Knowledge-Based Systems 245 (2022) 108568

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Generating customer's credit behavior with deep generative models

Rogelio A. Mancisidor [a,*], Michael Kampffmeyer [b,c], Kjersti Aas [c], Robert Jenssen [b,c]

[a] Department of Data Science and Analytics, BI Norwegian Business School, Nydalsveien 37, 0484 Oslo, Norway
[b] Department of Physics and Technology, Faculty of Science and Technology, UiT The Arctic University of Norway, Hansine Hansens veg 18, Tromsø 9037, Norway
[c] Norwegian Computing Center, P.O. Box 114, Blindern, Oslo, Norway

ARTICLE INFO

Article history:
Received 23 September 2020
Received in revised form 7 March 2022
Accepted 9 March 2022
Available online 17 March 2022

Keywords:
Multi-modal learning
Credit scoring
Deep generative models
Representation learning

ABSTRACT

Banks collect data $\boldsymbol{x}_1$ in loan applications to decide whether to grant credit and accepted applications generate new data $\boldsymbol{x}_2$ throughout the loan period. Hence, banks have two measurement-modalities, which provide a complete picture about customers. If we can generate $\boldsymbol{x}_2$ conditioned on $\boldsymbol{x}_1$ keeping the relationship between these two modalities, credit and behavior scoring may be enabled simultaneously (at the time $\boldsymbol{x}_1$ is obtained) to support cross-selling, launching of new products or marketing campaigns. Therefore, we develop a novel conditional bi-modal discriminative (CBMD) model for credit scoring, which is able to generate $\boldsymbol{x}_2$ based on $\boldsymbol{x}_1$ and can classify the outcome of loans in an unified framework. The idea behind CBMD is to learn joint (among modalities) latent representations that are useful to generate $\boldsymbol{x}_2$ using the available data $\boldsymbol{x}_1$ during the application process. The classifier model introduced in CBMD encourages the generative process to generate $\boldsymbol{x}_2$ accurately. Further, CBMD optimizes a novel objective function introduced in this research, which maximizes mutual information between the latent representation $\boldsymbol{z}$ and the modality $\boldsymbol{x}_2$ to improve the generative process in the model. We benchmark the generative process of our proposed model and CBMD outperforms other multi-learning models. Similarly, the classification performance of CBMD is tested under different scenarios and it achieves higher or on a par model performance compared to the state-of-the-art in multi-modal learning models.

1. Introduction

Retail banks model the relationship between customer's information $\boldsymbol{x}$ and the outcome y of a loan to decide whether to grant credit, where $y = 0$ if a customer repays the loan otherwise $y = 1$. Traditionally, $\boldsymbol{x}$ has been limited to information captured during the application process, even though banks have access to more data that is generated by granted applications throughout the loan period, e.g. repayment or purchase behavior. Therefore, banks have two measurement-modalities that provide complementary information about a given customer. The first data modality, or view of data, is generated before the loan is granted and we denote it as $\boldsymbol{x}_1$. The second modality is generated throughout the loan period and we called this modality $\boldsymbol{x}_2$, see Fig. 1. Commonly, banks use $\boldsymbol{x}_1$ to develop credit scoring models, while $\boldsymbol{x}_2$ can be used to develop behavior models or to support cross-selling activities, launching of new products or marketing campaigns in banks.

Multi-modal learning designs models that utilize different measurements-modalities of the same object to learn joint data representations between modalities. Examples of multi-modalities, or views of data, are audio, video, and text, words and context, or credit data before and after the application process. A traditional application for multi-modal learning is downstream classification in a two-steps approach [1,2]. That is, a joint data representation is learned in the first stage and then, in the second stage, it is used to train a classifier model. The two steps approach has two major shortcomings. First, it can become a burden for practitioners if domain-specific classifiers need to be used, e.g. hidden Markov classifier with Gaussian mixtures as in [2]. Second, it uses a disjoint optimization for data representations and classification, which discards any possible synergy between these two.

Some multi-modal learning models are able to generate the input modalities using autoencoder-like architectures,[1] which clearly requires that all modalities are available at test time. This is not the case in the context of credit scoring, where $\boldsymbol{x}_2$ is not available at the same time as $\boldsymbol{x}_1$. If we can generate $\boldsymbol{x}_2$ conditioned

* Corresponding author.
E-mail addresses: rogelio.a.mancisidor@bi.no (R.A. Mancisidor), michael.c.kampffmeyer@uit.no (M. Kampffmeyer), kjersti@nr.no (K. Aas), robert.jenssen@uit.no (R. Jenssen).

[1] Such an architecture is designed to reconstruct the input data, i.e. $f(\boldsymbol{x}_2) = \boldsymbol{z}$ and $f(\boldsymbol{z}) = \hat{\boldsymbol{x}}_2$ where $f(\cdot)$ is a neural network.

https://doi.org/10.1016/j.knosys.2022.108568

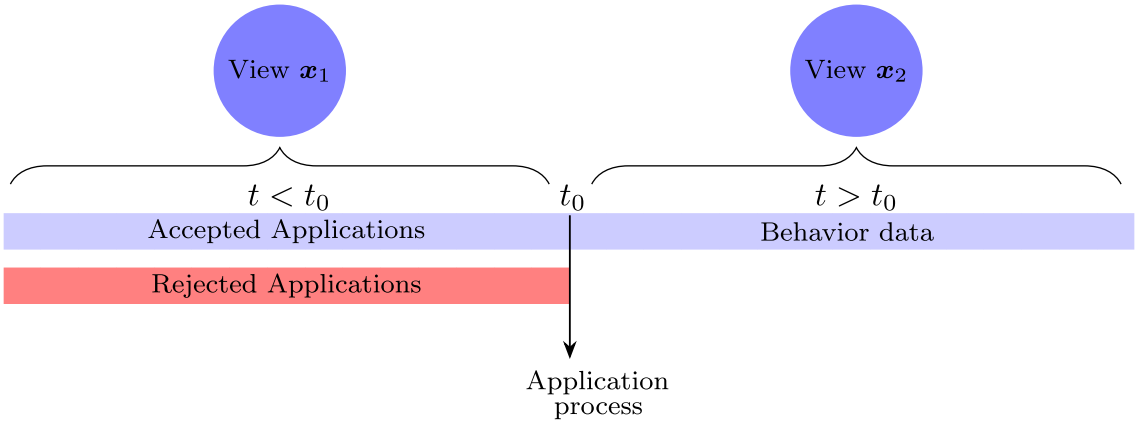


Fig. 1. Bi-modal credit data. At the time of the applications process t_0, only $\boldsymbol{x}_1$ is available. This data modality, which commonly is composed of socio-demographic features, is generated during $t < t_0$ and is used in credit scoring models. After the loan is granted, a new data modality $\boldsymbol{x}_2$ is generated, providing complementary information about the customer. Modality $\boldsymbol{x}_2$ is used to develop behavior models or to support cross-selling activities among others.

on $\boldsymbol{x}_1$ keeping the relationship between these two modalities, credit and behavior scoring may be enabled simultaneously (at the time $\boldsymbol{x}_1$ is obtained) to support cross-selling, launching of new products or marketing campaigns. Therefore, the main motivation for this research is to develop a novel bi-modal methodology that generates the modality $\boldsymbol{x}_2$ based on $\boldsymbol{x}_1$, which is our best source of information for future customer behavior. In other words, we use conditional distributions to keep the relation between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ since it makes sense to anchor the prediction of future bureau scores to its current value for example.

To that end, we develop a conditional bi-modal discriminative (CBMD) model that (i) learns to generate $\boldsymbol{x}_2$ conditioned on modality $\boldsymbol{x}_1$ together with data representations $\boldsymbol{z}$, and (ii) can classify class labels y using the learned data representations. The reason to include a classifier model into CBMD is twofold. First, to improve the generative process through the optimization of the classifier in an unified framework, which creates a synergy between representation learning and classifier training as in [3]. Second, to enable downstream classification with data representations using a classifier model that is relatively simple. This makes our proposed CBMD model useful for downstream generative and classification tasks in scenarios where only $\boldsymbol{x}_1$ is available at test time.

The contributions of this paper are as follows: (i) we develop the first bi-modal learning methodology for credit scoring, which generates the modality $\boldsymbol{x}_2$ conditioned on modality $\boldsymbol{x}_1$ and can classify the outcome of loans using latent representations, (ii) we show how can we utilize the generative properties of our proposed CBMD model to generate future credit data, and (iii) we introduce a novel objective function that maximizes mutual information between the common latent representation $\boldsymbol{z}$ and modality $\boldsymbol{x}_2$, which helps to improve the generative process of our proposed CBMD model.

The rest of the paper is organized as follows. Section 2 reviews the related work on multi-modal learning and Section 3 presents the proposed model. Further, Section 4 explains the data sets used in this research and presents the benchmark results. Finally, Section 6 discusses the main findings of this research.

2. Related work

This section reviews the research on multi-modal learning focusing on the development from the seminal canonical correlation analysis (CCA) [4] to models that optimize a variational lower bound and use neural networks to do amortized inference for model parameters. To facilitate model comparison, we use a common notation for all models where different data modalities are represented by $\boldsymbol{x}$ and are distinguished with a subscript, common latent transformations are represented by $\boldsymbol{z}$, private latent representations are denoted by $\boldsymbol{h}$ and a subscript referring to their data modality. Finally, labels are denoted by y. The plate notation for variational-based models included in this section are shown in Table 1.

Canonical correlation analysis finds linear projections by maximizing correlation between the transformations in multi-modal data. The objective is to learn the underlying semantic in the different modalities [5]. Originally, CCA deals only with linear projections of the data, but a kernel version of CCA was introduced in [5–9] to handle non-linearities.[2]

Both CCA and kernel-CCA maximize

$$\{f, g\} = \arg\max_{f,g} \frac{cov(f(\boldsymbol{x}_1), g(\boldsymbol{x}_2))}{\sqrt{var(f(\boldsymbol{x}_1)) \cdot var(g(\boldsymbol{x}_2))}}, \tag{1}$$

where $f(\boldsymbol{x}_1)$ and $g(\boldsymbol{x}_2)$ are the projections of modalities $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, subject to the constraints that $f_j(\boldsymbol{x}_1)$ is uncorrelated with $f_i(\boldsymbol{x}_1)$, $g_j(\boldsymbol{x}_2)$ is uncorrelated with $g_i(\boldsymbol{x}_2)$, and $f_j(\boldsymbol{x}_1)$ is uncorrelated with $g_i(\boldsymbol{x}_2)$ for all $i \neq j$. The difference between CCA and kernel-CCA is that the former assumes linear projections i.e. $f(\boldsymbol{x}_1) = \boldsymbol{v}^T\boldsymbol{x}_1$, while the latter uses linear combinations of the kernel k_1 evaluated at the data set, i.e. $f(\boldsymbol{x}_1) = \sum_{i=1}^{N} \alpha_i k_1(\boldsymbol{x}_1, \boldsymbol{x}_{1,i})$, where α_i determines the direction of the projections. Similar functions are used for the projection $g(\boldsymbol{x}_2)$.

A probabilistic interpretation of CCA is presented in [10]. The modalities $\boldsymbol{x}_1 \in \mathbb{R}^{d_1}$ and $\boldsymbol{x}_2 \in \mathbb{R}^{d_2}$ are generated given a common latent representation $\boldsymbol{z}$, that is

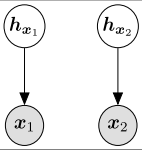
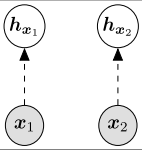
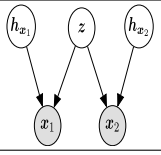
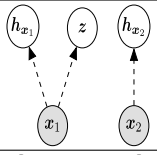
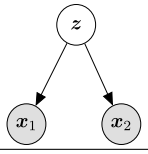
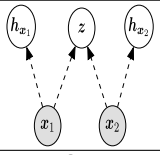
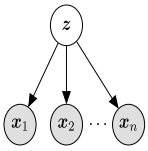
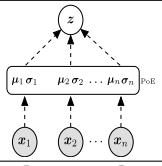
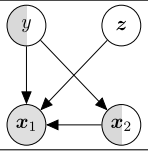
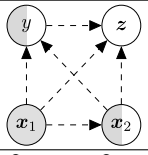
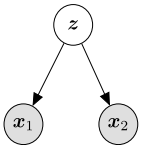
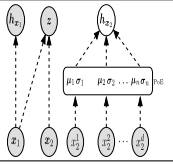
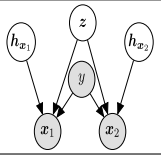
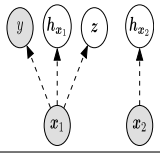
$$\begin{aligned}
\boldsymbol{z} &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d),\\
\boldsymbol{x}_1|\boldsymbol{z} &\sim \mathcal{N}(\boldsymbol{W}_1\boldsymbol{z} + \boldsymbol{\mu}_1,\ \boldsymbol{\Psi}_1),\\
\boldsymbol{x}_2|\boldsymbol{z} &\sim \mathcal{N}(\boldsymbol{W}_2\boldsymbol{z} + \boldsymbol{\mu}_2,\ \boldsymbol{\Psi}_2),
\end{aligned}$$

where $\min(d_1, d_2) \geq d \geq 1$ and $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, $\boldsymbol{\Psi}_1$, and $\boldsymbol{\Psi}_2$ are parameters defining a Gaussian distribution $\mathcal{N}(\cdot)$. These parameters are commonly estimated using the expectation–maximization (EM) algorithm [11] and their updating equations can be found in [10]. Furthermore, [10] show that linear discriminant analysis (LDA) [12] is a special case of CCA where one of the views is the label y.

Deep canonical correlation analysis [17] (DCCA) couple together deep neural networks and CCA with the objective to train neural networks able to maximize the correlation $\rho(f(\boldsymbol{x}_1), g(\boldsymbol{x}_2))$

[2] The method presented in [9] is an approximation based on random Fourier features.

Table 1
Overview over some generative and inference models presented in Section 2. We have harmonize the notation in all previous models with the one used in this paper. That is, given a bi-modal data, modality $\boldsymbol{x}_1$ is available during training and test time, while modality $\boldsymbol{x}_2$ is only available during training. Furthermore, common latent variables are denoted by $\boldsymbol{z}$, while private latent representations are represented by $\boldsymbol{h}_{\boldsymbol{x}_1}$ and $\boldsymbol{h}_{\boldsymbol{x}_2}$.

(Year) Author	Generative model	Inference model	Learning approach
(2015) Wang W. [1]			• Unsupervised representation learning • Loss function: AE + ACC • Training: SGD
(2016) Wang W. [2]			• Unsupervised representation learning • Loss function: VI lower bound • Training: SGD
(2016) Suzuki M. [13]			• Unsupervised representation learning • Loss function: VI lower bound • Training: SGD
(2018) Wu M. [14]			• Unsupervised representation learning • Loss function: VI lower bound with product of experts (PoE) • Training: SGD
(2018) Du C. [15]			• Semi-supervised classification • Loss function: VI lower bound • Training: SGD
(2018) Vedantam R. [16]			• Supervised representation learning • Loss function: VI lower bound • Training: SGD
(2019) Du F. [3]			• Supervised classification • Loss function: VI lower bound • Training: SGD

between modality $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. DCCA cannot only handle non-linearities, but can also capture high-level abstractions of the data in each of the multiple hidden layers. Note that the correlation objective function is a function of the entire data set, i.e. it is a fully batch objective function, and therefore it can be costly for large data sets. In a similar approach, [1] develop a model called deep canonically correlated autoencoder (DCCAE), where the objective function minimizes reconstruction error for both modalities (as in regular autoencoders) and optimizes canonical correlation between the learned representations (as in CCA). The main difference between DCCA and DCCAE is that the latter can reconstruct both modality $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, and DCCAE scales to large data sets using stochastic gradient descent to optimize its objective function.

A problem with DCCAE is that the CCA term in its objective function dominates the optimization procedure [1]. As a consequence, the reconstruction of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is poor. To overcome this problem, [2] use variational inference and deep generative models to generate latent representations of the input modalities and to reconstruct them. The authors in [2] present a model called variational CCA (VCCA) that uses a common latent variable to generate both modalities. In a second version, VCCA uses common and private latent variables to generate modality $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Note that when only common latent variables are used, it is not clear how to specify the inference model, i.e. $q(\boldsymbol{z}|\boldsymbol{x}_1)$ or $q(\boldsymbol{z}|\boldsymbol{x}_2)$. Therefore, the authors propose the objective function $\mathcal{L} = \mu\mathcal{L}_{q(\boldsymbol{z}|\boldsymbol{x}_1)} + (1-\mu)\mathcal{L}_{q(\boldsymbol{z}|\boldsymbol{x}_2)}$, where $\mathcal{L}_{q(\boldsymbol{z}|\boldsymbol{x}_1)}$ ($\mathcal{L}_{q(\boldsymbol{z}|\boldsymbol{x}_2)}$) is the loss function when $q(\boldsymbol{z}|\boldsymbol{x}_1)$ ($q(\boldsymbol{z}|\boldsymbol{x}_2)$) defines the inference model and $\mu \in [0, 1]$ is a weight parameter controlling the importance of each term in the objective function.

A supervised extension of VCCA is proposed by [3], which combines multi-modal learning and classification in one unified framework. The authors propose a discriminative multi-modal deep generative model (DMDGM) that generates both modalities based on private and common hidden variables. Unlike most approaches for downstream classification, DMDGM uses the available modalities at test time for classification, e.g $q(y|\boldsymbol{x}_1)$ or $q(y|\boldsymbol{x}_1, \boldsymbol{x}_2)$. This is not the only model where classification is addressed in a unified objective function, [15] develops a semi-supervised deep generative model for missing modalities where the latent variable is shared across modalities. To further improve the flexibility of the latent space, the authors model the inference process as a Gaussian mixture model (GMM). However, it is worth mentioning that modeling the inference process as GMM harms

the tightness of the lower bound since the entropy of a GMM is intractable.

The joint multimodal variational autoencoder (JMVAE) is introduced in [13]. The first model presented by the authors replaces missing modalities with zeros, e.g. $q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2) \approx q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{0})$ if $\boldsymbol{x}_2$ is missing. The second model presented in [13] includes two individual inference models $q(\boldsymbol{z}|\boldsymbol{x}_1)$ and $q(\boldsymbol{z}|\boldsymbol{x}_2)$, and one global inference model $q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)$. Further, the objective function includes two Kullback–Leibler (KL) divergence terms, $KL[q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)||q(\boldsymbol{z}|\boldsymbol{x}_1)]$ and $KL[q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)||q(\boldsymbol{z}|\boldsymbol{x}_2)]$, which force $q(\boldsymbol{z}|\boldsymbol{x}_1)$ and $q(\boldsymbol{z}|\boldsymbol{x}_2)$ to be close to $q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)$. The authors argue that including these two KL terms is equivalent to minimizing the lower bound of variation of information (VaI). This is not the only model optimizing the information theoretic measure VaI, [18] use restricted Boltzmann machines to develop a multi-modality model, which objective function is fully derived from a VaI perspective.

All previous models in this section assume data with only two modalities. A model that generalizes to more than two modalities is presented in [14]. Their deep generative model assumes that the posterior distribution $p(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)$ is proportional to the product of individual posteriors $p(\boldsymbol{z}|\boldsymbol{x}_1)\cdots p(\boldsymbol{z}|\boldsymbol{x}_n)$ normalized by the prior distribution $p(\boldsymbol{z})$. Additionally, they assume that individual posteriors are approximated by variational densities $q(\boldsymbol{z}|\boldsymbol{x}_i)$ for $i = 1, \cdots, n$. Hence, the joint posterior distribution is a product of experts (PoEs). Another model using PoEs is presented in [16]. However, in this case, the authors use a PoEs to deal with missing modalities, i.e. $q(\boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_2) \propto p(\boldsymbol{h}_{\boldsymbol{x}_2})\prod_{k\in\mathcal{O}} q(\boldsymbol{h}_{\boldsymbol{x}_2}|x_2^k)$, where $\mathcal{O}$ are the observed attributes in modality $\boldsymbol{x}_2$.

Objective functions optimizing a mutual information (MI) term have been introduced in infoGAN [19] and infoVAE [20], which are uni-modal unsupervised learning methods. infoGAN approximates MI by using the variational information maximization approach, which is a variational lower bound, and optimizes a minimax game based on generative adversarial networks [21]. On the other hand, infoVAE adds a MI term to the objective function to learn amortized inference distribution and to learn representations that embed information about $\boldsymbol{z}$. [20] show, in a two-steps classification experiment with latent representations, that infoVAE achieves the same classification as an unregularized autoencoder using a latent space with more than 10 dimensions. Meaning that the learned representations embed information about $\boldsymbol{x}$.

Our proposed CBMD model uses a prior distribution $p(\boldsymbol{z}|\boldsymbol{x}_1)$ that is conditioned on modality $\boldsymbol{x}_1$ to generate the modality $\boldsymbol{x}_2$ using the generative process $p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{z})$. Such a generative mechanism keeps the relationship between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ and allows us to generate $\boldsymbol{x}_2$ at the same time a loan application is received. CBMD, unlike infoGAN, optimizes a novel objective function based on variational inference, which maximizes MI between latent representations $\boldsymbol{z}$ and modality $\boldsymbol{x}_2$ to effectively learn amortized inference distributions and to generate accurate $\boldsymbol{x}_2$ samples. However, unlike infoVAE, our motivation to include a MI term stems from the restriction imposed by the variational lower bound on MI.

3. Conditional bi-modal discriminative model

Before we introduce our proposed CBMD model,[3] we define some variables that are used throughout this section. Let $\boldsymbol{x}_1$ be the data modality available at the time a loan application is received. Common features in this modality are: age, income, gender, geographical location, etc. Once an application is approved, customers generate new information constituting modality $\boldsymbol{x}_2$. The sort of information in this modality can be updated values for features in $\boldsymbol{x}_1$, e.g. latest income, current age, latest marital status etc. Other kind of features in $\boldsymbol{x}_2$ can be repayment or purchase behavior. In the context of this research, we have access to class labels y, where $y = 0$ denotes if a customer repaid a loan, otherwise $y = 1$. Finally, we assume that there is a common latent representation $q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)$ with prior $p(\boldsymbol{z}|\boldsymbol{x}_1)$ and a private posterior representation $q(\boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_2)$ with prior $p(\boldsymbol{z}_2)$. Both latent representations contain high-level information of both data modalities providing complementary information about the outcome of the loan.

3.1. Deriving the CBMD lower bound

We observe labeled bi-modal data $\{(\boldsymbol{x}_1^{(1)}, \boldsymbol{x}_2^{(1)}, y^{(1)}), \cdots, (\boldsymbol{x}_1^{(N)}, \boldsymbol{x}_2^{(N)}, y^{(N)})\}$ that is generated at different point in times, where only $\boldsymbol{x}_1$ is available at application time. Further, the modality $\boldsymbol{x}_2$ and class label y are generated after a loan application is granted.

We focus on learning a joint latent representation $\boldsymbol{z}$ and a private representation $\boldsymbol{h}_{\boldsymbol{x}_2}$ that can be used for downstream classification and to generate $\boldsymbol{x}_2$. For that purpose, we assume a conditional prior distribution $p(\boldsymbol{z}|\boldsymbol{x}_1)$ for the modality that is available at test time and an uninformative private distribution $p(\boldsymbol{h}_{\boldsymbol{x}_2})$ for the modality $\boldsymbol{x}_2$, which is missing at test time. Under this scenario, the joint generative process is given by

$$p(\boldsymbol{x}_2, \boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_1) = p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{z})p(\boldsymbol{z}|\boldsymbol{x}_1)p(\boldsymbol{h}_{\boldsymbol{x}_2}), \tag{2}$$

where $p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{z})$ is the generative process for future credit scoring data. Note that the posterior distribution of the latent variable, which is exactly the joint latent representation that we want to learn,

$$p(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{p(\boldsymbol{x}_2, \boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_1)}{\iint p(\boldsymbol{x}_2, \boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_1)d\boldsymbol{z}d\boldsymbol{h}_{\boldsymbol{x}_2}} \tag{3}$$

requires a marginal distribution that is not available in closed form. Therefore, we approximate the true posterior distribution $p(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)$ in Eq. (3) with the variational distribution $q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)$.

Taking the log of the marginal distribution in Eq. (3) we obtain the lower bound

$$\begin{aligned}
\log p(\boldsymbol{x}_2|\boldsymbol{x}_1) &= \log \iint p(\boldsymbol{x}_2, \boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_1)d\boldsymbol{z}d\boldsymbol{h}_{\boldsymbol{x}_2} \\
&= \log \iint q(\boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_1, \boldsymbol{x}_2)\frac{p(\boldsymbol{x}_2, \boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_1)}{q(\boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_1, \boldsymbol{x}_2)}d\boldsymbol{z}d\boldsymbol{h}_{\boldsymbol{x}_2} \\
&= \log \mathbb{E}_{q(\boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_1, \boldsymbol{x}_2)}\left[\frac{p(\boldsymbol{x}_2, \boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_1)}{q(\boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_1, \boldsymbol{x}_2)}\right] \\
&\geq \mathbb{E}_{q(\boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_1, \boldsymbol{x}_2)}\left[\log \frac{p(\boldsymbol{x}_2, \boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_1)}{q(\boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_1, \boldsymbol{x}_2)}\right],
\end{aligned} \tag{4}$$

where the inequality is a result of the concavity of log and Jensen's inequality. Eq. (4) is the variational lower bound $\mathcal{L}(\boldsymbol{x}_2, \boldsymbol{x}_1)$ on the conditional log-likelihood $\log p(\boldsymbol{x}_2|\boldsymbol{x}_1)$, which in principle can be optimized using the stochastic variational gradient Bayes (SVGB) approach introduced in [22].

Expanding the lower bound in Eq. (4) and assuming $q(\boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_1, \boldsymbol{x}_2) = q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)q(\boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_2)$, we get that

$$\begin{aligned}
\mathcal{L}(\boldsymbol{x}_2, \boldsymbol{x}_1) =& \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)}[\log p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{z}) + \log p(\boldsymbol{z}|\boldsymbol{x}_1) - \log q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)] \\
&+ \mathbb{E}_{q(\boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_2)}[\log p(\boldsymbol{h}_{\boldsymbol{x}_2}) - \log q(\boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_2)] \\
=& \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)}[\log p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{z})] - KL[q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)||p(\boldsymbol{z}|\boldsymbol{x}_1)] \\
&- KL[q(\boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_2)||p(\boldsymbol{x}_2)].
\end{aligned} \tag{5}$$

While in some cases optimizing Eq. (5) should be sufficient to do amortized inference and to reconstruct $\boldsymbol{x}_2$ correctly, it has been shown that this formulation of the lower bound has two main problems [20,23]. First, it can fail to learn an amortized inference

[3] https://github.com/rogelioamancisidor/cbmd.

distribution $q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)$ that correctly approximates $p(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)$. Second, the model can focus on reconstructing $\boldsymbol{x}_2$ ignoring the latent data representation $\boldsymbol{z}$, which implies that $\boldsymbol{z}$ does not depend on $\boldsymbol{x}_2$. This problem is called posterior collapse [24] and we attempt to solve it by the explicit optimization of the mutual information between $\boldsymbol{x}_2$ and $\boldsymbol{z}$.

To solve the aforementioned challenges, we propose a new objective function that maximizes mutual information between $\boldsymbol{z}$ and $\boldsymbol{x}_2$. Note that, assuming the factorizations $q(\boldsymbol{x}_1)q(\boldsymbol{x}_2|\boldsymbol{x}_1)q(\boldsymbol{z}|\boldsymbol{x}_1)$ and $q(\boldsymbol{z}|\boldsymbol{x}_2, \boldsymbol{x}_1)q(\boldsymbol{x}_2|\boldsymbol{x}_1) = q(\boldsymbol{x}_2, \boldsymbol{z}|\boldsymbol{x}_1)$, the conditional mutual information $I(\boldsymbol{x}_2, \boldsymbol{z}|\boldsymbol{x}_1)$ can be written as

$$
\begin{aligned}
I(\boldsymbol{x}_2, \boldsymbol{z}|\boldsymbol{x}_1) =& \mathbb{E}_{q(\boldsymbol{x}_2,\boldsymbol{z},\boldsymbol{x}_1)}\left[\log \frac{q(\boldsymbol{x}_2, \boldsymbol{z}|\boldsymbol{x}_1)}{q(\boldsymbol{x}_2|\boldsymbol{x}_1)q(\boldsymbol{z}|\boldsymbol{x}_1)}\right] \\
=& \mathbb{E}_{q(\boldsymbol{x}_2,\boldsymbol{z},\boldsymbol{x}_1)}\left[\log \frac{q(z|\boldsymbol{x}_1, \boldsymbol{x}_2)q(\boldsymbol{x}_2|\boldsymbol{x}_1)}{q(\boldsymbol{x}_2|\boldsymbol{x}_1)q(\boldsymbol{z}|\boldsymbol{x}_1)}\right] \\
=& \mathbb{E}_{q(\boldsymbol{x}_2,\boldsymbol{z},\boldsymbol{x}_1)}[\log q(z|\boldsymbol{x}_1, \boldsymbol{x}_2) - \log q(\boldsymbol{z}|\boldsymbol{x}_1) \\
&+ \log p(\boldsymbol{z}|\boldsymbol{x}_1) - \log p(\boldsymbol{z}|\boldsymbol{x}_1)] \\
=& \mathbb{E}_{q(\boldsymbol{x}_2,\boldsymbol{x}_1)}\big[KL[q(z|\boldsymbol{x}_1, \boldsymbol{x}_2)||p(\boldsymbol{z}|\boldsymbol{x}_1)]\big] \\
&- \mathbb{E}_{q(\boldsymbol{x}_1)}\big[KL[q(\boldsymbol{z}|\boldsymbol{x}_1)||p(\boldsymbol{z}|\boldsymbol{x}_1)]\big],
\end{aligned} \tag{6}
$$

where $q(\boldsymbol{x}_2, \boldsymbol{x}_1)$ and $q(\boldsymbol{x}_1)$ are estimated using the empirical data distribution. Hence, adding the mutual information term $(1-\omega)I(\boldsymbol{x}_2, \boldsymbol{z}|\boldsymbol{x}_1)$ to Eq. (5) we obtain the objective function for a single data point

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{x}_1, \boldsymbol{x}_2) =& \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x}_1,\boldsymbol{x}_2)}[\log p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{z})] - KL[q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)||p(\boldsymbol{z}|\boldsymbol{x}_1)] \\
&- KL[q(\boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_2)||p(\boldsymbol{x}_2)] + (1-\omega)[KL[q(z|\boldsymbol{x}_1, \boldsymbol{x}_2)||p(\boldsymbol{z}|\boldsymbol{x}_1)] \\
&- KL[q(\boldsymbol{z}|\boldsymbol{x}_1)||p(\boldsymbol{z}|\boldsymbol{x}_1)]] \\
=& \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x}_1,\boldsymbol{x}_2)}[\log p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{z})] - KL[q(\boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_2)||p(\boldsymbol{h}_{\boldsymbol{x}_2})] \\
&- \omega KL[q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)||p(\boldsymbol{z}|\boldsymbol{x}_1)] + (1-\omega)KL[q(\boldsymbol{z}|\boldsymbol{x}_1)||p(\boldsymbol{z}|\boldsymbol{x}_1)],
\end{aligned} \tag{7}
$$

where $\omega \in [0, 1]$ is a weight hyperparameter. The first two KL divergence terms in Eq. (7) have an analytical solution. However, the last KL divergence is intractable, due to the marginal distribution $q(\boldsymbol{z}|\boldsymbol{x}_1)$, but can be replaced by any strict divergence term [20], e.g. maximum mean discrepancy divergence (MMD) [25]. We choose the non-parametric squared MMD that can be estimated numerically and is given by

$$
\begin{aligned}
\text{MMD}[\mathcal{F}, p, q] =& \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{x}')}[k(\boldsymbol{x}, \boldsymbol{x}')] - 2\mathbb{E}_{p(\boldsymbol{x}),q(\boldsymbol{z})}[k(\boldsymbol{x}, \boldsymbol{z})] \\
&+ \mathbb{E}_{q(\boldsymbol{z},\boldsymbol{z}')}[k(\boldsymbol{z}, \boldsymbol{z}')],
\end{aligned} \tag{8}
$$

where $\mathcal{F}$ be a unit ball in a universal reproducing kernel Hilbert space $\mathcal{H}$, p and q are Borel probability measures and $k(\cdot, \cdot)$ is a universal kernel. We use a Gaussian kernel in our proposed model to obtain the objective function

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{x}_1, \boldsymbol{x}_2) =& \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x}_1,\boldsymbol{x}_2)}[\log p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{z})] - KL[q(\boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_2)||p(\boldsymbol{h}_{\boldsymbol{x}_2})] \\
&- \omega KL[q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)||p(\boldsymbol{z}|\boldsymbol{x}_1)] \\
&+ (1-\omega)\lambda MMD[q(\boldsymbol{z}|\boldsymbol{x}_1)||p(\boldsymbol{z}|\boldsymbol{x}_1)],
\end{aligned} \tag{9}
$$

where λ counteracts the loss imbalance between $\mathcal{X}_2$ and $\mathcal{Z}$ spaces. Eq. (9) give us more flexibility to reconstruct all features in modality $\boldsymbol{x}_2$ utilizing the joint latent representation $\boldsymbol{z}$ and to learn amortized inference distributions $q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)$.

It is worth analyzing Eq. (5) and (6). Given that the KL divergence is non-negative, Eq. (6) implies that (for one observation) $KL[q(z|\boldsymbol{x}_1, \boldsymbol{x}_2)||p(\boldsymbol{z}|\boldsymbol{x}_1)] \geq I(\boldsymbol{x}_2, \boldsymbol{z}|\boldsymbol{x}_1)$. In other words, the divergence $KL[q(z|\boldsymbol{x}_1, \boldsymbol{x}_2)||p(\boldsymbol{z}|\boldsymbol{x}_1)]$ is an upper bound on the conditional mutual information. Further, note that the upper bound is the same KL divergence as in Eq. (5). Hence, optimizing the regular lower bound imposes an upper bound on $I(\boldsymbol{x}_2, \boldsymbol{z}|\boldsymbol{x}_1)$, which can result in the undesired posterior collapse problem.

However, we are interested in developing a model that, in addition to generate modality $\boldsymbol{x}_2$, can also classify the outcome of the loan. Further, given that we have a supervised data set, we want to use label information to learn joint latent representations. Hence, we add a classification loss $q(y|\boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2})$ and replace $q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)$ by $q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2, y)$ in Eq. (9) to obtain the following final loss function in our proposed model

$$
\mathcal{J} = -\mathcal{L}(\boldsymbol{x}_1, \boldsymbol{x}_2, y) - \alpha \log q(y|\boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}), \tag{10}
$$

where α controls the importance of the classification loss in the objective function, and its plate notation is shown in Fig. 2.

We minimize Eq. (10) using SVGB and automatic differentiation routines in Theano [26]. Note that the reconstruction term of Eq. (9) can be efficiently estimated using the *reparameterization trick* [22], the KL divergence term has a closed-form expression [22,27], and the MMD divergence is approximated numerically by sampling from $q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2, y)$ and $p(\boldsymbol{z}|\boldsymbol{x}_1)$ for a given mini-batch of data as suggested by [25].

Finally, we assume the following density functions in our proposed CBMD model

$$
\begin{aligned}
p(\boldsymbol{h}_{\boldsymbol{x}_2}) \sim& \mathcal{N}(\mathbf{0}, \mathbf{1}) \\
p(\boldsymbol{z}|\boldsymbol{x}_1) \sim& \mathcal{N}(\boldsymbol{z}|\boldsymbol{x}_1; \boldsymbol{\mu} = f_\theta(\boldsymbol{x}_1), \boldsymbol{\sigma}^2 = f_\theta(\boldsymbol{x}_1)), \\
p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{z}) \sim& \mathcal{N}(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{z}; \boldsymbol{\mu} = f_\theta(\boldsymbol{x}_1, \boldsymbol{z}), \boldsymbol{\sigma}^2 = f_\theta(\boldsymbol{x}_1, \boldsymbol{z})), \\
q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2, y) \sim& \mathcal{N}(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2, y; \boldsymbol{\mu} = f_\phi(\boldsymbol{x}_1, \boldsymbol{x}_2, y), \boldsymbol{\sigma}^2 = f_\phi(\boldsymbol{x}_1, \boldsymbol{x}_2, y)), \\
q(\boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_2) \sim& \mathcal{N}(\boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_2; \boldsymbol{\mu} = f_\phi(\boldsymbol{x}_2), \boldsymbol{\sigma}^2 = f_\phi(\boldsymbol{x}_2)), \\
q(y|\boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}) \sim& \text{Bernoulli}(y|\boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2}; \pi_{y|\boldsymbol{z},\boldsymbol{h}_{\boldsymbol{x}_2}} = f_\phi(\boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2})),
\end{aligned} \tag{11}
$$

where $\mathcal{N}$ denotes the Gaussian distribution and $f(\cdot)$ is a multilayer perceptron (MLP) network [28]. That is, the density parameters $\boldsymbol{\mu}$, $\boldsymbol{\sigma}^2$, and $\pi_{y|\boldsymbol{z},\boldsymbol{h}_{\boldsymbol{x}_2}}$ are parametrized using neural networks with learnable parameters denoted by $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$.

The first density in Eq. (11) is non-informative about the future credit data, while the second equation learns a latent representation ($\boldsymbol{z}$) based on the available information ($\boldsymbol{x}_1$) during the loan application process. In other words, $p(\boldsymbol{z}|\boldsymbol{x}_1)$ represents our prior beliefs about the joint representation $\boldsymbol{z}$ and it is based on information available during the application process. The third density learns a data generating process to draw future credit scoring data ($\boldsymbol{x}_2$) based on available information ($\boldsymbol{x}_1$) and the joint representation ($\boldsymbol{z}$). The fourth density function, where we add the class label information (y), learns the posterior latent representation for credit data. The fifth density learns a latent representation for future credit data. Finally, the last density function classifies the outcome of a loan y using latent representations ($\boldsymbol{z}$ and $\boldsymbol{h}_{\boldsymbol{x}_2}$) for credit scoring data, and encourages latent representations to capture higher-level of abstractions that are useful for classification and to generate modality $\boldsymbol{x}_2$.

4. Experiments and results

The motivation for the experiments is threefold. First, we compare the generative process of our proposed methodology with existing multi-modal learning models using two modalities. Second, we show how financial institutions can utilize the generative network in the CBMD model to generate future data. Third, we compare the predictive power of the learned data representation for all models. In all experiments, we assume that only $\boldsymbol{x}_1$ is available at test time to generate joint representations $\boldsymbol{z}$ which are further used for downstream generative and classification tasks

The models included in this section are CCA [4], KCCA [9], DCCA [17], and DCCAE [1], which all are based on canonical correlation. We also include VCCA [2] and JMVAE [13] that are

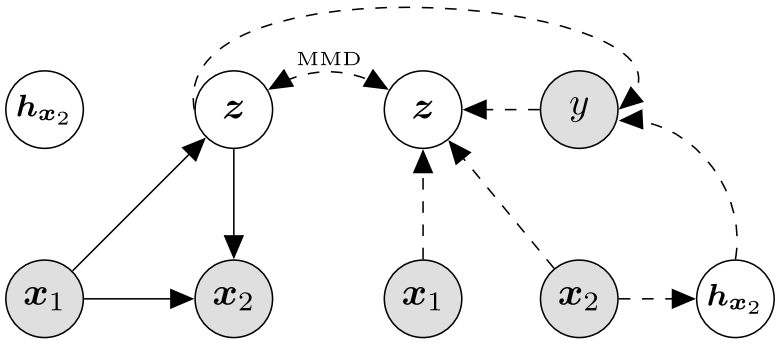


Fig. 2. Plate notation for our proposed bi-modality discriminative model for credit scoring. The left side shows the generative model, where the prior distribution of $\boldsymbol{z}$ is condition on the modality $\boldsymbol{x}_1$. The right side shows the inference model, where we explicitly optimize maximum mean discrepancy to minimize the information preference problem.

variational-based methods.[4] To allow a fair comparison to CBMD, all models are tested without pre-trained weights as in [1] or without adding generative adversarial networks to further improve reconstructed values as in [13]. In the classification experiments, we use fixed variance parameters in the generative networks for VCCA and JMVAE as suggested in their original papers. Otherwise, downstream classification is poor. It is worth mentioning that, in our experiments, VCCA is more prone to poor classification than JMVAE if the variance parameters are learned during the optimization process.

In order to test the generalization properties of our proposed model, we include a real data set for purchase prediction containing 200 features. Hence, we can test CBMD on scenarios with large number of missing features at test time. Note, data modalities do not need to be time-dependent. Therefore, creating a bi-modal data set based on the predictive power for each feature, which we explain in the next section, is a valid approach.

4.1. Data description

We use two real and publicly available data sets in this section.[5] The first data set corresponds to customers at Banco Santander and it contains 200 (anonymized) numerical features for purchase prediction, i.e. which customer will make a future transaction regardless of the amount. A training and test data set are available, but we only use the training data set since the test data set has no label information. The training data set contains 200 000 observations and there are 20 098 customers that made a purchase, which corresponds to 10.05% of customers. Given that behavioral models have higher model performance than credit scoring models [29], we assume that features with high predictive power[6] correspond to modality $\boldsymbol{x}_2$. Therefore, in the experiments conducted in Section 4.3.2, we select the top 50 features as modality $\boldsymbol{x}_2$, while the rest of the features correspond to view $\boldsymbol{x}_1$. Given the number of features in this data set, we also tested all models under a more challenging scenario where modality $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ contain 100 features each.

The second data set consists of peer-to-peer loan applications from January 2009 to December 2013 at Lending Club.[7] We only include accepted loans with 36-months maturity and some observations have been excluded using the same criteria as in [27,31]. This data set contains 89 998 accepted applications, where 10 896 are defaulted loans, i.e. default rate is 12.11%. Further, we choose the modality $\boldsymbol{x}_1$ to be all common features in accepted and rejected applications, which are only 5 features. All categorical features in modality $\boldsymbol{x}_1$ are transformed to one-hot-encoders, resulting in a 18D vector. On the other hand, the modality $\boldsymbol{x}_2$ contains 72 features and we select only features that are both continuous and with empirical distributions resembling Gaussian densities. Hence, we select 8 features for the modality $\boldsymbol{x}_2$ in the experiments conducted in Section 4.3.2. This choice is driven by the fact that modality $\boldsymbol{x}_1$ has only 5 original features. Details about data modalities in the Lending Club data set are shown in Appendix A.

4.2. Model training and testing

We use MLP networks with softplus activation functions in all hidden layers to parametrise $\boldsymbol{\mu}$, $\boldsymbol{\sigma}^2$ and $\pi_{y|\boldsymbol{z},\boldsymbol{h}_{\boldsymbol{x}_2}}$ in Eq. (11). For the output layers parameterizing $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$, we use linear activation functions, while for the classifier we use a softmax activation function. The minimization of the loss function is done using the Adam optimizer [32] with learning rate equal to 1e-4. The final architectures that we used in our proposed model, as well as all architectures used in the grid-search to tune the MLPs, are shown in Table B.1 in the Appendix B and are chosen based on both classification and generative performance. All CCA-based and variational-based models are trained with similar architectures to CBMD for a fair comparison. Further, for DCCAE we tune the λ parameter by grid search as suggested in [1]. Similarly, we tune the α and variance parameters by grid search in JMVAE and VCCA respectively. Finally, both data sets are scaled between 0 and 1 for better training stability.

During training we have a supervised data set containing both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, as well as the class label y. At test time we assume that only modality $\boldsymbol{x}_1$ is available. Therefore, at training time we draw samples from $q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2, y)$ to reconstruct modality $\boldsymbol{x}_2$ using the generative process $p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{z})$ in our proposed CBMD model. While at test time, we need to rely on the conditional prior distribution $p(\boldsymbol{z}|\boldsymbol{x}_1)$ to draw $\boldsymbol{z}$. Then, we use that representation to generate $\boldsymbol{x}_2$ using $p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{z})$. In other words, we generate future credit data ($\boldsymbol{x}_2$) based on current information about the loan application ($\boldsymbol{x}_1$) and based on the prior distribution ($p(\boldsymbol{z}|\boldsymbol{x}_1)$) of the joint latent representation. Note that the conditional prior distribution in our proposed model is more informative than the classical choice $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

We observed in our experiments that training the classifier with $\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2, y)$ leads to unstable classification performance. This problem arises because we assume that $\boldsymbol{x}_2$ is missing

[4] In our experiments, we use the implementations for CCA, KCCA, DCCA, DCCAE, and VCCA at https://ttic.uchicago.edu/ wwang5/. While, results for JMVAE are based on our own implementation.

[5] Banco Santander data set: https://www.kaggle.com/c/santander-customer-transaction-prediction/data. Lending Club data set: https://github.com/nateGeorge/preprocess_lending_club_data.

[6] We use the method introduced in [30] to estimate feature importance.

[7] Lending Club is the world's largest peer-to-peer lending company and it was the first peer-to-peer lender to register its offerings as securities with the Securities and Exchange Commission in the U.S., and to offer loan trading on a secondary market.

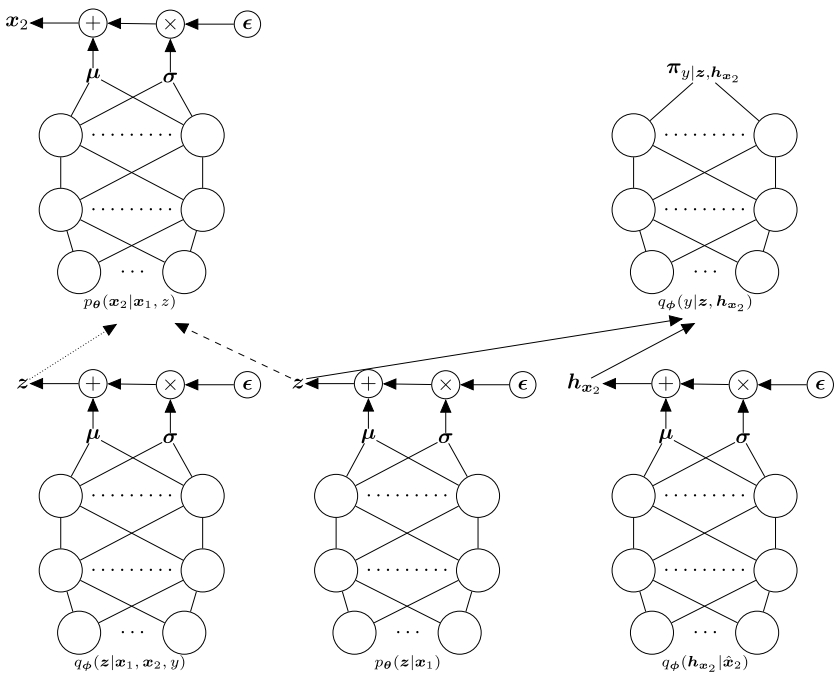


Fig. 3. Forward propagation in our proposed model. The dotted arrow indicates a forward pass during training, which is replaced by the dashed arrow at test time. Solid arrows depict a common forward propagation during training and test.

at test time and we only can draw $\boldsymbol{z}$ from the prior, i.e. we need to test the classifier using representations from the prior distribution. We hypothesize that the prior and posterior are characterized by different statistical properties, e.g. different kind of representation and correlational structure. Hence, the prior distribution reproduces the test scenario more accurately compared to the posterior distribution.[8] On the other hand, we generate $\boldsymbol{h}_{\boldsymbol{x}_2}$ from $q(\boldsymbol{h}_{\boldsymbol{x}_2}|\hat{\boldsymbol{x}}_2)$ at training and test time, drawing $\hat{\boldsymbol{x}}_2$ from the generative process $p(\boldsymbol{x}_2|\boldsymbol{x}_1,\boldsymbol{z})$. For clarity, Fig. 3 shows the forward propagation during training and test time in our proposed methodology.

Inspired by JMVAE, we tried to bring together the private latent representation $q(\boldsymbol{h}_{\boldsymbol{x}_2}|\boldsymbol{x}_2)$ and the joint representation $q(\boldsymbol{z}|\boldsymbol{x}_1,\boldsymbol{x}_2,y)$, but using MMD as divergence measure and the sampling approach described at the end of Section 3.1. While we do not see a clear benefit in the generative process of the model or in the predictive power of it, we see faster model convergence.

4.3. Experimental design

We use 70% of the data to learn a common data representation for both data modalities, which is further used to generate the modality $\boldsymbol{x}_2$ and to train a multilayer perceptron (MLP) classifier. Note that CBMD trains a classifier at the same time as it learns shared data representations and generates $\boldsymbol{x}_2$. For this 70% of the data, we down-sample the majority class ($y = 0$) to balance both class labels. Further, we use 25% of the data to test the predictive power of the classifier for all models and the quality of the reconstructed modality $\boldsymbol{x}_2$ using JMVAE, DCCAE and CBMD. Note that the test data set preserves the original balance between the two classes and that it is only used for testing purposes. Finally, we use the remaining 5% of the data to calibrate class probabilities using the beta calibration approach [33]. We report average values over 10 different runs.

[8] In [2] latent representations conditioned on the available modality at test time also give relatively stable performance.

4.3.1. Generating modality $\boldsymbol{x}_2$ — Lending club

Accuracy

Of all models tested in this research, only DCCAE, JMVAE and CBMD are able to generate modality $\boldsymbol{x}_2$ based on the available modality $\boldsymbol{x}_1$ during test time. Models with autoencoder-like architectures, e.g. VCCA or DMDGM, learn to reconstruct $\hat{\boldsymbol{x}}_2$ based on $\boldsymbol{x}_2$ and therefore cannot be used under the test scenario in this research. Note that both JMVAE and CBMD estimate a posterior distribution for modality $\boldsymbol{x}_2$. Hence, using a quadratic loss function $\mathcal{L} = (\boldsymbol{x}_2 - \hat{\boldsymbol{x}}_2)^2$ we obtain a point estimate $\hat{\boldsymbol{x}}_2^* = \arg\min \mathbb{E}[\mathcal{L} = (\boldsymbol{x}_2 - \hat{\boldsymbol{x}}_2)^2|\boldsymbol{x}_1,\boldsymbol{z}]$. Taking the first derivative of the expectation with respect to $\hat{\boldsymbol{x}}_2$ and forcing the result equal to 0, we obtain $\hat{\boldsymbol{x}}_2^* = \mathbb{E}[\boldsymbol{x}_2|\boldsymbol{x}_1,\boldsymbol{z}]$. This expectation is exactly what JMVAE and CBMD parametrise with MLPs (see Eq. (11)), and it is our choice for a point estimate in this section. On the other hand, DCCAE utilizes deterministic neural networks to generate $\boldsymbol{x}_2$, hence its output is a single point estimate. Note that to draw $\boldsymbol{x}_2$ with DCCAE, we use latent representations generated with $\boldsymbol{x}_1$.

Table 2 shows true and average and standard deviation values for all generated features in the modality $\boldsymbol{x}_2$ for the test data set. Interestingly, all models estimate highly accurate the support of the empirical distribution for each feature. However, JMVAE clearly fails at recognizing the dispersion in each feature. This results is most likely due to the information preference problem, meaning that $p(\boldsymbol{x}_2|\boldsymbol{z})$ is basically the same for all $\boldsymbol{z}$ [20]. Similarly, DCCAE does not match the empirical standard deviation for all features. On the other hand, our proposed CBMD model matches the variation for most of the features in modality $\boldsymbol{x}_2$. For features 1, 4, and 6, CBMD fails to capture the dispersion in the features. A possible solution to overcome this problem is to choose a model architecture for CBMD based only on its generative performance. Another alternative to further improve the generative process in CBMD is to use generative adversarial networks like in JMVAE.

Fig. 4 shows histograms for all true (solid curve) and generated features in modality $\boldsymbol{x}_2$. The generated features are depicted by the dashed and dotted curves, and the dotted vertical line, which correspond to CBMD, JMVAE, and DCCAE respectively. It is interesting to see that CBMD centers it mass in the main mode

Table 2
Average and standard deviation values for the true and reconstructed $\boldsymbol{x}_2$ features in the test data set using CBMD, JMVAE, and DCCAE. All models are able to capture the empirical mean for each feature. However, JMVAE and DCCAE fail at capturing the variation across different customers.

Feature name	True $\boldsymbol{x}_2$		CBMD $\hat{\boldsymbol{x}}_2$		JMVAE $\hat{\boldsymbol{x}}_2$		DCCAE $\hat{\boldsymbol{x}}_2$	
	Average	Std. deviation	Average	Std. deviation	Average	Std. deviation	Average	Std. deviation
Feature 1	0.17728	0.11037	0.18052	0.03246	0.18232	1.49011e−08	0.17613	0.01057
Feature 2	0.68461	0.15871	0.71402	0.14718	0.68072	1.19209e−07	0.70614	0.05208
Feature 3	0.74140	0.14333	0.75308	0.13515	0.74729	5.96046e−08	0.74132	0.02919
Feature 4	0.19370	0.08624	0.19388	0.03549	0.19306	1.48926e−08	0.16939	0.01109
Feature 5	0.46439	0.20650	0.41057	0.21793	0.46315	1.98431e−08	0.39902	0.09241
Feature 6	0.23878	0.12198	0.24361	0.03986	0.23649	1.49216e−08	0.21474	0.02407
Feature 7	0.20347	0.15674	0.20166	0.12713	0.20162	6.12372e−09	0.21303	0.03387
Feature 8	0.22988	0.19139	0.23108	0.16712	0.22704	1.49011e−08	0.23958	0.04155

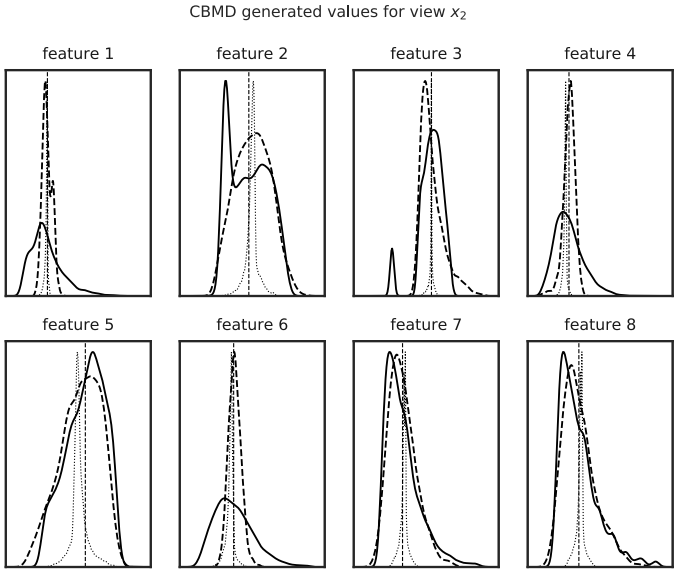


Fig. 4. Solid curves show the true empirical distributions for all features in modality $\boldsymbol{x}_2$ in the Lending Club test data set, while the dashed and dotted curves show the empirical distributions for the generated features using CBMD and DCCAE respectively. The dotted vertical line shows generated values using JMVAE.

of complex densities such as those for feature 2 and 3. Further, skewed densities like feature 5, 7, and 8 are reconstructed highly accurate. On the other hand, both JMVAE and DCCAE fail to capture the dispersion across different customers.

Ablation experiments

To further analyze the quality of the drawn $\boldsymbol{x}_2$ variables, we create 5 equally-sized groups with different risk profiles based on posterior class probabilities estimated with $q(y|\boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2})$. Group A has the lowest class posterior probability, while group E has the highest class posterior probability. Table 3 shows these 5 groups, together with true and generated average values for all features in the test data set. True values are shown in the first row for each group, while in the second and third row we generate $\boldsymbol{x}_2$ using the optimal ω^* value and $\omega = 1$ in our proposed objective function (Eq. (9)). The latter corresponds to the classical lower bound in deep generative models (Eq. (4)). We can see that for all groups, but A, the optimal ω^* value generates relatively more accurate features as suggested by the root mean squared error (rmse), showing the positive effect of optimizing the mutual information term in our proposed model. For some features in some groups the generated $\boldsymbol{x}_2$ values are highly accurate. Note that for group A the high rmse for ω^* is mainly driven by feature 2. The last row in each group, $\hat{\boldsymbol{x}}_2$ *CBM*(ω^*), shows the average of the generated features with our proposed model and with the optimal ω^* value, but without the **d**iscriminative model (hence the CBM name). We observe that the classifier in CBMD encourages the generative model to draw $\boldsymbol{x}_2$ accurately.

Table 3 shows from another perspective why models should not use fixed variance parameters in the generative process, as is the case for JMVAE and VCCA. Such a practice impedes a model to capture the variability among customers. Similarly, using deterministic neural networks to generate $\boldsymbol{x}_2$, as in DCCAE, makes it more challenging to capture the variation across customers.

The 5 groups that we created are shown in the left panel of Fig. 5, which are two-dimensional t-sne [34] components of the latent space $\boldsymbol{z} \sim p(\boldsymbol{z}|\boldsymbol{x}_1)$ for the test data set. Note that the 5 groups that we have defined are clustered in a well-defined structure with minimal overlap. Furthermore, the right panel of Fig. 5 shows a colormap of the same t-sne components where the color is given by the posterior class probability estimated using $q(y|\boldsymbol{z}, \boldsymbol{h}_{\boldsymbol{x}_2})$. Note that there is a smooth transition across the two dimensions. This is a characteristic of the learned latent space with deep generative models, which preserves the spatial coherence of creditworthiness [35].

Business application

Financial institutions use repayment data or behavior data, which is generated after $\boldsymbol{x}_1$ is obtained, for launching new products, cross-selling or marketing campaigns. This section presents an alternative approach where we use the modality $\boldsymbol{x}_1$ at test time and the generative process of the trained CBMD model of Section 4.2 to generate future data $\boldsymbol{x}_2$. To that end, we define *anchor customers*, which serve as point of reference.

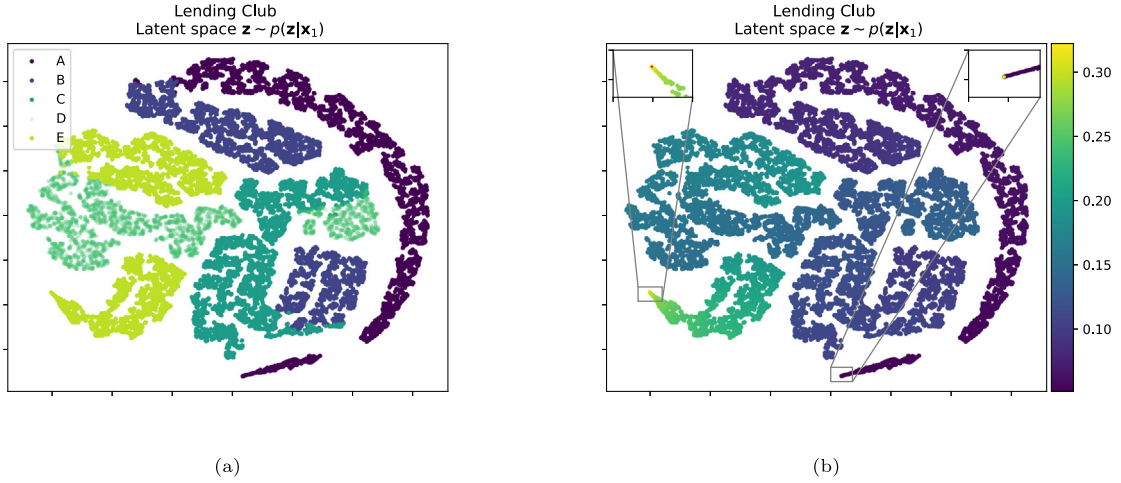
Suppose a bank wants to launch a new private loan for high-risk customers. At test time, we define as anchor customer the client in group E (the group with lowest creditworthiness) with the highest posterior class probability. This customer is depicted in the right panel of Fig. 5 by a red scatter point in the zoom box at the top-left corner. Then, we use $\boldsymbol{x}_1$ for the anchor customer to draw the latent representation $\boldsymbol{z} \sim p(\boldsymbol{z}|\boldsymbol{x}_1)$. Further, we use the generated representation $\boldsymbol{z}$ together with $\boldsymbol{x}_1$ to generate the future credit data using the generative process $p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{z})$.

Note that CBMD assumes per-observation latent variables and density functions, hence the Gaussian generative process for the i'th anchor customer is given by the distribution $\mathcal{N}(\boldsymbol{x}_2^{(i)}|\boldsymbol{x}_1^{(i)}, \boldsymbol{z}^{(i)}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)})$. That is, CBMD estimates a density function for the i'th anchor customer and hence we can draw several values for $\boldsymbol{x}_2$. At the bottom of Table 3, we show the average of 100 different $\boldsymbol{x}_2$ values. We can see that on average the anchor customer will have, just as expected, a low risk score (feature 3). Similarly, the bottom row in Table 3 shows average values for a different anchor customer, this time an anchor customer with the lowest class probability (see the yellow scatter in the zoom box at the top-right corner of Fig. 5). Note that instead of looking at average values for the i'th anchor customer, banks can utilize any value from the whole distribution, e.g. top or bottom quantiles, depending on the task at hand.

Another possibility to choose anchor customers is to select a set of customers within a given segment of interest. In this

Table 3
We use estimated class probabilities using CBMD to create 5 equally-sized groups (A–E). Further, we show true $\boldsymbol{x}_2$ average values and generated $\hat{\boldsymbol{x}}_2$ average values using our proposed lower bound and the classical lower bound denoted by ω^* and $\omega = 1$ respectively. The last column shows root mean squared error.

Group & model		Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7	Feature 8	rmse
A	true $\boldsymbol{x}_2$	0.1992	0.6304	0.8565	0.1531	0.2349	0.2222	0.2447	0.3129	
	$\hat{\boldsymbol{x}}_2(\omega^*)$	0.1904	0.7332	0.8291	0.1559	0.2171	0.2119	0.2352	0.3096	0.0386
	$\hat{\boldsymbol{x}}_2(\omega = 1)$	0.2134	0.6481	0.7885	0.1661	0.2600	0.2070	0.2398	0.2976	0.0284
	$\hat{\boldsymbol{x}}_2$ *CBM*(ω^*)	0.1938	0.6600	0.7845	0.1562	0.2466	0.2039	0.2300	0.2878	0.0304
B	true $\boldsymbol{x}_2$	0.1852	0.6242	0.8199	0.1602	0.4077	0.2142	0.2627	0.3239	
	$\hat{\boldsymbol{x}}_2(\omega^*)$	0.1830	0.6729	0.7515	0.1681	0.3627	0.2150	0.2466	0.3067	0.0348
	$\hat{\boldsymbol{x}}_2(\omega = 1)$	0.1997	0.6317	0.7260	0.1804	0.4090	0.2060	0.2465	0.2880	0.0373
	$\hat{\boldsymbol{x}}_2$ *CBM*(ω^*)	0.1754	0.6399	0.7108	0.1611	0.4011	0.19	0.2411	0.2767	0.0441
C	true $\boldsymbol{x}_2$	0.1826	0.6317	0.7973	0.1636	0.4764	0.2134	0.2674	0.3205	
	$\hat{\boldsymbol{x}}_2(\omega^*)$	0.1771	0.6342	0.7191	0.1743	0.4287	0.2147	0.2469	0.2952	0.0347
	$\hat{\boldsymbol{x}}_2(\omega = 1)$	0.1922	0.6274	0.6966	0.1859	0.4780	0.2047	0.2368	0.2669	0.0428
	$\hat{\boldsymbol{x}}_2$ *CBM*(ω^*)	0.1659	0.6299	0.6808	0.1619	0.4706	0.1816	0.2407	0.263	0.0487
D	true $\boldsymbol{x}_2$	0.1768	0.6362	0.7834	0.1694	0.5044	0.2140	0.2576	0.3016	
	$\hat{\boldsymbol{x}}_2(\omega^*)$	0.1682	0.6068	0.6967	0.1788	0.4766	0.2130	0.2392	0.2762	0.0359
	$\hat{\boldsymbol{x}}_2(\omega = 1)$	0.1855	0.6250	0.6733	0.1914	0.5293	0.2048	0.2264	0.2475	0.0467
	$\hat{\boldsymbol{x}}_2$ *CBM*(ω^*)	0.1581	0.6241	0.6608	0.1628	0.5208	0.1753	0.232	0.2441	0.0516
E	true $\boldsymbol{x}_2$	0.1663	0.6396	0.7646	0.1805	0.5322	0.2240	0.2318	0.2638	
	$\hat{\boldsymbol{x}}_2(\omega^*)$	0.1585	0.5829	0.6766	0.1890	0.5278	0.2212	0.2177	0.2406	0.0385
	$\hat{\boldsymbol{x}}_2(\omega = 1)$	0.1790	0.6240	0.6477	0.2071	0.5889	0.2151	0.2108	0.2203	0.0505
	$\hat{\boldsymbol{x}}_2$ *CBM*(ω^*)	0.1582	0.6229	0.6462	0.1728	0.5735	0.1825	0.2086	0.2105	0.0515
Highest $\boldsymbol{\pi}$		0.1310	0.5704	0.6526	0.2138	0.5884	0.2518	0.1280	0.1260	
Lowest $\boldsymbol{\pi}$		0.1798	0.6656	0.9791	0.1146	0.0122	0.1547	0.0348	0.0471	



(a) (b)

Fig. 5. Two-dimensional t-sne components of the latent space $\boldsymbol{z} \sim p(\boldsymbol{z}|\boldsymbol{x}_1)$ for the Lending Club test data set. The left panel shows the 5 groups that we created based on average values for posterior class probabilities, while the right panel shows a colormap of the same t-sne components where the color is given by the posterior class probability estimated by the CBMD model. Note the smooth transition across the two dimension.

case, it might be preferable to use the expectation for each $\boldsymbol{x}_2^{(i)}$ in the set of the selected customer, which is the $\boldsymbol{\mu}^{(i)}$ parameter in $\mathcal{N}(\boldsymbol{x}_2^{(i)}|\boldsymbol{x}_1^{(i)}, \boldsymbol{z}^{(i)}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)})$ and that has been parametrized by a neural network in CBMD. Given the flexibility of this approach to generate the future modality $\boldsymbol{x}_2$, anchor customers and their generated $\boldsymbol{x}_2$ data can be used as support when designing marketing campaigns, cross-selling strategies or launching new financial products. However, a detailed development of real-life strategies require access to more customer's information, e.g. target variables in actual cross-selling strategies, that we do not have at hand.

4.3.2. Classification using data representations

Even though the main motivation to include a classifier model in our proposed methodology is to generate accurate features in modality $\boldsymbol{x}_2$, Table 4 compares classification performance for all benchmark models in terms of AUC, GINI, and H-measure to provide different angles from which to examine the classification performance.[9] Given that the Santander Bank data set has 200 input features, we train all models in two different scenarios. In the first scenario $\boldsymbol{x}_1$ has 150 features and $\boldsymbol{x}_2$ has 50 features, while in the second scenario both modalities have 100 features. Model M-$\boldsymbol{x}_1$ provides a baseline for the traditional credit scoring approach where only $\boldsymbol{x}_1$ is used for training and testing.

Our experiments show that on average CCA-based models perform better for credit scoring than VCCA and JMVAE. This result has been explained in [1] and it happens when the modalities in the data sets are uncorrelated. Remember that the objective function in CCA-based models maximize canonical correlation. Further, it is interesting to see that both CCA and KCCA have slightly higher performance than the base model for the Lending Club data set. On the other hand, DCCA, DCCAE, and VCCA have the lowest model performance for the Lending Club data set.

[9] In credit scoring models, score-specific performance metrics, e.g. recall or precision, are not common to use since banks use the probabilities $\boldsymbol{\pi}_{y|\boldsymbol{z},\boldsymbol{h}_{\boldsymbol{x}_2}}$ to rank customers.

Table 4
The first model M-$\boldsymbol{x}_1$ uses only modality $\boldsymbol{x}_1$ to classify y with a MLP model. All CCA-based models, VCCA and JMVAE use shared data representations to classify y in a two-stage approach. On the other hand, our proposed CBMD model classifies labels in a unified framework. Average AUC, GINI, and H-measure are shown in the above table.

Model name	Lending Club ($\boldsymbol{x}_1$: 18 $\boldsymbol{x}_2$: 8)			Santander Bank ($\boldsymbol{x}_1$: 150 $\boldsymbol{x}_2$: 50)			Santander Bank ($\boldsymbol{x}_1$: 100 $\boldsymbol{x}_2$: 100)		
	AUC	GINI	H-measure	AUC	GINI	H-measure	AUC	GINI	H-measure
M-$\boldsymbol{x}_1$	0.61986	0.23972	0.04720	0.73844	0.47688	0.18509	0.63245	0.26490	0.06035
CCA	0.62004	0.24009	0.04733	0.73299	0.46597	0.17779	0.63141	0.26282	0.05919
KCCA	0.61996	0.23993	0.04684	0.74495	0.48989	0.19382	0.63152	0.26303	0.05822
DCCA	0.60783	0.21566	0.03787	0.74002	0.48004	0.18740	0.62420	0.24841	0.05246
DCCAE	0.60798	0.21597	0.03797	0.73756	0.47511	0.18273	0.62282	0.24564	0.05169
VCCA	0.60909	0.21818	0.04062	0.73621	0.47243	0.18211	0.63060	0.26120	0.05801
JMVAE	0.61920	0.23840	0.04654	0.68974	0.37948	0.11839	0.59354	0.18708	0.03200
CBMD	0.62049	0.24098	0.04764	0.74014	0.48028	0.18764	0.63395	0.26790	0.06146

However, DCCA achieves on-pair model performance compared to the baseline model for the Santander data set with 50 features in modality $\boldsymbol{x}_2$. It is important to note that [1] used pre-trained weights for DCCAE. We do not follow such practices to allow a fair comparison with CBMD. Hence, it might be possible to improve DCCAE performance by doing so.

Our proposed CBMD model performs slightly better than the baseline in all 3 experiments. Similarly, we also observe that CBMD achieves higher performance than most benchmark models. The only model with higher performance than CBMD is KCCA, which achieves the highest performance for the Santander data set with 150 features in modality $\boldsymbol{x}_1$. However, when we increase the number of features in $\boldsymbol{x}_2$ to 100, CBMD has a marginal improvement in performance compared to both KCCA and the baseline model. The fact that none of the benchmark models are able to achieve a significant improvement over the baseline, may suggest that the data modalities are not conditional independent given the data representations, which is an assumption in downstream classification tasks with multi-modal learning models [2].

It is important to mention that CBMD does not need to use fixed values for the variance parameters in the generative network $p(\boldsymbol{x}_2|\boldsymbol{x}_1, \boldsymbol{z})$, as opposed to VCCA and JMVAE, since CBMD is able to learn these parameters during the optimization procedure. It is also worth mentioning that we use the same model architecture and hyperparameter values in the experiment where both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ have 100 features as in the experiments where $\boldsymbol{x}_1$ has 150 features. If we tune the ω parameter in CBMD for the experiments with 100 features in both modalities, we obtain an average AUC of 0.63414. It would be interesting to see if pre-trained weights, as done in [2], can improve the classification performance. Likewise, adding dropout layers to the classifier might help to use representation from the posterior distribution $q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2, y)$ to train the classifier, which can lead to higher classification performance.

Finally, Fig. 6 shows average AUC as a function of ω in Eq. (9). A value $\omega = 1$ corresponds to the classical lower bound (Eq. (4)), while $\omega = 0$ maximizes mutual information $I_{q(\boldsymbol{x}_2,\boldsymbol{z}|\boldsymbol{x}_1)}(\boldsymbol{x}_2, \boldsymbol{z})$, and values between 0 and 1 maximize our proposed lower bound in Eq. (9). We can see that our proposed objective function achieves higher AUC compared to the classical lower bound, both for the Santander (solid lines) and Lending Club (dashed line) data sets. These results show that optimizing the mutual information term in our proposed model not only improves the generative process but also the classification performance.

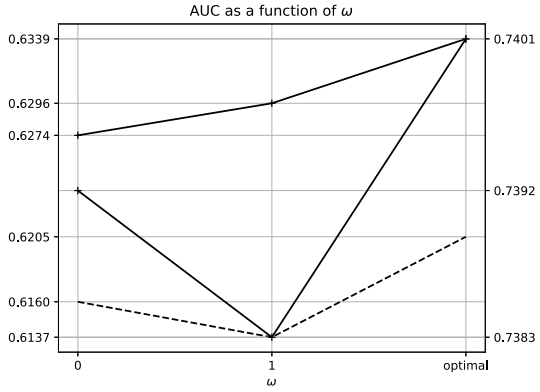


Fig. 6. Average AUC performance for the Santander (solid lines) and Lending Club (dashed line) data set. For $\omega = 1$ CBMD optimizes the classical lower bound in generative models, while $\omega = 0$ optimizes mutual information between $\boldsymbol{z}$ and $\boldsymbol{x}_2$. For ω values in between, CBMD optimizes the lower bound introduced in this paper. Note that the *optimal* ω value, 0.8 for Lending club and 0.05 for Santander data set, achiever AUC.

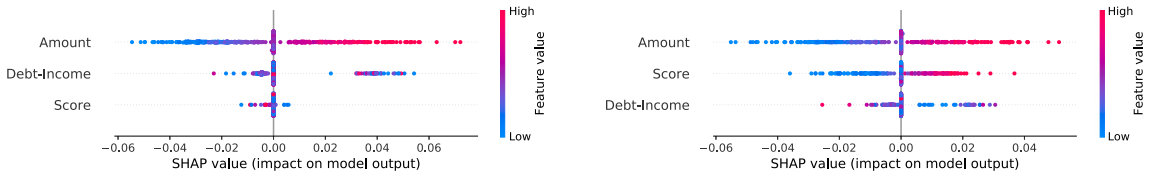
5. Model interpretability

Model performance in advanced deep generative models, like CBMD, comes at the cost of model interpretability. Fortunately, in the last decade, there has been an increasing interest in designing approaches to explain these advanced models. [36] introduce a unified approach, Shapley Additive Explanations (SHAP), for interpreting any model prediction. The SHAP values for a given feature is the average expected marginal contribution of this feature after all possible feature combinations have been considered. Hence, it considers both the effect the feature has by itself and in combination with the other features in the model. SHAP values offer an intuitive approach for model interpretability, providing useful insight to understand the CBMD output. Such information is valuable in different applications such as credit scoring, healthcare, and insurance to name a few.

We estimate SHAP values for the Lendig Club data set using the Kernel SHAP method introduced in [36] and utilize the python library developed by the same authors.[10] Fig. 7 shows SHAP values[11] for the classification prediction (a panel) and the generative process for the feature revolving utilization in the modality $\boldsymbol{x}_2$ (b panel). Each point in each of the two panels is the SHAP value for one specific feature and one specific observation in a subsample of the test data set. The x-axis shows the SHAP values and the color represents values of the feature from low to high. As can be seen from the figure, the loan amount has the largest impact on the classification performance and generative process of CBMD. Larger loan amounts increase both the output of the classifier and the revolving utilization of the credit line. There is no clear pattern on the effect of debt-to-income and bureau score values for the classification prediction. This result implies that both features reflect customers creditworthiness at the time

[10] https://github.com/slundberg/shap.

[11] We only plot SHAP values for the continuous features in modality $\boldsymbol{x}_1$ and not for the ones converted to one-hot-encoding. SHAP values for the rest of features in $\boldsymbol{x}_2$ can be found in Appendix C.



(a) Classification. (b) Revolving utilization feature in $\boldsymbol{x}_2$.

Fig. 7. SHAP values for the classification output in CBMD (a panel) and for the CBMD generated revolving utilization feature in modality $\boldsymbol{x}_2$ (b panel).

of application and can deteriorate with time. On the other hand, bureau score is the second most import feature for generating the revolving utilization feature. The higher the score, the larger revolving utilization amount.

6. Conclusion

In this research, we develop a novel conditional bi-modal discriminative (CBMD) model that learns a joint representation $\boldsymbol{z}$ and generates the modality $\boldsymbol{x}_2$ conditioned on data representations and the modality $\boldsymbol{x}_1$, which is our best source of information about future customer behavior. CBMD is not only able to generate $\boldsymbol{x}_2$ but also can classify the outcome of loans using the joint representations $\boldsymbol{z}$. Further, its generative process keeps the relationship between the modalities $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ for each customer and it is useful in scenarios where only one modality is available at test time. We show, under a simple scenario, the potential use of CBMD in launching new products. With access to the right data, CBMD can be used to design effective real-life cross-selling and marketing strategies or to analyze the difference in default probabilities by incorporating future behavior data.

Our proposed CBMD model optimizes a novel objective function that maximizes mutual information between the latent data representation $\boldsymbol{z}$ and the modality $\boldsymbol{x}_2$. This loss function learns an amortized inference distribution for $q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2, y)$, which contributes to an efficient generative model for $\boldsymbol{x}_2$. Therefore, we do not need to fix the variance parameters in the generative process as VCCA and JMVAE do. To further improve the generative process, we introduce a classifier model that encourages the generative model to draw $\boldsymbol{x}_2$ accurately. Our empirical results suggest that including the classification loss and the mutual information term in the objective function effectively improve the accuracy of generated features in $\boldsymbol{x}_2$. Finally, our proposed objective function also achieves higher AUC compared to the classical lower bound in generative models.

To the best of our knowledge, this research presents the first credit scoring model based on bi-modal learning able to generate future credit data $\boldsymbol{x}_2$ and therefore it opens an interesting avenue for future research. Likewise, our proposed methodology offers new possibilities on how banks could implement the use of generated $\boldsymbol{x}_2$ values in their activities that involve the prediction of customer's credit behavior.

CRediT authorship contribution statement

Rogelio A. Mancisidor: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Michael Kampffmeyer:** Conception and design of study, Writing – original draft, Writing – review & editing. **Kjersti Aas:** Conception and design of study, Writing – original draft, Writing – review & editing. **Robert Jenssen:** Conception and design of study, Writing – original draft, Writing – review & editing.

Table A.1
Lending Club data modalities.

	Variable name
Modality $\boldsymbol{x}_1$	Loan amount
	Fico score
	Address state
	Debt to income ratio
	Employment length
Modality x_2	(feature 1) days_earliest_cr_line
	(feature 2) days_last_pymnt_d
	(feature 3) last_risk_score
	(feature 4) open_acc
	(feature 5) revolv_util
	(feature 6) total_acc
	(feature 7) total_pymnt
	(feature 8) total_rec_prncp

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank Santander Consumer Bank Nordics and the Research Council of Norway [grant number 260205 and 276428] for financial support for this research. All authors approved the version of the manuscript to be published.

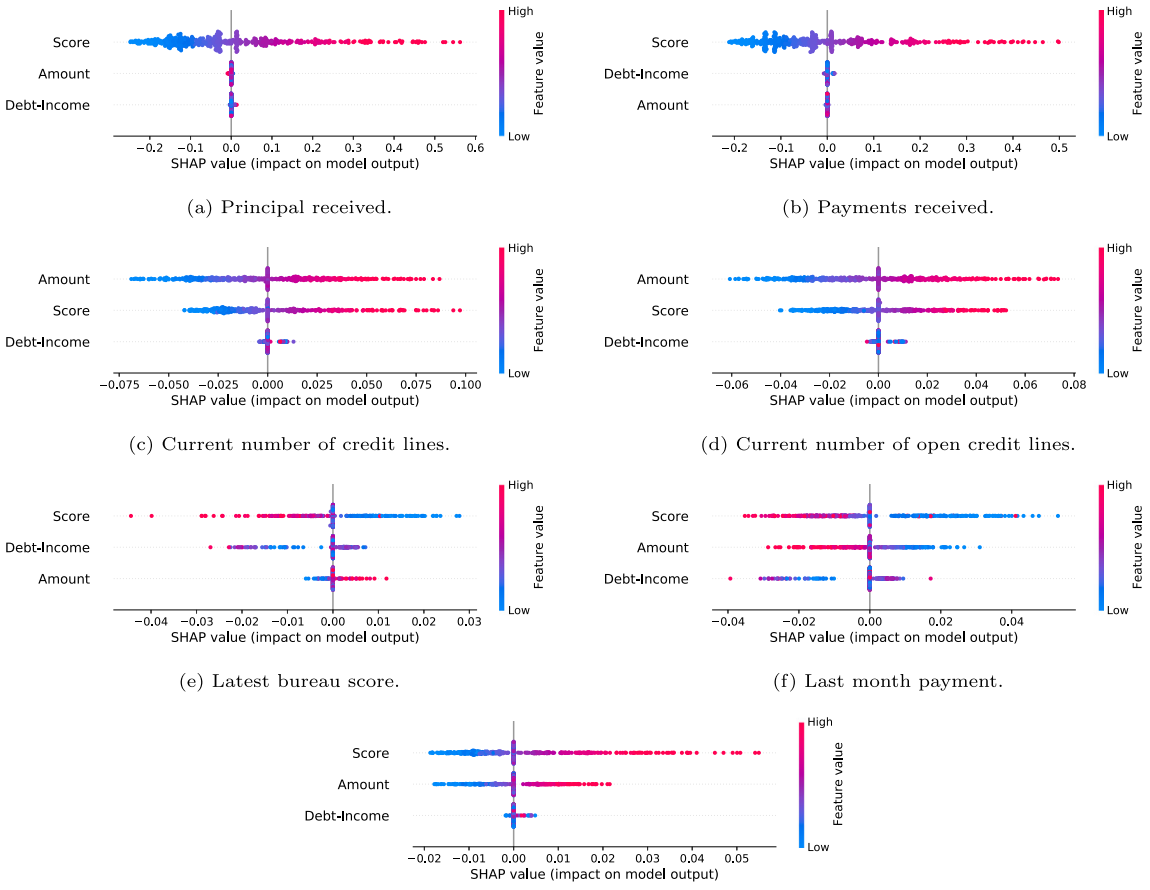
Appendix A. Data sets

We select modality $\boldsymbol{x}_1$ for the Lending Club data set using the common features for accepted and rejected applications, since this is the case in real loan application process. These features are loan amount, Fico scores, address state, debt to income ratio, and employment length. Further, we follow the practice as in [27,31] and create 4 different groups using address state, which are further transformed to one-hot encoders. Similarly, given that employment length has 11 different categories, we also convert it to one-hot encoders. Therefore, modality $\boldsymbol{x}_1$ has 18 features.

From the remaining 72 features for accepted applications, we select those variables whose empirical distribution resembles a Gaussian density. Remember that our proposed CBMD model assumes a multivariate Gaussian distribution for modality $\boldsymbol{x}_2$. Given that we only have 5 original features for modality $\boldsymbol{x}_1$, we select 8 features for modality $\boldsymbol{x}_2$ and can be found in Table A.1.

Appendix B. Model architectures

Table B.1 shows all architectures tested for hyperparameter optimization for our proposed CBMD model, JMVAE, VCCA, and DMDGM model. We use the notation for CBMD to specify the different MLP networks, but all models have a similar network just with different inputs. For example, JMVAE uses $q(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)$ as inference network. For models with two inference or generative networks, e.g. JMVAE has $p(x_1|\boldsymbol{z})$ and $p(x_2|\boldsymbol{z})$, we use the same architecture for both networks.



(a) Principal received.

(b) Payments received.

(c) Current number of credit lines.

(d) Current number of open credit lines.

(e) Latest bureau score.

(f) Last month payment.

(g) Earliest reported open credit line.

Fig. C.1. SHAP values for the CBMD generated features in modality $\boldsymbol{x}_2$.

Table B.1
Grid for hyperparameter optimization for CBMD, JMVAE, and VCCA. The numbers within brackets specify the number of neurons in each hidden layers, i.e. [10 10] means two hidden layers with 10 neurons each. Superscripts *, **, *** show the final architecture for CBMD, JMVAE, and VCCA, respectively.

Lending Club	
MLP network	Number of hidden layers and dimensions
$p(\boldsymbol{x}_2\|\boldsymbol{x}_1,\boldsymbol{z})$	[50], [60], [70], [80], [100], [120], [150], [200] [50 50], [60 60], [70 70], [80 80], [100 100]***, [120 120], [150 150]**, [200 200]*, [50 50 50], [60 60 60], [70 70 70], [80 80 80], [100 100 100], [120 120 120], [150 150 150], [200 200 200]
$p(\boldsymbol{z}\|\boldsymbol{x}_1)$	[20], [30], [40], [50], [60], [70], [80], [100]*, [120], [150]
$q(\boldsymbol{z}\|\boldsymbol{x}_1,\boldsymbol{x}_2,y)$	[40]***,[50], [60], [70], [80], [100]*,**, [120], [150], [200] [50 50], [60 60], [70 70], [80 80], [100 100], [120 120], [150 150], [200 200], [50 50 50], [60 60 60], [70 70 70], [80 80 80], [100 100 100], [120 120 120], [150 150 150], [200 200 200]
$q(y\|\boldsymbol{z})$	[50], [60], [70],[80], [100], [120], [150], [50 50], [60 60], [70 70],[80 80], [100 100]*, [120 120], [150 150]
Parameter/hyperparameter	Value
$\boldsymbol{z}$ dimension	10, 20, 30, 40, 50*,**,***,****, 70, 90, 110, 130, 150, 170
ω	0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8*, 0.9, 1
λ	1000, 2000, 3000, 4000*
α	1, 5, 10, 15, 20*,****, 30, 40, 50
Santander Bank	
$p(\boldsymbol{x}_2\|\boldsymbol{x}_1,\boldsymbol{z})$	[100 100], [200 200], [300 300], [500 500], [700 700], [900 900], [100 100 100],[200 200 200], [300 300 300],[500 500 500], [700 700 700], [900 900 900]*
$p(\boldsymbol{z}\|\boldsymbol{x}_1)$	[100], [200], [300]*, [400], [500]
$q(\boldsymbol{z}\|\boldsymbol{x}_1,\boldsymbol{x}_2,y)$	[100 100], [200 200], [300 300], [500 500], [700 700], [900 900], [100 100 100], [200 200 200], [300 300 300],[500 500 500], [700 700 700]*, [900 900 900]
$q(y\|\boldsymbol{z})$	[100], [200], [300], [400], [500], [700]*, [900]
Parameter/hyperparameter	Value
$\boldsymbol{z}$ dimension	100, 200, 300, 400*,**,***, 500, 600, 700, 800, 900
ω	0, 0.1*, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1
λ	1000, 2000, 3000, 4000*
α	1, 5, 10, 15, 20*,***, 30, 40, 50

Table B.2
Average root mean squared errors between the generated features by CBMD and their true values in the test data set. We generate the features in the modality $\boldsymbol{x}_2$ using the tanh, sigmoid and softplus activation function in the hidden layers of CBMD. The last row shows the AUC in the test data set for each of the activation functions.

Group	Activation function		
	tanh	sigmoid	softplus
A	0.2396	0.6845	0.0285
B	0.2224	0.5014	0.0252
C	0.2142	0.4286	0.0248
D	0.2104	0.4028	0.0275
E	0.2046	0.3969	0.0310
	Classification performance		
	tanh	sigmoid	softplus
AUC	0.6190	0.6147	0.6229

Activation function. Table B.2 shows the 5 different groups created in Section 4.3.1 and the average root mean squared errors between the generated features by CBMD and their true values in the test data set. We generate features using the tanh, sigmoid and softplus activation function in the hidden layers of our proposed CBMD model. Further, the last row in Table B.2 shows the AUC for each activation function. Note that the rest of hyperparameters are the same as in Table A.1.

Appendix C. SHAP values

Fig. C.1 shows SHAP values for the generative process in CBMD. Each point is the SHAP value for one specific feature and one specific observation in a sub-sample of the test data set for Lending Club. The x-axis shows the SHAP values and the color represents values of the feature from low to high.

References

[1] Weiran Wang, Raman Arora, Karen Livescu, Jeff Bilmes, On deep multi-view representation learning, in: International Conference on Machine Learning, 2015, pp. 1083–1092.
[2] Weiran Wang, Xinchen Yan, Honglak Lee, Karen Livescu, Deep variational canonical correlation analysis, 2016, arXiv preprint arXiv:1610.03454.
[3] Fang Du, Jiangshe Zhang, Junying Hu, Rongrong Fei, Discriminative multi-modal deep generative models, Knowl.-Based Syst. 173 (2019) 74–82.
[4] Harold Hotelling, Relations between two sets of variates, Biometrika 28 (3/4) (1936) 321–377.
[5] David R. Hardoon, Sandor Szedmak, John Shawe-Taylor, Canonical correlation analysis: An overview with application to learning methods, Neural Comput. 16 (12) (2004) 2639–2664.
[6] Shotaro Akaho, A kernel method for canonical correlation analysis, 2006, ArXiv Preprint Cs/0609071.
[7] Pei Ling Lai, Colin Fyfe, Kernel and nonlinear canonical correlation analysis, Int. J. Neural Syst. 10 (05) (2000) 365–377.
[8] Thomas Melzer, Michael Reiter, Horst Bischof, Nonlinear feature extraction using generalized canonical correlation analysis, in: International Conference on Artificial Neural Networks, Springer, 2001, pp. 353–360.
[9] David Lopez-Paz, Suvrit Sra, Alex Smola, Zoubin Ghahramani, Bernhard Schölkopf, Randomized nonlinear component analysis, in: International Conference on Machine Learning, 2014, pp. 1359–1367.
[10] Francis R. Bach, Michael I. Jordan, A Probabilistic Interpretation of Canonical Correlation Analysis, Technical Report 688, Department of Statistics UC, Berkeley, 2005.
[11] Arthur P. Dempster, Nan M. Laird, Donald B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. Ser. B Stat. Methodol. 39 (1) (1977) 1–22.
[12] Ronald A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. 7 (2) (1936) 179–188.
[13] Masahiro Suzuki, Kotaro Nakayama, Yutaka Matsuo, Joint multimodal learning with deep generative models, 2016, arXiv preprint arXiv:1611.01891.
[14] Mike Wu, Noah Goodman, Multimodal generative models for scalable weakly-supervised learning, in: Advances in Neural Information Processing Systems, 2018, pp. 5575–5585.
[15] Changde Du, Changying Du, Hao Wang, Jinpeng Li, Wei-Long Zheng, Bao-Liang Lu, Huiguang He, Semi-supervised deep generative modelling of incomplete multi-modality emotional data, in: 2018 ACM Multimedia Conference on Multimedia Conference, ACM, 2018, pp. 108–116.
[16] Ramakrishna Vedantam, Ian Fischer, Jonathan Huang, Kevin Murphy, Generative models of visually grounded imagination, 2017, arXiv preprint arXiv:1705.10762.
[17] Galen Andrew, Raman Arora, Jeff Bilmes, Karen Livescu, Deep canonical correlation analysis, in: International Conference on Machine Learning, 2013, pp. 1247–1255.
[18] Kihyuk Sohn, Wenling Shang, Honglak Lee, Improved multimodal deep learning with variation of information, in: Advances in Neural Information Processing Systems, 2014, pp. 2141–2149.
[19] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, Pieter Abbeel, Infogan: Interpretable representation learning by information maximizing generative adversarial nets, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 2180–2188.
[20] Shengjia Zhao, Jiaming Song, Stefano Ermon, Infovae: Information maximizing variational autoencoders, 2017, arXiv preprint arXiv:1706.02262.
[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[22] Diederik P. Kingma, Max Welling, Auto-encoding variational bayes, 2013, arXiv preprint arXiv:1312.6114.
[23] Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, Pieter Abbeel, Variational lossy autoencoder, 2016, arXiv preprint arXiv:1611.02731.
[24] James Lucas, George Tucker, Roger B Grosse, Mohammad Norouzi, Don't blame the elbo! a linear vae perspective on posterior collapse, Adv. Neural Inf. Process. Syst. 32 (2019) 9408–9418.
[25] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, Alex J Smola, A kernel method for the two-sample-problem, in: Advances in Neural Information Processing Systems, 2007, pp. 513–520.
[26] The Theano Development Team, Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, Anatoly Belikov, et al., Theano: A python framework for fast computation of mathematical expressions, 2016, arXiv preprint arXiv:1605.02688.
[27] Rogelio A Mancisidor, Michael Kampffmeyer, Kjersti Aas, Robert Jenssen, Deep generative models for reject inference in credit scoring, Knowl.-Based Syst. (2020) 105758.
[28] David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams, Learning Internal Representations by Error Propagation, Technical Report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
[29] Raymond Anderson, The Credit Scoring Toolkit: Theory and Practice for Retail Credit Risk Management and Decision Automation, Oxford University Press, 2007.
[30] Pierre Geurts, Damien Ernst, Louis Wehenkel, Extremely randomized trees, Mach. Learn. 63 (1) (2006) 3–42.
[31] Zhiyong Li, Ye Tian, Ke Li, Fanyin Zhou, Wei Yang, Reject inference in credit scoring using semi-supervised support vector machines, Expert Syst. Appl. 74 (2017) 105–114.
[32] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[33] Meelis Kull, Telmo Silva Filho, Peter Flach, Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers, in: Artificial Intelligence and Statistics, 2017, pp. 623–631.
[34] Geoffrey E. Hinton, Sam T. Roweis, Stochastic neighbor embedding, in: Advances in Neural Information Processing Systems, 2003, pp. 857–864.
[35] Rogelio A Mancisidor, Michael Kampffmeyer, Kjersti Aas, Robert Jenssen, Learning latent representations of bank customers with the variational autoencoder, Expert Syst. Appl. (2020) 114020.
[36] Scott M. Lundberg, Su-In Lee, A unified approach to interpreting model predictions, Adv. Neural Inf. Process. Syst. 30 (2017).