Robust Multi-Output Learning with Highly Incomplete Data via Restricted Boltzmann Machines

Giancarlo Fissore*, **Aurélien Decelle***, **Cyril Furtlehner**†
*Université Paris-Saclay, Laboratoire de recherche en informatique, 91405, Orsay, France.
†Inria, Inria Saclay-Île-de-France, 91120, Palaiseau, France.
Yufei Han - Nortonlifelock Research Group, yfhan.hust@gmail.com

Abstract

In this work we tackle the challenging problem of multi-output classification with partially observed features and labels. We show that a simple Restricted Boltzmann Machine can be trained with an adapted algorithm based on mean-field equations to efficiently solve problems of inductive and transductive learning in which both features and labels are missing at random. The effectiveness of the approach is demonstrated empirically on various datasets, with particular focus on a real-world Internet-of-Things security dataset.

1 Introduction

Modern machine learning models usually require large sets of fully observed data to be trained, which are often not available in real-world applications. Often a random subset of features is absent (e.g. failed sensors of a monitoring system) and the data are insufficiently annotated (limitation due to human annotation) meaning that a lot of labels are missing. Consequently, it is highly important for machine learning systems to tolerate co-occurrence of missing features and partially observed labels for robust learning in practical use.

We study both **transductive** and **inductive** multi-output classification. Multi-output classification, including **multi-class** and **multi-label** learning tasks, outputs class labels with higher dimensional representation. They thus define a more sophisticated learning scenario compared to binary classification. The former associates an input instance to one class of a finitely defined class set, while the latter allows one instance to be associated to multiple labels simultaneously. In the transductive case, the learning objective is to infer the missing features and labels based on the observed ones (no separate test set is required). In the inductive case, the model is trained over a set of **incomplete training instances** and the classification is performed by inferring the labels associated to **incomplete test instances**. In our work, we consider that both features and labels are missing completely at random, which means that the mask of missing information is assumed to be statistically independent on the data distribution.

State-of-the-art accuracies for classification with incomplete or noise-corrupted features and partially observed labels are given by **CLE** [HSSZ18], **NoisyIMC** [CHS15] and **MC-1** [ClTCB15]. In particular, **CLE** and **NoisyIMC** are able to conduct both transductive and inductive learning for multi-label classification, while **MC-1** is a transductive-only method. **NoisyIMC** and **MC-1** can work in the semi-supervised learning scenario. Notably, in [LZG15] a method similar to ours is proposed to deal with incomplete labels. However, none of the

In: Rudolph S., Marreiros G. (eds.): Proceedings of STAIRS, Santiago de Compostela, Spain, august 2020, published at http://ceur-ws.org

four methods above can handle all the challenges co-currently raised in our work. First, all of these assume the testing instances to have fully observed feature profiles. They don't consider coping with incomplete testing instances by design. Second, all of them are designed specifically for multi-label learning and adapting them to multi-class classification is not straightforward.

More recently, methods based on Deep Latent Variable Models (DLVM) have been proposed to deal with missing data. In [MF19], the Variational Autoencoder [KW14] has been adapted to be trained with missing data and a sampling algorithm for data imputation is proposed. Other approaches based on Generative Adversarial Networks (GAN) by [GPAM$^+$14] are proposed in [YJvdS18] and [LJM19]. Impressive results on image datasets are displayed for these models, at the price of a rather high model complexity and the need for a large training set. In addition these works are focused on features reconstruction, and additional specifications and fine-tuning are required to be able to take partially observed labels into account. The models specifications are quite involved and any new specificity of the dataset may increase both the cost and the difficulty in training (especially for the approaches based on GANs).

In this paper we choose to address this problem in a more economical and robust manner. We consider the old and simple architecture of the Restricted Boltzmann Machine and adapt it to the multi-output learning context (RBM-MO) with missing data. The **RBM-MO** method serves as a generative model which collaboratively learns the marginal distribution of features and label assignments of input data instances, despite the incomplete observations. Building on the ideas expressed in [NK94, GJ94] we adapt the approach to the more effective contrastive divergence training procedure [Hin02] and provide results on various real-world datasets. The advantage of the RBM-MO model is that of providing a robust and flexible method to deal with missing data, with little additional complexity with respect to the classic RBM. Indeed, the trained model can be naturally applied to both transductive and inductive scenarios, achieving superior multi-output classification performance then state-of-the-art baselines. Moreover, it works seamlessly with multi-class and multi-label tasks, providing a unified framework for multi-output learning.

2 Overview of Restricted Boltzmann Machines

An RBM is a Markov random field with pairwise interactions defined on a bipartite graph formed by two layers of non-interacting variables: the visible nodes represent instances of the input data while the hidden nodes provide a latent representation of the data instances. $\mathcal{V}$ and $\mathcal{H}$ will denote respectively the sets of visible and hidden variables. In our setting, the visible variables will further split into two subsets $\mathcal{V}_f$ and $\mathcal{V}_\ell$ corresponding respectively to features and labels, such that $\mathcal{V} = \mathcal{V}_f + \mathcal{V}_\ell$. The visible variables form an explicit representation of the data and are noted $\boldsymbol{v} = \{v_i, i \in \mathcal{V}\}$. The hidden nodes $\boldsymbol{h} = \{h_j, j \in \mathcal{H}\}$ serve to approximate the underlying dependencies among the visible units.

In this paper, we will work with binary hidden nodes $h_j \in \{0, 1\}$. The variables corresponding to the visible features will be either real with a Gaussian prior or binary, depending on the data to model, and labels variables will always be binary ($v_i \in \{0, 1\}$). The joint probability distribution over the nodes is defined through an energy function

$$P(\boldsymbol{v}, \boldsymbol{h}) = \frac{e^{-E(\boldsymbol{v},\boldsymbol{h})}}{Z} p_{\text{prior}}(\boldsymbol{v}), \quad E(\boldsymbol{v}, \boldsymbol{h}) = -\sum_{i\in\mathcal{V}, j\in\mathcal{H}} v_i w_{ij} h_j - \sum_{i\in\mathcal{V}} a_i v_i - \sum_{j\in\mathcal{H}} b_j h_j \tag{1}$$

where a_i and b_j are biases acting respectively on the visible and hidden units and w_{ij} is the weight matrix that couples visible and hidden nodes. p_{prior} is in product form and encodes the nature of each visible variable, either with a Gaussian prior $p_{\text{prior}} = \mathcal{N}(0, \sigma_v^2)$ or a binary prior $p_{\text{prior}}(v) = \delta(s^2 - s)$. $Z = \sum_{\boldsymbol{v},\boldsymbol{h}} p_{\text{prior}}(\boldsymbol{v}) e^{-E(\boldsymbol{v},\boldsymbol{h})}$ is the partition function. The classical training method consists in maximizing the marginal likelihood over the visible nodes $P(\boldsymbol{v}) = \sum_{\boldsymbol{h}} P(\boldsymbol{v}, \boldsymbol{h})$ by tuning the RBM parameters $\theta = \{w_{ij}, a_i, b_j\}$ via gradient ascent of the log likelihood $\mathcal{L}(\boldsymbol{v}; \theta)$.

The tractability of the method relies heavily on the fact that the conditional probabilities $P(\boldsymbol{v}|\boldsymbol{h})$ and $P(\boldsymbol{h}|\boldsymbol{v})$ are given in closed forms. In our case these read:

$$P(\boldsymbol{v}|\boldsymbol{h}) = \prod_{i\in\mathcal{V}_f} \frac{e^{\sum_{j\in\mathcal{H}} v_i w_{ij} h_j + a_i v_i} p_{\text{prior}}(v_i)}{\sum_{v_i} e^{\sum_{j\in\mathcal{H}} v_i w_{ij} h_j + a_i v_i} p_{\text{prior}}(v_i)} \prod_{i\in\mathcal{V}_\ell} \sigma\Big(\sum_{j\in\mathcal{H}} v_i w_{ij} h_j + a_i v_i\Big), \tag{2}$$

$$P(\boldsymbol{h}|\boldsymbol{v}) = \prod_{j\in\mathcal{H}} \sigma\Big(\sum_{i\in\mathcal{V}} v_i w_{ij} + b_j\Big), \tag{3}$$

where $\sigma(x) = 1/(1+e^{-x})$ is the logistic function. The gradient of the likelihood w.r.t. the weights (and similarly w.r.t. the fields a_i and b_j) is given by

$$\frac{\partial \mathcal{L}(\boldsymbol{v};\theta)}{\partial w_{ij}} = \langle v_i h_j p(h_j|\boldsymbol{v})\rangle_{\text{data}} - \langle v_i h_j\rangle_{\text{RBM}} \tag{4}$$

where the brackets $\langle\rangle_{\text{data}}$ and $\langle\rangle_{\text{RBM}}$ respectively indicate the average over the data and over the distribution (1). The positive term is directly linked to the data and can be estimated exactly with (3), while the negative term is intractable. Many strategies are used to compute this last term: the *contrastive divergence* (CD) approach [Hin02] consists in estimating the term over a finite number of Gibbs sampling steps, starting from a data point and making alternate use of (2) and (3); in its *persistent* version (PCD) the chain is maintained over subsequent mini-batches; using mean-field approximation [MTK15] the term is computed by means of a low-couplings expansion.

3 Learning RBM with incomplete data

The RBM is a generative model able to learn the joint distribution of some empirical data given as input. As such, it is intrinsically able to encode the relevant statistical properties found in the training data instances that relate features and labels, and this makes the RBM particularly suitable to be used in the multi-output setting in the presence of incomplete observations. In this sense, the most natural way to deal with incomplete observations is to marginalize over the missing variables; in this section we show how the contrastive divergence algorithm can be adapted to compute such marginals.

Given a partially-observed instance $\boldsymbol{v}$, we have a new partition of the visible space $\mathcal{V} = \mathcal{O} + \mathcal{M}$, where $\mathcal{O}$ is a subset of observed values of $\boldsymbol{v}$ that can correspond both to features and labels. $\boldsymbol{v}_o = \{v_i, i \in \mathcal{O}\}$ and $\boldsymbol{v}_m = \{v_i, i \in \mathcal{M}\}$ denote respectively the observed and missing values of $\boldsymbol{v}$. The probability over the observed variables $\boldsymbol{v}_o$ is given by (θ representing the parameters of the model)

$$P(\boldsymbol{v}_o) = \frac{Z_{\mathcal{O}}[\theta]}{Z_{\emptyset}[\theta]}, \qquad Z_{\mathcal{O}}[\theta] = \int \prod_{i\in\mathcal{M}} p_{\text{prior}}(v_i)dv_i \times e^{\sum_{k\in\mathcal{V}} a_k v_k} \prod_{j\in\mathcal{H}} \left(1 + \exp\Big(\sum_{k\in\mathcal{V}} w_{kj} v_k + b_j\Big)\right)$$

Taking the log-likelihood and then computing the gradient with respect to the weight matrix element w_{ij} (also similarly for the fields a_i and b_j), we obtain two different expressions for $i \in \mathcal{O}$ and $i \in \mathcal{M}$.

$$\frac{\partial \log Z_{\mathcal{O}}[\theta]}{\partial w_{ij}} = v_i \sum_{h_j} h_j p(h_j|\boldsymbol{v}_o) \quad i \in \mathcal{O}, \qquad \frac{\partial \log Z_{\mathcal{O}}[\theta]}{\partial w_{ij}} = \sum_{h_j} \int dv_i v_i h_j p(v_i, h_j|\boldsymbol{v}_o) \quad i \in \mathcal{M} \tag{5}$$

The gradient of the LL over the weights (4) now reads

$$\frac{\partial \mathcal{L}(\boldsymbol{v};\theta)}{\partial w_{ij}} = \Big\langle I_o(i) v_i \sum_{h_j} h_j p(h_j|\boldsymbol{v}_o)\Big\rangle_{\text{data}} + \Big\langle \big(1 - I_o(i)\big) \sum_{h_j} \int dv_i v_i h_j p(h_j|\boldsymbol{v}_o)\Big\rangle_{\text{data}} - \langle v_i h_j\rangle_{\text{RBM}} \tag{6}$$

where I_o is the indicator function of the samples dependent set $\mathcal{O}$. The observed variables $v_i, i \in \mathcal{O}$ are pinned to the values given by the training samples. In terms of our model, the pinned variables play the role of an additional bias over the hidden variables of a RBM where the ensemble of visible variables is reduced to the missing ones.

With respect to the non-lossy case where $p(h_j|\boldsymbol{v})$ is given in closed form, here we need to sum over the missing variables in order to estimate $p(h_j|\boldsymbol{v}_o)$. This means that also the positive term of the gradient (6) is now intractable and we need to approximate it. For CD training, we can simply perform Gibbs sampling over the missing variables (keeping fixed the observed variables). Details are reported in Alg. 1.

We note that the extra computational burden of Lossy-CD with respect to standard CD is due only to the extra Gibbs sampling steps in the positive term. Given that the observed variables strongly bias the sampling procedure speeding up convergence, only few sampling steps are needed to compute this term. Indeed, in our experiments we observed that a single sampling step (Lossy-CD1) is enough, making the additional complexity minimal. Finally, we note that the same method can be applied to PCD and mean-field training procedures. In the first case, it is sufficient to keep track of an additional persistent chain, which requires little extra memory

and no extra computational complexity. In the second case, we only need to substitute Gibbs sampling with iterative mean-field equations.

Algorithm 1: Lossy-CDk (RBM training with Incomplete data)

Data: a training set of N data vectors
Randomly initialize the weight matrix **W**
for t = 0 to T (# of epochs) **do**
    Divide the training set in m minibatches
    for all minibatches m **do**
     Positive term:
        pin variables $v_i, i \in \mathcal{O}$ to their correct value
        initialize $v_i, i \in \mathcal{M}$ randomly
        sample $\boldsymbol{h}, \boldsymbol{v}_m$ using $p(\boldsymbol{v}_m \mid \boldsymbol{h})$ and $p(\boldsymbol{h} \mid \boldsymbol{v})$ for k steps
        compute the positive terms in (5)
     Negative term:
        initialize $\boldsymbol{v}$ randomly
        iterate eq. (2), (3) (k steps) to compute $\langle v_i h_j \rangle_{model}$
     Full update:
        update **W** with equation (6)
    end for
end for

4 Mean-field based imputation with RBM

As a generative model, the trained RBM can be used to sample new data. For imputation of missing features and labels we just need to use the observed portions of our data to bias the sampling procedure in the same way as for the computation of the positive term in Alg. 1. Namely, we estimate $p(\boldsymbol{v_m}|\boldsymbol{v_o})$ by pinning the observed variables and iterating CD/PCD or mean-field to approximate the equilibrium values of the missing variables. In case of a high percentage of missing observations, however, we might expect the observed variables to be correlated to many different equilibrium configurations, such that the sampling could be biased towards the wrong sample. To overcome this problem, we simply average over multiple mean-field imputations for each incomplete data instance.

More in details, let $\{p_i, i \in \mathcal{V}_\ell\}$ and $\{q_j, j \in \mathcal{H}\}$ be the marginal probabilities respectively of visible labels and hidden variables to be activated and $\{m_i, i \in \mathcal{V}_f\}$ the marginal expectation of the visible features variables. Mean-field equations at lowest order ($\mathcal{O}(1/N)$, N being the size of the system) express self-consistent relations among these quantities

$$m_i = \Big(\sum_{j \in \mathcal{H}} w_{ij} q_j + a_i \Big) \sigma_v^2 \quad \forall i \in \mathcal{V}_f \backslash \mathcal{O} \qquad p_i = \sigma\Big(\sum_{j \in \mathcal{H}} w_{ij} q_j + a_i \Big) \quad \forall i \in \mathcal{V}_\ell \backslash \mathcal{O} \tag{7}$$

$$q_j = \sigma\Big(\sum_{i \in \mathcal{V}_f} w_{ij} m_i + \sum_{i \in \mathcal{V}_\ell} w_{ij} p_i + b_j \Big) \tag{8}$$

Higher order terms corresponding to TAP equations are discarded [MÍ7]. These equations can be efficiently solved by iteration starting from random configurations until a fixed point is reached. Observed variables are simply introduced by pinning their corresponding probabilities (0 or 1 for label variables) or their marginal expectation (for feature variables) to the observed values. In practice we run these fixed-point equations $N_f \sim 10$ times and the imputations are obtained by simple average

$$\hat{m}_i = \frac{1}{N_f} \sum_{n=1}^{N_f} m_i^{(n)} \qquad p_i = \frac{1}{N_f} \sum_{n=1}^{N_f} p_i^{(n)}.$$

In the multi-label setting, the predictor is the indicator function $\hat{p}_i = (p_i > t)$ (t is learned, it is chosen to maximize the accuracy for known labels), while for class labels we have $\hat{p}_i = 1$ if $i = \text{argmax}_k(p_k)$

Model	RMSE			Averaged AUC			Accuracy		
q_{mc}%	30%	50%	80%	30%	50%	80%	30%	50%	80%
RBM-MO(q_{fea}% =50%)	**0.183**	**0.182**	**0.185**	**0.969**	**0.971**	**0.929**	**0.950**	**0.912**	**0.822**
CLE(q_{fea}% =50%)	0.195	0.195	0.195	0.686	0.718	0.742	0.256	0.232	0.282
NoisyIMC(q_{fea}% =50%)	0.209	0.210	0.210	0.621	0.578	0.552	0.225	0.232	0.192
MC-1(q_{fea}% =50%)	0.334	0.335	0.337	0.495	0.493	0.500	0.110	0.111	0.112
RBM-MO(q_{fea}% =80%)	**0.209**	**0.213**	**0.211**	**0.938**	**0.932**	**0.906**	**0.920**	**0.852**	**0.733**
CLE(q_{fea}% =80%)	0.206	0.208	0.206	0.673	0.678	0.625	0.230	0.215	0.220
NoisyIMC(q_{fea}% =80%)	0.212	0.211	0.213	0.652	0.577	0.537	0.230	0.217	0.210
MC-1(q_{fea}% =80%)	0.334	0.334	0.335	0.500	0.501	0.500	0.112	0.110	0.110

Table 1: Transductive test on *MNIST* multi-class data set (our method in bold, best result in red)

Model	RMSE			Micro-AUC			Hamming-Accuracy		
q_{ml}%	30%	50%	80%	30%	50%	80%	30%	50%	80%
RBM-MO(q_{fea}% =50%)	**0.131**	**0.137**	**0.123**	**0.943**	**0.934**	**0.888**	**0.919**	**0.907**	**0.873**
CLE(q_{fea}% =50%)	0.130	0.130	0.131	0.905	0.893	0.898	0.885	0.871	0.878
NoisyIMC(q_{fea}% =50%)	0.132	0.133	0.133	0.865	0.863	0.858	0.845	0.841	0.848
MC-1(q_{fea}% =50%)	0.258	0.255	0.267	0.522	0.528	0.527	0.826	0.817	0.824
RBM-MO(q_{fea}% =80%)	**0.160**	**0.158**	**0.158**	**0.875**	**0.867**	**0.826**	**0.856**	**0.858**	**0.832**
CLE(q_{fea}% =80%)	0.129	0.129	0.128	0.913	0.897	0.899	0.889	0.875	0.876
NoisyIMC(q_{fea}% =80%)	0.133	0.134	0.134	0.853	0.857	0.849	0.839	0.835	0.826

Table 2: Transductive test on *Scene* multi-label data set (our method in bold, best result in red)

5 Experimental Study

5.1 Experimental configuration

To evaluate the efficiency of RBM-MO we compare its performance against **CLE**, **NoisyIMC** and **MC-1**, which provide state-of-the-art baselines.

For the transductive experiments we randomly hide features and labels of the whole dataset to generate incomplete data for training, and we compute appropriate scores for the reconstruction of missing features and labels. In the inductive test, instead, we split the whole dataset into non-overlapping training and testing sets. Concerning the training set the same protocol is used as in the transductive test. For the test set the difference is that now all labels are hidden. Once the classifier is trained, it is applied on the test set to predict the labels. We still randomly hide the entries of test features vectors, so as to form an **incomplete testing set**. Finally, in the splitting we use 70% of the data instances for training and the remaining 30% for testing.

We denote by q_{fea}, q_{ml} and q_{mc} the percentage of masked features, labels and classes labels respectively. Note that a masked class label means that all binary variables attached to the classes of a given label are masked together. These rates of masking are kept identical in the learning and test sets.

In the transductive test, we compute the **Root Mean Squared Error (RMSE)** to measure the reconstruction accuracy with respect to the missing feature values. Furthermore, for the reconstructed labels we calculate **Micro-AUC** scores and **Hamming-accuracy** [GKG12] in the multi-label scenario, and **Averaged AUC** plus **Accuracy** [LY15] in the multi-class case. In the tables, we define **Hamming-accuracy** as **1-Hamming loss** to keep a consistent variation tendency with the AUC scores. In the inductive test we only compute the scores on the reconstructed labels, since reconstructing missing features is not the goal of inductive classification.

We run the test as described 10 times with different realizations of the missing features and labels. Average and

Model	Averaged AUC			Accuracy		
q_{mc}%	30%	50%	80%	30%	50%	80%
RBM-MO(q_{fea}% =50%)	**0.887**	**0.914**	**0.910**	**0.533**	**0.673**	**0.660**
CLE(q_{fea}% =50%)	0.785	0.791	0.791	0.297	0.256	0.268
NoisyIMC(q_{fea}% =50%)	0.780	0.771	0.781	0.302	0.272	0.265
RBM-MO(q_{fea}% =80%)	**0.891**	**0.909**	**0.889**	**0.562**	**0.682**	**0.647**
CLE(q_{fea}% =80%)	0.768	0.664	0.622	0.271	0.200	0.176
NoisyIMC(q_{fea}% =80%)	0.748	0.687	0.615	0.264	0.220	0.178

Table 3: Inductive test on *Pendigits* multi-class dataset (our method in bold, best result in red)

Model	Micro-AUC			Hamming-Accuracy		
$q_{ml}\%$	30%	50%	80%	30%	50%	80%
RBM-MO($q_{fea}\%$ =50%)	**0.839**	**0.826**	**0.793**	**0.970**	**0.970**	**0.965**
CLE($q_{fea}\%$ =50%)	0.705	0.707	0.706	0.700	0.724	0.719
NoisyIMC($q_{fea}\%$ =50%)	0.704	0.702	0.700	0.710	0.717	0.718
RBM-MO($q_{fea}\%$ =80%)	**0.759**	**0.791**	**0.766**	**0.964**	**0.964**	**0.967**
CLE($q_{fea}\%$ =80%)	0.693	0.688	0.694	0.718	0.706	0.718
NoisyIMC($q_{fea}\%$ =80%)	0.689	0.688	0.685	0.705	0.704	0.704

Table 4: Inductive test on *EventCat* multi-label data set (our method in bold, best result in red)

Dataset	**No. of Instances**	**No. of Features**	**No.of Labels**	**No. of Classes**
Scene	2,407	294	6	-
Pendigits	10992	16	-	10
MNIST	70,000	784	-	10
EventCat	5,93	72	6	-

Table 5: Summary of 4 public multi-label and multi-class data sets.

standard deviation of the computed scores are recorded to compare the overall performances. In the tables, we use red fonts to denote the best reconstruction and classification performances among all the algorithms involved in the empirical study. The bold black font is used to highlight the performance of the proposed **RBM-MO** method.

For the baselines, we used grid search to choose the optimal parameter combination following the suggested ranges of parameters as in [HSSZ18].

The RBM-MO is trained following the guidelines in [Hin10]. We always use binary variables for the hidden layer, while in the visible layer we use binary variables for MNIST and Gaussian variables for the other datasets. In all the simulations, we fix the number of hidden nodes to 100. The learning rate η is fixed to 0.001 and the size of the mini-batches to 10. During training the number of Gibbs steps is set to $k = 1$ while for imputation we iterate the mean-field equations 10 times. As a stopping condition, we considered the degradation of the transductive AUC scores with a look-ahead of 500 epochs

5.2 Summary of datasets

We consider 3 publicly available datasets related to image processing. These datasets cover both multi-label and multi-class learning tasks, and they are popularly used as benchmark datasets in multi-output learning research.

In addition, we consider the challenging scenario of abnormality detection on IoT devices. The relevant dataset, that we call *EventCat*, consists in security telemetry data collected from various network appliances (e.g. smart watches, smartphones, driving assistance systems...), each reporting a features vector whose entries indicate the occurring frequency of a specific type of alert (e.g. downloading suspicious files, login failures, unfixed vulnerabilities...). Multiple labels are assigned to each device in the collected dataset, corresponding to a variety of categories of security threats.

Some details about the datasets are reported in Table.5.

5.3 Qualitative results on MNIST

A qualitative evaluation of the performance of the RBM-MO model is given by looking at features reconstruction for the MNIST dataset, as reported in Fig. 1. The model at hand has been trained over a dataset in which 50% of the features were missing. To assess the robustness of the method, we computed the reconstructions in the highly challenging case in which 80% of the features were missing. Apart from some smoothing due to the employment of mean-field imputations, the reconstructed samples look reasonably realistic. In general, from the qualitative point of you the results are comparable to those obtained with more complex and expensive DLVMs like MIWAE and MisGAN [MF19, LJM19].

5.4 Empirical results

The transductive results for MNIST (multi-class) and Scene (multi-label) datasets are reported in tables 1 and 2. Going into the details, we first observe that **RBM-MO** is by a large margin more efficient than all of the

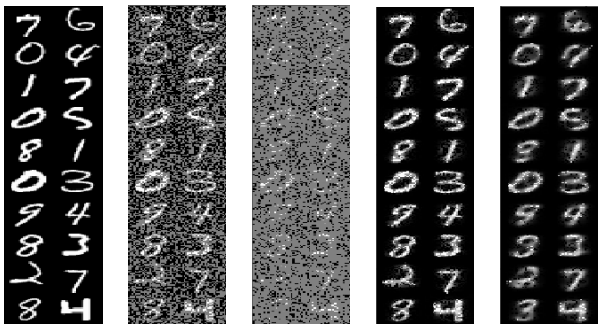

Figure 1: Features reconstruction by RBM-MO trained over an incomplete dataset with 50% missing-at-random features, whose classification accuracy has been measured to be around 91%. The first block shows some complete testing instances. The second and third block show the same testing instances after hiding respectively 50% and 80% of the pixels. The last two columns show the results of the mean-field imputations over the incomplete testing instances.

baselines for the inference of class labels (table 1), probably because it is able to encode more complex statistical properties.

On the multi-label problems, the situation is still in favour of **RBM-MO** but with less margin (table 2), in particular at a larger percentage of missing features.

Now if we look at the reconstruction error on these datasets we observe that **RBM-MO** generally achieves a higher reconstruction accuracy than the other opponents, especially on the MNIST dataset. The results verify empirically the basic motivation of using a generative model such as the RBM: **incomplete features and labels can provide complementary information to each other, so as to better recover the missing elements**. The variance of the results is omitted in the tables by lack of space. For **RBM-MO** the standard deviation of the derived RMSE, AUC and accuracy scores is not larger than 0.01 over the different datasets. Although the RMSE scores reported by the baseline methods look comparable to the RBM-MO ones, and in certain cases they are better, they also come with a slightly higher variance, such that the RBM-MO seems to be more efficient and robust for features reconstruction.

Except **MC-1**, all the baseline methods are used for inductive learning. As in the transductive test, we show only the mean of the derived metrics in the tables. Nevertheless, we have similar variance ranges for the computed scores as reported in the transductive test. Clearly **RBM-MO** is much better adapted to this setting than the baseline methods both for multi-class (table 3) and multi-label learning. The baseline inductive methods **CLE** and **NoisyIMC** are specifically designed for multi-label learning and their performance deteriorates significatively in the multi-class scenario. By comparison, **RBM-MO** can be adapted seamlessly to multi-class and multi-label learning, producing consistently good performances.

For the *EventCat* dataset, inductive results are reported in table 4. Even with highly incomplete training data, **RBM-MO** produces the best predictions over partially observed testing data instances.

6 Conclusion

Machine learning is witnessing a race to high complexity models eager for large data and computational power. In the context of multi-output classification in a challenging scenario - (i) learning with highly incomplete features and partially observed labels; ii) applying the learnt classifier with incomplete testing instances) - we advocate instead for simple probabilistic and interpretable models. After refining the learning of the RBM model, we give empirical evidences that it can be efficiently adapted to this context on a great variety of datasets. Experiments are conducted on both public databases and a real-world IoT security dataset, showing various sizes of training sets as well as features and labels vectors. Our approach consistently outperforms the state-of-the-art robust multi-class and multi-label learning approaches with imperfect training data, indicating good usability for practical applications.

References

[CHS15] Kai-Yang Chiang, Cho-Jui Hsieh, and Inderjit S.Dhillon. Matrix completion with noisy side information. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 3447–3455, Cambridge, MA, USA, 2015. MIT Press.

[ClTCB15] R. Cabral, F. D. l. Torre, J. P. Costeira, and A. Bernardino. Matrix completion for weakly-supervised multi-label image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1):121–135, Jan 2015.

[GJ94] Zoubin Ghahramani and Michael I. Jordan. Supervised learning from incomplete data via an em approach. In *Advances in Neural Information Processing Systems 6*, pages 120–127. Morgan Kaufmann, 1994.

[GKG12] G.Madjarov, D. Kocev, and D. Gjorgjevikj. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, pages 3083–3104, 2012.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[Hin02] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14:1771–1800, 2002.

[Hin10] Geoffrey Hinton. A practical guide to training restricted Boltzmann machines. *Momentum*, 2010.

[HSSZ18] Yufei Han, Guolei Sun, Yun Shen, and Xiangliang Zhang. Multi-label learning with highly incomplete data via collaborative embedding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '18, pages 1494–1503, 2018.

[KW14] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, pages 4413–4423, 2014.

[LJM19] Steven Cheng-Xian Li, Bo Jiang, and Benjamin Marlin. Learning from incomplete data with generative adversarial networks. In *International Conference on Learning Representations*, 2019.

[LY15] Yue Wu Li, Lin and Mao Ye. Experimental comparisons of multi-class classifiers. *Informatica*, 2015.

[LZG15] Xin Li, Feipeng Zhao, and Yuhong Guo. Conditional Restricted Boltzmann Machines for Multi-label Learning with Incomplete Labels. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 635–643, San Diego, California, USA, 09–12 May 2015. PMLR.

[MÍ7] M. Mézard. Mean-field message-passing equations in the Hopfield model and its generalizations. *Phys. Rev. E*, 95:022117, 2017.

[MF19] Pierre-Alexandre Mattei and Jes Frellsen. MIWAE: Deep generative modelling and imputation of incomplete data sets. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4413–4423, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[MTK15] G. Marylou, E.W. Tramel, and F. Krzakala. Training restricted Boltzmann machines via the Thouless-Anderson-Palmer free energy. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 640–648, 2015.

[NK94] M. J. Nijman and Kappen. Using boltzmann machines to fill in missing values, 1994.

[YJvdS18] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: Missing data imputation using generative adversarial nets. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5689–5698, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.