

the cluster 1 is composed by male users (59% of the cluster), in particular students aged between 18 and 24 years. The mainly genres watched by these users are drama and comedy. Another group of users (cluster 4) contains women aged between 25 and 34 years employed in the health sector. The preferred genres are comedy and romance. The cluster 2 is characterized by male users (68% of the cluster) between 25 and 34 years who prefer action movies. Many users belonging to this cluster are artists. Another interesting group is cluster 9. In this case, the 80% of the users are young men (25-34), who work as programmers and watch thriller movies. The largest cluster is made up of retirees, teachers or engineers and most part of users belong mainly to two age groups: 25-34 years and over-56. It is the largest cluster so it is difficult to identify a predominant gender (male / female). In this case, comedy is the most watched genre.

These results prove the effectiveness of the algorithm in profiling and identifying groups of users that exhibit similar behavior. The obtained profiles describe mainly young, with high level of education and who are big “consumers” of movies. This demographic composition depends on how the data were collected. MovieLens portal is a website used by many “movie-goers”, and in particular in the reference period (1997-98), the typical user of Internet was a person with a high level of education.

5 Conclusion

In this paper we proposed a novel approach for the Human Behavior Computing based on Multidimensional Latent Factor Modeling. We defined an extension of the well-known approach LDA, namely Multidimensional Latent Dirichlet Allocation (MDLDA), capable of deal with arbitrary multivariate elements. An experimental experience has been shown in order to prove the utility of a multidimensional perspective on human behavior. This paper represents a preliminary work in this direction, that, with further investigation, seems to be a promising branch of research for a better fitting of models for human actions.

References

1. Agarwal, D., Chen, B.C.: Regression-based latent factor models. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 19–28. KDD '09, ACM, New York, NY, USA (2009)
2. Barbieri, N., Costa, G., Manco, G., Ortale, R.: Modeling item selection and relevance for accurate recommendations: A bayesian approach. In: Proceedings of the Fifth ACM Conference on Recommender Systems. pp. 21–28. RecSys '11, ACM, New York, NY, USA (2011), `http://doi.acm.org/10.1145/2043932.2043941`
3. Barbieri, N., Manco, G., Ortale, R., Ritacco, E.: Balancing prediction and recommendation accuracy: Hierarchical latent factors for preference data. In: Proceedings of the Twelfth SIAM International Conference on Data Mining, Anaheim, California, USA, April 26-28, 2012. pp. 1035–1046
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
5. Cao, L., Yu, P.S.: Behavior Computing: Modeling, Analysis, Mining and Decision. Springer Publishing Company, Incorporated (2014)
6. Cheng, J., Yuan, T., Wang, J., Lu, H.: Group latent factor model for recommendation with multiple user behaviors. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 995–998. SIGIR '14, ACM, New York, NY, USA (2014)
7. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society, Series B 39(1), 1–38 (1977)
8. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National academy of Sciences 101(suppl 1), 5228–5235 (2004)
9. Griffiths, T.: Gibbs sampling in the generative model of latent dirichlet allocation (2002)
10. Guerzhoy, M., Hertzmann, A.: Learning latent factor models of travel data for travel prediction and analysis. In: Advances in Artificial Intelligence - 27th Canadian Conference on Artificial Intelligence, Canadian AI 2014, Montréal, QC, Canada, May 6-9, 2014. Proceedings. pp. 131–142 (2014)
11. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. 5(4), 19:1–19:19 (Dec 2015)
12. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the Fourth ACM Conference on Recommender Systems. pp. 135–142. RecSys '10, ACM, New York, NY, USA (2010)
13. Minka, T.: Estimating a dirichlet distribution (2000)
14. Pritchard, J.K., Stephens, M., Donnelly, P.: Inference of population structure using multilocus genotype data. Genetics 155(2), 945–959 (2000)
15. Skondral, A., Rabe-Hesketh, S.: Latent variable modelling: A survey. Scandinavian Journal of Statistics 34(4), 712–745 (2007)
16. Urtasun, R., Darrell, T.: Discriminative gaussian process latent variable model for classification. In: Proceedings of the 24th International Conference on Machine Learning. pp. 927–934. ICML '07, ACM, New York, NY, USA (2007), `http://doi.acm.org/10.1145/1273496.1273613`