JMLR: Workshop and Conference Proceedings 20 (2011) 213–229 Asian Conference on Machine Learning

Mixed-Variate Restricted Boltzmann Machines

Truyen Tran T.TRAN2@CURTIN.EDU.AU
Dinh Phung D.PHUNG@CURTIN.EDU.AU
Svetha Venkatesh S.VENKATESH@CURTIN.EDU.AU
Curtin University

Editor: Chun-Nan Hsu and Wee Sun Lee

Abstract

Modern datasets are becoming heterogeneous. To this end, we present in this paper *Mixed-Variate Restricted Boltzmann Machines* for simultaneously modelling variables of multiple types and modalities, including *binary* and *continuous* responses, *categorical* options, *multicategorical* choices, *ordinal* assessment and *category-ranked* preferences. Dependency among variables is modeled using latent binary variables, each of which can be interpreted as a particular hidden aspect of the data. The proposed model, similar to the standard RBMs, allows fast evaluation of the posterior for the latent variables. Hence, it is naturally suitable for many common tasks including, but not limited to, (a) as a pre-processing step to convert complex input data into a more convenient vectorial representation through the latent posteriors, thereby offering a dimensionality reduction capacity, (b) as a classifier supporting binary, multiclass, multilabel, and label-ranking outputs, or a regression tool for continuous outputs and (c) as a data completion tool for multimodal and heterogeneous data. We evaluate the proposed model on a large-scale dataset using the world opinion survey results on three tasks: feature extraction and visualization, data completion and prediction.

1. Introduction

Restricted Boltzmann Machines (RBM) (Hinton and Sejnowski, 1986; Freund and Haussler, 1993) have recently attracted an increasing attention for their rich capacity in a variety of learning tasks, including multivariate distribution modelling, feature extraction, classification, and construction of deep architectures (Hinton and Salakhutdinov, 2006; Salakhutdinov and Hinton, 2009a). An RBM is a two-layer Markov random field in which the visible layer represents observed variables and the hidden layer represents latent aspects of the data. Pairwise interactions are only permitted for units between layers. As a result, the posterior distribution over the hidden variables and the probability of the data generative model are easy to evaluate, allowing fast feature extraction and efficient sampling-based inference (Hinton, 2002). Nonetheless, most existing work in RBMs implicitly assumes that the visible layer contains variables of the same modality. By far the most popular input types are binary (Freund and Haussler, 1993) and Gaussian (Hinton and Salakhutdinov, 2006). Recent extension includes categorical (Salakhutdinov et al., 2007), ordinal (Truyen et al., 2009), Poisson (Gehler et al., 2006) and Beta (Le Roux et al., 2011) data. To the best of our knowledge, *none* has been considered for multicategorical and category-ranking data, *nor* for a mixed combination of these data types.

In this paper, we investigate a generalisation of the RBM for variables of multiple modalities and types. Take, for example, data from a typical survey, where a person is asked a variety of questions in many styles ranging from yes/no to multiple choices and preference statements. Typically, there are six question/answer types: (1) binary responses (e.g., satisfied vs. unsatisfied), (2) categorical options (e.g., one of employed, unemployed or retired), (iii) multicategorical choices (e.g., any of family, education or income), (iv) continuous information (e.g. age), (v) ordinal assessment (e.g., one of good, neural or bad), and (vi) category-ranked preferences (e.g., in the decreasing order of importance: children, security, food and money). As the answers in a response come from the same person, they are inherently correlated. For instance, a young American is likely to own a computer, whilst a typical Chinese adult may concern more about their children's education. However, modelling the direct correlation among multiple types is difficult. We show, on the other hand, a two-layer RBM is well-suited for this problem. First, its undirected graphical structure allows a great flexibility to encode all six data types into the same probability distribution. Second, the binary hidden layer pools information from visible units and redistributes to all others, thereby introducing dependencies among variables. We term our model the *Mixed-Variate Restricted Boltzmann Machines* (MV.RBM).

The MV.RBM has the capacity of supporting a variety of machine learning tasks. Its posteriors can be used as a vectorial representation of the data hiding away the obscured nature of the observed data. As the result, we can use MV.RBM for data pre-processing, visualisation, and dimensionality reduction. Given the hidden layer, the original and missing observables can also be reconstructed through the generative data model. By splitting the observed data into an input and output sets, predictive models can be learnt to perform classification, ranking or regression. These capacities are demonstrated in this paper on a large-scale international opinion survey across 44 nations involving more than 38 thousand people.

2. Mixed-Variate Restricted Boltzmann Machines

In this section we present Mixed-Variate Restricted Boltzmann Machines (MV.RBM) for jointly modelling variables of multiple modalities and types. For ease of following the text, we include a notation description in Table 1.

2.1. Model Definition

Denote by $\boldsymbol{v} = (v_1, v_2, ..., v_N)$ the set of *mixed-variate* visible variables where each v_i can be one of the following types: *binary*, *categorical*, *multicategorical*, *continuous*, *ordinal* or *category-ranked*. Let $\boldsymbol{v}_{disc}$ be the joint set of discrete elements and $\boldsymbol{v}_{cont}$ be the continuous set, and thus $\boldsymbol{v} = (\boldsymbol{v}_{disc}, \boldsymbol{v}_{cont})$. Denoting by $\boldsymbol{h} = (h_1, h_2, ..., h_K) \in \{0, 1\}^K$ the hidden variables, the model distribution of MV.RBM is defined as

$$P(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} \exp\{-E(\boldsymbol{v}, \boldsymbol{h})\}, \tag{1}$$

v_i	Single visible variable	$G_i(v_i)$,$H_{ik}(v_i)$	Functions of an input variable
$\boldsymbol{v}$	A set of visible variables	U_i,U_{im},U_{id}	Input bias parameters
h_k	Single hidden variable	V_{ik},V_{imk},V_{idk}	Input-hidden parameters
$\boldsymbol{h}$	A set of hidden variables	w_k	Hidden bias parameter
$Z(\cdot)$	Normalising function	a_{im}	Activation indicator
$\succ$,$\succeq$,$\triangleright$	Ordinal relations	$\mathbb{S}_i$	Set of categories
$\simeq$	Indifference	M_i	The number of categories
N	Number of visible units	c_{im}	Category member of set $\mathbb{S}_i$
K	Number of hidden units	$\delta_m[v_i]$,$\mathbb{I}[\cdot]$	Indicator functions
$P(\cdot)$	Probability distribution	C	Index of a subset of variables
$E(\cdot)$	Energy function	$\mathcal{L}$	Data log-likelihood

Table 1: Notations used in this paper.

where $E(\boldsymbol{v}, \boldsymbol{h})$ is the model energy, Z is the normalisation constant. The model energy is further decomposed into a sum of singleton and pairwise energies:

$$E(\boldsymbol{v}, \boldsymbol{h}) = \sum_{i=1}^{N} E_i(v_i) + \sum_{k=1}^{K} E_k(h_k) + \sum_{i=1}^{N}\sum_{k=1}^{K} E_{ik}(v_i, h_k),$$

where $E_i(v_i)$ depends only on the i-th visible unit, $E_k(h_k)$ on the k-th hidden unit, and $E_{ik}(v_i, h_k)$ on the interaction between the i-th visible and k-hidden units. The MV.RBM is thus a 2-layer mixed-variate Markov random field with pairwise connectivity across layers.

For the distribution in Eq. (1) to be properly specified, we need to keep the normalisation constant finite. In other words, the following integration

$$Z = \int_{\boldsymbol{v}_{cont}} \left(\sum_{\boldsymbol{v}_{disc}} \sum_{\boldsymbol{h}} \exp\{-E(\boldsymbol{v}_{disc}, \boldsymbol{v}_{cont}, \boldsymbol{h})\} \right) d(\boldsymbol{v}_{cont})$$

must be bounded from above. One way is to choose appropriate continuous variable types with bounded moments, e.g., Gaussian. Another way is to explicitly bound the continuous variables to some finite ball, i.e., $\|\boldsymbol{v}_{cont}\| \leq R$.

In our MV.RBM, we further assume that the energies have the following form:

$$E_i(v_i) = -G_i(v_i); \qquad E_k(h_k) = -w_k h_k; \qquad E_{ik}(v_i, h_k) = -H_{ik}(v_i) h_k, \tag{2}$$

where w_k is the bias parameter for the k-th hidden unit, and $G_i(v_i)$ and $H_{ik}(v_i)$ are functions to be specified for each data type. An important consequence of this energy decomposition is the factorisation of the *posterior*:

$$P(\boldsymbol{h} \mid \boldsymbol{v}) \;=\; \prod_k P(h_k \mid \boldsymbol{v}); \qquad P(h_k^1 \mid \boldsymbol{v}) = \frac{1}{1 + \exp\{-w_k - \sum_i H_{ik}(v_i)\}}, \tag{3}$$

where h_k^1 denotes the assignment $h_k = 1$. This posterior is efficient to evaluate, and thus the vector $\big(P(h_k^1 \mid \boldsymbol{v}), P(h_k^2 \mid \boldsymbol{v}), ..., P(h_k^K \mid \boldsymbol{v})\big)$ can be used as extracted features for mixed-variate input $\boldsymbol{v}$.

Similarly, the *data model* $P(\boldsymbol{v}|\boldsymbol{h})$ has the following factorisation

$$P(\boldsymbol{v} \mid \boldsymbol{h}) = \prod_i P_i(v_i \mid \boldsymbol{h}); \qquad P_i(v_i \mid \boldsymbol{h}) = \frac{1}{Z(\boldsymbol{h})} \exp\{G_i(v_i) + \sum_k H_{ik}(v_i)h_k\}, \quad (4)$$

where $Z(\boldsymbol{h}) = \sum_{v_i} \exp\{G_i(v_i) + \sum_k H_{ik}(v_i)h_k\}$ if v_i is discrete and $Z(\boldsymbol{h}) = \int_{v_i} \exp\{G_i(v_i) + \sum_k H_{ik}(v_i)h_k\} d(v_i)$ if v_i is continuous, assuming that the integration exists. Note that we deliberately use the subscript index i in $P_i(\cdot \mid \boldsymbol{h})$ to emphasize the heterogeneous nature of the input variables.

2.2. Type-specific Data Models

We now specify $P_i(v_i|\mathbf{h})$ in Eq. (4), or equivalently, the functionals $G_i(v_i)$ and $H_{ik}(v_i)$. Denote by $\mathbb{S}_i = (c_{i1}, c_{i2}, ..., c_{iM_i})$ the set of categories in the case of discrete variables. In this section, for continuous types, we limit to Gaussian variables as they are the by far the most common. Interested readers are referred to (Le Roux et al., 2011) for Beta variables in the context of image modelling. The data model and related functionals for binary, Gaussian and categorical data types are well-known, and thus we provide a summary here:

	$G_i(v_i)$	$H_{ik}(v_i)$	$P_i(v_i\|\boldsymbol{h})$
–Binary	$U_i v_i$	$V_{ik} v_i$	$\frac{\exp\{U_i v_i + \sum_k V_{ik} h_k v_i\}}{1+\exp\{U_i + \sum_k V_{ik} h_k\}}$
–Gaussian	$-v_i^2/2\sigma_i^2 + U_i v_i$	$V_{ik} v_i$	$\mathcal{N}\left(\sigma_i^2 \left(U_i + \sum_k V_{ik} h_k\right); \sigma_i\right)$
–Categorical	$\sum_m U_{im} \delta_m[v_i]$	$\sum_{m,k} V_{imk} \delta_m[v_i]$	$\frac{\exp\{\sum_m U_{im}\delta_m[v_i] + \sum_{m,k} V_{imk}\delta_m[v_i]h_k\}}{\sum_l \exp\{U_{il} + \sum_k V_{ilk} h_k\}}$

where $m = 1, 2, ..., M_i$; $U_i, V_{ik}, U_{im}, V_{imk}$ are model parameters; and $\delta_m[v_i] = 1$ if $v_i = c_{im}$ and 0 otherwise.

The cases of multicategorical, ordinal and category-ranking variables are, however, much more involved, and thus some further simplification may be necessary. In what follows, we describe the specification details for these three cases.

2.2.1. Multicategorical Variables

An assignment to a multicategorical variable has the form of a subset from a set of categories. For example, a person may be interested in games and music from a set of offers: games, sports, music, and photography. More formally, let $\mathbb{S}_i$ be the set of categories for the i-th variable, and $\mathcal{P}_i = 2^{\mathbb{S}_i}$ be the power set of $\mathbb{S}_i$ (the set of all possible subsets of $\mathbb{S}_i$). Each variable assignment consists of a non-empty element of $\mathcal{P}_i$, i.e. $v_i \in \{\mathcal{P}_i \backslash \emptyset\}$. Since there are $2^{M_i} - 1$ possible ways to select a non-empty subset, directly enumerating $P_i(v_i|\boldsymbol{h})$ proves to be highly difficult even for moderate sets. To handle this state explosion, we first assign each category c_{im} with a binary indicator $a_{im} \in \{0, 1\}$ to indicate whether the m-th category is active, that is $v_i = (a_{i1}, a_{i2}, ..., a_{iM_i})$. We then assume the following factorisation:

$$P_i(v_i|\boldsymbol{h}) = \prod_{m=1}^{M_i} P_i(a_{im}|\boldsymbol{h}). \quad (5)$$

Note that this does not says that binary indicators are independent in their own right but given the knowledge of the hidden variables $\boldsymbol{h}$. Since they hidden variables are never observed, binary indicators are therefore interdependent. Now, the probability for activating a binary indicator is defined as

$$P_i(a_{im} = 1|\boldsymbol{h}) = \frac{1}{1 + \exp(-U_{im} - \sum_k V_{imk} h_k)}. \tag{6}$$

Note that this specification is equivalent to the following decomposition of the functionals $G_i(v_i)$ and $H_{ik}(v_i)$ in Eq. (2):

$$G_i(v_i) = \sum_{m=1}^{M_i} U_{im} a_{im}; \qquad H_{ik}(v_i) = \sum_{m=1}^{M_i} V_{imk} a_{im}.$$

2.2.2. Ordinal Variables

An ordinal variable receives individual values from an ordinal set $S_i = \{c_{i1} \prec c_{i2} \prec ..., \prec c_{iM_i}\}$ where $\prec$ denotes the order in some sense. For example, c_{im} can be a numerical rating from a review, or it can be sentimental expression such as love, neutral and hate. There are two straightforward ways to treat an ordinal variable: (i) one is simply ignoring the order, and considering it as a multinomial variable, and (ii) another way is to convert the ordinal expression into some numerical scale, for example, $\{-1, 0, +1\}$ for the triple {love,neutral,hate} and then proceed as if it is a continuous variable. However, in the first treatment, substantial ordinal information is lost, and in the second treatment, there is no satisfying interpretation using numbers.

In this paper, we adapt the Stereotype Ordered Regression Model (SORM) by Anderson (1984). More specifically, the SORM defines the conditional distribution as follows

$$P(v_i = m \mid \boldsymbol{h}) = \frac{\exp\{U_{im} + \sum_{d=1}^{D} \sum_{k=1}^{K} V_{idk} \phi_{id}(m) h_k\}}{\sum_l \exp\{U_{il} + \sum_{d=1}^{D} \sum_{k=1}^{K} V_{idk} \phi_{id}(l) h_k\}}$$

where U_{im}, V_{idk} are free parameters, $D \leq M_i$ is the dimensionality of the ordinal variable[1] v_i, and $\phi_{id}(m)$ is the monotonically increasing function of m:

$$\phi_{id}(1) < \phi_{id}(2) < ... < \phi_{id}(M_i)$$

A shortcoming of this setting is that when $\boldsymbol{h} = \boldsymbol{0}$, the model reduces to the standard multiclass logistic, effectively removing the ordinal property. To deal with this, we propose to make the input bias parameters order dependent:

$$P(v_i = m \mid \boldsymbol{h}) \propto \exp\left\{\sum_{d=1}^{D} \phi_{id}(m) \left(U_{id} + \sum_{k=1}^{K} V_{idk} h_k\right)\right\} \tag{7}$$

where U_{id} is the newly introduced parameter. Here we choose $D = M_i$, and $\phi_{id}(m) = {}^{(m-d)}/_{(M_i-1)}$.

1. This should not be confused with the dimensionality of the whole data $\boldsymbol{v}$.

2.2.3. Category-ranking Variables

In category ranking, a variable assignment has the form of a ranked list of a set of categories. For example, from a set of offers namely games, sports, music, and photography, a person may express their preferences in a particular decreasing order: sports $\succ$ music $\succ$ games $\succ$ photography. Sometimes, they may like sports and music equally, creating a situation known as *ties* in ranking, or *indifference* in preference. When there are no ties, we can say that the rank is *complete*.

More formally, from a set of categories $\mathbb{S}_i = \{c_{i1}, c_{i2}, ..., c_{iM_i}\}$, a variable assignment without ties is then a permutation of elements of $\mathbb{S}_i$. Thus, there are $M_i!$ possible complete rank assignments. When we allow ties to occur, however, the number of possible assignments is extremely large. To see how, let us group categories of the same rank into a partition. Orders within a partition are not important, but orders between partitions are. Thus, the problem of rank assignment turns out to be choosing from a set of all possible schemes for partitioning and ordering a set. The number of such schemes is known in combinatorics as the *Fubini's number* (Mureşan, 2008, pp. 396–397), which is extremely large even for small sets. For example, Fubini $(1) = 1$, Fubini $(3) = 13$, Fubini $(5) = 541$ and Fubini $(10) = 102,247,563$. Directly modelling ranking with ties proves to be intractable.

We thus resort to approximate methods. One way is to model just pairwise comparisons: we treat each pair of categories separately *when conditioned on the hidden layer*. More formally, denote by $c_{il} \succ c_{im}$ the preference of category c_{il} over c_{im}, and by $c_{il} \simeq c_{im}$ the indifference. We replace the data model $P_i(v_i|\boldsymbol{h})$ with a product of pairwise comparisons $\prod_l \prod_{m>l} P_i(c_{il} \triangleright c_{im}|\boldsymbol{h})$, where $\triangleright$ denotes preference relations (i.e., $\succ$, $\prec$ or $\simeq$). This effectively translates the original problem with Fubini's number complexity to $M_i(M_i-1)/2$ pairwise sub-problems, each of which has only three preference choices. The drawback is that this relaxation loses the guarantee of *transitivity* (i.e., $c_{il} \succeq c_{im}$ and $c_{im} \succeq c_{in}$ would entail $c_{il} \succeq c_{in}$, where $\succeq$ means *better* or *equal-to*). The hope is that the hidden layer is rich enough to absorb this property, that is, the probability of preserving the transitivity is sufficiently high.

Now it remains to specify $P_i(c_{il} \triangleright c_{im}|\boldsymbol{h})$ in details. In particular, we adapt the Davidson's model (Davidson, 1970) of pairwise comparison:

$$\begin{aligned} P_i(c_{il} \succ c_{im}|\boldsymbol{h}) &= \frac{1}{Z_{iml}(\boldsymbol{h})}\varphi(c_{il}, \boldsymbol{h}) \\ P_i(c_{il} \simeq c_{im}|\boldsymbol{h}) &= \frac{1}{Z_{iml}(\boldsymbol{h})}\gamma\sqrt{\varphi(c_{il}, \boldsymbol{h})\varphi(c_{im}, \boldsymbol{h})} \\ P_i(c_{il} \prec c_{im}|\boldsymbol{h}) &= \frac{1}{Z_{iml}(\boldsymbol{h})}\varphi(c_{im}, \boldsymbol{h}) \end{aligned} \tag{8}$$

where $Z_{ilm}(\boldsymbol{h}) = \varphi(c_{il}, \boldsymbol{h}) + \varphi(c_{im}, \boldsymbol{h}) + \gamma\sqrt{\varphi(c_{il}, \boldsymbol{h})\varphi(c_{im}, \boldsymbol{h})}$, $\gamma > 0$ is the tie parameter, and

$$\varphi(c_{im}, \boldsymbol{h}) = \exp\{\frac{1}{M_i}(U_{im} + \sum_k V_{imk}h_k)\}.$$

The term $1/M_i$ normalises the occurrence frequency of a category in the model energy, leading to better numerical stability.

3. Learning and Inference

In this paper, we consider two applications of the MV.RBM: *estimating data distribution* and *learning predictive model*s. Estimating data distribution is to learn a generative model that generates the visible data. This can be useful in many other applications including dimensionality reduction, feature extraction, and data completion. On the other hand, a predictive model is a classification (or regression) tool that predicts an output given the input co-variates.

3.1. Parameter Learning

We now present parameter estimation for $\{w_k, U_i, U_{im}, V_{ik}, V_{imk}\}$, which clearly depend on the specific applications.

3.1.1. Estimating Data Distribution

The problem of estimating a distribution from data is typically performed by maximising the data likelihood $\mathcal{L}_1 = \sum_{\boldsymbol{v}} \tilde{P}(\boldsymbol{v}) \log P(\boldsymbol{v})$, where $\tilde{P}(\boldsymbol{v})$ denotes the empirical distribution of the visible variables, and $P(\boldsymbol{v}) = \sum_{\boldsymbol{h}} P(\boldsymbol{v}, \boldsymbol{h})$ is the model distribution. Since the MV.RBM belongs to the exponential family, the gradient of $\mathcal{L}_1$ with respect to parameters takes the form of difference of expectations. For example, in the case of binary variables, the gradient reads

$$\frac{\partial \mathcal{L}_1}{\partial V_{ik}} = \langle v_i h_k \rangle_{\tilde{P}(v_i, h_k)} - \langle v_i h_k \rangle_{P(v_i, h_k)}$$

where $\tilde{P}(h_k, v_i) = P(h_k|\boldsymbol{v})\tilde{P}(v_i)$ is the empirical distribution, and $P(h_k, v_i) = P(h_k|\boldsymbol{v})P(v_i)$ the model distribution. Due to space constraint, we omit the derivation details here.

The empirical expectation $\langle v_i h_k \rangle_{\tilde{P}(v_i, h_k)}$ is easy to estimate due to the factorisation in Eq. (3). However, the model expectation $\langle v_i h_k \rangle_{P(v_i, h_k)}$ is intractable to evaluate exactly, and thus we must resort to approximate methods. Due to the factorisations in Eqs. (3,4), Markov Chain Monte Carlo samplers are efficient to run. More specifically, the sampler is alternating between $\left\{\widehat{h}_k \sim P(h_k|\boldsymbol{v})\right\}_{k=1}^{K}$ and $\{\widehat{v}_i \sim P(v_i|\boldsymbol{h})\}_{i=1}^{N}$. Note that in the case of multicategorical variables, make use of the factorisation in Eq. (5) and sample $\{a_{im}\}_{m=1}^{M_i}$ simultaneously. On the other hand, in the case of category-ranked variables, we do not sample directly from $P(v_i|\boldsymbol{h})$ but from its relaxation $\{P_i(c_{il} \triangleright c_{im}|\boldsymbol{h})\}_{l,m>l}$ - which have the form of multinomial distributions. To speed up, we follow the method of Contrastive Divergence (CD) (Hinton, 2002), in which the MCMC is restarted from the observed data $\boldsymbol{v}$ and stopped after just a few steps for every parameter update. This has been known to introduce bias to the model estimate, but it is often fast and effective for many applications.

For the data completion application, in the data we observed only some variables and others are missing. There are two ways to handle a missing variable during training time: one is to treat it as hidden, and the other is to ignore it. In this paper, we follows the latter for simplicity and efficiency, especially when the data is highly sparse[2].

2. Ignoring missing data may be inadequate if the missing patterns are not at random. However, treating missing data as zero observations (e.g., in the case of binary variables) may not be accurate either since it may introduce bias to the data marginals.

3.1.2. Learning Predictive Models

In our MV.RBM, a predictive task can be represented by an output variable conditioned on input variables. Denote by v_i the i-th output variable, and $\boldsymbol{v}_{\neg i}$ the set of input variables, that is, $\boldsymbol{v} = (v_i, \boldsymbol{v}_{\neg i})$. The learning problem is translated into estimating the conditional distribution $P(v_i \mid \boldsymbol{v}_{\neg i})$.

There are three general ways to learn a predictive model. The *generative* method first learns the joint distribution $P(v_i, \boldsymbol{v}_{\neg i})$ as in the problem of estimating data distribution. The *discriminative* method, on the other hand, effectively ignores $P(\boldsymbol{v}_{\neg i})$ and concentrates only on $P(v_i \mid \boldsymbol{v}_{\neg i})$. In the latter, we typically maximise the conditional likelihood $\mathcal{L}_2 = \sum_{v_i} \sum_{\boldsymbol{v}_{\neg i}} \tilde{P}(v_i, \boldsymbol{v}_{\neg i}) \log P(\boldsymbol{v}_i \mid \boldsymbol{v}_{\neg i})$. This problem is inherently easier than the former because we do not have to make inference about $\boldsymbol{v}_{\neg i}$. The learning strategy is almost identical to that of the generative counterpart, except that we *clamp* the input variables $\boldsymbol{v}_{\neg i}$ to their observed values. For tasks whose size of the output space is small (e.g., standard binary, ordinal, categorical variables) we can perform exact evaluations and use any non-linear optimisation methods for parameter estimation. The conditional distribution $P(v_i \mid \boldsymbol{v}_{\neg i})$ can be computed as in Eq. (11). We omit the likelihood gradient here for space limitation.

It is often argued that the discriminative method is more preferable since there is no waste of effort in learning $P(\boldsymbol{v}_{\neg i})$, which we do not need at test time. In our setting, however, learning $P(\boldsymbol{v}_{\neg i})$ may yield a more faithful representation[3] of the data through the posterior $P(\boldsymbol{h} \mid \boldsymbol{v}_{\neg i})$. This suggests a third, *hybrid* method: combining the generative and discriminative objectives. One way is to optimise a hybrid likelihood:

$$\mathcal{L}_3 = \lambda \sum_{\boldsymbol{v}_{\neg i}} \tilde{P}(\boldsymbol{v}_{\neg i}) \log P(\boldsymbol{v}_{\neg i}) + (1 - \lambda) \sum_{\boldsymbol{v}_i} \sum_{\boldsymbol{v}_{\neg i}} \tilde{P}(\boldsymbol{v}_i, \boldsymbol{v}_{\neg i}) \log P(\boldsymbol{v}_i \mid \boldsymbol{v}_{\neg i}),$$

where $\lambda \in (0, 1)$ is the hyper-parameter controlling the relative contribution of generative and discriminative components. Another way is to use a 2-stage procedure: first we *pre-train* the model $P(\boldsymbol{v}_{\neg i})$ in an unsupervised manner, and then *fine-tune* the predictive model[4] $P(\boldsymbol{v}_i \mid \boldsymbol{v}_{\neg i})$.

3.2. Prediction

Once the model has been learnt, we are ready to perform prediction. We study two predictive applications: completing missing data, and output labels in predictive modelling. The former leads to the inference of $P(\boldsymbol{v}_C \mid \boldsymbol{v}_{\neg C})$, where $\boldsymbol{v}_{\neg C}$ is the set of observed variables, and $\boldsymbol{v}_C$ is the set of unseen variables to be predicted. Ideally, we should predict all unseen variables simultaneously but the inference is likely to be difficult. Thus, we resort to estimating $P(v_i|\boldsymbol{v}_{\neg C})$, for $i \in C$. The prediction application requires the estimation of $P(v_i|\boldsymbol{v}_{\neg i})$, which is clearly a special case of $P(v_i|\boldsymbol{v}_{\neg C})$, i.e., when $C = \{i\}$. The output is

3. As we do not need labels to learn $P(\boldsymbol{v}_{\neg i})$, this is actually a form of *semi-supervised learning*.
4. We can also avoid tuning parameters associated with $\boldsymbol{v}_{\neg i}$ by using the posteriors as features and learn $P\left(v_i \mid \hat{\boldsymbol{h}}\right)$, where $\hat{h}_k = P\left(h_k^1 \mid \boldsymbol{v}_{\neg i}\right)$.

predicted as follows

$$\hat{v}_i = \arg\max_{v_i} P(v_i|\boldsymbol{v}_{\neg C}) = \arg\max_{v_i} \sum_{\boldsymbol{h}} P(v_i, \boldsymbol{h}|\boldsymbol{v}_{\neg C}); \quad \text{where} \tag{9}$$

$$P(v_i, \boldsymbol{h}|\boldsymbol{v}_{\neg C}) = \frac{1}{Z(\boldsymbol{v}_{\neg C}, \boldsymbol{h})} \exp\left\{ G_i(v_i) + \sum_k w_k h_k + \sum_{j\in\{\neg C, i\}} \sum_k H_{jk}(v_j) h_k \right\}, \tag{10}$$

where $Z(\boldsymbol{v}_{\neg C})$ is the normalising constant. Noting that $h_k \in \{0, 1\}$, the computation of $P(v_i|\boldsymbol{v}_{\neg C})$ can be simplified as

$$P(v_i|\boldsymbol{v}_{\neg C}) = \frac{1}{Z(\boldsymbol{v}_{\neg C})} \exp\{G_i(v_i)\} \prod_k \left[1 + \frac{\exp\{H_{ik}(v_i)\}}{1/P(h_k^1|\boldsymbol{v}_{\neg C}) - 1} \right] \tag{11}$$

where $P(h_k^1|\boldsymbol{v}_{\neg C})$ is computed using Eq. (3) as

$$P(h_k^1|\boldsymbol{v}_{\neg C}) = \frac{1}{1 + \exp\{-w_k - \sum_{j\in\neg C} H_{jk}(v_j)\}}.$$

For the cases of binary, categorical and ordinal outputs, the estimation in Eq. (9) is straightforward using Eq. (11). However, for other output types, suitable simplification must be made:

- For multicategorical and category-ranking variables, we do not enumerate over all possible assignments of v_i, but rather in an indirect manner:
  - For multiple categories (Section 2.2.1), we first estimate $\{P_i(a_{im} = 1|\boldsymbol{v}_{\neg i})\}_{m=1}^{M_i}$ and then output $a_{im} = 1$ if $P_i(a_{im} = 1|\boldsymbol{v}_{\neg i}) \geq \nu$ for some threshold[5] $\nu \in (0, 1)$.
  - For category-ranking (Section 2.2.3), we first estimate $\{P_i(c_{il} \succ c_{im}|\boldsymbol{v}_{\neg i})\}_{l,m>l}$. The complete ranking over the set $\{c_{i1}, c_{i2}, ..., c_{iM_i}\}$ can be obtained by aggregating over probability pairwise relations. For example, the score for c_{im} can be estimated as $s(c_{im}) = \sum_{l\neq m} P_i(c_{im} \succ c_{il}|\boldsymbol{v}_{\neg i})$, which can be used for sorting categories[6].

- For continuous variables, the problem leads to a non-trivial nonlinear optimisation: even for the case of Gaussian variables, $P(v_i|\boldsymbol{v}_{\neg C})$ in Eq. (11) is no longer Gaussian. For efficiency and simplicity, we can take a *mean-field* approximation by substituting $\hat{h}_k = P(h_k^1|\boldsymbol{v}_{\neg C})$ for h_k. For example, in the case of Gaussian outputs, we then obtain a simplified expression for $P(v_i|\boldsymbol{v}_{\neg C})$:

$$P(v_i|\boldsymbol{v}_{\neg C}) \propto \exp\left\{ -\frac{v_i^2}{2\sigma_i^2} + U_i v_i + \sum_k V_{ik} v_i \hat{h}_k \right\},$$

  which is also a Gaussian. Thus the optimal value is the mean itself: $\hat{v}_i = \sigma_i^2 \left(U_i + \sum_k V_{ik} \hat{h}_k \right)$. Details of the mean-field approximation is presented in Appendix A.2.

5. Raising the threshold typically leads to better precision at the expense of recall. Typically we choose $\nu = 0.5$ when there is no preference over recall nor precision.
6. Note that we do not estimate the event of ties during prediction.

4. A Case Study: World Attitudes

4.1. Setting

In this experiment, we run the MV.RBM on a large-scale survey of the general world opinion, which was published by the Pew Global Attitudes Project[7] in the summer of 2002. The survey was based on interviewing with people in 44 countries in the period of 2001–2002. Some sample questions are listed in Appendix A.1. After some pre-processing, we obtain a dataset of 38,263 people, each of whom provides answers to a subset of 189 questions over multiple topics ranging from globalization, democracy to terrorism. Many answers are deliberately left empty because it may be inappropriate to ask certain type of questions in a certain area or ethnic group. Of all answers, 43 are binary, 12 are categorical, 3 are multicategorical, 125 are ordinal, 2 are category-ranking, and 3 are continuous. To suppress the scale difference in continuous responses, we normalise them to zeros means and unit variances [8].

We evaluate each data type separately. In particular, let u be the user index, $\hat{v}_i$ be the predicted value of the i-th variable, and N_t is the number of variables of type t in the test data, we compute the prediction errors as follows:

$$\begin{aligned}
&\text{–Binary} && : \frac{1}{N_{bin}} \sum_u \sum_i \mathbb{I}\left[v_i^{(u)} \neq \hat{v}_i^{(u)}\right],\\
&\text{–Categorical} && : \frac{1}{N_{cat}} \sum_u \sum_i \mathbb{I}\left[v_i^{(u)} \neq \hat{v}_i^{(u)}\right],\\
&\text{–Multicategorical} && : 1 - {}^{2\mathrm{RP}}/_{(\mathrm{R}+\mathrm{P})},\\
&\text{–Continuous} && : \sqrt{\frac{1}{D_{cont}} \sum_u \sum_i \left(v_i^{(u)} - \hat{v}_i^{(u)}\right)^2},\\
&\text{–Ordinal} && : \frac{1}{N_{ord}} \sum_u \sum_i \frac{1}{M_i - 1} \left|v_i^{(u)} - \hat{v}_i^{(u)}\right|,\\
&\text{–Category-ranking} && : \frac{1}{D_{rank}} \sum_u \sum_i \frac{2}{M_i(M_i-1)} \sum_{l,m>l} \mathbb{I}\left[(\pi_{il}^{(u)} - \pi_{im}^{(u)})(\hat{\pi}_{il}^{(u)} - \hat{\pi}_{im}^{(u)}) < 0\right],
\end{aligned}$$

where $\mathbb{I}[\cdot]$ is the identity function, $\pi_{im} \in \{1, 2, ..., M_i\}$ is the rank of the m-th category of the i-th variable, R is the recall rate and P is the precision. The recall and precision are defined as:

$$\mathrm{R} = \frac{\sum_u \sum_i \frac{1}{M_i} \sum_{m=1}^{M_i} \mathbb{I}\left[a_{im}^{(u)} = \hat{a}_{im}^{(u)}\right]}{\sum_u \sum_i \frac{1}{M_i} \sum_{m=1}^{M_i} a_{im}^{(u)}}; \qquad \mathrm{P} = \frac{\sum_u \sum_i \frac{1}{M_i} \sum_{m=1}^{M_i} \mathbb{I}\left[a_{im}^{(u)} = \hat{a}_{im}^{(u)}\right]}{\sum_u \sum_i \frac{1}{M_i} \sum_{m=1}^{M_i} \hat{a}_{im}^{(u)}},$$

where $a_{im} \in \{0, 1\}$ is the m-th component of the i-th multicategorical variable. Note that the summation over i for each type only consists of relevant variables.

To create baselines, we use the MV.RBM without the hidden layer, i.e., by assuming that variables are independent[9].

7. http://pewglobal.org/datasets/
8. It may be desirable to learn the variance structure, but we keep it simple by fixing to unit variance. For more sophisticated variance learning, we refer to a recent paper (Le Roux et al., 2011) for more details.
9. To the best of our knowledge, there has been no totally comparable work addressing the issues we study in this paper. Existing survey analysis methods are suitable for individual tasks such as measuring

	Baseline	$K = 20$	$K = 50$	$K = 100$	$K = 200$	$K = 500$
Binary	32.9	23.6	20.1	16.3	13.2	9.8
Categorical	52.3	29.8	22.0	17.0	13.2	7.1
Multicategorical	49.6	46.6	42.2	36.9	29.2	23.8
Continuous(*)	100.0	89.3	84.1	78.4	69.5	65.5
Ordinal	25.2	19.5	16.2	13.5	10.9	7.7
Category ranking	19.3	11.7	6.0	5.0	3.2	2.3

Table 2: Error rates (%) when reconstructing data from posteriors. The baseline is essentially the MV.RBM without hidden layer (i.e., assuming variables are independent). (*) The continuous variables have been normalised to account for different scales between items, thus the baseline error will be 1 (i.e., the unit variance).

4.2. Results

4.2.1. Feature Extraction and Visualisation

Recall that our MV.RBM can be used as a feature extraction tool through the posterior projection. The projection converts a multimodal input into a real-valued vector of the form $\hat{\boldsymbol{h}} = \left(\hat{h}_1, \hat{h}_2, ..., \hat{h}_K\right)$, where $\hat{h}_k = P\left(h_k = 1 \mid \boldsymbol{v}\right)$. Clearly, numerical vectors are much easier to process further than the original data, and in fact the vectorial form is required for the majority of modern data handling tools (e.g., for transformation, clustering, comparison and visualisation). To evaluate the faithfulness of the new representation, we reconstruct the original data using $\hat{v}_i = \arg\max_{v_i} P\left(v_i \mid \hat{\boldsymbol{h}}\right)$, that is, in Eq. (4), the binary vector $\boldsymbol{h}$ is replaced by $\hat{\boldsymbol{h}}$. The use of $P\left(v_i \mid \hat{\boldsymbol{h}}\right)$ can be reasoned through the mean-field approximation framework presented in Appendix A.2. Table 2 presents the reconstruction results. The trends are not surprising: with more hidden units, the model becomes more flexible and accurate in capturing the data content.

For visualisation, we first learn our MV.RBM (with $K = 50$ hidden units) using randomly chosen 3,830 users, with the country information removed. Then we use the t-SNE (van der Maaten and Hinton, 2008) to project the posteriors further into 2D. Figure 1 shows the distribution of people's opinions in 10 countries (Angola, Argentina, Bangladesh, Bolivia, Brazil, Bulgaria, Canada, China, Czech Republic, and Egypt). It is interesting to see how opinions cluster geographically and culturally: Europe & North America (Bulgaria, Canada & Czech Republic), South America (Argentina, Bolivia, Brazil), East Asia (China), South Asia (Bangladesh), North Africa (Egypt) and South Africa (Angola).

4.2.2. Data Completion

In this task, we need to fill missing answers for each survey response. Missing answers are common in real survey data because the respondents may forget to answer or simply ignore the questions. We create an evaluation test by randomly removing a portion $\rho \in (0, 1)$

pairwise correlation among variables, or building individual regression models where complex co-variates are coded into binary variables.

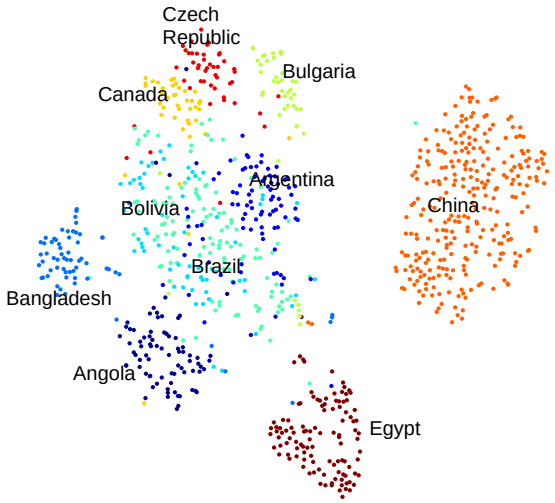

Figure 1: t-SNE projection of posteriors ($K = 50$) with country information removed. Each point is a person from one of the 10 countries: Angola, Argentina, Bangladesh, Bolivia, Brazil, Bulgaria, Canada, China, Czech Republic, and Egypt. Each colour represents a country. Best viewed in colour.

	Baseline	$K = 20$	$K = 50$	$K = 100$	$K = 200$	$K = 500$
Binary	32.7	26.0	24.2	23.3	22.7	22.3
Categorical	52.1	34.3	30.0	28.2	27.5	27.1
Multicategorical	49.5	48.3	45.7	43.6	42.4	42.0
Continuous(*)	101.6	93.5	89.9	87.9	87.3	87.9
Ordinal	25.1	20.7	19.3	18.6	18.2	17.9
Category ranking	19.3	15.4	14.7	14.2	14.1	13.9

Table 3: Completion error rates (%) $\rho = 0.2$ answers missing at random. (*) See Table 2.

of answers for each person. The MV.RBM is then trained on the remaining answers in a generative fashion (Section 3.1.1). Missing answers are then predicted as in Section 3.2. The idea here is that missing answers of a person can be interpolated from available answers by other persons. This is essentially a multimodal generalisation of the so-called collaborative filtering problem. Table 3 reports the completion results for a subset of the data.

4.2.3. Learning Predictive Models

We study six predictive problems, each of which is representative for a data type. This means six corresponding variables are reserved as outputs and the rest as input co-variates. The predictive problems are: (i) satisfaction with the country (*binary*), (ii) country of origin (*categorical*, of size 44), (iii) problems facing the country (*multicategorical*, of size 11), (iv)

	Baseline	$K = 3$	$K = 5$	$K = 10$	$K = 15$	$K = 20$	$K = 50$
Satisfaction (*bin.*)	26.3	18.0	17.7	17.7	17.8	18.0	18.0
Country (*cat.*)	92.0	70.2	61.0	21.6	11.0	9.9	5.9
Probs. (*multicat.*)	49.6	47.6	41.9	39.2	38.8	39.1	39.2
Age (*cont.**)	99.8	67.3	67.6	66.3	66.4	65.8	66.3
Life ladder (*ord.*)	16.9	12.2	12.2	11.9	11.9	12.2	11.8
Dangers (*cat.-rank*)	31.2	27.1	24.6	24.0	23.2	23.0	22.5

Table 4: Predictive error rates (%) with 80/20 train/test split. (*) See Table 2.

age of the person (*continuous*), (v) ladder of life (*ordinal*, of size 11), and (vi) rank of dangers of the world (*category-ranking*, of size 5). All models are trained discriminatively (see Section 3.1.2). We randomly split the users into a training subset and a testing subset. The predictive results are presented in Table 4. It can be seen that learning predictive models requires far less number of hidden units than the tasks of reconstruction and completion. This is because in discriminative training, the hidden layer acts as an information filter that allows relevant amount of bits passing from the input to the output. Since there is only one output per prediction task, the number of required bits, therefore number of hidden units, is relatively small. In reconstruction and completion, on the other hand, we need many bits to represent all the available information.

5. Related Work

The most popular use of RBMs is in modelling of individual types, for example, binary variables (Freund and Haussler, 1993), Gaussian variables (Hinton and Salakhutdinov, 2006; Ranzato and Hinton, 2010), categorical variables (Salakhutdinov et al., 2007), rectifier linear units (Nair and Hinton, 2010), Poisson variables (Gehler et al., 2006), counts (Salakhutdinov and Hinton, 2009b) and Beta variables (Le Roux et al., 2011). When RBMs are used for classification (Larochelle and Bengio, 2008), categorical variables might be employed for labeling in additional to the features. Other than that, there has been a model called Dual-Wing RBM for modelling both continuous and binary variables (Xing et al., 2005). However, there have been no attempts to address all *six* data types in a single model as we do in the present paper.

The literature on ordinal variables is sufficiently rich in statistics, especially after the seminal work of (McCullagh, 1980). In machine learning, on the other hand, the literature is quite sparse and recent (e.g. see (Shashua and Levin, 2002; Yu et al., 2006)) and it is often limitted to single ordinal output (given numerical input co-variates). An RBM-based modelling of ordinal variables addressed in (Truyen et al., 2009) is similar to ours, except that our treatment is more general and principled.

Mixed-variate modelling has been previously studied in statistics, under a variety of names such as *mixed outcomes*, *mixed data*, or *mixed responses* (Sammel et al., 1997; Dunson, 2000; Shi and Lee, 2000; McCulloch, 2008). Most papers focus on the mix of ordinal, Gaussian and binary variables under the *latent variable* framework. More specifically, each observed variable is assumed to be generated from one or more underlying continuous latent

variables. Inference becomes complicated since we need to integrate out these correlated latent variables, making it difficult to handle hundreds of variables and large-scale datasets.

In machine learning, the problem of predicting a single multicategorical variable is also known as multilabel learning (e.g., see (Tsoumakas and Katakis, 2007)). Previous ideas that we have adapted into our context including the shared structure among labels (Ji et al., 2008). In our model, the sharing is captured by the hidden layer in a probabilistic manner and we consider many multicategorical variables at the same time. Finally, the problem of predicting a single category-ranked variable is also known as label-ranking (e.g., see (Dekel et al., 2003; Hüllermeier et al., 2008)). The idea we adopt is the pairwise comparison between categories. However, the previous work neither considered the hidden correlation between those pairs nor attempted multiple category-ranked variables.

6. Conclusion

We have introduced Mixed-Variate Restricted Boltzmann Machines (MV.RBM) as a generalisation of the RBMs for modelling correlated variables of multiple modalities and types. Six types considered were: binary, categorical, multicategorical, continuous information, ordinal, and category-ranking. We shown that the MV.RBM is capable of handling a variety of machine learning tasks including feature exaction, dimensionality reduction, data completion, and label prediction. We demonstrated the capacity of the model on a large-scale world-wide survey.

We plan to further the present work in several directions. First, the model has the capacity to handle multiple related predictive models simultaneously by learning a shared representation through hidden posteriors, thereby applicable to the setting of multitask learning. Second, there may exist strong interactions between variables which the RBM architecture may not be able to capture. The theoretical question is then how to model inter-type dependencies directly without going through an intermediate hidden layer. Finally, we plan to enrich the range of applications of the proposed model.

Acknowledgment: We thank anonymous reviewers for insightful comments.

References

J.A. Anderson. Regression and ordered categorical variables. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–30, 1984.

R.R. Davidson. On extending the Bradley-Terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65(329):317–328, 1970.

O. Dekel, C. Manning, and Y. Singer. Log-linear models for label ranking. *Advances in Neural Information Processing Systems*, 16, 2003.

D.B. Dunson. Bayesian latent variable models for clustered mixed outcomes. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, pages 355–366, 2000.

Y. Freund and D. Haussler. Unsupervised learning of distributions on binary vectors using two layer networks. *Advances in Neural Information Processing Systems*, pages 912–919, 1993.

P.V. Gehler, A.D. Holub, and M. Welling. The rate adapting Poisson model for information retrieval and object recognition. In *Proceedings of the 23rd international conference on Machine learning*, pages 337–344. ACM New York, NY, USA, 2006.

G.E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:1771–1800, 2002.

G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

G.E. Hinton and T.J. Sejnowski. Learning and relearning in Boltzmann machines. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1:282–317, 1986.

E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 2008.

S. Ji, L. Tang, S. Yu, and J. Ye. Extracting shared subspace for multi-label classification. In *KDD*. ACM New York, NY, USA, 2008.

H. Larochelle and Y. Bengio. Classification using discriminative restricted Boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, pages 536–543. ACM, 2008.

N. Le Roux, N. Heess, J. Shotton, and J. Winn. Learning a generative model of images by factoring appearance and shape. *Neural Computation*, 23(3):593–650, 2011.

P. McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 109–142, 1980.

C. McCulloch. Joint modelling of mixed outcome types using latent variables. *Statistical Methods in Medical Research*, 17(1):53, 2008.

M. Mureşan. *A concrete approach to classical analysis*. Springer Verlag, 2008.

V. Nair and G.E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proc. 27th International Conference on Machine Learning*, 2010.

M.A. Ranzato and G.E. Hinton. Modeling pixel means and covariances using factorized third-order Boltzmann machines. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2551–2558. IEEE, 2010.

R. Salakhutdinov and G. Hinton. Deep Boltzmann machines. In *Proceedings of The Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS'09*, volume 5, pages 448–455, 2009a.

R. Salakhutdinov and G. Hinton. Replicated softmax: an undirected topic model. *Advances in Neural Information Processing Systems*, 22, 2009b.

R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 791–798, 2007.

M.D. Sammel, L.M. Ryan, and J.M. Legler. Latent variable models for mixed discrete and continuous outcomes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(3):667–678, 1997.

A. Shashua and A. Levin. Ranking with large margin principle: Two approaches. *Advances in Neural Information Processing Systems*, 15, 2002.

J.Q. Shi and S.Y. Lee. Latent variable models with mixed continuous and polytomous data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(1):77–87, 2000.

T.T. Truyen, D.Q. Phung, and S. Venkatesh. Ordinal Boltzmann machines for collaborative filtering. In *Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, Montreal, Canada, June 2009.

G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

E. Xing, R. Yan, and A.G. Hauptmann. Mining associated text and images with dual-wing harmoniums. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI-2005)*. Citeseer, 2005.

S. Yu, K. Yu, V. Tresp, and H.P. Kriegel. Collaborative ordinal regression. In *Proceedings of the 23rd international conference on Machine learning*, page 1096. ACM, 2006.

Appendix A. Additional Materials

A.1. Sample Questions

- **Q1** (*Ordinal*): How would you describe your day today—has it been a typical day, a particularly good day, or a particularly bad day?
- **Q7** (*Binary*): Now thinking about our country, overall, are you satisfied or dissatisfied with the way things are going in our country today?
- **Q5** (*Multicategorical*): What do you think is the most important problem facing you and your family today? {Economic problems / Housing / Health / Children and education/Work/Social relations / Transportation / Problems with government / Crime / Terrorism and war / No problems / Other / Don't know / Refused}

- **Q10,11** (*Category-ranking*): In your opinion, which one of these poses the greatest/second greatest threat to the world: {the spread of nuclear weapons / religious and ethnic hatred/AIDS and other infectious diseases / pollution and other environmental problems / or the growing gap between the rich and poor}?
- **Q74** (*Continuous*): How old were you at your last birthday?
- **Q91** (*Categorical*): Are you currently married or living with a partner, widowed, divorced, separated, or have you never been married?

A.2. Mean-field Approximation

We present here a simplification of $P(v_i \mid \boldsymbol{v}_{\neg C})$ in Eq. (11) using the mean-field approximation. Recall that $P(v_i \mid \boldsymbol{v}_{\neg C}) = \sum_{\boldsymbol{h}} P(v_i, \boldsymbol{h} \mid \boldsymbol{v}_{\neg C})$, where $P(v_i, \boldsymbol{h} \mid \boldsymbol{v}_{\neg C})$ is defined in Eq. (10). We approximate $P(v_i, \boldsymbol{h} \mid \boldsymbol{v}_{\neg C})$ by a fully factorised distribution

$$Q(v_i, \boldsymbol{h} \mid \boldsymbol{v}_{\neg C}) = Q(v_i \mid \boldsymbol{v}_{\neg C}) \prod_k Q(h_k \mid \boldsymbol{v}_{\neg C}).$$

The approximate distribution $Q(v_i, \boldsymbol{h} \mid \boldsymbol{v}_{\neg C})$ is obtained by minimising the Kullback-Leibler divergence

$$\mathcal{D}_{KL}(Q(v_i, \boldsymbol{h} \mid \boldsymbol{v}_{\neg C}) \parallel P(v_i, \boldsymbol{h} \mid \boldsymbol{v}_{\neg C})) = \sum_{v_i} \sum_{\boldsymbol{h}} Q(v_i, \boldsymbol{h} \mid \boldsymbol{v}_{\neg C}) \log \frac{Q(v_i, \boldsymbol{h} \mid \boldsymbol{v}_{\neg C})}{P(v_i, \boldsymbol{h} \mid \boldsymbol{v}_{\neg C})}$$

with respect to $Q(v_i \mid \boldsymbol{v}_{\neg C})$ and $\{Q(h_k \mid \boldsymbol{v}_{\neg C})\}_{k=1}^K$. This results in the following recursive relations:

$$\begin{aligned} Q(v_i \mid \boldsymbol{v}_{\neg C}) &\propto \exp\left\{ G_i(v_i) + \sum_k H_{ik}(v_i) Q(h_k \mid \boldsymbol{v}_{\neg C}) \right\}, \\ Q(h_k \mid \boldsymbol{v}_{\neg C}) &= \frac{1}{1 + \exp\{-w_k - \sum_{v_i} H_{ik}(v_i) Q(v_i \mid \boldsymbol{v}_{\neg C}) - \sum_{j \in \neg C} H_{ik}(v_j)\}}. \end{aligned}$$

Now we make a further assumption that $\left|\sum_{v_i} H_{ik}(v_i) Q(v_i \mid \boldsymbol{v}_{\neg C})\right| \ll \left|\sum_{j \in \neg C} H_{ik}(v_j)\right|$, e.g., when the set $\neg C$ is sufficiently large. This results in $Q(h_k \mid \boldsymbol{v}_{\neg C}) \approx P(h_k \mid \boldsymbol{v}_{\neg C})$ and

$$Q(v_i \mid \boldsymbol{v}_{\neg C}) \propto \exp\left\{ G_i(v_i) + \sum_k H_{ik}(v_i) P\left(h_k^1 \mid \boldsymbol{v}_{\neg C}\right) \right\},$$

which is essentially the data model $P(v_i \mid \boldsymbol{h})$ in Eq. (4) with h_k being replaced by $P\left(h_k^1 \mid \boldsymbol{v}_{\neg C}\right)$.

The overall complexity of computing $Q(v_i \mid \boldsymbol{v}_{\neg C})$ is the same as that of evaluating $P(v_i \mid \boldsymbol{v}_{\neg C})$ in Eq. (11). However, the approximation is often numerically faster, and in the case of continuous variables, it has the simpler functional form.