Learning Latent Representations of Bank Customers With The Variational Autoencoder

Rogelio A. Mancisidor[a,b,*] rogelio.a.mancisidor@uit.no
Michael Kampffmeyer [a] michael.c.kampffmeyer@uit.no
Kjersti Aas [c] kjersti@nr.no
Robert Jenssen [a] robert.jenssen@uit.no

[a]Machine Learning Group, Department of Physics and Technology, Faculty of Science and Technology, UiT - The Artic University of Norway, Hansine Hansens veg 18, Tromsø 9037, Norway
[b]Credit Risk Models, Santander Consumer Bank AS, Strandveien 18, Lysaker 1325, Norway
[c]Statistical Analysis, Machine Learning and Image Analysis
Norwegian Computing Center, Gaustadalleen 23a, Oslo 0373, Norway

*Corresponding author

August 19, 2020

Abstract. Learning data representations that reflect the customers' creditworthiness can improve marketing campaigns, customer relationship management, data and process management or the credit risk assessment in retail banks. In this research, we show that it is possible to steer data representations in the latent space of the Variational Autoencoder (VAE) using a semi-supervised learning framework and a specific grouping of the input data called Weight of Evidence (WoE). Our proposed method learns a latent representation of the data showing a well-defied clustering structure. The clustering structure captures the customers' creditworthiness, which is unknown a priori and cannot be identified in the input space. The main advantages of our proposed method are that it captures the natural clustering of the data, suggests the number of clusters, captures the spatial coherence of customers' creditworthiness, generates data representations of unseen customers and assign them to one of the existing clusters. Our empirical results, based on real data sets reflecting different market and economic conditions, show that none of the well-known data representation models in the benchmark analysis are able to obtain well-defined clustering structures like our proposed method. Further, we show how banks can use our proposed methodology to improve marketing campaigns and credit risk assessment.

Keywords: Variational Autoencoder, Data Representations, Clustering, Machine Learning

1 Introduction

Banks need to estimate the creditworthiness of both customers and applicants to improve marketing campaigns, customer relationship management, data and process management or the credit risk assessment (Anderson, 2007). Further, Anderson (2007) suggests that customer segmentation can improve

the aforementioned bank activities. Therefore, it is important to learn a data representation of bank customers that has the ability to express the natural clustering of the data, and that can be used in marketing campaigns, product offering or in improving the credit risk assessment.

The Variational Autoencoder (VAE) (Kingma and Welling, 2013; Rezende et al., 2014) has shown promising results in different research domains. The powerful information embedded in its latent space has been documented e.g., in health analytics (Rampasek and Goldenberg, 2017; Titus et al., 2018; Way and Greene, 2017a,b), in speech emotion recognition (SER) (Latif et al., 2017), and in natural language processing (NLP) (Bowman et al., 2015; Su et al., 2018), among others. Additionally, research has been conducted where the VAE has been modified to improve its feature learning properties, e.g. Bouchacourt et al. (2018); Higgins et al. (2017); Hsu et al. (2017); Su et al. (2018). However, to the best of our knowledge there is no previous work on data representations of bank customers using a modified version of the VAE and therefore this is the first research to focus on the development of a data representation framework suitable for the bank industry.

Inspired by the previous results in other research fields and the lack of research on learning data representations of bank customers, we adopt the VAE and the Auto Encoding Variational Bayesian (AEVB) algorithm (Kingma and Welling, 2013) and propose a new framework that effectively learns a data representation that is useful to support the aforementioned banking activities. Our proposed method is able to steer the latent embeddings in the VAE by transforming the input data into a meaningful representation, and by creating a specific grouping of the data. Hence, the focus in this research is use the effective manifold learning capabilities of the VAE (Goodfellow et al., 2016) and develop a new framework which is able to capture valuable information in the latent space for bank activities.

The main advantage of our method is that it learns a data representation in the low-dimensional latent space generated by the VAE, which can be visualized and suggests well-defined clustering structures. Therefore, our method reveals the number of groups in the natural clustering structure of the data. Further, these clusters are well suited for the bank industry given that they encapsulate different risk profiles, which are unknown a priori and cannot be identified in the input space that is high-dimensional and with complex relationships. In addition, the latent representations not only encapsulate creditworthiness, but also preserve its spatial coherence. Using the generative properties of the VAE, we can draw the latent space of unseen customers and map them into an existing cluster without the need of further supervision. Finally, our empirical results, based on real data sets reflecting different market and economic conditions, show that the data representations obtained with our proposed method are able to obtain well-defined clustering structures capturing the customers' creditworthiness, unlike some well-known data representation models, and we also show how banks can use our proposed methodology to improve marketing campaigns and credit risk assessment.

This paper is organized as follows. Section 2 reviews the related work where the VAE has been used to learn data representations in different research fields, while Section 3 introduces variational inference and the VAE. In Section 4 we explain the data transformation used to learn latent representations of bank customers and Section 5 presents our experiments and findings. Finally, Section 7 presents the main conclusions in this paper.

2 Related Work

Methods to learn data representations from the input data can be divided into probabilistic graphical models (PGMs) and neural network-based models (Bengio et al., 2013). Data representations play an important role in the results we can achieve in detection or classification tasks (Bengio et al., 2013; LeCun et al., 2015; Zhong et al., 2016). The ability to express general-purpose priors, such as natural clustering or spatial coherence, among others, is what make data representations to be good (Bengio et al., 2013).

Further, PGMs aim to learn latent representations $\boldsymbol{z}$, which are able to describe the input data $\boldsymbol{x}$. This is done by modelling their joint distribution $p(\boldsymbol{z}, \boldsymbol{x})$. Depending on how this joint distribution is constructed, PGMs can be divided in directed or undirected graphical models (Bengio et al., 2013).

The Variational Autoencoder (Kingma and Welling, 2013; Rezende et al., 2014) is an influential (unsupervised) directed probabilistic graphical model, which has been widely used to learn meaningful latent representations of the input data. For example, latent representations of gene expression data are used in Way and Greene (2017a) for cancer prediction. The results show that the VAE latent features are useful to predict cancer and its predictive power is similar to other data transformation methods, e.g. principal component analysis (PCA) (Pearson, 1901).

Latent representations in the VAE have also been used for predictions in a semi-supervised context. In Rampasek and Goldenberg (2017), latent representations for pre-treatment and post-treatment gene expression are use to predict drug response. Their proposed model achieves higher performance relative to Ridge logistic regression (Hoerl and Kennard, 1970) using the original input data. In addition, PCA transformations are used in three different classifiers to predict drug responses, but their performance, in most of the experiments, is not better relative to Ridge regression and the VAE model.

Classification of speech emotion is another example where latent representations of the input data have been successfully used for classification. Using Long Short Term Memory (LSTM) networks to classify emotion, Latif et al. (2017) compare the predictive power of data transformations using the VAE and regular Auto Encoders. Speech emotion prediction is more accurate when the latent representations in the VAE are used as predictors. The classification results are further improved by using latent representations obtained with conditional VAE (Sohn et al., 2015).

In another classification study, Titus et al. (2018) train logistic regression models, on t-SNE (Hinton and Roweis, 2003) embeddings of high-dimensional VAE latent variables, to classify tumours. Their results show that the latent embeddings in the VAE learn a biological relevant information and successfully classify disease sub-types. Both works in Latif et al. (2017); Titus et al. (2018) build upon the *Tybalt* model (Way and Greene, 2017b). The Tybalt exploits the data transformation capabilities of the VAE to generate latent representations of gene expression data.

The VAE has also been used in the natural language processing field. Studying bilingual word embeddings, Su et al. (2018) use the VAE to generate latent representations, which explicitly induce the underlying semantics of bilingual text. Their model is able to learn a hidden representation of paired bag-of-words sentences. Furthermore, in Bowman et al. (2015) recurrent neural networks are combined with the VAE to model text data. The latent transformations are able to generate coherent sentences. In addition, the proposed model in this research is able to impute missing words in text corpus.

Research has also been conducted on modifying the original VAE aiming to improve the quality of the learned latent representations. In Higgins et al. (2017), for example, the authors add an hyperparamter β to the VAE, which limits the capacity of the latent information channel and impose an emphasis on learning statistically independent latent factors. Hence, the model is able to learn disentangled factors of variation.

In Bouchacourt et al. (2018) the concept of supervision in VAE is introduced. The authors group the input data, aiming to learn representations of the data that reflect the semantics behind a specific grouping of the data. In other words, the grouping makes it possible to learn a semantically useful data transformation. Similarly, Hsu et al. (2017); Su et al. (2018) use supervision but in the latent space. Both works Hsu et al. (2017); Su et al. (2018), manipulate the latent representations arithmetically to decompose the latent representation into different attributes.

There has been some work on segmentation in both the financial industry and in the marketing area. Hand et al. (2005) studies whether credit risk assessment can be improved by creating segments, which are created using a bipartite model. Bijak and Thomas (2012) use decision trees and chi-squared automation interaction detection trees for identifying customer segments. Further, they proposed a unified framework where segmentation and credit risk assessment is optimized simultaneously. Aurifeille (2000) uses a genetic algorithm and linear regressions to identify clusters in a marketing data set. More recently, Xiao et al. (2016) use an ensemble approach where k-means is used for segmentation. Similarly, Lim and Sohn (2007) use k-means to develop a dynamic model for credit risk assessment. However, the focus of the research in this paper is on representation learning of bank customers and not on coupling existing clustering and classification techniques.

In this research, as in Bouchacourt et al. (2018); Hsu et al. (2017) and Su et al. (2018), we introduce a

supervision stage into the VAE. In this stage, we form groups that share a common factor of variation. The difference in our method is that the grouping is derived from the class label, see Section 4. This means that our proposed method is a semi-supervised representation learning model where we indirectly steer the data transformation using a specific grouping of the input data. Finally, we only focus on learning a data representation of bank customers' data that is able to capture the customers' creditworthiness in the latent space of the VAE, and not in the predictive power of such representations.

3 The Variational Autoencoder

3.1 Variational Inference

In the rest of the paper we use the following notation. We consider i.i.d. data $\{\boldsymbol{x}_i\}_{i=1}^n$ where $\boldsymbol{x}_i \in \mathbb{R}^{d_x}$ is the customers data, e.g. income, age, marital status etc. Further, the latent variables $\{\boldsymbol{z}_i\}_{i=1}^n$ where $\boldsymbol{z}_i \in \mathbb{R}^{d_z}$ are the data transformation of $\boldsymbol{x}_i$. The subscript i is dropped whenever the context allows for it.

The latent variable in the joint density $p(\boldsymbol{x}, \boldsymbol{z})$ is drawn from a prior density $p(\boldsymbol{z})$ and then it is linked to the observed data through the likelihood $p(\boldsymbol{x}|\boldsymbol{z})$. Inference amounts to conditioning on data and computing the posterior $p(\boldsymbol{z}|\boldsymbol{x})$ (Blei et al., 2017).

The problem is that the posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$ is intractable in most cases. Note that

$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{z}, \boldsymbol{x})}{p(\boldsymbol{x})}, \tag{1}$$

involves the marginal distribution $p(\boldsymbol{x}) = \int p(\boldsymbol{z}, \boldsymbol{x}) d\boldsymbol{z}$. This integral, called the *evidence*, in some cases requires exponential time to be evaluated since it considers all configurations of latent variables. In other instances, it is unavailable in a closed form (Blei et al., 2017).

Variational Inference (VI) copes with this kind of problem by minimizing the Kullback-Leibler (KL) divergence[1] between the true posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$ and a parametric function $q(\boldsymbol{z}|\boldsymbol{x})$, which is chosen among a set of densities $\Im$ (Blei et al., 2017). This set of densities is parameterized by *variational parameters* and they should be flexible enough to capture a density close to $p(\boldsymbol{z}|\boldsymbol{x})$ and, in addition, be simple for efficient optimization. The parametric density which minimizes the KL divergence is

$$q^*(\boldsymbol{z}|\boldsymbol{x}) = \underset{q(\boldsymbol{z}|\boldsymbol{x}) \in \Im}{\arg\min}\, KL[q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|\boldsymbol{x})]. \tag{2}$$

Unfortunately, Equation 2 cannot be optimized directly since it requires computing a function of $p(\boldsymbol{x})$. To see this, let us expand the KL divergence using the Bayes' theorem and noting that $p(\boldsymbol{x})$ does not depend on $\boldsymbol{z}$

$$\begin{aligned} KL[q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|\boldsymbol{x})] =& E_{\boldsymbol{z}\sim q}[\log q(\boldsymbol{z}|\boldsymbol{x}) - \log p(\boldsymbol{z}|\boldsymbol{x})] \\ =& E_{\boldsymbol{z}\sim q}[\log q(\boldsymbol{z}|\boldsymbol{x}) - \log p(\boldsymbol{x}, \boldsymbol{z})] + \log p(\boldsymbol{x}). \end{aligned} \tag{3}$$

Given that Equation 3 cannot be optimized directly, VI optimizes the alternative objective function

$$\begin{aligned} E_{\boldsymbol{z}\sim q}[\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{x})] =& E_{\boldsymbol{z}\sim q}[\log p(\boldsymbol{z}) + \log p(\boldsymbol{x}|\boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{x})] \\ =& E_{\boldsymbol{z}\sim q}[\log p(\boldsymbol{x}|\boldsymbol{z})] - KL[q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})] \\ =& ELBO. \end{aligned} \tag{4}$$

From Equations 3 and 4 we have that

$$\log p(\boldsymbol{x}) = KL[q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|\boldsymbol{x})] + ELBO. \tag{5}$$

[1] The KL divergence $KL[q(\cdot)||p(\cdot)]$ is a measure of the proximity between two densities and it is commonly measured in bits. It is non-negative and it is minimized when $q(\cdot) = p(\cdot)$.

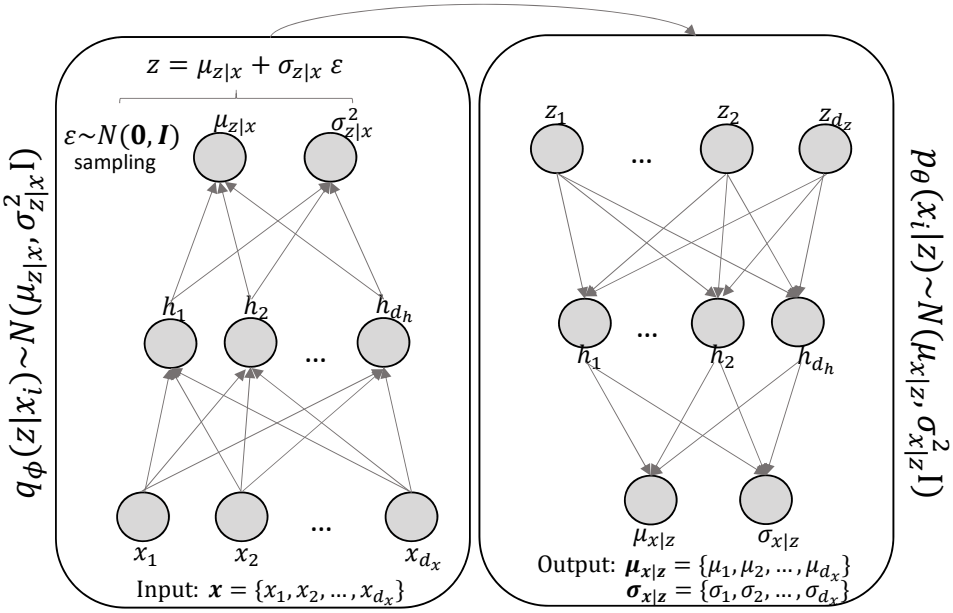

Figure 1: Graphical representation of the VAE. The multilayer perceptron network to the left corresponds to the probabilistic encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ where $\boldsymbol{x} \in \mathbb{R}^{d_{\boldsymbol{x}}}$ is the network input. The output of the network are the parameters in $q_\phi(\boldsymbol{z}|\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{x}}, \boldsymbol{\sigma}^2_{\boldsymbol{z}|\boldsymbol{x}}\boldsymbol{I})$. Note that $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$ is drawn outside the network in order to use gradient descent and backpropagation optimization techniques. Similarly, the feedforward network to the right corresponds to the probabilistic decoder $p_\theta(\boldsymbol{x}|\boldsymbol{z})$. In this case, the input are the latent variables $\boldsymbol{z} \in \mathbb{R}^{d_{\boldsymbol{z}}}$ and the network output are the parameters in $p_\theta(\boldsymbol{x}|\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}|\boldsymbol{z}}, \boldsymbol{\sigma}^2_{\boldsymbol{x}|\boldsymbol{z}}\boldsymbol{I})$. For readability purposes we do not specify the weights $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ in the decoder and encoder respectively. However, these parameters are represented by the lines joining the nodes in the networks plus a bias term attached to each node.

Since the KL divergence is non-negative, the expression in Equation 4 is called *the evidence lower bound* (ELBO). Noting that the ELBO is the negative KL divergence in Equation 3 plus the constant term $\log p(\boldsymbol{x})$, it follows that maximizing the ELBO leads to minimizing Equation 2.

It is worth mentioning that the term $KL[q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})]$ makes the variational density to be close to the prior distribution, while the term $E_{\boldsymbol{z}\sim q}[\log p(\boldsymbol{x}|\boldsymbol{z})]$ encourages densities that place their mass on configurations of the latent variables that explain the observed data. The interested reader is referred to Blei et al. (2017); Doersch (2016) for further details.

3.2 The Variational Autoencoder and AEVB algorithm

The Variational Autoencoder, see Figure 1, is a generative model, which aims to learn the distribution of the input data $\boldsymbol{x}$. This means that the VAE can sample from a distribution that it is similar to the one that have generated $\boldsymbol{x}$. In addition, the VAE assumes that latent variables $p(\boldsymbol{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ govern the distribution of $\boldsymbol{x}$. In this research, the input data $\boldsymbol{x}$ represents the customer data, or a specific grouping of it, and the data transformation of such data is generated by $q(\boldsymbol{z}|\boldsymbol{x})$. This data representation of the customer data should capture the customers' creditworthiness. In this section, we will show how the VAE approximates Equation 5 by maximizing the ELBO. This is done using multilayer perceptron (MLP) networks and stochastic gradient optimization.

The MLPs, which optimze the ELBO, estimate the parameters $\boldsymbol{\mu}_\cdot$ and $\boldsymbol{\sigma}^2_\cdot$ in the density functions $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ and $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, i.e. $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}|\boldsymbol{z}}, \boldsymbol{\sigma}^2_{\boldsymbol{x}|\boldsymbol{z}}\boldsymbol{I})$ and $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{x}}, \boldsymbol{\sigma}^2_{\boldsymbol{z}|\boldsymbol{x}}\boldsymbol{I})$. Note that given that the output of the MLPs are $\boldsymbol{\mu}_\cdot$ and $\boldsymbol{\sigma}^2_\cdot$, the stochastic gradient optimization is on $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, which are the weights in the MLPs. By doing this, the VAE learns the values of $\boldsymbol{\mu}_\cdot$ and $\boldsymbol{\sigma}^2_\cdot$ that maximize the ELBO[2].

[2]It is possible to specify other distributions for $p(\cdot)$ and $q(\cdot)$. However, Gaussian distributions are appropriate for our data sets, and we assume a diagonal covariance matrix as in the original VAE.

Specifically, assuming the set of i.i.d vectors $\{\boldsymbol{x}_i, ..., \boldsymbol{x}_n\}$, the Auto Encoding Variational Bayesian (AEVB) algorithm (Kingma and Welling, 2013) learns the parameters $\boldsymbol{\theta}, \boldsymbol{\phi}$ jointly using MLP networks, and by performing stochastic gradient descent on the

$$ELBO(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{x}_i) = E_{\boldsymbol{z}\sim q}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z})] - KL[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_i)||p(\boldsymbol{z})] \tag{6}$$

for the i'th customer. Therefore, the MLPs for $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ in Figure 1 have the following form

$$\begin{aligned}
\boldsymbol{h} &= \tanh(\boldsymbol{W}_1\boldsymbol{x}_i + \boldsymbol{b}_1), & \boldsymbol{h} &= \tanh(\boldsymbol{W}_4\boldsymbol{z}_i + \boldsymbol{b}_4), \\
\boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{x}} &= \boldsymbol{W}_2\boldsymbol{h} + \boldsymbol{b}_2, & \boldsymbol{\mu}_{\boldsymbol{x}|\boldsymbol{z}} &= \boldsymbol{W}_5\boldsymbol{h} + \boldsymbol{b}_5, \\
\log \boldsymbol{\sigma}^2_{\boldsymbol{z}|\boldsymbol{x}} &= \boldsymbol{W}_3\boldsymbol{h} + \boldsymbol{b}_3, & \log \boldsymbol{\sigma}^2_{\boldsymbol{x}|\boldsymbol{z}} &= \boldsymbol{W}_6\boldsymbol{h} + \boldsymbol{b}_6, \\
\boldsymbol{z}_i &= \boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{x}} + \boldsymbol{\sigma}_{\boldsymbol{z}|\boldsymbol{x}} \odot \boldsymbol{\epsilon}, & \hat{\boldsymbol{x}}_i &= \boldsymbol{\mu}_{\boldsymbol{x}|\boldsymbol{z}} + \boldsymbol{\sigma}_{\boldsymbol{x}|\boldsymbol{z}} \odot \boldsymbol{\epsilon},
\end{aligned} \tag{7}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\odot$ is the element-wise product, $\boldsymbol{\phi} = \{\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3, \boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3\}$ and $\boldsymbol{\theta} = \{\boldsymbol{W}_4, \boldsymbol{W}_5, \boldsymbol{W}_6, \boldsymbol{b}_4, \boldsymbol{b}_5, \boldsymbol{b}_6\}$ are the unknown parameters in the MLPs for $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ respectively.

It is worth mentioning that the latent variable $\boldsymbol{z}$ has been reparametrized as a deterministic and differentiable system. The reason is that we need to backpropagate the term $E_{\boldsymbol{z}\sim q}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z})]$ in Equation 6. Without the reparametrization, $\boldsymbol{z}$ would be inside a sampling operation which cannot be propagated. This means that the AEVB algorithm actually takes the gradient of $E_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_i = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\epsilon})]$ (Kingma and Welling, 2013).

Note that $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ generates latent variables given $\boldsymbol{x}$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ converts them into its original representation. Hence, the former is referred as probabilistic encoder and the latter as probabilistic decoder.

4 Learning Latent Representations

In this section we introduce the motivation for the specific grouping of data that we use to steer a data representation, which encapsulates the customers' creditworthiness in the latent space of the VAE. The presumption is that given that the AEVB algorithm has converged to the optimal variational density $q^*(\boldsymbol{z})$, the latent space should have learned a data representation, which encapsulates the customers' creditworthiness. Otherwise, the reconstruction would have failed, and the algorithm would not have converged to $q^*(\boldsymbol{z})$ in the first place.

To quantify creditworthiness, let us first define the ground truth class

$$y = \begin{cases} 1 & \text{if at least 90 days past due} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

At least 90 days past due, or just 90+dpd, refers to the customers' payment status, which is known after the performance period is over [3]. This definition is aligned with the Basel II regulatory framework (Anderson, 2007).

Let $C_j = \{c_{j,1}, c_{j,2}, ..., c_{j,n_j}\}$ be the *j'th* set of customers with class labels $Y_j = \{y_{j,1}, y_{j,2}, ..., y_{j,n_j}\}$. Hence,

$$dr_{C_j} = \frac{\sum_i^{n_j}[y_{j,i} = 1]}{n_j}, \tag{9}$$

where $[\cdot]$ is the Iverson bracket, is the default rate of the j'th group of customers.

Given that $dr_{C_j} > dr_{C_l}$, we say that the group C_j has lower creditworthiness compared to group C_l. In other words, customers in C_j have, on average, higher probability of default. Therefore, in order to identify L segments with a different propensity to fall into financial distress, we need to find segments

[3] The performance period is the time interval in which if customers are at any moment 90+dpd, then their ground truth class is $y = 1$. Frequently, 12 and 24 months are time intervals used for the performance period. Further, the performance period starts at the moment an applicant signs the loan contract.

Fine classing approach								
Age	Count	Distribution all	Goods	Distribution goods	Bads	Distribution bads	Bad Rate	WoE
Missing	1 000	2.5%	860	2.38%	140	3.65%	14.00 %	-0.4272
18-22	4 000	10%	3 040	8.41%	960	25.00%	24.00 %	-1.0898
23-26	6 000	15%	4 920	13.61%	1 080	28.13%	18.00 %	-0.7261
27-29	9 000	22.5%	8 100	22.40%	900	23.44%	10.00 %	-0.0453
30-35	10 000	25.0%	9 500	26.27%	500	13.02%	5.00 %	0.7019
36-44	7 000	17.5%	6 800	18.81%	200	5.21%	2.86 %	1.2839
44+	3 000	7.5%	2 940	8.13%	60	1.56%	2.00 %	1.6493
Total	40 000	100%	36 160	100%	3 840	100%	9.60 %	
Coarse classing approach								
Missing	1 000	2.5 %	860	2.38 %	140	3.65 %	14.00%	-0.4272
18-29	19 000	47.5 %	16 060	44.41%	2 940	76.56 %	15.47 %	-0.5445
30-44+	20 000	50%	19 240	53.20%	760	19.79 %	3.80 %	0.9889
Total	40 000	100%	36160	100 %	3840	100 %	9.60%	

Table 1: Weight of Evidence transformation of the variable age. The top panel shows the fine classing approach, while the bottom panel shows the coarse approach where only three groups are created.

where the average probability of default is different from the rest of the groups. Mathematically, we want to learn a data representation that satisfies

$$dr_{C_j} \neq dr_{C_l}, \quad for \quad j, l = 1, 2, ..., L \; and \; j \neq l. \tag{10}$$

Now it should be clear that the data transformation $f(\cdot)$ that we are looking for, needs to incorporate the class label y. In this way, the latent space in the VAE should generate codes that also contain information about y. Otherwise, those codes will fail to reproduce $f(\boldsymbol{x}|y)$.

One such transformation is the Weight of Evidence[4] (WoE) (Anderson, 2007; Siddiqi, 2012), which is defined as

$$\log \frac{Pr(\boldsymbol{x}|y=0)}{Pr(\boldsymbol{x}|y=1)}. \tag{11}$$

4.1 The Weight of Evidence

The WoE transformation has been used in credit scoring for a long time (Abdou, 2009), and it has become the standard in credit scoring models. The way to estimate it, given that the m'th feature x_m is continuous, is by dividing its values into K bins $B_1, B_2, ..., B_K$. In the case of categorical variables, the different categories are already these bins. Hence, the WoE for the k'th bin of the m'th feature is

$$WoE_{k,m} = \log \frac{Pr(x_m \in B_k|y=0)}{Pr(x_m \in B_k|y=1)} = \log \frac{\frac{1}{n}\sum_{i=1}^{n}[x_{i,m} \in B_{k,m} \text{ and } y_i = 0]}{\frac{1}{n}\sum_{i=1}^{n}[x_{i,m} \in B_{k,m} \text{ and } y_i = 1]}, \tag{12}$$

where n is the total number of observations. Note that the number of bins can vary for different features. See chapter 16.2 in Anderson (2007) or chapter 6 in Siddiqi (2012) for more details. Table 1 shows the difference between fine and coarse classing. In the fine classing approach, we create K bins, which provide the finest granularity. Then, fine bins with similar risk are binned into smaller groups resulting in the coarse classing. See Anderson (2007) for more details.

We use the coarse classing WoE transformation[5] of the input data $\boldsymbol{x}$ to tilt the latent space in the VAE towards configurations which encapsulate the propensity to fall into financial distress.

[4]Originally, the WoE was introduced by Irving John Good in 1950 in his book *Probability and the Weighing of Evidence* and it has been used in the logistic regression and Naïve Bayes literature, among others.

[5]We will simply called it as WoE in the remaining of the paper for brevity.

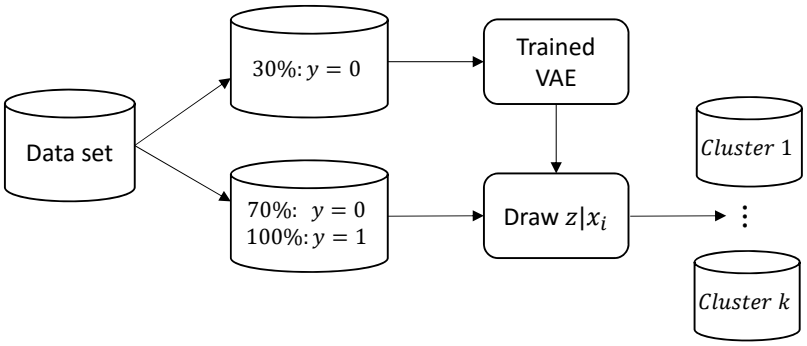

Figure 2: Development methodology: We use 30% of the majority class data for training the VAE. Once it is trained, we generate the latent variables for the remaining data.

5 Experiments and Results

Our goals in this section are: i) to show how can we reveal the natural clustering structure of financial data, which is unknown a priori, using our proposed framework, ii) to analyze some of the properties of the learned data representations with our method, iii) to benchmark our proposed methodology with some well-known representation learning methodologies, such as PCA (Pearson, 1901), kernel PCA (Schölkopf et al., 1998), isomaps (Tenenbaum et al., 2000), and t-SNE (Hinton and Roweis, 2003), and iv) to show that our proposed supervision stage is able to steer representation learning for bank data.

5.1 Data description

We use real data sets from three different geographical regions, reflecting different market and economic conditions, allowing us to test the generalization properties of our proposed methodology for data representations of bank customers. The data set used in this section are a Norwegian and a Finnish car loan data set provided by Santander Consumer Bank Nordics and a public data set used in the Kaggle competition *Give me some credit*[6]. These data sets show applicants' status, financial and demographic factors at the time of application as well as the class label. The performance period for the real data sets is 12 months, while for the public data set it is 24 months. More details about the data sets can be found in Tables A1, A3 and A4.

5.2 Training the VAE and Generating Latent Representations

We train the VAE using the WoE as the input data, and using only the majority class ($y = 0$) data. The reason is because we want to have a robust estimate for the default rate in the data representation that we are learning. In addition, using observations from the minority class ($y = 1$) did not change the data representation in the latent space in our experiments, which is probably explained by the strong class imbalance in the three data sets.

Hence, we use 30% of the majority class to train the VAE. During training, we generate the latent space for the remaining 70% of the majority class and 100% of the minority class data, see Figure 2. Based on the optimization of the ELBO, together with a heuristic visual comparison of the latent space in the training and test data sets, we select the optimal network architecture as well as the stopping criteria. It is worth mentioning that we observe that the shapes and proportions of the clusters in the training data are similar to the ones in the test data. This is a good indicator that our proposed model learns data representations that generalizes to unseen customers.

The VAE architectures that we tested are shown in Table A2, and the final architecture IDs that we use are arch4, arch4 and arch1 for the Norwegian, Finnish and Kaggle data sets respectively. In addition,

[6]Website `https://www.kaggle.com/c/GiveMeSomeCredit`

Algorithm 1: Labelling the latent data representation of bank customers.

Input : $\boldsymbol{z}$,n_{min},ρ
Output: cluster labels
pending_data = $\{\boldsymbol{z}\}$;
labels = ones(length($\boldsymbol{z}$)) ;
while *EOF(pending_data)==FALSE* **do**
  for *item in pending_data* **do**
    labels = HierarchicalAlgorithm(pending_data[item], k = 2) ;
    get centroids c1 and c2 ;
    split pending_data[item] into C1 and C2 using labels ;
    if $n1 > n_{min}$ *AND* $n2 > n_{min}$ *AND* $||c1 - c2|| > \rho$ **then**
      update labels ;
      pending_data.append = {C1,C2}
    end
  end
end
return labels

we use tanh activations in all hidden layers, linear and sigmoid activations in the μ output layer for the encoder and decoder respectively, and linear activations in all log σ^2 layers[7]. The MLP models are trained with the adagrad optimizer (Duchi et al., 2011) using constant 0.01 learning rate and 0.001 momentum.

Finally, we use the expectation over the latent space for the i'th customer

$$E[\boldsymbol{z}|\boldsymbol{x}_i] = \int_{-\infty}^{\infty} \boldsymbol{z} q_\phi(\boldsymbol{z}|\boldsymbol{x}_i) d\boldsymbol{z} = \boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{x}} \tag{13}$$

as the data representation of bank customers in the latent space of the VAE. Note that it is simply the output in the encoder MLP network, see Equation 7. We also tried the Monte Carlo version of Equation 13 using 100 samples of $\boldsymbol{z}_i$ and Equation 7 and the results do not change. At this point it is worth mentioning that our proposed methodology only utilizes the class label y during the supervision stage. After our model has been trained, we are able to transform the input data for new customers into the WoE and then transform the WoE into a data representation that lies in one of the clusters in the latent space.

In what follows, we provide some practical considerations to train a VAE with our proposed methodology that help to reveal useful data representations for bank customers:

- Create business intuitive and monotonic coarse classing WoE, and make sure that the final WoE groups are not small.
- Tune the hyperparameters of the model by grid search.
- Choose simple network architectures for the encoder and decoder.
- During training, plot the latent space often, e.g. every fifth epoch, for both the training and testing data set.
- Based on the optimization of the lower bound and the data representation for the test data set decide a stopping criteria.

To further analyze the learned data representation of our method, we assign labels to the structure in the two-dimensional latent space. This task can be done manually using a set of *if/else* rules given the well-defined clustering structure in the latent space. However, we propose an automated version, which is presented in Algorithm 1. The idea is to use the hierarchical clustering algorithm iteratively, generating only two clusters in each iteration. Always preserving the clustering structure in the learned data representation. For this purpose, Algorithm 1 specifies the minimum number of observations in each cluster, denoted by n_{min}. Similarly, the minimum Euclidean distance between the centroids in the

[7] We need to use different activation functions depending on the kind of variable that the MLP is handling. See chapter 6 in Goodfellow et al. (2016) for more details.

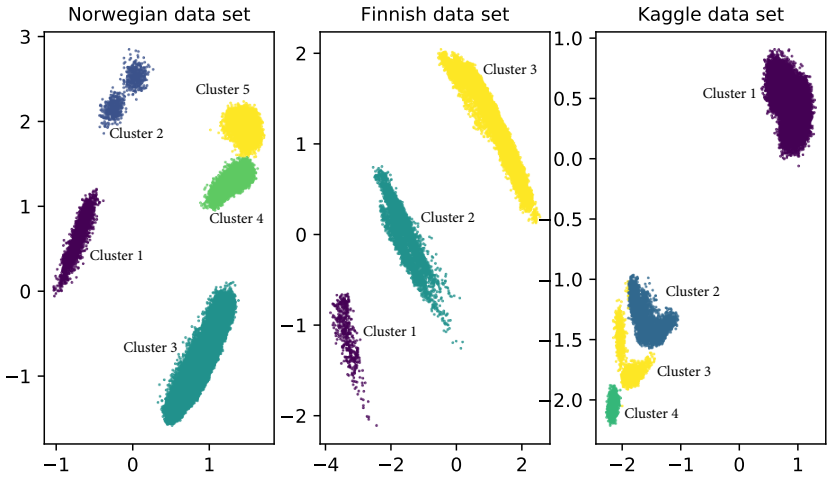

Figure 3: Latent representation of bank customers.

	Norwegian data set				Finnish data set				Kaggle data set			
Cluster	dr	CI	Obs	$y = 1$	dr	CI	Obs	$y = 1$	dr	CI	Obs	$y = 1$
1	5.30%		2 206	117	5.93%		438	26	5.47%	(***)	97 434	5 327
2	11.24%	***	774	87	3.76%	***	6 067	228	33.13%	***(***)	6 121	2 028
3	1.39%	(***)	109 969	1539	0.91%	(***)	75 536	685	55.68%	***(***)	2 026	1 128
4	2.89%	***(***)	11 450	331					63.58%	***	2 427	1 543
5	5.30%		9 106	483								

Table 2: Default rates dr_{C_j} for the different clusters C_j in the data representation of bank customers are given in the first column. The 99% confidence intervals (CI) are shown in the second column for each data set, where non-overlapping lower bounds are denoted outside parenthesis and non-overlapping upper bounds are within parenthesis. Note that for the cluster with the lowest (highest) default rate we do not need to verify the lower (upper) bound. Finally, the total number of customers and the number of default customers are shown in the third and last column respectively.

two clusters needs to be specified, and it is denoted by ρ. These two parameters are data dependent and should be selected in such a way that Algorithm 1 assigns cluster labels preserving the clustering structure in the latent space. The advantage of this approach is that the labelling happens automatically while we train the VAE. Finally, the results of our proposed data representations are shown in Figure 3 and Table 2.

5.3 Properties of the Learned Data Representations

The first important result to highlight is that using the WoE transformation we learned a data representation with well-defined clusters in the latent space for all three data sets. Analyzing the Norwegian car loan data, we see that about 82% of all customers are in cluster 3, which is the cluster with the smallest default rate. This makes sense since the data set contains only 2 557 customers from the minority class. On the other hand, about 18% of the customers are in clusters with relatively high default rate. Finally, we check whether the default rates are significantly different using the 99% confidence interval for a binomial variable, i.e. $\hat{p} \pm 2.57\sqrt{\hat{p}(1-\hat{p})/n}$, where $\hat{p}$ is the default rate in each cluster, 2.57 is the corresponding critical value and n is the total number of observations in the cluster. With the exception of clusters 1 and 5, the default rate for the other clusters are statistically different. See Table 2.

For the Finnish and Kaggle data sets we observe the same pattern. The majority of the customers are in the cluster with the smallest default rate. However, about 10% of the customers in the Kaggle data are in three clusters with very high default rates. Note that all default rates in the Kaggle data are significantly different. On the other hand, the confidence intervals for the default rates in cluster 1 and 2 for the Finnish data set overlap each other. This is driven by the relatively small number of defaults

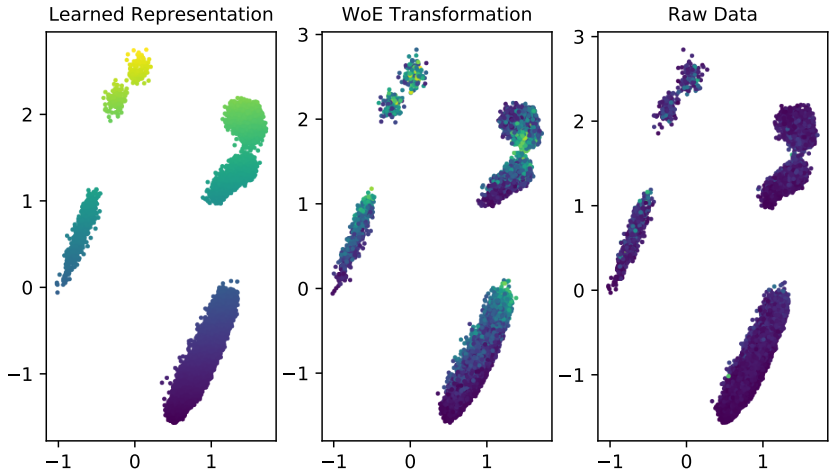

Figure 4: Best viewed in color. We estimate the default probability for 30% of customers in the Norwegian data using the learned representation, WoE, and raw data. Further, we generate the latent space for these customers using the trained VAE. Finally, we use the three estimate values for default probability to create a colormap. Note that the left panel shows a smooth color transition.

in the cluster 1, which increases the variance of their default rate estimate.

We estimate the default probability for customers in the Norwegian data set using three different input data: i) the learned data representation of the VAE, ii) the WoE transformation, and iii) the raw data. We use 70% of the data to estimate the default probability of the remaining 30% of the data. Further, we use the trained VAE from Section 5.2 to generate the latent space of the customers for whom we estimated the default probability. In Figure 4, we show the learned data representation for these customers and we use the three estimated values for the default probability to create a colormap. It is interesting to see that the default probability estimated with the learned data representation reveals a smooth color transition. On the other hand, when the default probability is estimated with the WoE or with the raw data, the colormaps show a relatively random pattern. This result shows that our proposed method is not only able to learn a data representation of customers data, which shows a well-defined clustering structure and captures the customers' creditworthiness, but which also ranks the default probability across the two dimensions of the latent space.

The well-defined clustering structure of the data representation in the latent space and its ability to capture customers' creditworthiness, allows our proposed method to generate good representations. These representations express general priors that are particularly useful in the bank industry. Specifically, our proposed data representation is able to learn the natural clustering and spatial coherence of creditworthiness in the customers data.

5.4 Representation Learning Benchmark

We use the k-means (Lloyd, 1982), affinity propagation (Frey and Dueck, 2007), hierarchical (Ward Jr, 1963), birch (Zhang et al., 1996) and GMM algorithms to cluster the WoE transformations for the Norwegian data set in the original input space. Note that for all of these algorithms, the number of cluster must be specified. Hence, we use as input parameter the number of cluster suggested by our proposed methods, which is 5 for the Norwegian data set, to make results comparable.

After we cluster the Norwegian WoE data using the original input space, we apply different dimensionality reduction techniques to visualize the clusters that we obtained in a two dimensional space. Specifically, we use PCA, kernel PCA, isomaps, and t-SNE for dimensionality reduction. Figure 5 shows both the clusters obtained in the first step, represented by different colors, and the two dimensional data representation

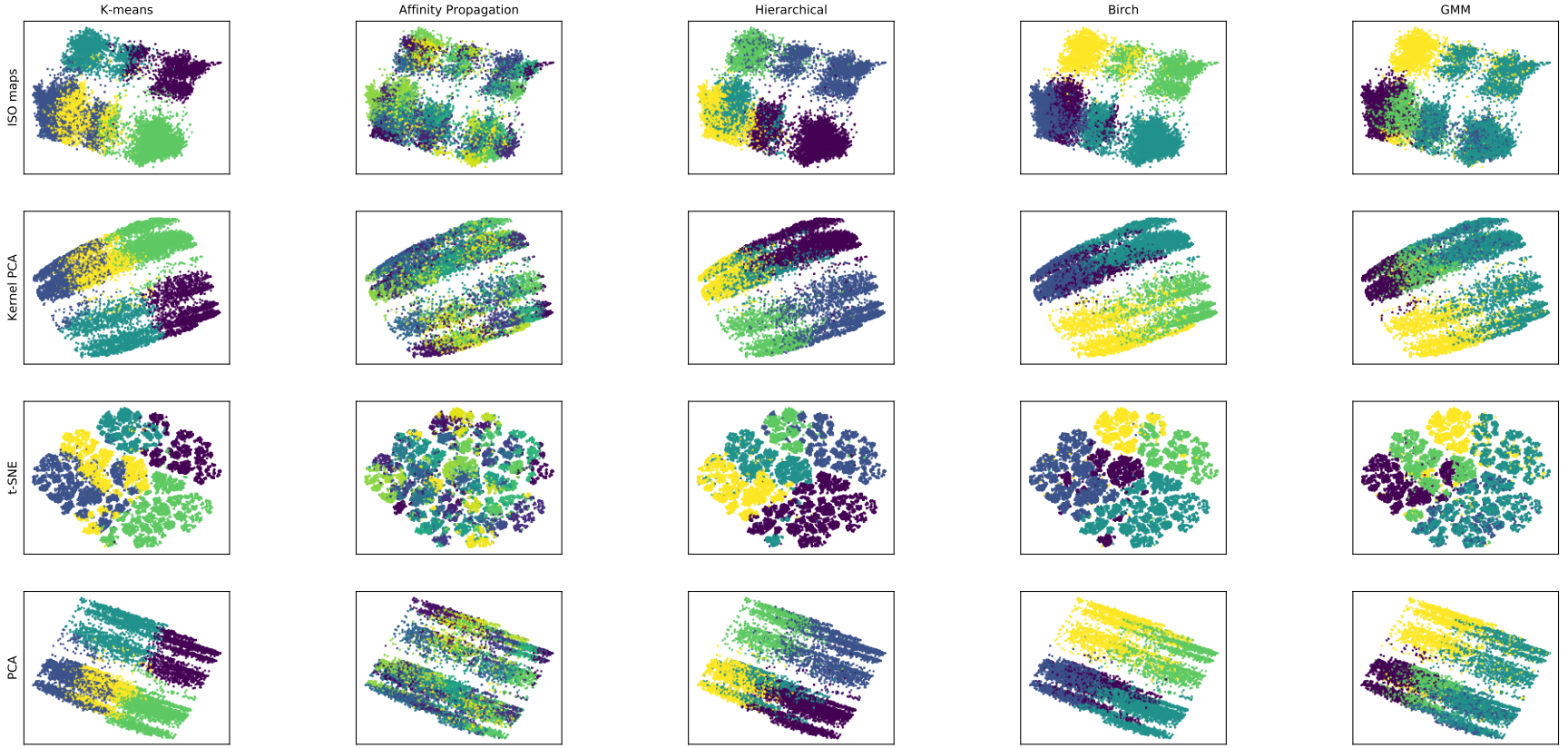

Figure 5: Best viewed in color. We use the k-means, affinity propagation, hierarchical, birch, and GMM algorithms to cluster the WoE transformations for the Norwegian data, specifying five clusters. Then, we reduce the original dimensional space for the WoE to two dimensions using isomaps, kernel PCA, t-SNE and PCA. Cluster labels are given by the colors.

that we obtained. As can be seen from the figure, none of the the existing state-of-the-art data representation methods that we benchmark are able to generate a well-defined clustering structure.

Note that we could use the representation learning algorithms in the original input space and then cluster the data representation obtained. However, this ordering does not change the learned data representation of the input data, it will only change the clustering results in the two dimensional space. Given that none of the representation learning methods that we benchmark are able to generate a well-defined structure, the clustering algorithms will never find such non-overlapping clusters as the ones that we obtain with our proposed method.

5.5 Grouping of the Input Data

Now we want to show that the supervision step in our proposed methodology has valuable information about the input data, which is captured in well-defined clustering structures. Hence, we use different data transformations and, for each of these transformations, we train a new VAE, i.e. for each data transformation we learn a data representation using the same network architectures as the ones used to learn the data representations in Figure 3. Specifically, we generate the latent space for the following data transformations:

1. PCA: The input data is transformed using principal component analysis with all the principal components, i.e. there is no dimensionality reduction.
2. Standardization: The input data is standardized by removing the mean and scaling to unit variance.
3. Fine classing WoE: The input data is transformed into WoE by creating bins with an approximately equal number of customers, i.e. no coarse classing is done.
4. Input data: Raw data without any transformation.

Figure 6 shows the resulting latent spaces for the data transformations explained above. Interestingly, three of these transformations do not show any clustering structure at all. For the standardized transformation, the clusters have practically the same default rate. Hence, by identifying appealing data transformations and a useful grouping of the input data, it is possible to steer data representations in the latent space of the VAE. In this particular case, these representations are well-defined clusters with considerably different risk profiles.

Note that the right-most scatter in Figure 6 represents the latent space for a traditional training approach where the input data is fed into the encoder network without any manipulation. It should be clear now that the supervision stage in our proposed methodology, together with the WoE transformation that encapsulates customers' creditworthiness, makes it possible to steer data representations in the latent space of the VAE. As a result, we obtain clusters with different risk profiles, which are unknown a priori and cannot be identified in the input space.

6 Business Application

In this section, we present two business applications of our proposed methodology for learning data representations of bank customers. First, we identify the salient dimensions in the clustering structure for the Norwegian data set and use those salient dimensions to obtain descriptive labels associated with each cluster. These labels can be used to find out which customers should be the target of a marketing campaign to sell a new product, for example. Second, we show how banks can improve the assessment of customers' creditworthiness using the clusters identified by our proposed method for the Norwegian data set.

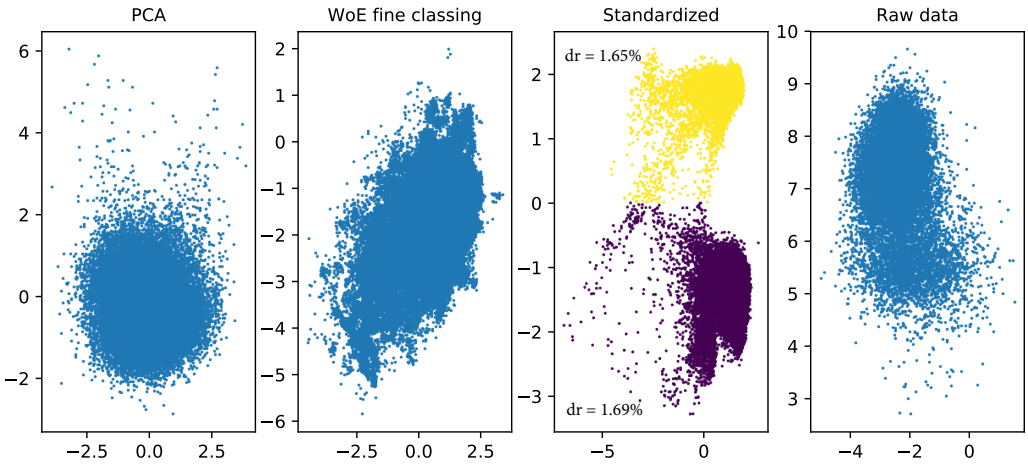

Figure 6: Latent space for four different data transformation for the Norwegian data set. The left panel shows a PCA transformation (preserving the original data dimensionality). The second panel shows the latent space for the fine classing WoE transformation. The third panel shows the latent space for the standardized data, and finally, the right panel shows the latent space for the raw data. Standardizing the data reveals two clusters in the latent space. However, these clusters have practically the same default rate (dr). The other three transformations do not show any clustering structure.

6.1 Customers Profiles

After the VAE model has been trained with our proposed methodology, we are able to map the input data for new customers into one of the clusters in the latent space, which have different risk profiles (see Table 2). Further, we assign descriptive labels to the clusters in the latent space so they can be used for marketing or product offering purposes. To that end, we adopt the salient dimension methodology presented in Azcarraga et al. (2005) and explained in Appendix B. This approach identifies features whose values are statistically significant in different clusters, and are called salient dimensions. In what follows, we analyse the salient dimensions for the Norwegian data set, and salient dimensions for the Kaggle and Finnish data sets can be found in Table A5.

The first interesting result in Figure 3 is the pattern of the latent variables for clusters 1 and 5 (both clusters have default rate = 5.30%), which are located on opposite sides of the two-dimensional space. The salient dimension *MaxBucket12* in cluster 1 shows that about 70% of the customers were between 30 and 60 days past due at the moment they applied for a new loan, i.e. they are existing customers applying for a new loan. Actually, all customers in cluster 1 are existing customers who are at least 30 days past due. On the other hand, about 51% of the customers in cluster 5 are new applicants willing to buy middle-age cars. Cluster 2 is actually composed only by existing customers, i.e. new applicants lie on the right side of Figure 3, while existing customers on the left hand side. Therefore, we can label cluster 1 as *existing customers in arrears applying for a car loan* and cluster 5 as *new applicants and existing customers in arrears applying for a loan to by a middle-age car.*

Now let us see what characterizes cluster 3, which is the cluster with the lowest default rate. Looking at one of its salient dimension namely *DownPayment%*, we can see that the average down payment in this cluster is about 20%. On the other hand, the average down payment for the rest of the clusters is less than 12%. In the bank industry, high down payments are linked to low default rates. Further, the salient dimension *AgeObject* shows that about 35% of the customers in cluster 3 are applying to buy relatively new cars. In contrast, the average percentage of customers applying to buy new cars, in the other clusters, is only 23%. Buyers of new cars are also associated with low default rates by bank risk analysts. Hence, we could label cluster 3 as *new applicants willing to buy new cars with high down payment amount.*

Cluster 2 has the highest default rate and can be explained by its salient dimension *MaxBucket12*. About 93% of customers in this cluster are between 1 and 90 days past due, while the percentage of customers in

Norwegian data set			
Performance metric	Cluster	Segment-based	Portfolio-based
Kolmogorov-Smirnov	1	0.4648	0.4098
	2	0.3441	0.3280
	3	0.4318	0.4199
	4	0.3821	0.3489
	5	0.3410	0.3299
Gini coefficient	1	0.5377	0.4860
	2	0.3582	0.3402
	3	0.5511	0.5412
	4	0.4790	0.4377
	5	0.4043	0.3846
H-measure	1	0.2774	0.2310
	2	0.1665	0.1453
	3	0.2174	0.2076
	4	0.1760	0.1471
	5	0.1302	0.1193
AUC	1	0.7688	0.7430
	2	0.6791	0.6701
	3	0.7756	0.7706
	4	0.7395	0.7188
	5	0.7021	0.6923

Table 3: Model performance for the segment-based and the classical credit scoring approach. Performance values are the average of a 10-cross-validation.

the other clusters in the same interval is only 15%. This cluster could have the label *existing customers in high arrears level.*

6.2 Improving Customers' Creditworthiness

To show whether the clustering structure revealed by our proposed methodology can improve the assessment of customers' creditworthiness, we train one multilayer perceptron for each of the 5 clusters found in the Norwegian data set (see Figure 3). We use the WoE as input features and we divide the data set for each cluster in 70% for training and 30% testing. Further, we compare the classification performance of the segment-based strategy with the traditional credit scoring approach where only one classifier is trained for the whole data set. To train the classical credit scoring model, we also use the WoE as input features and for the training data we aggregate all training data sets for the 5 clusters in the segment-based approach. Similarly, we test model performance of the classical credit scoring model on each of the 5 test data sets for the segment-based approach. Table 3 shows the values of 4 different performance metrics, which are commonly used in the financial industry to measure the discriminative power of credit scoring models, obtained for our approach and the standard credit scoring approach. By using our proposed methodology to identify clusters that encapsulated customers' creditworthiness and developing segment-based classifiers, banks can improve the assessment of creditworthiness.

7 Conclusion

In this paper, we show that it is possible to steer data representations in the latent space of the Variational Autoencoder using a semi-supervised learning framework and a specific grouping of the input data. We show that the Weight of Evidence transformation encapsulates the propensity for financial distress and by training a VAE with our proposed methodology we can learn a latent data representation that captures the natural clustering of the data and encapsulates the customers' creditworthiness.

The data representations generated with our proposed methodology have certain features that are particularly useful in the bank industry. The representations are not only able to learn the natural clustering of the data, which is unknown a priori and cannot be identified in the input space, but also the spatial coherence of customers' creditworthiness.

The main advantages of our proposed method are that it captures the natural clustering of the data, suggests the number of clusters, captures the spatial coherence of customers' creditworthiness, generates data representations of unseen customers, and assigns them to one of the existing clusters. Finally, our empirical results, based on real data sets reflecting different market and economic conditions, show that none of the well-known data representation models in the benchmark analysis are able to obtain well-defined clustering structures. Further, we show how banks can use our proposed methodology to improve marketing campaigns and credit risk assessment.

Acknowledgements

We would like to thank Santander Consumer Bank for financial support and the real data sets used in this research. This work was also supported by the Research Council of Norway [grant number 260205] and SkatteFUNN [grant number 276428].

Appendices

A Figures and Tables

Name	Cases	Features	Default rate
Norwegian data set	187 069	20	0.0137
Finnish data set	115 899	12	0.0081
Give me some credit	150 000	10	0.0668

Table A1: Summary of the three data sets used in the different experiments in this paper. Default rate for the j'th set of customers is defined as $dr_{C_j} = \frac{\sum_i^{n_j}[y_{j,i}=1]}{n_j}$, where n_j is the total number of customers and y_i is the class label.

Architecture ID	z dimension	Hidden Layers	Neurons	Learning Rate	Epochs
arch1	2	1	5	0.01	50
arch2	2	1	10	0.01	50
arch3	2	1	20	0.01	50
arch4	2	1	30	0.01	50
arch5	2	1	40	0.01	50
arch6	2	1	50	0.01	50
arch7	2	1	60	0.01	50
arch8	2	1	70	0.01	50
arch9	2	1	30	0.007	50
arch10	2	1	30	0.008	50
arch11	2	1	30	0.009	50
arch12	2	1	30	0.011	50
arch13	2	1	30	0.012	50
arch14	2	1	30	0.013	50
arch15	5	1	30	0.01	50
arch16	10	1	30	0.01	50
arch17	15	1	30	0.01	50
arch18	2	2	30	0.01	50
arch19	2	3	30	0.01	50
arch20	2	4	30	0.01	50
arch21	2	5	30	0.01	50

Table A2: Different architectures tested to train the VAE for the three different data sets. More complex architectures, with more hidden layers and different dimension in the latent spaces, were also tested. However, for the data sets under analysis relative complex architectures do not add any significant value.

Norwegian data set	
Variable Name	Description
BureauScoreAge	Matrix with bureau scores and applicants age
NetincomeStability	Net income stability index
RiskBucketHistory	Delinquency history
NumApps6M	Number of applications last 6 months
ObjectGroupCarMake	Car brand in the application
DownPaymentAgeObject	Matrix with down payment and car model year
CarPrice	Car price
NetIncomet0t1	Change in applicant's net income
MaxBucketSnapshot	Delinquency at the time of application
MaxMoB12	Months on books at the time of application
NetIncomeTaxt0	Ratio between net income and taxes
AgeObject	Car model year
AgePrimary	Age of primary applicant
BureauScoreUnsec	Bureau score unsecured
DownPayment	Own capital
MaxBucket12	Maximum delinquency in the past 12 months
TaxAmountt0	Tax amount paid
BureauScore	Bureau score generic
Taxt0t1	Change in applicant's taxes
Netincomet0	Net income at the time of application

Table A3: Variable name and description for all features in the Norwegian car loan data set.

Kaggle	
Variable Name	Description
RevolvingUtilizationOfUnsecuredLines	Total balance on credit lines
AgePrimary	Age of primary applicant
NumberOfTime3059DPD	Number of times borrower has been 30-59 dpd
Monthly debt payments divided by monthly gross income	MaritalStatus
Income	Monthly Income
NumberOfOpenCreditLines	Number of loans or credit cards)
NumberOfTimesDaysLate	Number of times borrower has been 90 dpd
NumberRealEstateLoansOrLines	Number of mortgage loans
NumberOfTime6089DPD	Number of times borrower has been 60-89 dpd
NumberOfDependents	Number of dependents in family
Finnish data set	
AgePrimary	Age of primary applicant
AgeObjectContractTerm	Matrix with car model year and number of terms
DownPayment	Own capital
Marital Status	DebtRatio
MaxBucket24	Maximum delinquency in the past 24 months
MonthsAtAddress	Number of months living at current address
Number2Rem	Number of 2nd reminders last year
NumberRejectedApps	Number of rejected applications
ObjectPrice	Car price
ResidentialStatus	Whether the applicant owns a house
ObjectMakeUsedNew	Matrix with car make and whether it is new or used
EquityRatio	Debt to equitity

Table A4: Variable name and description of all features in the Kaggle and Finnish data set data sets.

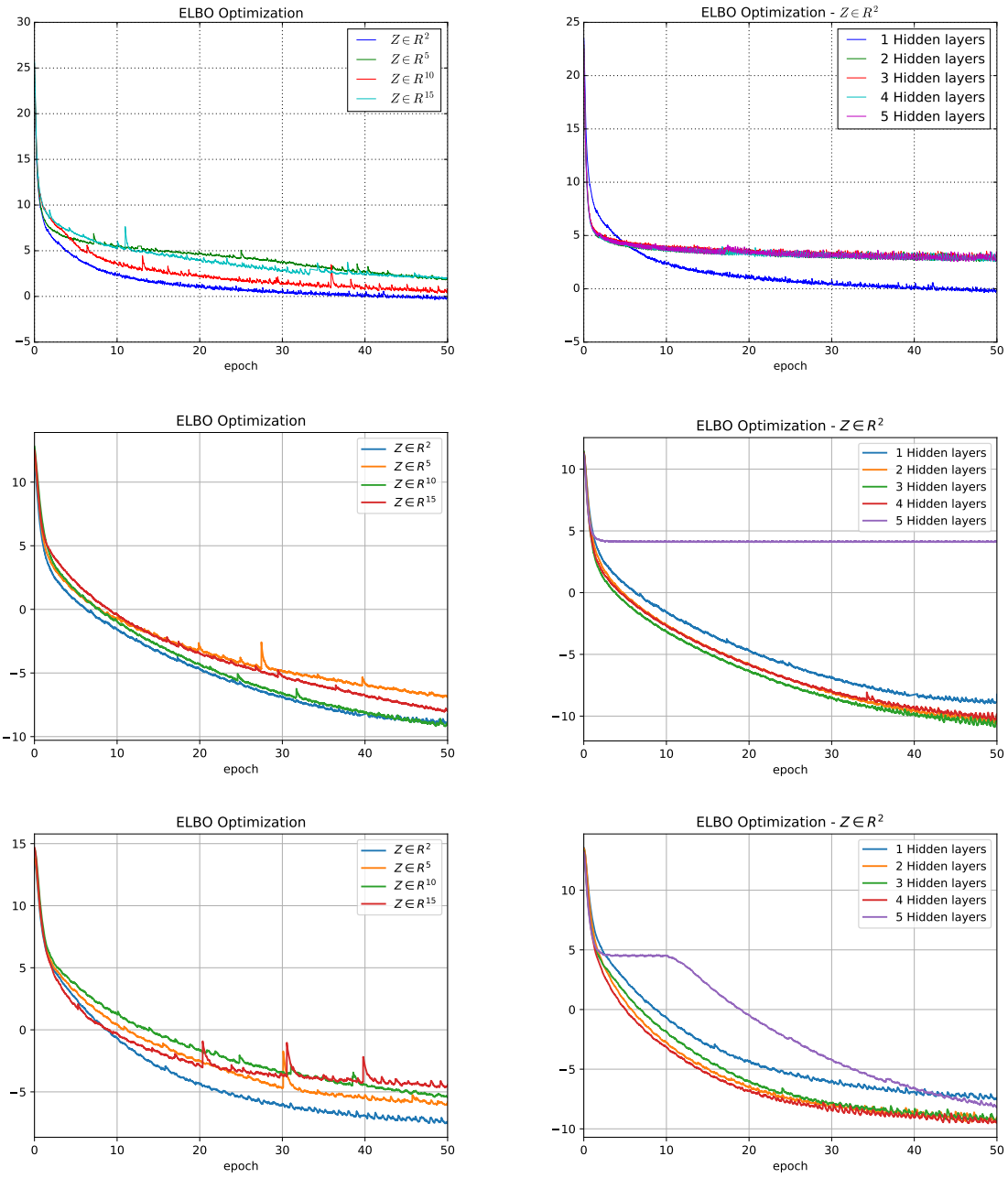

Figure A1: Panels to the left show the optimization of the negative ELBO for different dimensionalities in the latent space. For $z \in \mathbb{R}^2$, the AEVB algorithm converges faster to the optimal variational density $q^*(z)$ for all data sets (Norwegian data set top-left panel, Kaggle data set middle-left and Finnish data set bottom-left panel). Further, panels to the right also show the optimization of the ELBO but for $z \in \mathbb{R}^2$ and for a different number of hidden layers. The VAE for the Norwegian data set (top-right panel) with 1 hidden layer converges faster to $q^*(z)$. For the Kaggle data set (middle-right panel), 2-4 hidden layers converge faster to the optimal variational density. However, the resulting clustering structure in the latent space contains only two clusters. Similarly, for the Finnish data set (bottom-right panel) 2-4 hidden layers makes the algorithm converge faster. However, the resulting clustering structure contains four clusters. For this data set, it is not optimal to have four clusters.

Norwegian data set		**Kaggle**		**Finnish data set**	
Cluster	Salient Dimension	Cluster	Salient Dimension	Cluster	Salient Dimension
1	MaxBucket12	1	NumberOfTime3059DPD	1	AgePrimary
2	NetIncomet0t1	1	NumberOfTimesDaysLate	1	Number2Rem
2	MaxBucket12	1	NumberRealEstateLoansOrLines	1	NumberRejectedApps
3	AgeObject	1	NumberOfTime6089DPD	2	Number2Rem
3	NetIncomet0t1	2	RevolvingUtilizationOfUnsecuredLines	2	NumberRejectedApps
3	Taxt0t1	2	DebtRatio	3	DownPayment
3	DownPayment	3	NumberOfTime3059DPD	3	ResidentialStatus
4	NumApps6M	3	NumberOfTime6089DPD		
4	AgeObject	3	NumberOfDependents		
4	NetIncomet0t1	4	NumberOfTime3059DPD		
4	Taxt0t1	4	NumberOfTimesDaysLate		
4	DownPayment	4	NumberOfTime6089DPD		
5	AgeObject				
5	NetIncomet0t1				
5	DownPayment				

Table A5: Statistically significant salient dimensions for the Norwegian, Kaggle and Finnish data set. We use $s.d. = 1$ to define salient dimensions.

B Salient Dimensions

Let v be the *v'th* dimension of the *i'th* vector $x_{i,v}$, where $x \in R^{\ell}$. Further let $\Phi_{in}(k)$ be the set of in-patterns (within cluster k) and $\Phi_{out}(k)$ be the set of out-patterns (not within cluster k). Then compute the mean input values

$$\mu_{in}(k,v) = \frac{\sum_{x_i \in \Phi_{in}(k)} x_{i,v}}{|\Phi_{in}(k)|}, \tag{14}$$

$$\mu_{out}(k,v) = \frac{\sum_{x_i \in \Phi_{out}(k)} x_{i,v}}{|\Phi_{out}(k)|}, \tag{15}$$

where $|\{\cdot\}|$ returns the cardinality of $\{\cdot\}$. Further, compute the difference factors

$$df(k,v) = \frac{\mu_{in}(k,v) - \mu_{out}(k,v)}{\mu_{out}(k,v)}, \tag{16}$$

and their mean and standard deviations

$$\mu_{df}(k) = \frac{1}{\ell} \sum_{v}^{\ell} df(k,v), \tag{17}$$

$$\sigma_{df}(k) = \sqrt{\sum_{v}^{\ell} \left(df(k,v) - \mu_{df}(k)\right)^2 / \ell}. \tag{18}$$

Finally, we say that the *v'th* feature in cluster k is a salient dimension if

$$df(k,v) \leq \mu_{df}(k) - s.d.\ \sigma_{df}(k), \tag{19}$$

or

$$df(k,v) \geq \mu_{df}(k) + s.d.\ \sigma_{df}(k), \tag{20}$$

where $s.d.$ is the number of standard deviations to be used. The value for $s.d$ is defined based on the data set. We use $s.d. = 1$ for all three data sets under analysis.

References

Abdou, H. A. (2009). Genetic programming for credit scoring: The case of egyptian public sector banks. *Expert systems with applications*, 36(9):11402–11417.

Anderson, R. (2007). *The credit scoring toolkit*. Oxford University Press.

Aurifeille, J.-M. (2000). A bio-mimetic approach to marketing segmentation: Principles and comparative analysis. *European Journal of Economic and Social Systems*, 14(1):93–108.

Azcarraga, A. P., Hsieh, M.-H., Pan, S. L., and Setiono, R. (2005). Extracting salient dimensions for automatic som labeling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(4):595–600.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Bijak, K. and Thomas, L. C. (2012). Does segmentation always improve model performance in credit scoring? *Expert Systems with Applications*, 39(3):2433–2442.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Bouchacourt, D., Tomioka, R., and Nowozin, S. (2018). Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2015). Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Doersch, C. (2016). Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Frey, B. J. and Dueck, D. (2007). Clustering by passing messages between data points. *science*, 315(5814):972–976.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Hand, D. J., Sohn, S. Y., and Kim, Y. (2005). Optimal bipartite scorecards. *Expert Systems with Applications*, 29(3):684–690.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.

Hinton, G. E. and Roweis, S. T. (2003). Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 857–864.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hsu, W.-N., Zhang, Y., and Glass, J. (2017). Learning latent representations for speech generation and transformation. *arXiv preprint arXiv:1704.04222*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Latif, S., Rana, R., Qadir, J., and Epps, J. (2017). Variational autoencoders for learning latent representations of speech emotion. *arXiv preprint arXiv:1712.08708*.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

Lim, M. K. and Sohn, S. Y. (2007). Cluster-based dynamic scoring model. *Expert Systems with Applications*, 32(2):427–431.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Rampasek, L. and Goldenberg, A. (2017). Dr. vae: Drug response variational autoencoder. *arXiv preprint arXiv:1706.08203*.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*.

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319.

Siddiqi, N. (2012). *Credit risk scorecards: developing and implementing intelligent credit scoring*, volume 3. John Wiley & Sons.

Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491.

Su, J., Wu, S., Zhang, B., Wu, C., Qin, Y., and Xiong, D. (2018). A neural generative autoencoder for bilingual word embeddings. *Information Sciences*, 424:287–300.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323.

Titus, A. J., Bobak, C. A., and Christensen, B. C. (2018). A new dimension of breast cancer epigenetics - applications of variational autoencoders with dna methylation. In *Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 4: BIOINFORMATICS,*, pages 140–145. INSTICC, SciTePress.

Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

Way, G. P. and Greene, C. S. (2017a). Evaluating deep variational autoencoders trained on pan-cancer gene expression. *arXiv preprint arXiv:1711.04828*.

Way, G. P. and Greene, C. S. (2017b). Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *BioRxiv*.

Xiao, H., Xiao, Z., and Wang, Y. (2016). Ensemble classification based on supervised clustering for credit scoring. *Applied Soft Computing*, 43:73–86.

Zhang, T., Ramakrishnan, R., and Livny, M. (1996). Birch: an efficient data clustering method for very large databases. In *ACM Sigmod Record*, volume 25, pages 103–114. ACM.

Zhong, G., Wang, L.-N., Ling, X., and Dong, J. (2016). An overview on data representation learning: From traditional feature learning to recent deep learning. *The Journal of Finance and Data Science*, 2(4):265–278.