THE CONSTRAINED INFOMAX OBJECTIVE: A UNIFYING PRINCIPLE TO LEARN OPTIMAL REPRESENTATIONS IN BOTH UNSUPERVISED AND SUPERVISED SCENARIOS

VINCENZO CRESCIMANNA

Doctor of Philosophy

Division of Computing Science and Mathematics
University of Stirling

December 2020

DECLARATION

I hereby declare that this dissertation is entirely the result of my own work and includes nothing which is the outcome of any work done in collaboration except where specifically indicated within the text and bibliography.

I also declare that this dissertation (or any significant part thereof) is not substantially the same as any that I have previously submitted, or that is being concurrently submitted, for a degree or diploma or other qualification at the University of Stirling or similar institution.

This dissertation is a record of the work carried out at the University of Stirling between 2017 and 2020 under the supervision of Professor Bruce Graham and Dr Patrick Herbst.

Stirling, December 2020

Vincenzo Crescimanna

ABSTRACT

In the machine learning framework, the data *representation* is defined as a possible description of the visible data, e.g. to describe the number of balls in a table (nine) we can use two equivalent descriptions: either "9" or "IX". Both descriptions are universal representations of the same concept (the number) that can appear in different forms. From this simple example we can see that more than a single representation can be associated to the data. For this reason it is useful to define the characteristics that define the *optimal* representation, the most preferable between all the possible ones. The first property is the *non-ambiguity*, i.e. from a given representation it is possible to reconstruct only one class of visible data (e.g. if we write the digit nine too quickly it could be appear close to an eight, and so this representation is ambiguous); the second important property that characterises the optimal solution is the *shortness*, i.e. the representation should use the shortest code such that the non-ambiguity property is satisfied (e.g. a possible representation of a number is writing down as many points as the number indicates. If the number is one million, one million points have to be drawn: - this representation is not short!). In other words we can assert that the optimal representation of the visible data is the description that contains all the *relevant knowledge* about the data, and not more than that.

Although intuitively clear, the definition of the optimal representation is not unequivocal and it differs from context to context. In this work, utilising the basic concepts of Information Theory – a theory suitable to measure the knowledge stored within the data – we provide a definition of the optimal representation based on information that works in two of the main contexts of machine learning applications: Supervised and Unsupervised Learning. Such a concept is defined via an optimisation principle, the *Constrained InfoMax* (CIM): the optimal representation is from the ones sharing maximal information with the task class, and is the one storing the least knowledge. As this objective is unfeasible-to-compute, to evaluate it quantitatively we provide a variational approximation of the CIM, the *Variational InfoMax* (VIM), – a unique variational objective to learn data representation both in supervised and unsupervised settings. In both settings, the trained models with VIM outperform, on the standard metrics to evaluate the representation quality, the alternative variational models proposed in literature.

This work was firstly motivated by the need to define the optimal representation within the unsupervised setting, and so define the principle to learn it. Indeed, from the introduction of the Variational AutoEncoder (VAE) [37] – the first variational model to learn a possibly non-linear representation, trained optimising the Evidence Lower Bound (ELBO) – two main definitions of optimal representation have been provided: - *disentangled* [28] – separate out the generative factors of the data; and - *informative* [89] – the most informative description about the data.

Following the work done in [5], we observe that actually the two definitions are not in contrast, but only two faces of the same coin. Indeed, an optimal representation of the data has to store all the necessary information to generate the data but none of the generative factors have to be redundant. So the optimal representation, is both informative and disentangled, and it is the one maximising the Constrained InfoMax with respect to the data generation task.

With the help of basic geometric notions, we also observe that the Variational AutoEncoder trained with VIM is a pure inference model: the representation is defined by the encoder (i.e. the inference process) and not by the decoder (i.e. the generative process). To show this, we consider a special Variational AutoEncoder with linear decoder and non-linear encoder, and we observe that, differently from the ELBO trained models that are learning only a linear representation of the data, the VIM trained models are learning a possibly non-linear representation. In particular we observe that the solution of the ELBO trained model is equivalent to the Principal Component Analysis (PCA) one, whereas the VIM trained models are learning a Mixture PCA representation.

Finally, we consider the supervised setting, precisely the classification task, and we observe that the CIM objective is equivalent to the well-known Information Bottleneck [72]. However, its variational description, VIM, better approximates the theoretical objective than the Variational Information Bottleneck [4].

ACKNOWLEDGEMENTS

I would like to thank the University of Stirling CONTEXT Research Programme and the B2B Robo advisor based in Singapore, Bambu, for funding this research.

A special thanks to my supervisor Professor Bruce Graham for his constant support and guidance, and to Dr Patrick Herbst for his continuous encouragement in evaluating new applications and also to Professor Bill Phillips, for his contagious enthusiasm.

LIST OF PUBLICATIONS

The fourth chapter is based on the following peer-reviewed publications:

- Vincenzo Crescimanna, Bruce Graham. *An Information Theoretic Approach to the Autoencoder*. Proceedings of the International Neural Networks Society, vol 1. Springer, Cham. https://doi.org/10.1007/978-3-030-16841-4_10.
- Vincenzo Crescimanna, Bruce Graham. *The Variational Infomax AutoEncoder*. 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, United Kingdom, 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9207048.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Part I

INTRODUCTION

1

INTRODUCTION

1.1 REPRESENTATION LEARNING

In psychology, learning is usually defined as a relatively permanent change in behaviour due to the past experience [16]. This definition is somehow unexpected, since it is addressing visible learning consequences, and not those learning processes which occur inside the brain which, by nature, are hidden. Indeed, the *learning* process can be considered solely as a hypothetical construct as it cannot be directly observed, but rather only inferred from observable behaviour.

Many assumptions have been made in describing the learning process and held as a common belief within cognitive neuroscience, despite being only partially verified by experimental data. The central tenet is that the learning procedure can be described in terms of *mental representations*: a set of hypothetical internal mental symbols that collectively represent external reality. In a restrictive way, the learning process can be thought of as the cognitive process by which the sensory (external) data are linked to their associated representation(s) and then combined to create new concepts.

The first description of how the cognitive representations are made was provided by the English philosopher John Locke in trying to explain the complex ideas, namely those concepts that are not real but rather exist within our minds, such as chimeras [54]. As an illustrative example, let us consider the creation of one of these mythological creatures: the dragon. According to Locke, the imagination (the fundamental learning process) is subdivided into two stages: the representation learning, to associate the sensory data to the idea, and the generation, to combine the ideas to create something new. In the dragon imagination example, the two stages are respectively: first to learn the representations for each visible animal: the bat, and the lizard; and then to combine two of the learnt representations, the bat wings and the lizard body, to create the imaginary animal.

In a living system, the representation learning hypothesis is merely a philosophical assumption. It is difficult to verify since the brain structure is not entirely understood, and the sensory inputs are nearly infinite. In the Artificial Intelligence setting the construct differs, in that the input data is limited, and the architecture is known, the representations are visible, and they can be (at least partially) analysed. For this reason, henceforth, the term *representation* will refer to the visible data description *learnt* via a computational process.

REPRESENTATIONS IN ARTIFICIAL INTELLIGENCE In the Artificial Intelligence (AI) context the representation is a description of the sensory (input) data storing the different explanatory factors (*features*) of the data, that are useful to perform the task for which the AI

was trained. Then, coherently with the philosophical intuition, the representations are a clean picture of the sensory data, where all the possible data-noise or useless knowledge, which are possibly misleading for the task, are removed.

By those properties, as highlighted in [10], the performances of Machine Learning (ML) methods are heavily dependent on the choice of data representation on which they are applied. For this reason, a classic approach to train an AI to perform a specific task was constituted by two steps: to choose the optimal features of the data by hand or via some particular method, and then feed the learnt representations to a machine learning model performing the desired task.

More recently, with the introduction of the (Deep) Neural Networks the two steps are no longer necessary, since a Neural Network (NN), that has been trained to optimise a task-related objective, is learning on its own the representation to perform the task.

Although the NNs simplify the Representation Learning problem, they do not solve it. Indeed, a greedy trained NN is learning non-optimal representations storing too much knowledge about the training data. This is an undesirable scenario since the network is not able to deal with unseen data, i.e. the system is *over-fitting*. A practical way to avoid the over-fitting scenario is to add a regulariser in the learning objective, an additional term pushing the network to learn the representations storing only the most relevant information that are useful to perform the task, and so a minimal representation.

In light of the described behaviour of the neural networks, two fundamental questions of Representation Learning arise:

- What is the optimal representation?
- What is the objective to optimise in order to recover the optimal representation?

To better understand the importance of an optimal representation and the intrinsic difficulty in defining it within a general setting, it is useful to illustrate some simple examples of such representations.

EXAMPLES OF REPRESENTATION The representation was defined as a hypothetical symbol, a description of the data-point living in a different space from the visible one that could be shared between more than one data-point. One of the most important symbols in human history is given by the numerical representations.

Example 1.1.1 (The number)**.** The number is a kind of concept that appears in different forms in real life as any measure is associated with a number; but, in order to handle this entity it is necessary to represent it via a symbol. In our experience, we have observed many different representations. For instance, the naive one, where each natural number is represented by a collection of dots; although this representation is intuitive it is not very simple to handle. A smaller, easier to handle, representation is the Roman one. Still, in this case, it is not simple to perform even such simple operations as the sum. Finally, the most common representation, the Arabic one, that is both compact and useful for many operations, by its positional properties.

All the representations are good (*sufficient*): they store the relevant knowledge about the data since from any of these, it is possible to recover the number. However, depending on the context, we prefer one representation to another. For example, in teaching, the optimal is the dot representation. At the same time, for elegance, we prefer the Roman description, and for everyday use, we prefer the Arab numbers since they have a compact description and separates the factors, i.e. each element of a different class (the tens, the hundreds) are separated from each other, and so they are easy to recognise.

A slightly more complex example of widely used representation in numerical computations is the series approximation of functions.

Example 1.1.2 (Principal Factors)**.** By the Taylor theorem any smooth function $f : \mathbb{R} \to \mathbb{R}, x \mapsto f(x)$, can be represented, around the point x_o, by the Taylor sum:

$$f(x) = \sum_{i=0}^{\infty} \frac{d^i f(x_o)}{dx^i}(x - x_0)^i. \tag{1.1}$$

In general, this infinite sum is intractable to compute exact, but for practical reasons, a truncated sum is taken as a good approximation of the function around the point x_0.

The Taylor sum describes just locally the function f. If we are interested in a global approximation of the function, for instance we want to compute the integral, it is necessary to consider a different description. The classic approach, first used by Riemann, is to approximate locally the function by a constant map, i.e. $\hat{f}(x) = c_i$ for $x \in \Delta_i = [x_i, x_{i+1}]$, in this way the function f can be represented as

$$\hat{f}(x) = \sum_{i=0}^{N} c_i \delta(x \in \Delta_i). \tag{1.2}$$

From this example, we see that a good representation depends on the task, and, in general, it does not have to contain all the possible knowledge about the visible data. Indeed, we accept that we may lose some information to gain a compact description wherein just the principal components appear.

In the first two examples, we have shown that an optimal representation depends on the task. In general, a task is not provided, and the problem of representation learning can be described, geometrically, as a hidden manifold learning.

Example 1.1.3 (Hidden Manifold)**.** Data in the same set, by definition, share some common properties, and then intuitively, we can assert that they are connected: there is a line connecting them. If the line is often changing in direction it could be preferable to embed the line in a manifold, the *hidden generative* manifold of the data, storing all the visible data and their possible (feasible) combination of them. To illustrate that scenario, let us come back to the dragon generation example. The animals' data-set is a line that changes direction at least twice in the bat and the lizard directions. These two directions do not cover the the dragon, since it is not a real animal. Indeed, the dragon, as any other imaginary animal, lives in the space generated by the two directions, i.e. it lives in the embedding manifold.

From this example, we see the importance to learn the principal directions of the data, as the goal is not just to reconstruct the visible data, but also to generate possibly new, unseen, data.

By these experiments, we see that to formally define an optimal representation for any context formally and then define the objective to optimise the learning of such a representation is not a trivial process. Motivated by the relevance and the general difficulty of the problem, this thesis aims to address the fundamental questions of representation learning in the Neural Network setting.

1.2 CONTRIBUTION AND SUBDIVISION OF THE WORK

As seen by the simple examples cited above, representation learning is not a simple task. Indeed, there are many possible representations associated with the same entity, and the optimal one depends on the context. Thus, in principle, the optimal definition is an ill-posed problem.

In light of the problem complexity, within this thesis, we consider *unsupervised*, where no task is provided, and *supervised*, wherein the task is provided, as separate scenario.

By means of this separated description, we show that an unsupervised optimal representation is the least informative representation storing, as it does, all the knowledge to recover the data. The optimal representation in the supervised setting is an optimal unsupervised representation storing the information about the task. Equivalently, the optimal representation is the one satisfying the proposed *Constrained InfoMax* (CIM) principle: wherein the optimal representation is the one with the smallest entropy such that the information shared with the task is maximal, whereas the generation is considered the implicit task in the unsupervised setting.

The definition of the CIM principle, a unifying principle for optimal representation for both unsupervised and supervised setting, which is the main contribution of the thesis, leads us to the following collateral contributions:

- The Wasserstein distance (Wass) and the optimal β-ELBO, two objectives used to train the Variational AutoEncoder (VAE) are both a variational lower bound of the CIM.
- The optimal (unsupervised) representation is the one storing as little information as possible and not the one sharing as little information as possible of the visible data. In other words, to constrain the latent entropy in the Wass is more effective than increasing the β in the ELBO.
- The Wass and the ELBO are informative equivalents but not geometrically so. Indeed, the ELBO trained model is discovering only the closest plane fitting the data-points, while the Wass trained model is learning a generic latent manifold.
- In the supervised setting, the CIM is equivalent to the Information Bottleneck principle, but the two associated variational objectives Variational InfoMax (VIM) and Variational

Information Bottleneck (VIB) respectively, are not equivalent: the VIM representations are discovering the hidden data-structure, and the associated network has higher accuracy than the VIB trained network.

1.2.1 *Subdivision of the work*

The main findings as outlined above are derived throughout the thesis. After the first introductory chapters, wherein the general framework is described, we first derive the CIM within the unsupervised setting, describing its connection with such classic models as the Principal Components Analysis, and then it is extended to the supervised setting.

Going into further detail, the rest of this thesis is structured as follows.

- Chapter 2: Introduction to Bayesian and Variational inference, and the main concepts of Information Theory. The chapter is devoted to illustrating the basic concepts of the inference theory in the probabilistic setting and its connection with the information theory.

- Chapter 3: Review of the principal methods to learn representations. The first part describes the Principal Component Analysis, the Gaussian Process methods, and the relationship between them. The second part introduces the Neural Networks and, in particular, the AutoEncoder Network and its stochastic version: the Variational AutoEncoder (VAE), a network maximising the variational lower bound of the visible log-likelihood: the ELBO.

- Chapter 4: Information theoretical analysis of the ELBO-related issues, and introduction of the CIM principle. We observe that it is necessary to maximise the information arising between the visible data and the representation maintaining bounded the entropy of the latter, to have both optimal inference and generative performance. The theoretical description is useful to highlight the equivalence between the optimal ELBO and the Wass objectives and also the importance of a parameter that is often forgotten, namely the entropy of the prior. Indeed, with the help of computational experiments, we see that a fruitful strategy to learn more disentangled representations is to control the latent entropy.

- Chapter 5: Description of the Wass and ELBO solutions in terms of the classic ML methods. By means of a geometric informative analysis, we observe that, in the linear decoder VAE setting, the ELBO trained model is equivalent to the PCA. Instead, the Wass trained model is related to the Mixture of PCA. That means, geometrically, that for the ELBO the hidden manifold is always a plane; whereas the Wass approach allows the network to learn more flexible manifolds. The higher flexibility of the Wass objective is also highlighted from a Gaussian Process perspective, wherein it is observed that the Wass latent variables are associated with the optimal encoding kernel of the Kernel PCA.

- Chapter 6: Generalisation of the CIM principle to the supervised setting. Through an information analysis of the stochastic network, we observe that the CIM is equivalent to the Information Bottleneck principle; in that both are learning a Minimal Sufficient representation of the data with respect to the task. The advantage in considering the CIM principle is two-fold: theoretically, it is connecting the optimal solution with the optimal kernel of the associated GP problem and, computationally, the CIM allows us to define a novel variational objective: the Variational InfoMax (VIM), an objective that does not put any constraint upon the shape of the encoding map, differently from what was done by the variational objective associated to the Information Bottleneck.
- Chapter 7: Conclusion, final remarks and future work. After a summary of the novel contributions of the thesis is followed by its possible applications for future work In particular, we outline a possible extension of the CIM principle to life-long learning, the scenario where the network has to perform more than one task.

2

BAYESIAN AND VARIATIONAL INFERENCE: A SMOOTH INTRODUCTION

In this chapter, we introduce the fundamental concepts of Bayesian and Variational Inference and their relationship with Information theory. As an introductory chapter, devoted to the reader that is not familiar with these concepts, we follow a tutorial approach, introducing each new concept from an example. These descriptions are strongly based on two seminal publications: - Murphy [62] and - Tipping [77] , to which we point the reader for a complete description of that big field.

2.1 PRELIMINARIES AND NOTATION

We use calligraphic letters (i.e. $\mathcal{X}$) for sets, lower case Latin letters for vectors (i.e. $x \in \mathbb{R}^D$), and capital letters (i.e. X) for matrix with columns x_i (i.e. $X = [x_1^T, \dots, x_N^T]^T \in \mathbb{R}^{N \times D}$). As the matrix X can be seen as the generative matrix of the sample x_i, with abuse of notation the capital letter is used also to denote the random variable from which the data-points are sampled. We denote density distribution with the lower case letter p (i.e. $p(x)$), and we use the notation p_f to denote the distribution associated to the function f. For the variational density, we use the letter q, and only on the optimal case $q(x) = p(x)$.

2.2 FROM DETERMINISTIC TO BAYES INFERENCE

Given a dataset $\mathcal{D} = \{(x_i, y_i)\}$ the goal of a machine learning model is to find a function f modelling the distribution $p(y, x)$, such that $f(x) = p(y|x)$ for any x. In the assumption the function f is *parametric*, i.e. can be described by some parameters w, e.g. $f(x) = w \cdot \phi(x)$, with ϕ a certain prior function. In this setting, the goal is to learn the optimal (weights) parameters w^*, such that $w^* \cdot \phi(x) \sim p(y|x)$.

THE LINEAR MODEL Let us start with the assumption that there exists a deterministic map relating the visible data $\{x_i\}_i^N$ with the associated regressor output $\{y_i\}_i^N$, i.e. $y_i = f(x_i)$. For the sake of simplicity, let us assume known the feature function $\phi = [\phi_1, \dots \phi_k]$, and that the function f is linear in the parameter $w = [w_1, \dots, w_k]$: a function that is linear combination of the features ϕ_i,

$$f(x) = w \cdot \phi(x) = \sum_i^K w_i \phi_i(x). \tag{2.1}$$

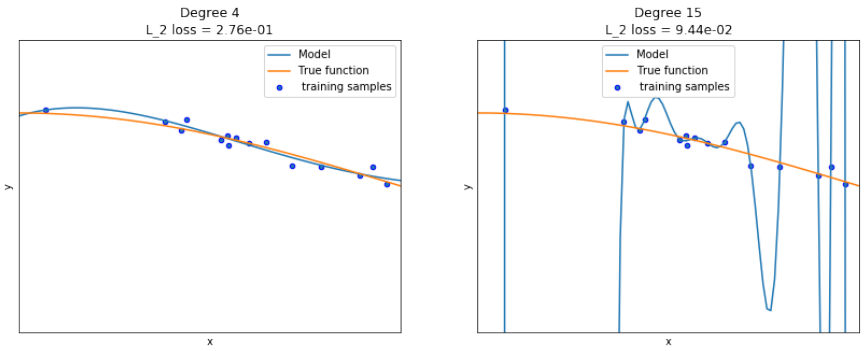


Figure 2.1 Example of over-fitting, assuming a no-noise data, the fitting is *following* the noise

The most intuitive idea to learn a parameter w^* such that for any x_i the function $f_{w^*}(x_i)$ is *close* to y_i, to learn a parameter w minimising the distance

$$L(w) = \sum_i^N \|y_i w \cdot \phi(x_i)\|_p^p, \tag{2.2}$$

where the norm $\|\cdot\|_p$ denotes the l_p-norm defined as $\|x\|_p^p = \sum_i |x_i|^p$. By its smoothness and equivalence with the euclidean distance, common choice is the L_2 distance $(p = 2)$, in this case the problem has a unique solution and the optimal parameter w^* has the closed form:

$$w^* = (\Phi^T \Phi)^{-1} \Phi^T y, \tag{2.3}$$

where Φ is the matrix having as (i, j) entries the quantity $\phi_j(x_i)$ and $y = [y_1, \ldots, y_n]$ the vector of the labels.

The approach described above is entirely deterministic (i.e. we are assuming that there is no noise in the data, and the relationship can be entirely described by the function f). Often that scenario is too optimistic and indeed is practically impossible in real life. The risk of the deterministic approach is, as we can see from figure 2.1, that the solution over-fits the data (i.e. the model predicts exactly the visible training data). It is also predicting the intrinsic noise, with few chances that such a model will work with new unseen (test) data-points.

REGULARISATION A model $\hat{f}$ that is overfitting the data, is considered a complex model; in that the model is describing the visible data better than the unknown original model f. For this reason, to avoid having to learn a too complex system, it is reasonable to add a penalty term, a *regulariser* $R(w)$, to the (2.2) objective in order to penalise the model complexity. In light of that observation, in practical cases, the objective to optimise has the following form

$$E(w) = L(w) + \lambda R(w), \tag{2.4}$$

where $\lambda > 0$ is an objective hyper-parameter, asserting how regular has to be the model; this parameter has to be tuned in the validation stage.

How to choose the regulariser term? A complex model is a function that is often changes its slope or, more formally, which has second derivatives larger than 0; conversely, we state that a model is simple if it changes directions a few times (zero second derivatives). For

this reason, a common choice for the regulariser is the one penalising the second derivative, $R(w) = \|\nabla^2 f(x)\|_2^2$, which serves as a measure of the roughness of the function.

In the following section, if not asserted differently, we will consider as a feature map the Radial Basis Function (RBF), $\phi_i(x_j) = \exp(-(x_j - x_i)^2)$. By this choice the second-derivative regulariser is constant equal to 0. Thus a common choice for the regulariser is the L_2-norm of the parameter, $R(w) = \|w\|_2^2$. Under this assumption the objective E in (2.4) is referred as the Penalised Least Squares (PLS) and it is optimised by

$$w_{PLS} = (\Phi^T \Phi + \lambda I)^{-1} \Phi y. \tag{2.5}$$

2.2.1 *A probabilistic approach*

Until now, we have discussed the deterministic setting: wherein the target y is a deterministic function of x, as the approach works only in the noise-less assumption, we have seen it is favourable to add a penalty term to have a smooth regressor and avoid over-fitting. The issue with this approach is that we do not have an estimation of how good the regression is, or in other words what is the probability (likelihood) of a correct prediction? A solution to overcome such an issue is to consider a probabilistic perspective.

A probabilistic regressor is a model defined as

$$y_i = w \cdot \phi(x) + \varepsilon, \tag{2.6}$$

where to the standard linear model was added a noise ε distributed according a certain distribution $p(\varepsilon)$.

Assuming the data has been generated independently and the noise normally distributed with variance σ^2, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, by linearity of Gaussian distribution, it is possible to define the *likelihood* of the data, $p(y|x, w.\sigma)$, as:

$$\begin{aligned} p(y|x, w, \sigma) &= \prod_i p(y_i|x_i, w, \sigma) \\ &= \prod_i \mathcal{N}(y|f(x_i), \sigma^2) \\ &= \prod_i (2\pi\sigma^2)^{-1/2} \exp\Big[-\frac{(y_i - f(x_i))^2}{2\sigma^2}\Big]. \end{aligned} \tag{2.7}$$

By the closed form of the likelihood to see y given x, the first idea is to learn the parameter maximising such a quantity, or better the logarithm of the likelihood (the log-likelihood) in order to have a convex optimisation problem. The log-likelihood of (2.7) is proportional (equal up to a constant) to the Least-Squares objective in (2.2), such that, as observed above, it is prone to over-fitting. This relationship between the MLE and the Least Square objective highlights that the log-likelihood is not the correct objective to optimise. Indeed, we have to take into account that the weights w are, in principle, distributed according to a specific prior distribution $p(w)$, and we therefore expect that the optimal weights are somehow close to the original form.

By this observation, assuming we know the prior distribution $p(w)$, it is possible to define the optimal weight as the one optimising the *posterior* $p(w|\mathcal{D})$, the probability to have the parameter w given the data-set $\mathcal{D}$. That quantity can in principle be computed thanks to the *Bayes formula*:

$$p(w|\mathcal{D}) = \frac{p(y|x,w)p(w)}{p(y)} = \frac{\texttt{likelihood} \times \texttt{prior}}{\sum_{w_i} p(y|w_i)p(w_i)} \tag{2.8}$$

Assuming that the prior be normally distributed:

$$p(w) = \prod_k^K \sqrt{\frac{\alpha}{2\pi}} \exp\left(\frac{\alpha}{2} w_k\right), \tag{2.9}$$

with the hyper-parameter α, measuring the variance of the prior, it is possible to compute easily the posterior.

Indeed, by means of the Gaussian conditional property, given the distributions

$$p(x) = \mathcal{N}(\mu_x, \Sigma_x), \qquad p(y|x) = \mathcal{N}(Ax + b, \Sigma_y) \tag{2.10}$$

the posterior is also Gaussian, $p(x|y) \sim \mathcal{N}(\mu, \Sigma)$ with

$$\begin{aligned} \mu &= \Sigma[A^T \Sigma_y^{-1}(y - b) + \Sigma_x^{-1}\mu_x] \\ \Sigma^{-1} &= \Sigma_x^{-1} + A^T \Sigma_y^{-1} A. \end{aligned} \tag{2.11}$$

Thus, in our case the posterior distribution $p(w|\mathcal{D})$ is Gaussian with mean μ and covariance Σ:

$$\begin{aligned} \mu &= (\Phi^T \Phi + \sigma^2 \alpha I)^{-1} \Phi^T y \\ \Sigma &= \sigma^2 (\Phi^T \Phi + \sigma^2 \alpha I)^{-1}. \end{aligned} \tag{2.12}$$

Before we proceed further, let us observe that the average value of the posterior (i.e. the most likely value), is precisely the one learned by the PLS in (2.4), wherein the hyper-parameter λ introduced without proper meaning, is substituted with the product of the variance term $\alpha\sigma^2$. This is perhaps not so surprising, as the prior condition in the probabilistic description it is enforcing aims to consider a smoother likelihood $p(y|x)$. For the same reason for which we introduced the (explicit) regulariser in the PLS objective (2.4).

The advantage of the probabilistic approach over the PLS is its interpretability, since the regulariser comes out naturally. Thanks to the Bayesian principle, we are also able to *infer* the distribution over w, additional information which is not made available via a deterministic process. Unfortunately, such an approach is often unfeasible to compute, as the inversion of matrix is not cheap, and the Gaussian assumption, which until now we have heavily employed does not hold in any context.

For this reason, most of the time we limit ourselves to finding the parameter maximising the posterior, or Maximum a Posteriori (MAP), w_{MAP}. Obviously, if we know the posterior distribution, the MAP estimate can be recovered, but it is possible to infer it without knowing the posterior. Indeed, since the posterior is proportional to the prior times the likelihood,

and the denominator does not depend upon this parameter, the weight optimising the log posterior $\log p(w|\mathcal{D})$ is the one maximising the sum of the log of prior and its likelihood

$$\begin{aligned}\log p(w|\mathcal{D}) &\propto \log p(y|x,w) + \log p(w) \\ &= -\frac{1}{2\sigma^2}\sum_i y_i - w\cdot\phi(x_i) - \frac{\alpha}{2}\|w\|_2^2.\end{aligned} \tag{2.13}$$

Unsurprising we achieve the same objective, up to sign, of the PLS in (2.4).

2.3 DENSITY ESTIMATION

In the previous section, we considered that the regression problem, taken with the classification task, is a classic example of supervised learning: namely the problem of ascribing to each data-point x a value y. In real-world setting, the correct label y is often not available, because labelling procedure is expensive and biased by the human annotator (e.g. two dog images could be labelled as "dog" by on annotator and with two different races by another annotator). For this reason, it is particular important to consider the setting where no labels are provided. Such a family of problems is collectively referred to as *'unsupervised'*. A particularly interesting unsupervised task is the density estimation wherein, given a set of data-points $\mathcal{D} = \{x_i\}_i^N$ independent identical distributed (iid), it is possible to estimate from which density $p(x)$ they are sampled, i.e. to find θ such that $x_i \sim p(x|\theta)$ for any $x_i \in \mathcal{D}$.

This task is not relevant solely from a theoretical standpoint. Indeed, the knowledge of the hidden distribution allows us to *generate* new data-points: and the set of visible points is just a small subset of all possible points. Knowing the parameter θ allows us to generate new data-points where necessary, and then to discover the generative factors of the data. One interesting application of such knowledge is anomaly detection, in which it is assumed that we know the generative distribution $p(x|\theta)$ for any given data-point x_i so we can check if the point is an anomaly, $p(x_i) < \delta$, or a legit point sampled by $p(x)$.

In the following section, in agreement with what we have done over and above using linear regression, we will limit ourselves to consider a density described by a linear model known as the univariate Gaussian density.

UNIVARIATE GAUSSIAN DENSITY ESTIMATION Let $\mathcal{D} = \{x_i\}$ represent a set of uni-dimensional data, $x_i \in \mathbb{R}$, which is iid sampled from distribution $p(\mathcal{D}) = \prod p(x_i)$. By means of the Central Limit Theorem, we can assume that it is sampled via a Gaussian distribution:

$$p(\mathcal{D}|\mu,\sigma) = \prod_i (2\pi\sigma^2)^{-1/2} \exp\Big[-\frac{1}{2\sigma^2}(x_i-\mu)^2\Big]. \tag{2.14}$$

Such a distribution depends on two parameters, namely μ and σ. Once inferred, we are able to completely describe their distribution $p(x)$. By that property, the parameter $\theta = (\mu,\sigma)$ is called *generative factor* since, given θ, it is possible to generate any data-point in $\mathcal{D}$.

The goal is to find a posterior $p(\theta|\mathcal{D})$, and select θ that maximises the posterior. To estimate the posterior we need firstly to define the prior $p(\theta)$.

In this manuscript, we assume that σ is known, and then it is necessary to estimate only the mean μ. For the sake of simplicity, as above, we assume the $p(\mu)$ Gaussian distributed with mean m_0 and variance τ^2:

$$p(\mu) = (2\pi\tau^2)^{-1/2} \exp\Big[-\frac{1}{2\tau^2}(\mu - \mu_0)^2\Big]. \tag{2.15}$$

The special selection of the prior allows us to compute a closed form for the posterior $p(\mu|\mathcal{D})$, which is also Gaussian $\mathcal{N}(m, s)$, with parameters:

$$\begin{aligned} m &= s\Big[\frac{1}{\sigma^2}\Big(\sum_i x_i\Big) + \frac{1}{\tau^2} m_0\Big] \\ s &= \Big(\frac{N}{\sigma^2} + \frac{1}{\tau^2}\Big)^{-1}, \end{aligned} \tag{2.16}$$

where the computation follows naturally by its Gaussian properties.

By the estimation above, we have determined that the MAP estimator for μ is the average value above, $\mu_{MAP} = m$.

2.3.1 *Mixture of models*

In the section above, we have seen how simple it is to perform a Bayes inference when the distribution to estimate is the univariate Gaussian. However, the univariate Gaussian distribution is not often optimal. For instance, if we consider any labelled data-set, say the set of handwritten digits, to assume that the handwritten digits are sampled by a normal distribution is unrealistic and not particularly informative. For example, what does μ mean? A more suitable distribution is a multi-variate one, wherein each data-point lies around a specific value, defining the mean value of each digit. To describe that distribution, we have to introduce *Mixture models*.

From another perspective, we are asserting that the identically distributed assumption (the second i in iid) is often too strong. Indeed, within the same data-set, the two classes of data-point are distributed differently.

From this point on, let us consider the more general setting wherein the visible data $\mathcal{D}$ is the union of K different subsets $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$ with each element in $\mathcal{D}_k$ sampled according the distribution $p_k(x)$ (i.e. the data-points are sampled from a *mixture distribution*, $p(x) = p_k(x)$ if $x \in \mathcal{D}_k$.)

Analytically, this is equivalent to stating that there exists a K-value random latent vector z, with $p(z) = \mathrm{Cat}(\vec{\pi})$ and $\sum_i \pi_i = 1$, such that

$$p(x|\theta) = \sum_i^K p(x|\theta_i) p(z_i) \tag{2.17}$$

Where, although each model $p(x|\theta_i)$ is independent, their sum depends by each latent factor z_i, because the *latent* variable, is shared by all models.

Let us note that the mixture of models $p(x|\theta)$ defines the distribution from which the data is sampled independently and identically; then it is possible to apply the Bayes inference

with respect the parameters $\theta = [\theta_i]$ and z. Indeed, in departure from the unimodal setting described above, we introduce a new parameter to estimate, namely the categorical parameter z.

The first idea of how to estimate the model is to proceed as we have done until now: i.e. to infer the two variables θ, z, although this is not feasible since the distribution $p(\theta, z|\mathcal{D})$ does not factorise as the product of $p(\theta|\mathcal{D})p(z|\mathcal{D})$: and thus it is necessary to find an approximated approach. Since the inference problem of the mixture of the model differs solely for the inference of the latent variable z, it is reasonable to think that knowing z, it is possible to estimate θ. But, as observed in [62] (11.3), in this case, the inference solution is undefined since there is no unique solution.

THE EXPECTATION MAXIMISATION ALGORITHM Since the pure Bayes inference is impossible to apply, another option is to limit ourselves to finding the MAP estimator θ. Indeed, the MAP task is more straightforward than the pure Bayesian approach since, as observed in (2.13), it is sufficient to optimise the likelihood $L(\theta)$ and the prior term $p(z)$. In the following section we will discuss the two terms separately. Let us start with the log-likelihood:

$$L(\theta) = \sum_{i}^{N} \log p(x_i|\theta) = \sum_{i}^{N} \log \Big[\sum_{z_i} p(x_i, z_i|\theta) \Big]. \tag{2.18}$$

This quantity, by the sum inside the log, is not easy to maximise, and it can be proved (e.g. [62] chap 11), that such an objective is not convex.

For that reason, it is necessary to consider an approximated MAP; let us start by considering the *complete data log-likelihood*

$$L_c = \log p(x, z|\theta) = \sum_{i}^{N} \log p(x_i, z_i|\theta). \tag{2.19}$$

That objective, assuming $p(x, z)$ belongs to the exponential family, is a convex function. Thus, to compute that quantity we need to guess z, an unknown variable. To overcome this issue, we might consider the Expectation Maximisation (EM) technique: a two-step iterative process which performs an expectation (E) step in order to avoid removing dependence upon the unknown variable, and a maximisation (M) step, which computes the parameters so as to maximise the expected log-likelihood.

In more detail, the EM firstly computes the *expected complete data log-likelihood* with respect the actual parameter θ_{t-1}:

$$Q(\theta, \theta_{t-1}) = \mathbb{E}_{q(z|\theta_{t-1})}[L_c(\theta)], \tag{2.20}$$

and then *maximise* the *auxiliary function* $Q(\theta, \theta_{t-1})$, that does not depend on z, to obtain the new parameter θ_t,

$$\theta_t = \arg\max_{\theta} Q(\theta, \theta_{t-1}). \tag{2.21}$$

Thanks to the EM algorithm, we have a way in which to learn the parameters that maximise the log-likelihood of the model. Since we are looking for the MAP estimator, and the log-prior $p(\theta)$ does not depend by z, it is sufficient to add the log-prior within the maximisation step:

$$\theta_t = \arg\max_\theta Q(\theta, \theta_{t-1}) + \log p(\theta). \tag{2.22}$$

EM FOR GAUSSIAN MIXTURE MODEL As an illustrative example we apply the EM for a Gaussian Mixture Model (GMM), a Mixture Model wherein the distribution functions are all Gaussian:

$$p_k(x|\theta) = \mathcal{N}(\mu_k, \Sigma_k). \tag{2.23}$$

The first step necessary to proceed with the EM is to compute the Auxiliary function $Q(\theta, \theta_{t-1})$,

$$\begin{aligned} Q(\theta, \theta_{t-1}) &= \mathbb{E}_{p(z)}\Big[\sum_i p(x_i, z_i|\theta)\Big] \\ &= \sum_i \mathbb{E}\Big[\log\Big[\prod_k \pi_k p_k(x_i|\theta)^{I(z_i=k)}\Big]\Big] \\ &= \sum_i \sum_k p(z_i = k|x_i, \theta_{t-1}) \log(\pi_k p_k(x_i|\theta)) \\ &= \sum_i \sum_k r_{ik}(\log \pi_k + \log p(x_i|\theta). \end{aligned} \tag{2.24}$$

The goal of E-step is to compute Q or, more precisely, $r_{ik} = p(z_i = k|x_i)$, being the only term that depends by θ_{t-1}. Here r_{ik} represents the *responsibility* that cluster k takes for data-point i, and it is defined as

$$r_{ik} = \frac{\pi_k p_k(x_i|\theta_{t-1})}{\sum_{k'} \pi_{k'} p_{k'}(x_i|\theta_{t-1})}. \tag{2.25}$$

Until now we have not used any information about the shape of the generative distributions, since the responsibility term has the form described above for any Mixture of Models.

The goal of the M-step is to optimise the auxiliary function Q for the parameters π and θ.

Let us start with π, since $Q[\pi] = \sum_i \sum_k r_{ik} \log \pi_k$, we have that the optimal π is

$$\pi_k = \frac{1}{N} \sum_i r_{ik}. \tag{2.26}$$

To estimate μ_k and Σ_k, it is necessary to rewrite the function Q in terms of these two variables:

$$\begin{aligned} Q[\mu_k, \Sigma_k] &= \sum_i \sum_k r_{ik} \log p_k(x_i|\theta) \\ &= -\frac{1}{2} \sum_k r_{ik}[\log|\Sigma_k| + (x_i - \mu_k)^T \Sigma_k^{-1}(x_i - \mu_k)], \end{aligned} \tag{2.27}$$

a function that is maximised by

$$\begin{aligned} \mu_k &= \frac{\sum_k r_{ik} x_i}{\sum_i r_{ik}} \\ \Sigma_k &= \frac{\sum_k r_{ik} x_i x_i^T}{\sum_i r_{ik}} - \mu_k \mu_k^T \end{aligned} \tag{2.28}$$

In the Mixture of Models setting, we observed that it is unfeasible to compute a posterior for the parameters ascribed, and we have to limit ourselves to consider a MAP estimate, one which is computed via the EM algorithm. But sometimes it could be useful to have at least an approximation of the posterior distribution. For example, let us consider the regression case. Given a new data-point x^*, we do not want to know just the predicted value, but also the expected accuracy of such a prediction $p(y^*|x^*)$, which is given by integrating the likelihood

$$p(y^*|x^*) = p(y^*|x^*, \mathcal{D}) = \int p(y^*|x^*, w, \mathcal{D}) p(w|\mathcal{D}). \tag{2.29}$$

Since the posterior $p(w|\mathcal{D})$ is not often feasible to compute (until now we have considered only a special case wherein the distribution is Gaussian and there exists a closed-form), the only feasible approach is to approximate it. A general class of approximations that works essentially within all contexts is *variational inference* in which we choose a distribution $q(\theta)$ from a known tractable family, $q \in \mathcal{Q}$, and then try to make this distribution as close as possible to the true posterior $p(\theta|\mathcal{D})$. Before proceeding further, let us give a brief overview of the possible metrics in a distribution space.

METRICS IN THE DISTRIBUTION SPACE In a classic normed vector space $V = (V, \|\cdot\|)$, we say that for any two vectors v and w the natural distance is defined as $\|v - w\|$, and we call the metric $d : d(v, w) = \|v - w\|$, the induced metric by the norm $\|\cdot\|$. The induced metric is not the only possible one, indeed a metric in V is a function $d : V \times V \to \mathbb{R}_+$ that satisfies the following three properties: [1] positive definiteness $d(v, w) \geqslant 0$ for any $v \neq w$; [2] symmetry $d(v, w) = d(w, v)$ and, [3] the triangle inequality $d(v, w) \leqslant d(v, u) + d(u, w)$. In the case of the distribution space, the space of the distributions in the vector space V is not a normed space, and the most popular metric between the distributions p and q is the k-Wasserstein distance W_κ defined as

$$W_\kappa(p, q) = \inf_{\gamma \in \Gamma(p,q)} \int_V \|x - y\|^\kappa d\gamma(x, y), \tag{2.30}$$

where $\Gamma(p, q)$ denotes the collection of all measures on $V \times V$ with marginals p and q on the first and second factors, respectively. With the infimum in the definition, the metric cannot be computed in real settings, for this reason in machine learning the most popular "distance" function is a divergence: the Kullback-Leibler (KL) divergence, which is a non-symmetric function that does not satisfy the triangle inequality. The KL-divergence defined as

$$D_{KL}(q\|p) = \int q(z) \log \frac{q(z)}{p(z)} dz, \tag{2.31}$$

could be computed analytically without too much effort and, as we will see in the following section, has a clear information interpretation.

THE EVIDENCE LOWER BOUND Letting $\mathcal{Q}$ be the space of all tractable distributions $q(\theta)$, the goal of Variational Inference (VI) is to find the closest $q \in \mathcal{Q}$ to the intractable $p(\theta|\mathcal{D})$:

$$q^*(\theta) = \arg\min_{q \in \mathcal{P}} D_{KL}(q(\theta)||p(\theta|\mathcal{D})). \tag{2.32}$$

Unfortunately, that objective is unfeasible to compute since it would require us to compute the posterior $p(\theta|\mathcal{D})$ point-wise. For this reason we have to optimise an approximation – an upper bound – of (2.32). The idea is to substitute the intractable $p(\theta|\mathcal{D})$ with the unnormalised distribution $p(\theta, \mathcal{D}) = p(\theta|\mathcal{D})p(\mathcal{D})$ so that, in general, it is simpler to compute the posterior. Thus, the new objective that is feasible to compute is:

$$J(\theta) = D_{KL}(q(\theta)||p(\theta, \mathcal{D})), \tag{2.33}$$

an upper bound of the original objective (2.32) yet the two minima coincide,

$$\begin{aligned} J(\theta) &= D_{KL}(q(\theta)||p(\theta, \mathcal{D})) \\ &= \int q(\theta) \log \frac{q(\theta)}{p(\theta|\mathcal{D})p(\mathcal{D})} \\ &= D_{KL}(q(\theta)||p(\theta|\mathcal{D})) - \log p(\mathcal{D}), \end{aligned} \tag{2.34}$$

where the equivalence of the minima is followed by the observation that $p(\mathcal{D})$ is a constant of the system.

Let us note that the objective in (2.34) is strictly related to the EM-algorithm where it is optimised to the auxiliary function $Q(\theta, \theta_{t-1}) = \mathbb{E}_{q(z|\theta_{t-1})}[\sum_i \log p(x_i, z_i|\theta)]$. Remembering that by Bayes formula $p(x_i, z_i|\theta) = p(x_i|z_i, \theta)p(z_i|\theta)$, the auxiliary formula can be rewritten equivalently as

$$\begin{aligned} Q(\theta, \theta_{t-1}) &= \mathbb{E}_{q(z|\theta_{t-1})}\Big[\sum_i \log p(x_i|z_i.\theta)p(z_i|\theta)\Big] \\ &= \mathbb{E}_{q(z|\theta_{t-1})}\Big[\sum_i \log p(x_i|z_i.\theta) + \log p(z_i|\theta)\Big] \\ &= \mathbb{E}_{q(z|\theta_{t-1})}\Big[\sum_i \log p(x_i|z_i.\theta)\Big] - D_{KL}(q(z|\theta)||\log p(z_i|\theta)), \end{aligned} \tag{2.35}$$

an expression really close to the objective in (2.34). By this observation, we can think of the VI inference as being a theoretical framework by which we may apply the EM in a single step.

As an introductory chapter, devoted to sketching the main ideas underlying Bayesian and Variational inference, we apply the VI in a specific context; namely a mean-field approximation for the univariate Gaussian estimation. For a general review, we refer to chapter 11 of [62].

VARIATIONAL INFERENCE FOR UNIVARIATE GAUSSIAN In the preceding section we discussed the density estimation and provided a posteriori for the mean, assuming that the variance σ^2 is a constant. Although it is possible to provide a full Bayesian approach to evaluate the posterior of σ in the following section, for illustrative purposes, we estimate the posterior via VI. Following the same notation in [62] let us define $\lambda = \sigma^{-2}$, and assume that the prior for the parameters μ, λ is

$$p(\mu, \lambda) = \mathcal{N}(\mu|\mu_0, (\kappa_0\lambda)^{-1})\Gamma(\lambda|a_0, b_0) \tag{2.36}$$

where $\Gamma(\cdot,\cdot)$ denotes the Gamma distribution, and the parameters defining the prior are supposed known. The variational distribution has to be tractable, for this reason a common assumption is to consider a factorized approximation, $q(\theta) = \prod_i q(\theta_i)$, that in our specific case means:

$$q(\mu,\lambda) = q(\mu)q(\lambda). \tag{2.37}$$

The choice of a factorised distribution, the so called *mean field* method, it is particular useful to have an easy to compute inference:

$$\log q(\theta_i) = \sum_{j\neq i} \mathbb{E}_{q(\theta_j)}[\log p(\theta,\mathcal{D})]. \tag{2.38}$$

In the following we will observe that this update holds for the univariate Gaussian, but a general derivation could be found in [62].

The approximated inference in (2.32) could be written as

$$\begin{aligned} L(q) &= \int q(\lambda)q(\mu)\Big[\log p(\mu,\lambda,\mathcal{D}) - \log q(\mu)q(\lambda)\Big]d\lambda d\mu \\ &= \int q(\lambda)\int q(\mu)\log p(\mu,\lambda,\mathcal{D})d\lambda d\mu \\ &- \int q(\lambda)\int q(\mu)\Big[\log q(\mu) + \log q(\lambda)\Big]d\lambda d\mu \\ &= \int q(\lambda)\mathbb{E}_{q(\mu)}[\log p(\lambda,\mu,\mathcal{D})] - \mathbb{E}_{q(\mu)}[\log q(\mu)] - \mathbb{E}_{q(\lambda)}[\log q(\lambda)]. \end{aligned} \tag{2.39}$$

From the equation above, we see that the objective is maximised when each marginal, say $q(\lambda)$, is equal to the average of the unnormalized posterior with respect the other variable,

$$\log q^*(\lambda) = \mathbb{E}_{q(\mu)}[\log p(\mu,\lambda,\mathcal{D})]. \tag{2.40}$$

In light of this useful property, in order to compute the variational inference it is sufficient to compute the unnormalized log posterior,

$$\begin{aligned} \log p(\mu,\lambda,\mathcal{D}) &= \log p(\mathcal{D}|\mu,\lambda) + \log p(\mu|\lambda) + \log p(\lambda) \\ &= \frac{N}{2}\log\lambda - \frac{\lambda}{2}\sum_i^N (x_i-\mu)^2 - \frac{\kappa_0\lambda}{2}(\mu-\mu_0)^2 + \\ &+ \frac{1}{2}\log(\kappa_0\lambda) + (a_0-1)\log(\lambda) - b_0\lambda + C, \end{aligned} \tag{2.41}$$

and compute the average with respect to one of the two parameters, respectively:

$$\begin{aligned} \log q^*(\mu) &= \mathbb{E}_{q(\lambda)}[\log p(\mathcal{D}|\mu,\lambda) + \log p(\mu|\lambda)] + C \\ &= \frac{\mathbb{E}_\lambda[\lambda]}{2}\Big[\kappa_0(\mu-\mu_0)^2 + \sum_i^N (x_i-\mu)^2\Big] + C, \end{aligned} \tag{2.42}$$

and

$$\begin{aligned} \log q^*(\lambda) &= \mathbb{E}_{q(\mu)}[\log p(\mathcal{D}|\mu,\lambda) + \log p(\mu|\lambda) + \log p(\lambda)] + C \\ &= (a_0-1)\log\lambda - b_0\lambda + \frac{1}{2}\log\lambda + \frac{N}{2}\log\lambda \\ &- \frac{\lambda}{2}\mathbb{E}_\mu\Big[\kappa_0(\mu-\mu_0)^2 + \sum_i^N (x_i-\mu)^2\Big] + C. \end{aligned} \tag{2.43}$$

From the description above we have established that the variational distributions associated to the two parameters are

$$q(\mu) = \mathcal{N}(\mu|\mu_N, \kappa_N^{-1}), \qquad q(\lambda) = \Gamma(\lambda|a_N, b_N), \tag{2.44}$$

where

$$\begin{aligned} \mu_N &= \frac{\kappa_0\mu_0 + N\bar{x}}{\kappa_0 + N} \\ \kappa_N &= (\kappa_0 + N)\mathbb{E}_\lambda[\lambda] \\ a_N &= a_0 + \frac{N+1}{2} \\ b_N &= b_0 + \frac{1}{2}\mathbb{E}_\mu\Big[\kappa_0(\mu-\mu_0)^2 + \sum_i^N (x_i - \mu)^2.\Big]. \end{aligned} \tag{2.45}$$

Thanks to this simple example, we have seen the power and the conceptual simplicity of variational inference, a general principle that can be applied to more challenging tasks wherein pure Bayesian inference is not feasible.

2.5 VARIATIONAL INFERENCE AND INFORMATION THEORY

2.5.1 *Basic introduction on Information theory*

ENTROPY Information Theory is a relatively new field which was developed initially by Shannon [71] to measure the quality of the communication in a (physical) channel. This theory, although in principle related to signal processing, has a strong relationship with Machine Learning [56, 61] and, as we will see in the following section, it will be particularly useful to interpret both the Bayes and Variational Inference. The body of literature pertaining to information theory is vast and so we refer to [18] for a relatively recent review. In the following section, we limit ourselves to defining those basic information theory concepts that will be useful in understanding our future analysis.

To have a clearer interpretation, let us start by considering the following basic problem: Alice has a set of pictures $\mathcal{X} = \{x_i\}_i^N$ with subject $\mathcal{Y} = \{y_i\}_i^N$ and she wants to inform Bob about the picture subjects, using a communication channel which allows her to send just a binary description Z. It is necessary to *encode* each image x_i in a channel code (*representation*) z_i. It is the assumption that the channel is noiseless and Bob knows all the possible subjects $\mathcal{Y}$ (without loss of generality $\mathcal{Y} = \{1, \dots, M\}$), and he knows how to *decode* the binary code z to the target y. The first idea is to map each data-point x_i with $y_i = j$ in a one-hot vector z (i.e. $z^{(j)} = 0$ for $j \neq y$ and $z^y = 1$). That description, although correct, is expensive, since the dimension of the code is growing linearly with the number M of subjects. If there are N data-points, Alice has to send a code Z that is $N \times M$ bytes long. A cheaper alternative is to consider a code of length $\log_2(M)$, if $M = 2^k$, otherwise the smallest integer is greater than $\log_2(M)$. That option is cheaper than the naive one, but this choice is not optimal and it could improved. If we know that some data-points are more likely than others then the idea could

be to give a smaller length code to the points with a higher probability. Thus, one idea would be to define the length of the data-point $l(x)$ as

$$l(x) = \log_2 \frac{1}{p(x)}, \tag{2.46}$$

in this setting the average length of the code Z will be $L(Z) = NH(X)$, where the H function, the *entropy* of X, is defined as

$$\begin{aligned} H(X) &= \sum_i p(x_i) l(x_i) \\ &= -\sum_i p(x_i) \log_2 p(x_i). \end{aligned} \tag{2.47}$$

It can be proven that the embedding described above is optimal and that the $NH(X)$ is the minimal code-length, so that this theorem is the so-called *Shannon Coding Theorem*.

To give an intuitive insight, let us consider the case $\mathcal{Y} = \{1, 2, 3, 4\}$ with a probability to occur respectively of $\left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\}$. If we consider the same code length for all the data, we should have a code z of length 2 for each data-point with a message length of $2N$ bits. Otherwise, choosing a length equal to the logarithm of the inverse probability, we determine that the average code length is $1.75N < 2N$. Let us notice that the difference between the two coding approaches is greater when the distribution is concentrated within a few terms. In the extreme case, the distribution is concentrated within just one element (Dirac distribution) and we do not need to encode anything. In contrast, if the distribution is equally distributed for all the points, the code length is the same for all the data points (i.e. we have the naive case).

We observed above that the entropy of a random variable defines a lower bound of the code-length that we need to use to describe the random variable itself: X cannot be described in less than $NH(X)$ bits. In light of the connection with the code-length of the data, we say that the entropy of a variable is a measure of the information stored within the variable itself: thus a variable X is said to be more informative than Y if we need more bits to describe X than Y.

In the description above, we restricted our attention to the binary code setting. Such a choice is arbitrary, and the space can have many possible bases b, with the entropy computed with respect the $\log_b$. Moreover, while the entropic definition holds for discrete distributions, as we have seen in the Bayes description wherein we heavily employ Gaussian properties, it would be preferable to extend the entropic definition to the continuous case. The entropy for continuous distributions $p(x)$ is called *differential entropy* and is defined as

$$h(X) = \int -p(x) \log p(x) dx. \tag{2.48}$$

In this setting the distribution $d(x)$ is discrete, and the differential entropy $h(X)$ coincides with the (classic) entropy $H(X)$ which follows, by the definition of integral sum, that in the discrete case it is just a summation.

The entropy of a continuous function does not satisfy the same properties as a discrete distribution. For example, in general, the entropy of a continuous distribution is not positive, voiding the original interpretation of entropy as being a quantity of information stored within the variable.

To avoid such an issue, and to maintain an informative description of the continuous distribution, in the following section we consider the entropy of a continuous function as being its n-bit quantisation, with n being arbitrarily large.

Given a distribution $p(x)$, the n-bit quantization $p_n(x)$ is the discrete distribution with n values, such that $p_n(x) = p_i$ for any $x \in [x_i, x_{i+1}]$, where $p_i = p(x_i)|[x_i, x_{i+1}]|$. Such a description, which at first sight could appear a little anomalous, is nonetheless consistent with the general idea that, in nature, nothing exists that is continuous and, therefore, continuity is merely an abstract mathematical construct within the definition of differential entropy. Indeed, it can be proven that, for $n \to \infty$, the quantised entropy is equal to the differential entropy.

From henceforth, in terms of the use of language, we refer to both classical and quantised entropy only as entropy.

MUTUAL INFORMATION Until this point, we have considered the unrealistic scenario in which there is no noise in the communication, and that X and Y have the same entropy, since there was no loss of information within the channel. Such an assumption, as previously stated, is unrealistic and therefore it is more precise to assume that there is some degree of noise inherent within the channel and that some information will therefore be lost. The quantity of information lost by Y about X in the channel is quantified by the *conditional entropy*

$$H(X|Y) = \sum_{i,j} -p(x_i, y_j) \log p(x_i|y_j). \tag{2.49}$$

Defined the quantity of information stored in the variable and the quantity of information lost in the message, we are able to define the quantity of information shared between the two variables, the *Mutual Information* $I(X;Y)$,

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \end{aligned} \tag{2.50}$$

i.e. the information stored by X, $H(X)$, minus the information lost in the channel by Y about X, $H(X|Y)$. The Mutual Information (MI), in agreement with the intuition, is a positive symmetric function.

Moreover, the mutual information can be written in terms of a KL divergence, such that

$$I(X,Y) = D_{KL}(p(x,y)||p(x)p(y)). \tag{2.51}$$

The definition via KL divergence shows a pure probabilistic definition of the MI, the distance between the joint distribution $p(x,y)$ and the product of marginals: $p(x)p(y)$, higher is the MI higher is the distance between the two distributions. Equivalently, by the properties of KL-divergence the MI can be rewritten as

$$I(X,Y) = \mathbb{E}_y[D_{KL}(p(x|y)||p(x))], \tag{2.52}$$

by this formulation it is also clearer as to what the KL divergence is measuring in terms of the expected number of extra bits that must, on average, be sent to identify Y if the value

of X is not already known to the receiver. By this description, we see that the maximal mutual information is obtained when $Y = X$ (i.e. $I(X;Y) = H(X) = H(Y)$), indeed in general $I(X;Y) \leqslant \min\{H(X), H(Y)\}$.

To show the final property of the MI, let us revert to the Alice-Bob channel. In a noise-free channel $I(X,Z) = I(Z,Y) = I(X,Y) = H(X)$, but in a noisy channel we expect that the information shared by X with Y is less than the one with Z because there could be some noise arising between Z and Y. This is true in terms of *Data Processing inequality* for any channel $X \to Z \to Y$, and, as such, the following inequality holds:

$$I(X,Z) \geqslant I(X,Y). \tag{2.53}$$

The data-processing inequality will be used heavily in the subsequent chapters, but for now it is relevant because it induces a form of triangular inequality,

$$I(X,Z) + I(Z,Y) \geqslant 2I(X,Y). \tag{2.54}$$

That last property, combined with the positivity and symmetry, allows us to show that the function

$$D(X,Y) = 1 - \frac{I(X;Y)}{\max\{H(X), H(Y)\}} \tag{2.55}$$

is a metric within the probabilistic space.

2.5.2 *Applications of Variational Inference*

In the Variational Inference example provided above, we observed that such a technique allows us to compute the posterior of the parameters, terms which the classic Bayesian approach cannot compute. In this section, after the description in [30], we aim to highlight the relationship arising between variational inference and information theory. We will observe that the heuristic Occam's razor principle can be formalised, within information-theoretic terms, as a Minimum Description Length (MDL) principle, and that the variational inference approach is a stochastic version of that principle.

Let us then consider a general problem with a data-set $\mathcal{D}$ and a model family $\mathcal{M}$ parametrised by θ. The MAP goal is to find the parameter θ_*, such that $p(\theta_*|\mathcal{D})$ is maximal. However, what if two parameters satisfy the same properties? Occam's razor asserts that, given two models describing the same data in the same way, $p(\mathcal{D}|\theta_1) = p(\mathcal{D}|\theta_2)$ we have to prefer the simpler option, but the definition of 'simple' is not clear. Commonly it is considered that the attribute 'simple' describes a model which uses the fewest possible parameters.

As we have seen, entropy provides a measure of the information of a random variable, and the logarithm of the inverse probability is the length of the specific term within the encoding space. In light of that description, it is reasonable to state that longer codes are more complex than shorter ones, and that a high entropy variable is more complex than a low entropy one. In this context, Occam's Razor can be written in terms of choosing the less

entropic random variable θ', such that the information lost in the communication $p(\mathcal{D}|\theta')$ is maintained bounded, i.e. we optimise the following MDL problem

$$\min_{\theta} L(\mathcal{D}|M) = -\log p(\mathcal{D}|M,\theta) - \log p(\theta|M), \tag{2.56}$$

finding the shortest parameter that describes both the data and the model itself.

The MDL objective in (2.56) is formally equivalent to the MAP objective (2.13), but the two are not the same. In MAP, the model is prefixed, and the learnt variables are also the optimal parameters for the model, whereas in the MDL, the model has to be selected. Assuming that we have a specially extra-flexible model, such that its parameters can describe any possible models, the MAP and MDL objectives are the same.

To further underline the connection between MAP and MDL, let us note that the objective (2.56) can be viewed as an approximation of the problem to minimise the complexity (code-length) of the approximated generating distribution $p(\mathcal{D}|M)$, which is defined as

$$p(\mathcal{D}|M) = \int p(\mathcal{D}|\theta)p(\theta|M). \tag{2.57}$$

Such a quantity is an integral with respect to a potentially infinite set of parameters and, in general, it is not easy to compute. To simplify the computation in an extreme fashion we can assume that the data is concentrated around a single parameter θ'. Within this setting, the integral has the form:

$$p(\mathcal{D}|M) = p(\mathcal{D}|\theta')p(\theta'|M), \tag{2.58}$$

and putting the expression (2.58) under the logarithmic function, we recover the objective in (2.56).

Underlining the relationship arising between the MAP and MDL estimator and their differences, in the following section we will assume implicitly that we are within the setting in which the parameter θ can describe any parameter. Thus we can simplify the notation, removing from $q(\theta|M)$ the dependence of M and simply writing the term $q(\theta)$

VARIATIONAL INFERENCE AND MDL In the MAP setting, we assume that we are considering only a single θ, onw which is sampled from the distribution $p(\theta)$. A reasonable improvement could be to consider more than just one parameter, chosen according to a distribution $q(\theta)$; wherein the integral in (2.57) is better approximated by a reduced sum of the most likely parameters either than considering only that factor with the most likely parameter. In that setting, since the parameters are more than one and sampled from the distribution $q(\theta)$, the length of the generative distribution $L(\mathcal{D}|\theta)$ cannot be computed explicitly and thus we can estimate the average code-length with respect the distribution $q(\theta)$:

$$\begin{aligned} L(\mathcal{D}|M) &= \mathbb{E}_q[l(\mathcal{D}|M)] \\ &= \mathbb{E}_q[l(\theta) + l(X|\theta)] \\ &= -\int q(\theta)[\log p(\theta|M) + \log p(\mathcal{D}|\theta, M)]. \end{aligned} \tag{2.59}$$

The derived average generating length $L(\mathcal{D}|M)$, is an upper bound of the MDL objective in (2.56), and is minimised by the single model minimising the deterministic MDL (i.e. the optimal solution would be $q(\theta) = \delta(\theta')$); thereby losing the advantage of considering a random θ. Indeed, by introducing the new distribution $q(\theta)$ with added redundancy, there is extra information within the system that has to be taken into account. Thus, the idea of taking into account the extra information adds to the objective in (2.59), e.g. the entropy of the distribution $q(\theta)$. In this way the new objective to optimise has the form:

$$\begin{aligned} L_q(\mathcal{D}) &= L(\mathcal{D}|M) - H_q(\theta) \\ &= \int q(\theta) \log \frac{q(\theta)}{p(\theta|\mathcal{D})p(\mathcal{D})}, \end{aligned} \tag{2.60}$$

which is exactly the variational objective optimised (2.32).

FREE ENERGY AND INFOMAX In the paragraph above we have seen how the variational inference is strictly connected to a model selection problem, namely the Minimum Description Length, which was initially described in information-theoretic terms. The connections between information theory and variational inference are not limited exclusively to the example cited above. Indeed, the variational inference objective (2.32) is the associated with the efficient coding theory that was first developed by Barlow [8] to describe the brain's behaviour in terms of maximising the available information arising between sensation and representation, also known as the InfoMax principle.

In terms of detail, the efficient coding theory idea is that, given a set of *signals* $\{x_i\}$, the brain has to encode these within some *representations* θ in an *efficient* way, and this is done via maximising the mutual information arising between sensations x_i and representations θ. In the original formulation, the representation θ was fixed, and thus the principle did not have a probabilistic interpretation. A probabilistic interpretation was provided in the first machine learning applications and, as observed in [9], in this revisited setting it is necessary to bound the representation entropy in order to avoid the representation being equivalent to the signal.

More recently, the free-energy principle [22] was proposed to describe how any self-organising system – and, in particular, the brain – works. The main idea underlying the free-energy principle is that a learning system is learning by acting, wherein the action is itself the ability to generate samples that are associated with the environment $p(\theta)$ in which it lives. More formally, the free-energy principle is a probabilistic model depending, as it does, on two distributions: [1] the representation density $q(\theta)$, and [2] the one generating the sensory samples $p(x|\theta)$ maximising the free-energy objective:

$$\mathbb{E}_{q(\theta,x)}[-\log p(x|\theta)] + D_{KL}(q(\theta)||p(\theta)), \tag{2.61}$$

where the first term represents the ability to generate samples associated with the environment, while the KL term represents the ability to learn the correct representation adequately describing the environment.

The Free-Energy objective in (2.61) is exactly the VI in (2.32) objective (with opposite sign) that, as seen in (2.60), can be written as a variational InfoMax: an InfoMax wherein the encoding representation $q(\theta)$ has to be bounded as stipulated in [9].

By virtue of this analysis, we have seen that Variational Inference is not only a way in which we can approximate the pure Bayesian Inference, it is also useful in describing the selection model objective, specifically the MDL, and the learning objective that derives from Neuroscience. As we will see in the following chapters, such precepts are widely used within the context of Machine Learning.

3

INTRODUCTION TO REPRESENTATION LEARNING

The last chapter observed how variational inference and information theory are used to describe the (representation) learning objective within the human brain framework. In the following chapter, starting from the basic PCA model, we will define what representation learning is within a purely Machine Learning context.

The first time we encountered the concept of representation was in the mixture model description, wherein the representation z, as described by a categorical distribution, defined which model of the mixture to use to generate a particular data-point. In the following chapter, we introduce a generalisation of the mixture of models, wherein the representation z is not more categorical, but is rather continually distributed. These models are called *Latent Variable Models* since the goal is not only to discover the weight parameter θ, but also the *latent* variables (aka representations) describing the data-set itself.

3.1 PRINCIPAL COMPONENT ANALYSIS

Let $\mathcal{D} = \{x_i\}_1^N$ be a set of points in $\mathbb{R}^D$, and assume we want to estimate the density from which they are sampled. Assuming that we know that the data are normally distributed, as we have discussed in the previous chapter, it is possible to estimate the mean and the $D \times D$ covariance matrix within a purely Bayesian framework. However, let us assume that an oracle tells us that the data are distributed around $K < D$ directions, i.e. the data lives around a K-dimensional hyper-plane embedded in D-dimensional space. In this setting, it is convenient to estimate the covariance matrix of the Gaussian for the hyper-plane axis and then to map the data in the D-space. In this way, the generative process of a new data-point x is composed of two steps, namely sampling a point $z \in \mathbb{R}^K$ and then mapping it in the D-space.

The latter approach is preferable for two primary reasons. Theoretically, we store only those relevant factors, the so-called *Principal Components*, thereby getting rid of possible noise, and also computationally, assuming $K \leqslant D\phi$, with $\phi = (1+\sqrt{5})/2$ the golden number, that the parameters required to estimate the Principal Component Analysis (PCA) are less than would arise within a pure Bayesian inference. Indeed, in the PCA, the parameters to be estimated are $K(D+K)$, i.e. - the embedding matrix $W \in \mathbb{R}^{D\times K}$, plus- the covariance matrix $C \in \mathbb{R}^{K\times K}$, instead in the Bayesian approaches the parameters to estimate are D^2. Thus, $K(D+K) < D^2$ if $K^2 + DK - D^2 < 0$, i.e. if $K \leqslant D\phi$.

ESTIMATION OF PARAMETERS We have seen that it is convenient to look at the principal components and describe the covariance matrix with respect to such components. However,

these components have to be recovered as they are not known in advance. By definition, the principal components are the directions wherein the data is changing increasingly or, more formally, where the variance is higher. Starting from that definition it is only natural to consider the principal components as the eigenvector with highest magnitude (eigenvalue) of the empirical covariance matrix $S = \sum_i (x_i - \bar{x})(x_i - \bar{x})^T$, where $\bar{x} = \frac{1}{N}\sum_i x_i$ is the empirical average. Indeed, we know from basic linear algebra that any positive definite square matrix S can be decomposed into

$$S = V\Lambda V^T, \tag{3.1}$$

with Λ the diagonal matrix of the eigenvectors λ_i describing the variance of $\mathcal{D}$ around the eigenvector v_i column of V.

Both the eigenvalues λ_i and the eigenvectors v_i are associated with the data and they are recovered directly from the data itself. In particular, defined $X = [x_1^T, \ldots, x_N^T]^T$, the $N \times n$ data matrix, by the Singular Value Decomposition (SVD) the data matrix is decomposed as follows:

$$X = U\Lambda V^T, \tag{3.2}$$

where U is a $N \times N$ matrix whose columns are orthonormal ($U^T U = I$), and the other matrices are as above.

By these SVD decomposition properties, in whuch we defined $\hat{D}$ as the diagonal matrix containing only the first K eigenvalues (i.e. the one with the highest magnitude), with $\hat{V}$ and $\hat{U}$ the associated truncated matrices, we can assert that the matrix data is approximately equal to

$$\hat{X} \approx \hat{U}\hat{\Lambda}\hat{V}^T, \tag{3.3}$$

where the approximation improves with the number of eigenvectors considered. Indeed, it can be proved that the error arising between the truncated approximation and the original data matrix is approximately equal to the highest eigenvalue that was not considered λ_{K+1}, [62] (12.2.3)

$$\|X - \hat{X}\|_F \approx \lambda_{K+1}. \tag{3.4}$$

Thanks to the SVD approximation, and remembering that the principal components are generated by the highest eigenvectors, defining the latent variable (representation) matrix $Z = \hat{U}\hat{\Lambda}$ and the mapping matrix $W = V^T$, we have recovered the PCA model.

By this description, the projection matrix mapping the visible data-points in the latent variable is the eigenvector matrix and the transpose of the generative one. The PCA setting wherein both projection and generation are linear and symmetric is a special one. In the following section, we will see more complex models, but that setting is particularly useful in describing the properties of the representation.

A GEOMETRIC PERSPECTIVE The most natural description of the PCA is a statistical one: finding the directions where the normal distribution has the highest variance. Another possible description is a geometric one, as it is useful to have an intuitive idea of what the Principal Components are. Assuming that the visible data are noisy observations of data-points living in a k-dimensional hyper-plane (a mathematical generalisation of the straight line, 1-plane, and the canonical plane, 2-plane), the goal of the PCA is to find such an hyper-plane, or better still the tangent vectors unequivocally defining such a plane.

For example, let us assume the trivial case of a 2-plane in $\mathbb{R}^3$, mapped by the function $f(x, y) = [x, y, x + y]^T$. In this setting, where we know the generative function, f, the two representations are the variables x, y: the so-called *generative factors* (i.e. the independent variables from which it is possible to generate any point of the plane), and the two eigenvectors which are defined respectively as

$$\partial_x f = [1, 0, 1]^T, \qquad \partial_y f = [0, 1, 1]^T, \tag{3.5}$$

(i.e. the partial derivatives of the generative function f with respect the representations, the *covariant vectors*). From that straightforward example, it is possible to see that the PCA is invariant with respect to these defined rotations. Indeed, considering any rotation of the representations x and y does not change the result, since the fundamental property of a PCA representation is that they are independent.

AN INFOMAX DESCRIPTION The first application of the InfoMax principle in machine learning [51] was to train the self-organising network maximising the information between the layers. This is a special network connecting the representations and the visible data (i.e. a network performing the linear operator $x = Wz$), and it was learning a representation equivalent to PCA. This is not surprising, given that the PCA solution is the one maximising the information arising between the representation and the visible data. To visualise this, we may consider the two terms of the mutual information separately, namely the latent entropy $H(X)$ and the conditional entropy $H(X|Z)$.

The first term is a constant of the system. Then, the only parameter we need to consider is the conditional entropy $H(X|Z)$, which is a function of the distance arising between the original and the reconstructed data. Let us assume that the generated data $\hat{X}$ is generated as above by WZ plus an arbitrary small noise $\mathcal{N}(0, \sigma^2 I)$. By means of this assumption, if we infer that $p(x|z)$ is Gaussian distributed as

$$\log p(x|z) \propto -\|x - Wz\|_2^2/\sigma^2, \tag{3.6}$$

then the conditional entropy is $H(X|Z) = \|X - \hat{X}\|_F$, as can be shown (followed by the SVD decomposition) that it is effectively minimised by the PCA solution.

Thanks to the alternative descriptions given above, we have observed that the PCA representations are not merely the projection in the highest magnitude of eigenspace. They are the orthogonal factors spanning the most *informative* subspace of dimension K with respect to the visible data-points $\{x_i\}_i$.

The PCA, as described above, is a deterministic model wherein the representations are linear projections of the data. However, with this model, is not possible to make any probabilistic inference in relation to the latent variables; a probabilistic description of the generative model; or a pure informative description without adding any arbitrary noise as in the InfoMax description provided above. For this reason, it is necessary to consider a probabilistic PCA [79].

The derivation of the probabilistic PCA is based on the observation that, since the visible data $\mathcal{D}$ is Gaussian distributed and the projection operator is a linear one, then the representation is normally distributed as well. Thus, without a loss of generality, we assume that it is zero-mean and unitary variance distributed, $p(z) \sim \mathcal{N}(0, I_K)$. The generated points $\hat{x}_i$ are sampled by the distribution $p(x|z, W, \sigma^2) = \mathcal{N}(Wz, \sigma^2 I)$, that depends on a linear generative operator W; the representation z (as in the deterministic case); and the variance of the distribution σ^2. That definition of the generator was made in order to be consistent with what was seen in the deterministic case. Indeed, for $\sigma \to 0$, the generative distribution $p(x|z, W, \sigma^2)$ becomes a deterministic one.

As with any probabilistic approach, the most intuitive way to learn the parameters is to maximise the log-likelihood of the generative sample

$$\mathcal{L}(\theta) = \sum_i \log p_\theta(x_i) = \log p(X|\theta), \tag{3.7}$$

where the generative distribution $p_\theta(x_i)$ is defined as

$$p_\theta(x) = p(x|W, \sigma^2) = \mathbb{E}_{p(z)}[p(x|z, W, \sigma^2)]. \tag{3.8}$$

In particular, as observed in [79] the generative log-likelihood is

$$\log p(X|W, \sigma^2) = \frac{N}{2}[n \log(2\pi) + \log|C| + \mathrm{tr}(C^{-1}S)], \tag{3.9}$$

where $C = WW^T + \sigma^2 I_n$, $S = N^{-1} \sum_i x_i x_i^T$, with $|\cdot|$ denoting the matrix determinant, and it is maximised by

$$W = V(\Lambda - \sigma^2 I)^{1/2} R, \tag{3.10}$$

where R is an arbitrary $K \times K$ rotation matrix, V is the $n \times K$ matrix whose columns are the first K eigenvectors of S, and Λ is the corresponding diagonal matrix of eigenvalues. Moreover, the optimal variance is given by

$$\sigma^2 = \frac{1}{n-K} \sum_{j=K+1}^{n} \lambda_j. \tag{3.11}$$

The optimal solution, consistent with the problem definition, for $\sigma \to 0$ is equivalent (equal up to rotation) to the solution as learned by the deterministic PCA. Moreover, the probabilistic description of the generator gives us, for free, an inference description of the representation projection, with latent posterior defined as

$$p(z|x, \theta) = \mathcal{N}(F^{-1} W^T x, \sigma^2 F^{-1}), \tag{3.12}$$

with $F = W^T W + \sigma^2 I$.

We conclude the analysis of the PCA by observing that we introduced the PCA as a generalisation of mixture models, but in manner that is fundamentally different from what was done with the Bayesian models as outlined within the previous chapter, in that the inference is done with respect to the representation and not the weight. This is not a real issue, per se, since, as described in [47], the PCA described above can be described equivalently within the terms of the Bayesian framework standards. Thus we elected to introduce this version to stress the importance of learning representation rather than weights, although the two problems are equivalent.

3.3 REPRESENTATION LEARNING IN THE SUPERVISED SETTING

The PCA is the most famous example of representation learning in the unsupervised setting, since its principles can be generalised for any, possibly non-linear, generator g. Indeed, as we will see in the following section, any representation learning model can be described as a generative model problem: find the generator parameter, maximising the log-likelihood. By this description it appears as though representation learning is an unsupervised problem, and the representations are simply a description of the visible data, storing the relevant information necessary to generate good samples (i.e. samples that are close to the original ones).

In this section, we show that the definition provided above is incomplete, since the optimal representations are task-dependent. Indeed, the representation learning problem can be read at any time as a supervised problem, and the unsupervised setting is just a special case wherein the task is the generation. In particular, in the following section, without the risk of a loss of generality, we study the supervised problem via the regression task. Indeed, the classification task is a special regression in which it is put as a softmax in the output.

Let us start by considering the regression problem as described in the previous chapter: given the model

$$y = w \cdot \phi(x) + \varepsilon, \tag{3.13}$$

find the parameter w maximising the the posterior $p(w|\mathcal{D})$, with prior $p(w) = \mathcal{N}(0, \sigma^2 I)$.

Assuming the feature map ϕ is known, the posterior $p(w|\mathcal{D})$ is maximised by

$$w_{PLS} = (\Phi^T \Phi + \sigma^2 I)^{-1} \Phi^T y, \tag{3.14}$$

an expression that depends upon the feature map ϕ, or equivalently on the model representations $Z = \phi(X) = \Phi$. By this observation, assuming that we have a free (to learn) feature map, we expect to learn a *better* function f. Let us note that the problem of estimating the feature map ϕ (or equivalently Z), is equivalent to an estimation model problem: i.e. find the optimal representation Z, such that $\arg\max p(w|\mathcal{D}, z)$ is maximal. Such a relationship would be useful in the following context to provide an objective through which to learn the optimal representation.

In the setting where both the weights w and the feature map ϕ are free parameters, it is convenient to rewrite the Bayes inference problem with respect to the function $f = w \cdot \phi$. In this thesis, we will restrict ourselves to considering only the MAP problem: i.e. to find the optimal function f, such that the posterior $p(f|\mathcal{D})$ is maximised. For a pure Bayesian inference analysis, we refer to [73].

GAUSSIAN PROCESSES Proceeding in the same vein as Chapter 2, in order to estimate the MAP it is first necessary to define the log-likelihood

$$\log p(\mathbf{y}|X, f) \propto \sum_i \|y - f(x_i)\|_2^2, \tag{3.15}$$

and the prior for the function f. To define the prior of f, it is first necessary to define the space in which the function has to live, space that is not more than $\mathbb{R}^n$, the vector space where the parameters w were living. A common assumption is to say that f lives in the continuous function space $\mathcal{C}(\mathbb{R}^n)$: the space of all functions g such that the L_2-norm is bounded, i.e.

$$\|g\|_2^2 = \int \sum_i^k g_i^2(x)dx, < \infty. \tag{3.16}$$

Having defined the function space we can assume that the prior distribution has the form

$$p(f) \propto e^{-\|f\|_2^2}, \tag{3.17}$$

a *special* Gaussian distribution in the function space.

The relationship with the Gaussian distribution is not just formal. Indeed, given a set of points $X = \{x_i\}_i^M$, the distribution $p(f(X)) = p(f(x_1), \ldots, f(x_M))$ is a Gaussian one, with mean 0, and covariance matrix $K = [K_{i,j} = k(x_i, x_j)]_{i,j}$

$$k(x_i, x_j) = \mathbb{E}[f(x_i)f(x_j)^T]. \tag{3.18}$$

Thanks to the special prior, ascertained in an analogous way as for the vector setting, it is possible to show that the log-likelihood is

$$\log p(y|f, D) \propto \|f\|_2^2 + \frac{1}{\sigma^2} \sum_i (y_i - f(x_i))^2, \tag{3.19}$$

a normal distribution with mean

$$k^T(\sigma^2 I + K)^{-1}\mathbf{y}, \tag{3.20}$$

and variance matrix

$$\kappa - k^T K^{-1} k, \tag{3.21}$$

where σ^2 represents the noise variance in the prediction, $p(y|f(x)) \sim \mathcal{N}(f(x), \sigma^2 I)$ and $\vec{k} = [k(x, x_1), \ldots, k(x, x_N)]^T$, and $\kappa = k(x, x)$.

By the property of the function f, the process $f(X)$ is called the *Gaussian Process* (GP). That kind of process is widely used in machine learning since it is the generalisation of the standard Bayes inference on the parameters (finite space) to the functions (infinite space).

Moreover, the MAP estimation of the function f, (3.20), resembles the optimal weight solution in (3.14), an expression that depends exclusively on the feature map. That behaviour suggests that the feature function already describes the fundamental property of the models. To better investigate this point and to describe the properties of a *good* feature map, let us consider the weight w and the feature map separately.

OPTIMAL FEATURE MAP Without a loss of generality, we assume that each component ϕ_i of $\phi : \mathbb{R}^n \to \mathbb{R}^m$ has the same norm (if it is not true it is sufficient to normalise all of the factors), in this way the L_2 norm of the function f is rewritten as,

$$\begin{aligned} \|f\|_2^2 &= \int_2 \sum_i (w_i \phi_i(x))^2 \\ &= \sum_i w_i^2 \int \phi_i^2 \\ &\|w\|_2^2 \|\phi\|_2^2, \end{aligned} \tag{3.22}$$

and then the prior has the form

$$p(f) \propto e^{-\|f\|_2^2} = e^{-\|w\|_2^2 \|\phi\|_2^2} = p(w)p(\phi), \tag{3.23}$$

with $p(w)$ having the same form assumed in the previous chapter. Observing that the Gaussian assumption in (3.17) induces the normality in the weights w, we can write the kernel K as a function of ϕ instead than f

$$\begin{aligned} k(x_i, x_j) &= \mathbb{E}[\phi(x_i)^T w w^T \phi(x_j)] \\ &= \mathbb{E}_\phi[\phi(x_i)^T \mathbb{E}_w[w w^T] \phi(x_j)] \\ &= \mathbb{E}_\phi[\phi(x_i)^T \alpha I \phi(x_j)] \\ &= \alpha k_\phi(x_i, x_j). \end{aligned} \tag{3.24}$$

From the equivalent Kernel description above we see that an implicit property of Gaussian Processes is that the kernel, and subsequently the behavioural (*dynamics*) of the process, has to be described completely by the feature map, also known as the representations. If so, then the optimal feature map ϕ has to be defined as the one with kernel $K = \phi(X)\phi(X)^T = ZZ^T$ such that the likelihood of the function $p(f(X))$ is maximised.

3.4 A GOOD REPRESENTATION

Introducing PCA, we have seen that its representations are a set of orthogonal vectors spanning the most informative subspace in relation to the one generated by the visible data. In the more general setting of the Gaussian Process, we have seen that a good representation has to have the same dynamics as the entire GP model: i.e. the kernel of the representation is the same as the kernel of the entire function. In this section, with the help of the PCA example, we show that an unsupervised problem can be described as a GP and that the two definitions of representations are in agreement. Thus, an informative disentangled representation describes the model dynamics.

PCA: A PARTICULAR REGRESSION PCA can be described as the problem by which to find a representation Z and a weight matrix W such that the reconstruction loss objective

$$J(W.Z) = \|X - WZ^T\|_F^2, \tag{3.25}$$

is minimised.

Comparing the objective in (3.25) with (3.19), we see that the PCA objective is optimising the deterministic log-likelihood of the GP $f(z) = Wz$. Considering a probabilistic version with the $p(f)$ Gaussian as above, the distribution $p(f(Z))$ is a zero mean with a variance given by $K = ZZ^T$, and marginal likelihood

$$\begin{aligned} p(X|Z, \sigma^2) &= \prod_i^N \int p(x_i|z_i, W, \sigma^2) p(W) \\ &= (2\pi)^{nN/2} |K|^{n/2} \exp\left(\frac{1}{2} tr(K_z^{-1} XX^T)\right), \end{aligned} \tag{3.26}$$

where $K_z = ZZ^T + \sigma^2 I$.

In this setting, the goal, as outlined above, is to find the Z optimising the log-likelihood $\log p(f(X))$. By its special structure, wherein the input data are the latent variables and the unsupervised Gaussian Processes (like the one here) and are called Gaussian Process Latent Variable Models (GPLVMs), first introduced in [47]. In particular, the objective in (3.26) is maximised by the latent variable having the form

$$Z = ULV^T, \tag{3.27}$$

where U is an $N \times k$ matrix, whose columns are the eigenvectors of XX^T; L is a $k \times k$ diagonal matrix with elements $l_{ii} = (\frac{\lambda_i}{n} - 1)^{-1/2}$, where λ_i is the i-th eigenvalue of XX^T; and V is an arbitrary $k \times k$ orthogonal matrix. The derived solution, as expected, is equivalent to the probabilistic PCA solution derived before. Indeed, the eigenvalues of the two matrices XX^T and X^TX are the same, and the eigenvector v of XX^T is given by X^Tv', where v' is an eigenvector of X^TX.

By this method, we have shown a dual definition of the probabilistic PCA, wherein the inference on the weights is now moved to the latent variables, enforcing the intuition that optimal representations are equivalent to optimal weights.

3.5 NEURAL NETWORKS

In the preceding analysis, we have seen that both supervised and unsupervised models can be described as GPs, and that the latent representation plays a fundamental role in the model performance, since the representation defines the kernel map K associated with the model dynamics. In classical machine learning theory, the feature maps are chosen in advance and the optimal one is obtained via a model selection technique.

An alternative approach, avoiding manually selecting the subset of feature maps to consider, is to define a special parametric class of models that potentially can describe any feature map ϕ, that model class is the *Neural Network* (NN) family. A (fully-connected) Neural Network

f_{NN} is a concatenation of layers $f_{NN} = L_n(L_{n-1}(\dots(L_0(x))))$, where a layer $L_i(z)$ is the output of a linear operator followed by a fixed non-linearity σ, i.e $L_i(z) = \sigma(W(z))$, where the parameters to learn are the entries of the matrices W_i. Technically, we define the width of the layer as the dimension of the number of units (features) used to describe the layer itself (i.e. if $z_1 = L(z) \in \mathbb{R}^d$, d is the width of the first layer). Such a relatively simple structure can, in principle, be used to learn any possible feature map. Indeed, a NN is a theoretical universal approximator [31] (i.e. for any $\varepsilon > 0$ given any function f there exists a network with enough number of layer and width such that $\|f - f_{NN}\| < \varepsilon$).

Such a theorem works only theoretically and, in general, many alternative architectures are used to describe the network layers that were proposed to fit better to the input data (e.g. convolutional [42] for static geometrically connected data-points, and recurrent [67] for sequential data), while maintaining the same idea and principles. Although some differences arise in the main structure, all NNs f_{NN} are optimised via Gradient Descent derived algorithms (e.g. Batch Gradient Descent, Stochastic Gradient Descent (SGD), Adam [36]); learning algorithms wherein the main idea is to optimise the old parameters θ_{old} within the gradient of the loss objective $J(\theta)$:

$$\theta_{new} = \theta_{old} - \nabla_\theta J(\theta_{old}), \tag{3.28}$$

where the objective loss J, in general, is the sum of a distance D arising between the label y and the predicted data $f_{NN}(x)$, plus some regulariser R,

$$J(\theta) = D(\hat{Y}, Y) + R(\theta). \tag{3.29}$$

3.5.1 *AutoEncoder: a neural net performing PCA*

In agreement with what has been achieved thus far in terms of the representation learning problem and the Gaussian Process models, as described within the special context of the PCA, we conclude the chapter by describing a special NN, the AutoEncoder, which learns the representation in an unsupervised way and, in its simplest structure, is learning the PCA parameters.

AUTOENCODER An AutoEncoder (AE) network f_{AE} is a composition of two NNs; specifically, an encoder e, $e : x \to z$ transforming the visible data-point(s) x into some representation z, and a decoder d, $d : z \to \hat{x}$, transforming the latent variable in some real data-point. The network is trained to minimise the objective loss

$$J(\theta) = D(x, \hat{x} = d(e(x))) + R(\theta), \tag{3.30}$$

with D, the reconstruction loss (e.g. the L_2-distance $\|x - \hat{x}\|_2^2$, or the cross-entropy $x \cdot \log(\hat{x})$ when the data-points are binary), and R a generic regulariser.

The AE is a network explicitly defined to learn the optimal representation within an unsupervised scenario, since the main idea underlying the network is that an optimal representation $z = e(x)$ is the one storing all the information necessary to restore the data.

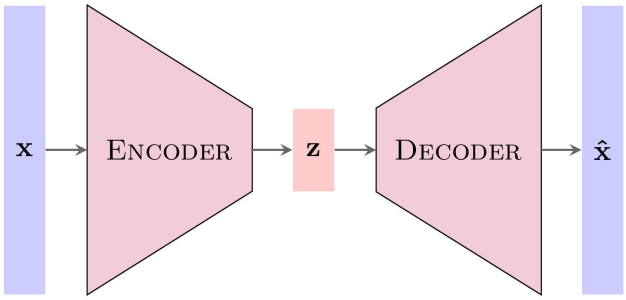


Figure 3.1 A schematic illustration of an AutoEncoder Network. In general, the representation z is smaller than the data-point x, in order to constrain the network to learn only that essential information pertaining to the data.

It is possible to prove (e.g. [6]) that, if the encoder and decoder are linear, and one is the transpose of the other (i.e. $e(x) = Wx$ and $d(z) = Vz$, with $W = V^T$), then the AE optimised to minimise the L_2 reconstruction loss with no auxiliary regulator is recovering the same weights of PCA. Such a theorem, which is in agreement with the regression description of the PCA summarised in (3.25), is useful to show that, at least in special cases, a NN can describe a Gaussian Process [50].

3.5.2 *Regularised AutoEncoders*

By the good properties of the linear AE, it is reasonable to consider more complex encoder and decoder networks to learn GP with kernels that differ from the linear one, and to discover those non-linear factors describing the data. To achieve this goal, it is insufficient to add more layers to these networks, but rather it is necessary to also change the reconstruction loss. Indeed, the direct reconstruction loss is not an excellent objective through which to learn the network parameters, since a deep network without a regulariser tends to over-fit, and the learnt representations are thus sub-optimal and not describing the data adequately.

In the over-fitting scenario, if a few features are more used than others, then an intuitive form of regulariser is the weight penalty $R(W) = \lambda \sum_{(i,j)} W_{i,j}^2$. In this way, all the weights have to be bounded, and thus no feature is used more widely than others. This approach is working optimally within a relatively shallow network. However, in a deep network architecture, wherein the number of parameters to control grows rapidly with the number of layers, it is no more feasible.

Starting from the observation that a regulariser on the weight is hard to define, and that a good representation is the one that is robust with respect to noise, in [84] it was suggested that it was necessary to minimise, as an objective, the reconstruction loss of the network wherein some input noise ν was added to the visible data (i.e. $J(\theta) = D(x, d(e(\tilde{x})))$, where $\tilde{x} = x + \nu$). The network optimising the revisited reconstruction loss is called the Denoising AutoEncoder

(DAE) because the network is trained to remove the noise from the data and then to learn a more robust representation that learns the fundamental properties of the data. Moreover, the addition of noise gives a probabilistic description of the DAE. Indeed, the reconstruction loss can be described as an approximation of the conditional entropy $H(X|Z)$, in that the extra noise in the model defines an implicit regulariser.

To have an explicit regulariser of the network, it is necessary to look at the small-noise setting, wherein the DAE is equivalent to the Contractive AutoEncoder (CAE) [66], a noise-free network with a regulariser bounding the Jacobian of the encoder map $J_e(x) = (\partial_x \sigma(e(x)))$, i.e. $R(\theta) = \lambda \|J_e\|_F^2$, with $\sigma = (1 + \exp(x))^{-1}$. The CAE description is useful for two main reasons. First, it shows that in a deep scenario it is more suitable to control the latent description rather than the network weights, and it also gives a probabilistic description of the latent representation. Indeed, by the non-linearity in the encoder, its derivative is $\sigma'(x) = p(x)$, with $p(x)$ a logistic distribution. Then, the regulariser $\|J_e\|_F^2 = \sum_i [\partial_{x_i} \sigma(e(x_i))]^2 = \sum_i p(z_i)^2 e'(x_i)^2$, is a term enforcing a high entropy representation (i.e. $p(z)$ has to be almost constant for any z_i).

THE VARIATIONAL AUTOENCODER The AutoEncoders described above are deterministic models where a probabilistic description was provided. As we have seen for the probabilistic PCA, a full probabilistic description of the AEs would be necessary to have a clear interpretation of the latent variable; a sample of the inference map; and of the decoder, the generative process associated to the data.

A full probabilistic AutoEncoder was provided by [37], and is called the *Variational AutoEncoder* (VAE). The VAE is a network wherein the encoder is modelling the inference $q_\phi(z|x)$ and the decoder the generative distribution $p_\theta(x|z)$. It was originally proposed to maximise the Evidence Lower Bound (ELBO)

$$\sum_i \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)\|p(z)), \tag{3.31}$$

objective which, as we have seen in the previous chapter, is the variational lower bound of the visible log-likelihood $\mathbb{E}[\log p_\theta(x)]$, the very same objective that was optimised by means of the Probabilistic PCA. By means of that description and the analogy seen by the AE and PCA, it would be not so surprising to know that the linear VAE optimising the ELBO is learning a solution equivalent to the one offered by the pPCA [55].

In the following chapter, we will analyse the VAE network, describe its issues, and try to provide a possible solution.

Part II

CONTRIBUTION

4

THE CAPACITY CONSTRAINED INFOMAX

Although the Variational AutoEncoder (VAE) is one of the most popular models, both for inference [74, 39] and generative purposes [86, 14], it presents two main issues: - the generated samples have higher variance than the visible ones, so that if the data are images that means blurred samples; and - the inference is often inaccurate: the representations do not describe the data accurately. These two issues, as we will see in the following sections, are strictly interrelated, and they are associated with the Evidence Lower Bound (ELBO) objective

$$\text{ELBO} = \sum_{i}^{N} \mathbb{E}_{q(z|x_i)}[\log p(x_i|z)] - D_{KL}(q(z|x_i)||p(z)), \tag{4.1}$$

and thus the two tasks of representation inference and generative modelling, seems impossible to be solved together.

In this chapter, we analyse from an information theoretic perspective the ELBO objective and the most relevant proposed variants: β-ELBO by Higgins et al.[28], Variational Wasserstein AutoEncoder by Tolstikhin et al. [83], TC-VAE by Chen et al. [13], and we observe that all of these are aiming to optimise the same (theoretical) objective:

$$\begin{cases} \max_Z & I_p(X;Z) \\ \text{s.t.} & I_q(X;Z) \leqslant H_p(Z). \end{cases}$$

This objective, that in the following we refer to as Constrained InfoMax (CIM), is asserting that the optimal inference, Z, is the most informative about the data X, while storing the least possible noise.

Computationally, that means that the problem is not the ELBO objective per se, but rather the chosen prior $p(z)$ for the representation. Indeed, bounding the latent (representation) entropy, i.e. choosing the correct inference variance, the optimal inference and generated samples are obtained.

The contribution of the present chapter can be summarised in the following points:

- The Minimal Description Length (MDL) representation of the data is the most robust to invariance
- Show the informative equivalence between two of the most popular variational objective for generative modelling: ELBO and Wasserstein distance, both are optimised by a MDL solution.
- The Wasserstein objective, by its definition, is more robust than ELBO to the hyper-parameter defining the respective object itself.

- Empirical observation that a representation with smaller entropy tends to decompose better the latent factor than the one with high entropy.

4.1 BACKGROUND AND RELATED WORK

4.1.1 *Notation*

The Evidence Lower Bound is described by two distributions: - the inference, q, and - the generative p one. They are associated respectively with the VAE encoder and decoder network. As q is the distribution modelling the visible data X, we assume that X is defined as $q(x) = \frac{1}{N}\sum_i \delta(x_i)$ and the generated distribution as $p_\theta(x) = \int p_\theta(x|z)p(z)$, where $p(z)$ is the prior distribution of the latent variable (parameter not learned during the training). The subscriber associated with the distributions denote the parameter set from which the distribution are modelled, e.g. $q_\phi(z|x) \sim \phi(x) + \varepsilon$, with $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$

4.1.2 *Variational AutoEncoder Issues*

Both the high entropy samples (*generative issue*) and the inaccurate representations (*inference issue*) are associated with the ELBO objective. In the following, we describe via an informative description of the network, the reason why the two issues arise, and we propose a possible solution to overcome both the issues.

BLURRINESS To understand the high-entropy issue it is necessary to underline that maximising the ELBO is equivalent to minimise the KL divergence between the joint real distribution $p(x, z)$ and inferred one $q(x.z)$: $D_{KL}(q(z, x)||p(z, x))$.

Indeed,

$$\begin{aligned} D_{KL}(q(z,x)||p(x,z)) &= -\mathbb{E}_{q(x,z)}[\log p_\theta(x,z) - \log q(z|x) - \log q(x)] \\ &= -\mathbb{E}_{q(x)}\Big[\mathbb{E}_{q(z|x)}[\log p_\theta(x|z)\log p(z) - \log q(z|x)] - \log q(x)\Big] \quad (4.2) \\ &= -\text{ELBO} - H_q(X), \end{aligned}$$

where the equivalence holds since the empirical entropy of the data X, $H_q(X) = \log(|\mathcal{D}|)$, is a constant of the problem.

The equivalence in (4.2) implies that the variance of the generative joint distribution $p(x, z)$ is higher (or equal) than the inference distribution $q(z, x)$. That issue follows directly from the Kullback-Leibler divergence property, that given two different distribution q and p, $D_{KL}(q\|p) \leqslant D_{KL}(p\|q)$ if p has higher variance than q. Indeed, for any given rare event x_r, the "pointwise" KL-divergence $q(x_r)\Big[\log \frac{q(x_r)}{p(x_r)}\Big]$ tends to infinity if $q(x_r) \ll p(x_r)$. Thus, a close to optimal ELBO tends to have larger generative variance than the exact one, i.e. $H_p(X) < H_q(X)$.

In order to avoid such an issue, two different alternative corrections were proposed: - choose a more flexible generator distribution $p_\theta(x|z)$, [26], or - consider a free-inference problem, relaxing the condition of an encoder $q(z|x)$ close to the prior $p(z)$, [25]. However, if the latter approach is really expensive because it is necessary to compute and evaluate the latent inferred variable (there is no more prior), the former one is often associated with the *posterior collapse* issue: the learnt representations do not share any information with the visible data, an extreme case of bed inference.

ENTANGLED REPRESENTATIONS The second issue affecting an ELBO learnt latent representation is the inaccurate description of the visible data, e.g. two data-points belonging to two different classes are not separated in the latent space, or the main factors describing the data (*generative factor*) are not distinguishable (*entangled* representation). The non-interpretability issue is often associated with two causes: either - the representation is not sharing the proper knowledge with the data, or - the latent representation is not close enough to the prior (assumed to have independent components). As we will see in the following sections, the two causes are strictly interrelated; indeed, intuitively the factors are separated if they are contained in the latent variable (the representation is informative), and if they appear in each feature once, i.e. the data knowledge has not to be redundant.

According to the description above, both the issues have an informative description: the generated data is blur if the generator is not informative enough, and the representation is not interpretable if the representation does not share the *correct* amount of information with the visible data. For this reason, in order to provide a solution to the issues listed above, it would be useful to provide a full informative description of the ELBO variational objective.

4.1.3 *Information theoretic description of the ELBO*

Following what has been presented in [5], let us consider the two terms defining the ELBO, the *distortion* $D = -\mathbb{E}_{q(x,z)}[\log p(x|z)]$ and the *rate* $R = \mathbb{E}_{q(x)}[D_{KL}(q(z|x)||p(z))]$ separately. The distortion D is the cross-entropy between the generative and inference distribution $H(q(x|z)||p(x|z))$ and can be rewritten as

$$\begin{aligned} D &= -\int q(x,z)\log p(x|z) \\ &= -\int q(x,z)\log\frac{q(x|z)p(x|z)}{q(x|z)} \\ &= H_q(X|Z) + \mathbb{E}_{q(z)}D(q(x|z)||p(x|z)), \end{aligned} \tag{4.3}$$

where $H_q(X|Z)$ is the conditional entropy with respect to the inference distribution $q(x,z)$. From the relationship above, and by non-negativity of the KL divergence we have that the distortion D defines a lower bound of the encoding information $I_q(X;Z)$,

$$H_q(X) - D \leqslant I_q(X;Z), \tag{4.4}$$

where we used the fact that the entropy of the visible data $H_q(X) = H$ is a constant of the system.

On the other hand, the rate R is an upper bound of the mutual information $I_q(X;Z)$, because

$$\begin{aligned}\mathbb{E}_{q(x)}[D_{KL}(q(z|x)||p(z))] &= \int q(x,z)\log\frac{q(z|x)}{p(z)}dxdz \\ &= \int q(x,z)\log\frac{q(z|x)q(z)}{p(z)q(z)}dxdz \\ &= D_{KL}(q(z)||p(z)) + I_q(X;Z).\end{aligned} \tag{4.5}$$

Thus, combining the two terms, we establish that the two ELBO terms are bounding the encoding information from below and above,

$$H - D \leqslant I_q \leqslant R. \tag{4.6}$$

An observation which suggests that we should think the ELBO as the objective looking for the optimal encoding information. The correct amount that guarantees both good quality samples and optimal inference. Such a description of ELBO shows once more, from a pure informative description, that the ELBO objective is (in principle) the correct one to learn both optimal inference and generative model. Indeed, the issues described in the previous section are caused by a wrong amount of information within the network.

TARGET RATE ELBO AND β-VAE In light of the informative description above, the ELBO can be equivalently described as the objective looking for the representation that shares the correct amount of information, say R_0, with the input data. In that scenario, both the inference and the generation are optimal.

For this reason in [5], assuming to know R_0, was suggested that the following Target-Rate ELBO be optimised

$$\mathbb{E}_{q(x,z)}[\log p(x|z)] - \gamma\Big|R_0 - \mathbb{E}_{q(x)}[D_{KL}(q(z|x)||p(z))]\Big|, \tag{4.7}$$

where $\gamma > 0$ is an hyper-parameter, to tune.

The objective in (4.7), due to the absolute value term, is not very tractable and moreover the correct target is unknown in general. For that reason in [5], it was suggested to consider the following relaxed version of (4.7):

$$\beta\text{-ELBO} = \mathbb{E}_{q(x)}\Big[\mathbb{E}_{q(z|x)}[\log p(x|z)] - \beta D_{KL}(q(z|x)||p(z))\Big], \tag{4.8}$$

where $\beta > 0$ is an hyper-parameter implicitly defininig the bound of the mutual information: the higher is the β the less is the information. The β-ELBO objective (4.8), was firstly proposed in [28], in the special case $\beta \gg 0$ to obtain disentangled representations.

Indeed, starting from the observation that, in general, an entangled representation contains too much information and the generative factors are stored in more than one feature, the idea of [28] was to bound the encoding information and push each generative factor in just one feature, in order to have independent factors. That approach, in general, gives us a disentangled representation but one which is not very informative (i.e. generative samples

of poor quality). For that reason, [11], was optimised the objective (4.12); where, during the training, the Target Rate increases to obtain a correct balance between inference and generative accuracy.

WASSERSTEIN DISTANCE The parameter β plays a fundamental role since it is an implicit definition of the target rate R_0 which, in general, is unknown. However, that parameter is often difficult to estimate, and, moreover, as observed in [28] for large β although the representation is disentangled, the generated samples are not good.

In order to avoid the selection of the β parameter, a possible choice is to remove the rate term, i.e. the same idea of the free-inference model, but where the latent prior is constrained to have a shape as close as possible to the inferred one.

Starting from a similar information-theoretic argument, as posited in [89], aiming to obtain an informative inference that fits the prior, an alternative objective for the VAE was proposed, wherein the information penalty upon the ELBO is removed.

Going into the details, combining (4.5) and (4.1), the ELBO objective can be rewritten as

$$\text{ELBO} = \sum_i \mathbb{E}_{q(z|x_i)}[\log p(x|z)] - D_{KL}(q(z)||p(z)) - I_q(X;Z), \tag{4.9}$$

then, by removing the $I(X;Z)$ term, the model is informative (i.e. no posterior collapse) and the inference model is still constrained to be close to the prior, $D_{KL}(q(z)||p(z)) < \varepsilon$. Moreover, observing that the KL divergence term $D_{KL}(q(z)||p(z))$ can be approximated with the simpler to compute Maximum Mean Discrepancy (MMD), $MMD(q(z)||p(z))$, in [89] was suggested to optimise the following objective

$$L_{Wass} = \mathbb{E}_{q(x,z)}[\log p(x|z)] - \lambda \cdot MMD(q(z)||p(z)), \tag{4.10}$$

with the KL divergence approximate with the MMD distance.

The same objective was derived independently in [57], where the KL divergence was approximated with an adversarial network.

More recently, [83] using an optimal transport argument, showed that the objective (4.10) is the variational upper bound of the Wasserstein distance arising between the visible and generated data. For that reason, henceforth, we will dub the objective in (4.10) as Wasserstein distance, or simply *Wass*, and the network optimising Wass will be termed Wasserstein AutoEncoder (WAE), in contraposition to the ELBO trained network that would be simply called VAE.

4.1.4 *Disentangled representation: A review*

We said that often the learnt representation by the ELBO trained model are entangled. And our goal is to learn a disentangled representation. If the entangled representation notion is relatively simple – the data representation is entangled if given two data-points belonging to two different classes the associated representations are not separate in the latent space – it is less clear what a disentangled representation is.

As in literature are provided many different definitions, see [53], In the following, we decide to consider the one provided in [28], because, for the authors of this manuscript, is the most intuitive one.

FRAMEWORK Let us assume that each data-point $x \in \mathcal{D}$ is generated by some independent $u \in \mathbb{R}^L$ and dependent $v \in \mathbb{R}^H$ generative factors with generative distribution $p(x|u,v)$. In this setting, the representation learning goal is to find a representation $z \in \mathbb{R}^K$, with $K \geqslant L$, such that $p(x|z) \approx p(x|u,v)$. In particular, we wish that the inferred representation $z \sim q(z|x)$ captures the independent latent factors u in a disentangled manner, (i.e. the variable z_i and z_j are independent for any $i \neq j$). The conditionally dependent data generative factors v can remain entangled within a separate subset of z that is not used for representing u. Equivalently, we can consider the factor v to be noise, wherein the goal of the representation is to separate such noise from the independent factors.

The independent factors by definition factorise (i.e. $q(u|x) = \prod_i q(u_i|x)$). For this reason, we expect that also the latent representation has to factorise, say for instance $p(z) \sim \mathcal{N}(0, \alpha I)$, $\alpha \in \mathbb{R}$. With the representation z that has to contain all the information about u, $H(Z) \approx I(U;X)$.

From that definition, we see that the disentanglement property is not directly related to any theoretical objective. It is rather more a computational aspect that is associated to the chosen prior $p(z)$, and on the metric evaluating the fitting of the inferred with the prior distribution, $D(q(z|x), p(z))$.

In light of the independence shown by the representation information, the objectives that are proposed to obtain disentangled representations are just a variant of the ELBO where to the standard objective it is added an external factor to guarantee the disentanglement of the factors. In particular, [35] added to the ELBO objective the Total Correlation (TC) term

$$TC(q(z)) = D_{KL}\Big(q(z)\|\prod q(z^{(j)})\Big), \tag{4.11}$$

which serves as a measure of the independence of the factors, which is null when all the factors are independent. Actually, as observed in [13] the TC is already minimised by the rate term, indeed

$$\begin{aligned} D_{KL}(q(z|x)\|p(z)) = D_{KL}(q(z|x)\|q(z)) + D_{KL}\Big(q(z)\|\prod q(z^{(j)})\Big) \\ + D_{KL}\Big(\prod q(z^{(j)})\|\prod p(z^{(j)})\Big), \end{aligned} \tag{4.12}$$

then the model in [35] can be considered as a refined ELBO. Moreover, the Rate decomposition explains the reason why in the standard β-VAE the β is chosen higher than 1: in order to constrain the TC more than the Mutual Information.

However, the decomposition in independent factors is not the only property of a good (disentangled) representation. Indeed, as observed in [58, 11] another property to take into account is the *latent overlapping*; in that the representation points have to lie neither too close no to far from each other; a property which is associated with the network information.

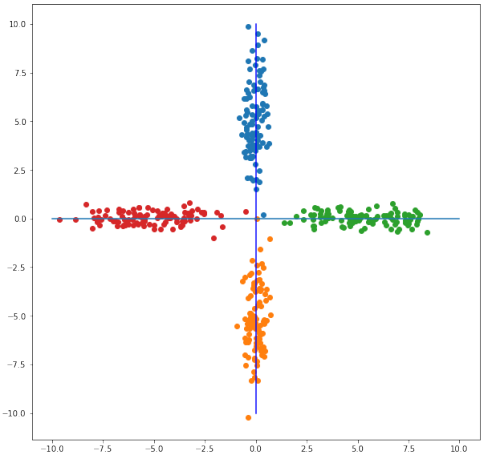


Figure 4.1 Example of disentangled representation of a four class data-set with Gaussian prior distribution. e.g. green class inference $q(z|x_g) = \mathcal{N}((-\mu, 0), diag(\sigma^2, \varepsilon))$

AN INFORMATION DESCRIPTION OF OVERLAPPING Each point in the latent space z_i, can be described in terms of its posterior distribution $q(z|x_i)$. By means of the posterior distribution, we can define the overlapping arising in the latent space as the area in common of the two distributions. By this definition we say that there is maximal overlap arising between z_1 and z_2 if $q(z|x_1) = q(z|x_2)$ for any z, and it is minimal if $q(z|x_i) = \delta(z_i)$, with $z_1 \neq z_2$. The two opposite scenarios correspond to the extreme cases: specifically, zero information for the maximal overlapping, and maximal information for the null overlapping.

By the description given above, in the optimal inference scenario, we expect that latent distributions of same-class points coincide, while latent distribution associated with different class data-points have null overlap. In the information theoretic setting, it is equivalent to state that $I(X; Z) = H(U)$ (i.e. the representation has to be informative as the generative (unknown) factors), and it has not to contain other information.

However, by empirical evidence, in agreement with the theoretical analysis, an unbounded network tends to learn an unnecessarily informative representation, $I(X; Z) > H(U)$, and for that reason, most of the representation learning objectives explicitly bound the encoding information.

As we will see in the following paragraph, the information bound has to be done properly, in order to avoid noise in the latent variables.

To better understand the concept of disentanglement and its information theoretic properties, we introduce an information based disentanglement metric: the Mutual Information Gap, firstly proposed in [13].

MUTUAL INFORMATION GAP The main idea behind the MIG is that a representation is disentangled if the information between each ground truth (independent) factor $u^{(k)}$ and a representation factor $z^{(j)}$ is maximal and zero for any other factors $z^{(j')}$, $j' \neq j$. Formalising the concept, a representation z is said to be disentangled if, for any ground truth factor $u^{(k)}$, there exists a representation factor $z^{(j)}$ such that $I(u^{(k)}; z^{(j)}) = H(z^{(k)})$ and $I(u^{(k)}; z^{(l)}) = 0$

for any $l \neq j$. This definition is in agreement with the prevailing notion of disentanglement and its geometric intuition of overlapping. Indeed, given any two data-points differing for just one generative factor, their representation has to overlap in all the axes apart from the one associated with the differing generating factor [1].

Assuming to know the correct generator $p(x|u)$, the joint distribution $p(z^{(j)}, u^{(k)})$ can be estimated as follows:

$$q(z^{(j)}, u^{(k)}) = \sum_{i}^{N} p(u^{(k)}) p(x_i|u^{(k)}) q(z^{(j)}|x_i), \tag{4.13}$$

and then estimate the MI $I(Z^{(j)}, U^{(k)})$,

$$I(Z^{(j)}, U^{(k)}) = \mathbb{E}_{q(z^{(j)}, u^{(k)})}\Big[\log \sum_{x_i \sim p(x|u^{(k)})} p(x_i|u^{(k)}) q(z^{(j)}|x_i)\Big] + H(Z^{(j)}). \tag{4.14}$$

Since we are assuming the ground-truth generative factors are discrete, we prefer to consider a normalised metric, (i.e. we divide the MI $I(Z^{(j)}, U^{(k)})$ by the generative factor entropy $H(U^{(k)})$).

Let us note that the normalised information serves only as an informative measure that is maximised for any factor $Z^{(j)}$, such that the encoding information $I(X; Z)$ is maximal, without taking into account the proper overlapping. To measure the overlap in the encoding space, the idea is to compute the Gap arising between the MI of the generative factor with the associated feature, $I(Z^{(j)}, U^{(k)})$, and the information shared with the most informative *wrong* feature, $I(Z^{(j')}, U^{(k)})$, with $Z^{(j')}$ the second most closely related variable to $U^{(k)}$. In this way the Mutual Information Gap (MIG) is defined as

$$\text{MIG} = \frac{1}{M} \sum_{m}^{M} (I(Z^{(j_m)}, U^{(m)})) - I(Z^{(j'_m)}, U^{(m)}))), \tag{4.15}$$

where $Z^{(j_m)} = \arg\max I(Z^{(j)}, U^{(k)})$ and $Z^{(j'_m)} = \arg_{j \neq j(m)} \max I(Z^{(j)}, U^{(k)})$.

Thus, that we define what a we mean with disentangled representation and we provide an information theoretic description of it we are ready to describe the Constrained InfoMax, the learning objective connecting the Wasserstein distance and the ELBO.

Observation. The MIG is only one of the possible disentanglement metrics, indeed have been proposed as many metrics as disentangled definitions. We select such a metric because is Information based and well describe the disentanglement property and moreover, because is one of the few that satisfies the two following properties [70]:

- the metric gives a high score to all representations that satisfy the characteristic that the metric reflects,
- the metric gives a low score for all representations that do not satisfy the characteristic that the metric reflects.

Two properties that we consider fundamental to define a metric a good one.

1 The metric is based on the assumption that both the generative factors U, and the generative map $p(x|u)$ are known. This assumption is quite strong and, since the choice of the ground truth factors is an arbitrary choice, the metric is intrinsically biased but, as observed in [53], all the disentangled metrics are biased.

4.2.1 *A unifying principle describing both the ELBO and the Wass objectives*

Starting from the same information analysis of the ELBO, two different objectives were proposed: Target-Rate ELBO (4.7) and the Wass (4.10). In the following, we will show that the two objectives are two different variational descriptions of the same information principle: the *Constrained InfoMax*.

From a purely theoretical information perspective, the target rate ELBO, (4.7), is learning the parameters such that both the encoding and decoding information coincide with R_0 and, the marginal $q(z)$ coincides with $z \sim p(z)$, or more formally it is equivalent to the following objective

$$\begin{cases} \max_q & I_q(X;Z) \\ \text{s.t.} & I_q(X;Z) = I_p(X;Z) \\ & I_q(X;Z) \leqslant R_0 \text{ and } H_q(Z) = H(Z), \end{cases} \tag{4.16}$$

where the InfoMax objective (1st term) and the first condition is associated with the distortion term, compare it with (4.3), and the last two terms with the rate term, see (4.5). Since the distortion term is a common term for all the VAE objectives, and then the first condition is common for all the objectives, from now on we simply write $I(X;Z)$, rewriting the objective in (4.16) as

$$\begin{cases} \max & I(X;Z) \\ \text{s.t.} & I(X;Z) \leqslant R_0 \text{ and } H_q(Z) = H(Z). \end{cases} \tag{4.17}$$

In this way it becomes clear that the objective in (4.16), is an Infomax objective bounded from above, by the Target-Rate R_0 and by the entropy $H(Z)$, indeed $I(X;Z) \leqslant H(Z)$.

THE OPTIMAL UPPER BOUND We have seen above that the Wass and the ELBO are two variational descriptions of the same objective: the Constrained InfoMax (4.17), where the constrain is, in principle, given by two terms: - the rate R_0 and the latent entropy $H(Z)$. Actually, the constrain is just one, the smallest of the two upper bounds, and as we will see the property of the learnt representation depends by the chosen upper bound. Let us consider the two cases, and see what are the optimal learnt representations in the two settings.

- $R_0 \ll H(Z)$: the objective is optimised by an inference distribution $q(z|x)$ such that $I(Z;X) = R_0$ and $H_q(Z) = H(Z)$, as the information shared beteeen the latent bariable and the generated data, R_0, is much smaller than the knowledge encoding information, we expect to have the generative issue described for the β-VAE, [28], whereupon the sample quality is poor. Moreover, if R_0 is too small, the encoding channel would be very noisy and so the learn representation cannot store all the generative factors (that issue is

common with the β-VAE representations, that are separating some generative factors but, not of all the generative factors are contained within the representation).

- $R_0 > H(Z)$: the encoding and decoding distribution do not coincide, since $I_q(X;Z) + D_{KL}(q(z)\|p(z)) = R_0 > H(Z)$, holds only if $D_{KL}(q(z)\|p(z)) > 0$. In this scenario we can expect good reconstruction performance (i.e. high information), but poor inference and generative samples, since the distribution from which to sample is unknown.

In light of the description above, since we are interested in yielding both good generative samples and optimal representations, the only scenario we should take into account is the one wherein the fixed rate R_0 is smaller than latent entropy $H(Z)$. In particular, as we know that the optimal encoding information has to be as close as possible to the latent prior entropy (noiseless channel), in the following we set $R_0 = H(Z)$.

By means of this analysis, in the setting $R_0 = H(Z)$ the objective in (4.17) can be further simplified in terms of:

$$\begin{cases} \max & I(X;Z) \\ \text{s.t.} & I(X;Z) \leqslant H(Z), \end{cases} \tag{4.18}$$

which is exactly the theoretical objective optimised by the Wass (4.10).

In this way, we have seen that for the optimal Target-Rate $R_0 = H(Z)$, the two variational objectives Target-Rate ELBO and Wasserstein Distance are equivalent. They are optimising the Constrained InfoMax (4.18), a theoretical information objective, learning an almost noise-free informative representation.

4.2.2 *Bound the latent Entropy to achieve disentangled representations*

By virtue of the preceding analysis, summarised in the equation (4.18), we have that the optimal scenario for the Target-Rate ELBO arises when the Target-Rate R_0 is chosen to be equal to the latent entropy $H(Z)$ and, by the disentanglement definition, we have observed that, in order to have a proper overlapping in the latent space it is first necessary to control the information in the network. By those two observations, we see that the choice of the (entropy) prior affects the quality of the learnt representation.

How to choose the correct representation entropy? The choice of the representation entropy, as seen in chapter 2, is equivalent to a model selection problem, in that the latent representation first defines the information inside the network and then its complexity. By the analogy between the model selection and the optimal representation problem, it is reasonable to state that *the optimal representation is the one optimising the Minimum Description Length (MDL) principle*.

The MDL principle, introduced in the second chapter, assert that the optimal representation is the smallest descriptive latent variable Z ($H(Z)$), such that the information lost in the channel, $H(X|Z)$, is deemed to be acceptable:

$$\min_{\theta, p(z)} H_\theta(X|Z) - \lambda H(Z). \tag{4.19}$$

Let us notice that the objective in (4.19) is similar to the Wass objective in (4.10) wherein the conditional entropy $H(X|Z)$ is substituted with the Distortion D, its lower-bound, and the entropy term, with the KL-divergence term, $D_{KL}(q(z)||p(z))$. Indeed, the Wass objective is the lower bound of (4.19) in the setting where $p(z)$ is fixed.

Although the MDL objective cannot be optimised directly, it will provide a framework by which to define the optimal network and then the optimal representation. Indeed, as we will see in the following section, the MDL solution retains the main properties of the optimal representation.

MINIMUM DESCRIPTION LENGTH FOR OPTIMAL SOLUTION We have seen that the optimal representation is the most informative about the data and with a proper overlapping. Thus, to show the optimality of the MDL solution, we have to consider the two properties separately. We do not ask to be disentangled, since the separation in independent factors depends on the choice of the prior $p(z)$.

The informativeness is simple, since the MDL has a minimal $H(X|Z)$, or equivalently maximal $I(Z;X)$ for any Z. The proper overlapping property is not easy to verify, so for this reason, we prefer to show that an optimal representation is maximally invariant to noise, a property closed related to the proper overlapping.

A representation Z is said to be *invariant* to nuisance ν in the input, if the two variables are independent, (i.e. $I(Z;\nu) = 0$). In alternative terms, a representation is said to be invariant if $q(z|x+\nu) = q(z|x)$ for any x. By that definition, we see that a complete overlapping representation is invariant (if any data-point is mapped into the same latent point). On the other hand, the most informative is highly susceptible to noise, since any visible point is mapped to a different latent point with zero overlapping, i.e. for any small noise ν, the representations of x_i and $x_i + \nu$ differ.

Excluding the trivial invariance for the total overlapping, $H(Z) = 0$, it follows from the optimal overlapping description above that the *maximally invariant representation to nuisance is the one with proper overlapping*. Indeed, the generative factors of x and $x+\nu$ are the same, and if the representation is affected by the nuisance is changing only on the active generative factors (that are a small subset of all the possible options).

By that intuitive relationship, to show that the representation is disentangled (it has the proper overlapping) it is enough to prove that the representation is robust to noise. Thus, from the following proposition we can assert that: *the optimal representation is the one optimising the MDL principle* (4.19).

PROPOSITION The MDL solution is the maximally invariant with respect to noise ν in the input data.

By the chain rule of the MI, the following equivalence holds: $I(X; Z, \nu) = I(X; \nu) + I(X; Z|\nu)$. Observing that the conditional information can be estimated as

$$\begin{aligned} I(Z; X|\nu) &= H(X|\nu) - H(X|Z, \nu) \\ &= H(X) - H(X|Z, \nu) \\ &\geqslant H(X) - H(X|Z) \\ &= I(X; Z), \end{aligned} \tag{4.20}$$

where the equivalence in the second line follows the independence between the noise ν and the data X, and the inequality in the third line follows from the Jensen inequality (we put an average inside the log). Moreover, remembering the entropy bounds of the MI, it is possible to bound the MI between the representation and the nuisance as follows

$$\begin{aligned} I(Z; \nu) &= I(Z; X, \nu) - I(Z; X|\nu) \\ &\leqslant H(Z) - I(X; Z). \end{aligned} \tag{4.21}$$

Remembering that the $H(X)$ is a constant of the system, the bound is exactly the opposite objective of the MDL solution (with Lagrangian equal to 1), then we derived that the MDL solution is also the most invariant to noise.

The good properties of the MDL solution confirm that the Constrained InfoMax solution is the optimal objective, wherein the latent prior has to be defined as the one with minimal entropy such that the decoding information is maximal.

4.3 EXPERIMENTS

In the theoretical section, we have seen that both the ELBO and the Wass are two variational objectives associated with the same principle, in that they are seeking the same representation, namely the one maximising the information available with the visible data such that no knowledge in the latent representation is redundant. In the following section, we want to verify whether the theoretical assertions are confirmed by the computational experiments.

In particular, in the following experiments, we compare the ELBO and Wass objective to show that:

- the optimal (β-)ELBO is equivalent to the Wass objective,
- to control the encoding information it is more fruitful to bound the latent entropy than the rate term in ELBO, and
- the MDL solution is the optimal one: i.e. disentangled and informative.

For this purpose, the section is divided in two parts: first we describe the effect of the β (the encoding information) in the ELBO trained solutions, where it is observed that it is necessary to find a balance between generative and inference task and that an optimal solution is not

associated with the most informative representation (i.e. $\beta \approx 0$); in the second part we describe the effect of the variance σ^2 in the Wass trained model, the WAE. Indeed, a way to bound the latent entropy is to bound each feature variance (σ^2), another possible approach is to reduce the number of latent feature, the latter approach would be not discussed in future works. From the WAE analysis, we observe that the Wass trained representations almost coincide with the optimal ELBO, and that bounding the latent entropy (the variance of the prior) is a way to achieve better, more disentangled and robust to nuisance, representations.

DATASETS AND CONFIGURATIONS The experiments are performed with different network architectures that are associated with different datasets. We consider 4 data-sets: the MNIST [48] a standard dataset consisting of 70k, 28×28 grey-scale, handwritten digits; the Omniglot [44] a set of 1623, 28×28 grey-scale, characters drawn online via Amazon's Mechanical Turk by 20 different people; the CelebA [52] a collection of more than 200k of 40×40 cropped celebrity images; and finally, the DSprites [59] a synthetic dataset of more than 70k, 64×64 grey-scale, images generated by six ground truth independent latent factors: rotation, translation in the x and y axis, shape, orientation, dimension. The respective network architectures are described in table 4.1.

For the greyscale data-sets, we pre-processed these to create binary data-sets, and we consider, as Distortion, the binary cross entropy

$$\mathbb{E}_{q(z|x)}[-\log p(x|z)] = x \cdot \log(x_g), \qquad x_g \sim p(x|z), \tag{4.22}$$

for the colour data-set, we instead consider the standard Gaussian entropy

$$\mathbb{E}_{q(z|x)}[-\log p(x|z)] = \|x - x_g\|_2^2, \qquad x_g \sim p(x|z). \tag{4.23}$$

Finally, for the ELBO, the rate R is computed as suggested in [37],

$$\mathbb{E}_x[D_{KL}(q(z|x)\|p(z))] = \frac{1}{2}\sum_i \sum_j (1 + \log((\sigma_i^{(j)})^2) - (\mu_i^{(j)})^2 - (\sigma_i^{(j)})^2), \tag{4.24}$$

and the KL divergence $D_{KL}(q(z)\|p(z))$ in the Wass with the Maximum Mean Discrepacy $MMD(q(z), p(z))$, defined as

$$MMD(q, p) = \| \int k(z, \cdot)p(z) - \int k(z, \cdot)q(z)\|_{H_k}, \tag{4.25}$$

with $k : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$, a kernel function living in the Reproducing Kernel Hilbert Space $\mathcal{H}_K$, the space of all possible kernels. In particular for all our experiments we consider the inverse multi-quadratic kernel $k(x, y) = C/(C + \|x - y\|_2^2)$, with $C = dim(\mathcal{Z})$.

As the model architecture is the same as the one utllised in [83], for consistency purpose we decided to consider the same training procedure and hyper-parameters utilised in [83].

4.3.1 *Balance between encoding and decoding information*

GENERATIVE PERFORMANCES The first issue of the VAE model was to generate blurred images, since the entropy of the generated data was greater than the actual data entropy.

Table 4.1 Network architectures of different VAEs

Dataset	Optimiser	Architecture	
MNIST Omniglot	Adam 10^{-3}	Encoder:	Conv(4×4, 128), BN + ReLu
			Conv(4×4, 256), BN + ReLu
			Conv(4×4, 512)
			BN + ReLu
			Conv(4×4, 1024), BN + ReLu
			FC 2K, $K = 10$
		Decoder:	FC $8 \times 8 \times 1024$
			FSConv(4×4, 512), BN + ReLu
			FSConv(4×4, 256), BN + ReLu
			FSConv(4×4, 128), BN + ReLu
			FSConv(4×4, 1)
CelebA	Adam 10^{-4}	Encoder:	Conv(4×4, 128), BN + ReLu
			Conv(4×4, 256), BN + ReLu
			Conv(4×4, 512)
			BN + ReLu
			Conv(4×4, 1024), BN + ReLu
			FC 2K, $K = 64$
		Decoder:	FC $8 \times 8 \times 1024$
			FSConv(4×4, 512), BN + ReLu
			FSConv(4×4, 256), BN + ReLu
			FSConv(4×4, 128), BN + ReLu
			FSConv(4×4, 1)
Dsprites	SGD 10^{-4}	Encoder:	FC 1200, BN + ReLu
			FC 1200, BN + ReLu
			FC 2K, $K = 10$
		Decoder:	FC 1200, BN + ReLu
			FC 1200, BN + ReLu
			FC 1200, BN + ReLu
			FC 4096

Table 4.2 Quality samples scores for ELBO trained VAEs. Negative Log-Likelihood for MNIST, Omniglot and DSprites, and Frechet Inception Distance for CelebA (smaller is better).

Dataset	$\beta(\log_{10})$				
	-2	-1	0	1	2
MNIST	81	**70**	120	190	210
Omniglot	121	**109**	111	153	186
DSprites	53	**44**	57	96	246
CelebA	161	**150**	218	271	298

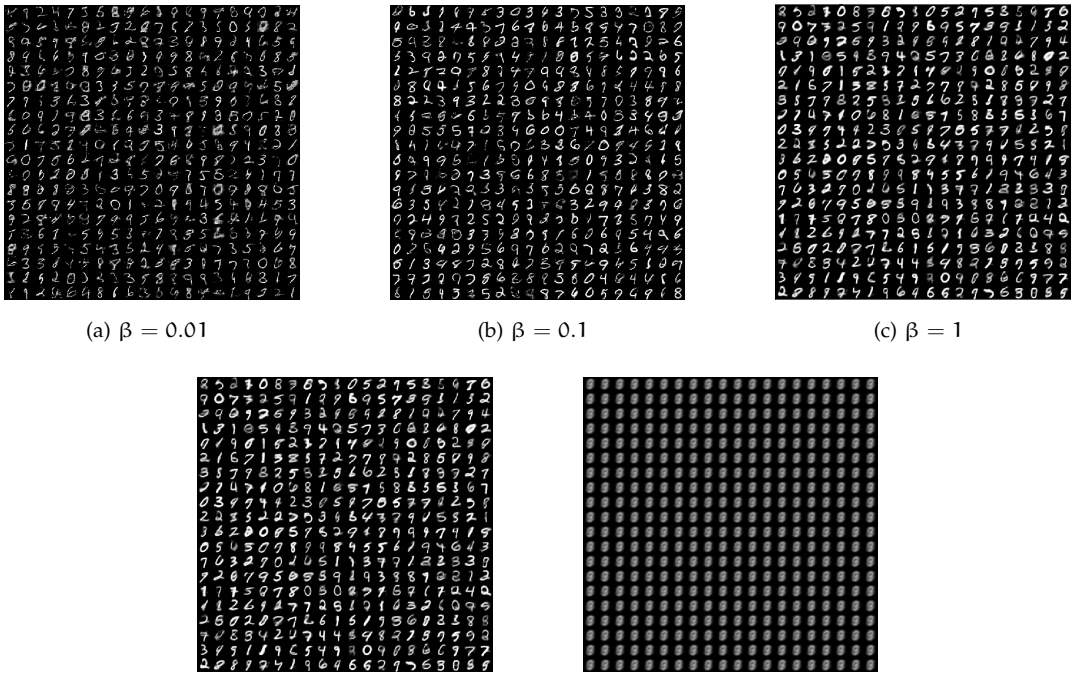
This phenomenon could be explained as an attempt to fill the gap between the decoding information with the real optimal information. Indeed, for any fixed representation, the only way to maximise the decoding information is to maximise the entropy of the generated data. For this reason, a good estimation of the decoding information, or better of the (generative) conditional entropy $H_\theta(X|Z)$ is the sample quality. In agreement with the intuition; for any fixed generative family of models, the better the generated samples, the more informative the representations are. Let us note, that this estimation is acceptable only in the case the posterior collapse is avoided (otherwise it is possible to obtain optimal generative performances with zero information between the inferred and generated data).

To evaluate the quality of the generated samples the natural metric is the Negative Log Likelihood (NLL), $\text{NLL}(X) = -\mathbb{E}_x[\log p_\theta(x)]$, upper bound of the KL divergence $D_{KL}(p(x)\|p_\theta(x))$, that can be estimated via importance sampling method [24]. The NLL serves, in general, as a good proxy when the data-structure is relativly simple and the visible data describes completely the possible data samples. In the more complex setting, in which the visible data are just a small sample of all the possible data-points, it is common assumption to evaluate the quality of the sampled images using other metrics. In specific terms, in [27] was suggested that for computer vision problem the Frechet Inception Distance (FID) be considered to measure the distance arising between those feature vectors calculated for real and generated images. The FID is formally defined as:

$$\text{FID} = \|\mu_v - \mu_g\|^2 + \text{Tr}(\Sigma_v + \Sigma_g - (\Sigma_v\Sigma_g)^{\frac{1}{2}}), \tag{4.26}$$

or equivalently the Wasserstein distance arising between two Gaussian distributions $X_{v,g} \sim \mathcal{N}(\mu_{v,g}.\Sigma_{v,g})$, describing respectively the visible and the generated data.

According to the relationship in (4.3) and (4.5), one way to maximise the information within the representation is to train a VAE with small β, since the smaller the β is, the higher the encoding information. But that naive approach, as we shall see qualitatively from figures 4.2, 4.3, 4.4 and quantitatively from table 4.2, is not the correct one. Indeed, the optimal generative performances are not obtained for the smallest β. This phenomenon shows, that the most



(a) $\beta = 0.01$ (b) $\beta = 0.1$ (c) $\beta = 1$

(d) $\beta = 10$ (e) $\beta = 100$

Figure 4.2 Samples generated by VAE with different βs, trained with MNIST. For small β the diversity is reduced since the latent does not fit with the prior. Instead for large β the samples are blurred since small amount of information are transferred.

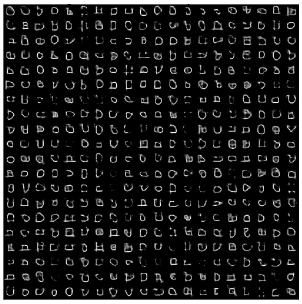
informative encoder is not associated with informative decoder, (i.e. the representations are storing not relevant data information and thus it is necessary to contain that quantity). If we view this kind of extra unnecessary information as noise, then we understand why the optimal samples are obtained for a relatively large β as, in a small information setting only the most relevant information is shared.

DISENTANGLEMENT The generative performance analysis is a phenotypical description of the learnt representation, and it is useful to describe the informativeness of the representations. However, as observed above, an optimal representation has to separate out the generative factors (i.e. an optimal representation has to be *disentangled*).

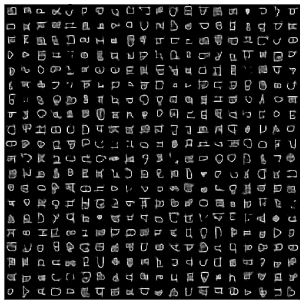
The analysis in this setting is not straightforward, because it depends on the dataset. Indeed, the disentangled metrics can be used only if the ground-truth generative factors are known.

Known generative factors. Let us start with the simple case in which the ground-truth factors are provided, and it is possible to explicitly compute the MIG score, the metric that we choose in order to evaluate the disentanglement. Of the datasets taken into consideration, the only one having this property is DSprites.

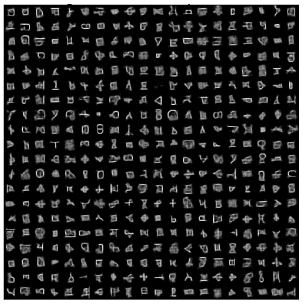
In the same way as performed previously for the generative classification, we compared the MIG associated values for each VAE trained with different βs. As we see from the MIG scores in figure 4.5, the disentangled property is related to the generative quality metric. Indeed, the two most disentangled representations are the ones with minimal NLL, respectively $\beta = 0.1, 1$.



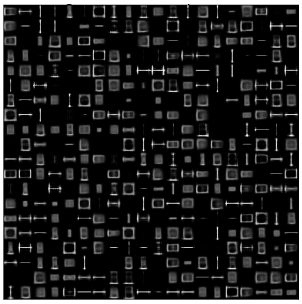
(a) $\beta = 0.01$



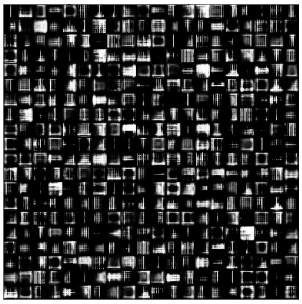
(b) $\beta = 0.1$



(c) $\beta = 1$



(d) $\beta = 10$

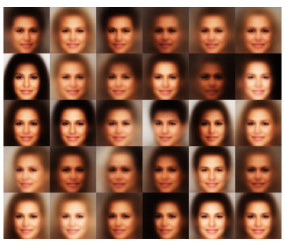


(e) $\beta = 100$

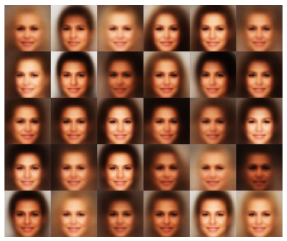
Figure 4.3 Samples generated by VAE with different βs, trained with Omniglot. For small β just one datum is sampled whereas, for large β the samples are blurred since small of information are transferred. In particular, for $\beta = 100$ the samples do not look like characters.

(a) $\beta = 0.01$

(b) $\beta = 0.1$



(c) $\beta = 1$



(d) $\beta = 10$

(e) $\beta = 100$

Figure 4.4 Samples generated by VAE with different βs, trained with CelebA. In all the settings the samples are not really diverse, and this is particularly visible in the $\beta = 100$ setting wherein the non-black samples are just few.

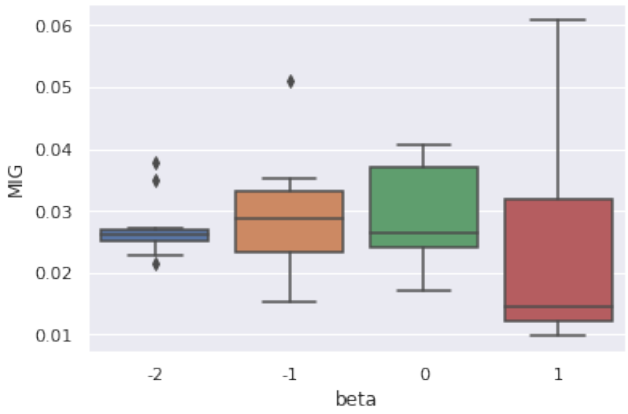


Figure 4.5 Comparison of MIG scores obtained by the same VAE trained with different ELBOs with the DSprites dataset. The larger the beta, the higher the variance, and the higher is the maximum value. To achieve a disentangled representation, it is necessary to bound the encoding information.

This behaviour, in agreement with the intuition, confirms that the generative performance is a benchmark for the representation quality.

The MIG scores summarised in figure 4.5 also highlight the importance of finding a balance between the encoding information and the sample quality. Indeed, if we look at the MIG maximum values, we see that the less informative it is the more disentangled, and the more informative it is the more entangled. This behaviour suggests that the most informative has representations that are highly redundant, and thus a generative factor can appear within more than a feature. Instead, the least informative representation has to contain knowledge about generative factors in just one feature. In the same way, the high variance associated with the minimal informative encoder shows that, for an high β, it is not guarantee that we should learn a representation sufficiently informative to generate good samples.

To better understand the balance between informative and disentangled representation, it is useful to continue our representation analysis with the other datasets.

DECOMPOSABILITY For real datasets, as the ground-truth generative factors are unknown, it is not possible to consider any metric to evaluate the disentanglement. However, we can still proceed with a decomposability analysis. The decomposability is a property firstly defined in [58] that generalises the disentanglement to the context in which the prior does not factorise. In detail, the representation is said to be decomposed if the latent variables:

- have an appropriate level of overlap,
- conform to the desired prior structure.

As we see, this definition differs from the disentanglement in one main aspect, in that the data representation should not be orthogonally distributed. Thus a disentangled representation is a decomposed one, but the opposite does not hold true.

Table 4.3 Decomposition metrics: listed pair (Hoyer, MMD), the more decomposed is closer to the origin (0,0). For a large β, since most of the features are zero, the representations are sparse, but these representations are not informative. This is visible from the generated samples that are far from the original ones, and by the high MMD distance: in that the latent entropy is smaller than the prior entropy.

Dataset	$\beta(\log_{10})$				
	-2	-1	0	1	2
Dsprites	(0.68, 0.25)	(0.57, 0.06)	**(0.42, 0.08)**	(0.49, 0.54)	-
MNIST	(0.71, 0.36)	(0.74, 0.06)	**(0.73, 0.05)**	(0.64, 0.36)	(0.78, 1.84)
Omniglot	(0.71, 0.62)	(0.73, 0.18)	**(0.71, 0.05)**	(0.73, 1.21)	(0.59, 1.56)
CelebA	(0.8, 3.09)	(0.72, 1.99)	**(0.67, 1.54)**	(0.71, 1.54)	(0.78, 1.92)

To provide a measure of the decomposition, let us remember that, given two data-points differing for just one factor, the associated representations overlap correctly if they are coincident in all the latent features but one. Then, in the case that the generative factor is just one for each data-point, the correct overlapping coincides with the *sparse* notion, a classic property of the optimal representation [15, 45, 49, 65]. Indeed, in a single-factor generated data, the representation is constant (say 0) in all the features but the correct one.

In light of the connection arising between the overlapping and the sparseness notion, following on from what was determined in [58], we decided to estimate the sparsity of the representation with the Hoyer distance [33]

$$\mathrm{Hoyer}(y) = \frac{\|y\|_1/\|y\|_2 - 1}{\sqrt{d} - 1}, \tag{4.27}$$

for any given point $y \in \mathbb{R}^d$, being 1 for a full dense vector and 0 for a full sparse vector. Since we are interested in computing the metric with respect to the learnt representations, in the following we consider the average measure, $\mathrm{Hoyer}(Z) = \frac{1}{N}\sum_i \mathrm{Hoyer}(\tilde{z}_i)$, where we decide to consider the normalised variance $\tilde{z} = z/\sigma(z)$, just to avoid the eventuality that a larger entropic distribution is considered more sparse than the one with small entropy.

Let us notice, that the Hoyer metric in (6.24), is not exactly the one introduced in [33], but an equivalent variant. That choice was made so as to describe the metric as a distance and to be consistent with the measure to evaluate the proper fitting.

To evaluate the conformity of the embedding to the prior, we considered the MMD distance between the prior $p(z)$ and the inferred distribution $q(z)$, as seen above, which is considered a special distance in the probability space.

The disentanglement results are summarised in table 5.1. First of all, to validate the decomposability analysis, let us note that the disentanglement and decomposability metrics are concordant. Indeed, in agreement with the MIG results listed in 4.5, the most decomposed representations for DSprites are those obtained by the VAE trained with $\beta = 0.1$ and 1. That

result confirms the disentanglement is a special case of the decomposability; and the two metrics estimating the decomposability serve as a good proxy to evaluate the disentanglement.

As we see from table 5.1, the optimal representation – sparse and fitting the prior – essentially does not exist; Indeed, the most sparse representation do not fit the prior representation. The balance between the two metrics is, in general, found for a $\beta \approx 1$. Moreover, from table 5.1, we see that the main discriminant is the MMD measure, and not the Hoyer which is similar for all the data; suggesting that the sparseness factor, which varies differently from the prior fitting, is not associated to the network information, but rather is more a data property.

The different prior learnt by the different models tested is motivated by the quantity of information shared with the data. For high (small) β the encoding information is small (high); the inferred entropy is smaller (higher) than the prior, and the the MMD value high. Thus, the most decomposed is the one sharing the correct amount of information (i.e. a quantity close to the latent entropy).

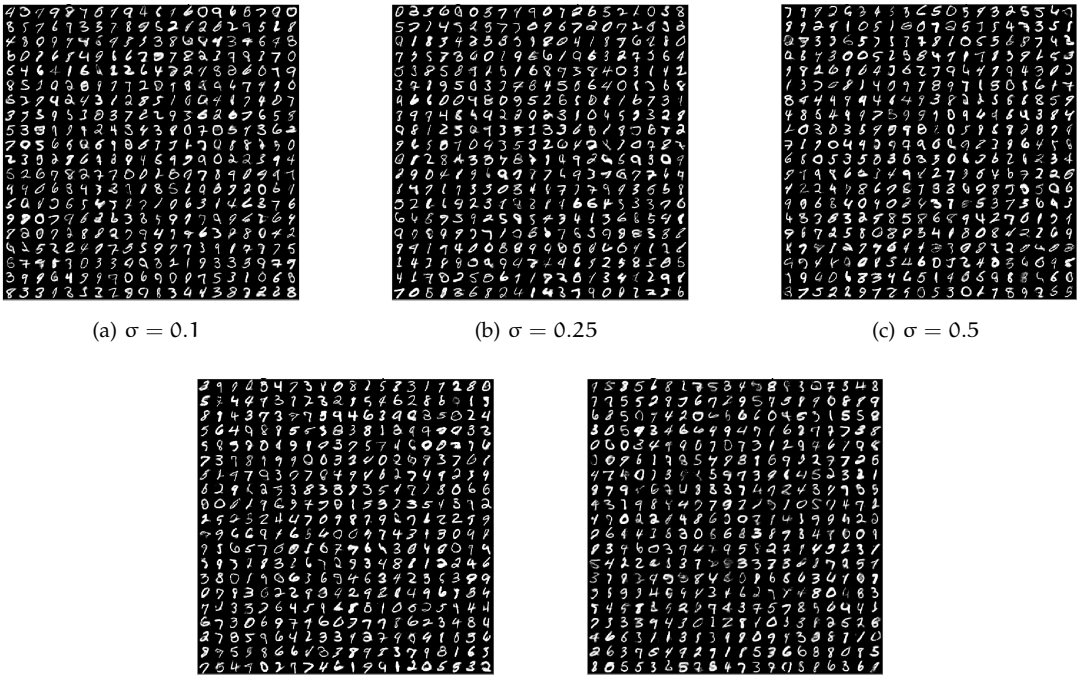
Let us conclude these experiments observing that in the ELBO it is difficult to find the optimal β. This difficulty is confirmed by the non-total agreement arising between the sample quality and decomposable performance. The models with optimal generative performances are not the one learning the best representations and vice-versa.

4.3.2 *InfoMax and the role of the prior*

In the experiments detailed above, we have observed that finding the optimal β from which are obtained the optimal generative and inference performances, is not an easy task. In this second experiment part, we want to show that the optimal solution is learnt without much effort by the WAE and also that the disentanglement property depends upon the entropy of the latent variable more than the encoding information. Latent entropy that is implicitly defined by the hyper-parameter σ. Indeed, as the prior is assumed to be Normally distributed, for any given number of latent units (K), the latent entropy ($H(\mathcal{N}(0, \Sigma)) = \log((2\pi e)^K \det(\Sigma))^{1/2}$) is defined by the covariance matrix Σ and, as the factors are supposed orthogonal and equally important, by the diagonal covariance matrix $\Sigma = \sigma^2 I$.

GENERATIVE PERFORMANCE The first task for which we compare the different VAEs objectives is the generative one. For this, we start the description of the WAEs. As we see from table 4.4 and from the samples presented in figures 4.6, 4.7 and 4.8 the sample quality of the different WAEs is comparable to or better than the best VAE models.

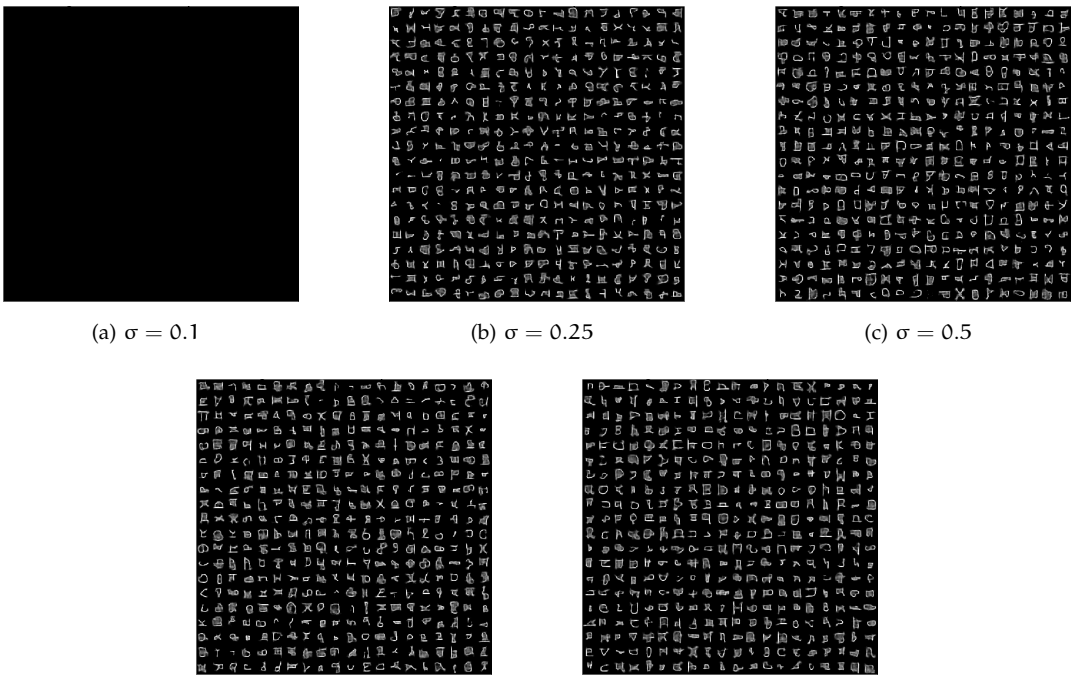
In particular, if we compare the $\sigma = 1$ WAE with the (optimal) VAE, the two models yield very similar results, with the WAE model having slightly better results with the only exception being the DSprites setting, wherein the VAE samples are of better quality that the WAE ones. That observation is in agreement with the theoretical results, wherein the optimal VAE and the Wass should share the same information and then have similar generative performance.



(a) $\sigma = 0.1$ (b) $\sigma = 0.25$ (c) $\sigma = 0.5$

(d) $\sigma = 1$ (e) $\sigma = 1.5$

Figure 4.6 Samples generated by WAE, trained with MNIST, with different σs, (i.e. different latent entropies). There are no visible differences arising between the samples generated by the different models, suggesting that unnecessary information is transmitted in the high variance setting.



(a) $\sigma = 0.1$ (b) $\sigma = 0.25$ (c) $\sigma = 0.5$

(d) $\sigma = 1$ (e) $\sigma = 1.5$

Figure 4.7 Samples generated by WAE, trained with Omniglot, with different σs. No samples are generated with $\sigma = 0.1$, since the shared information was not sufficient (Omniglot has larger entropy than MNIST). Instead for the other models there are no visible differences arising between the generated samples.

(a) $\sigma = 0.1$ (b) $\sigma = 0.25$ (c) $\sigma = 0.5$

(d) $\sigma = 1$ (e) $\sigma = 1.5$

Figure 4.8 Samples generated by WAE, trained with CelebA, with different σs. Small σ are associated with worse samples, suggesting that, in that case, the latent representations do not share sufficient information with the visible data.

Table 4.4 NLL (FiD for CelebA) obtained by WAE trained with different priors and different data-sets

Dataset	σ (Wass)					ELBO
	0.1	0.25	0.5	1	1.5	β^*
Dsprites	72	48	52	48	53	**44**
MNIST	70	74	**63**	67	83	70
Omniglot	-	109	**104**	107	108	109
CelebA	102	100	**98**	101	99	150

Moreover, from table 4.4, it is possible to infer that the choice of the prior has no real effect upon the generative performance, suggesting that not all the information stored in the latent variable is necessary for the task itself.

It is also important to note that, in the CelebA setting, the samples generated by a low sigma prior do not appear to be of better quality than those generated by the optimal VAE (see figure 4.8); although the FID score in figure 4.4 does not highlight that difference. This phenomenon can be explained if we remember that the FID score takes into account the diversity of the samples. In other words, the optimal VAE samples, although of decent quality, are not diverse as those generated by a small entropy prior.

DISENTANGLEMENT As observed for the VAE analysis, generative accuracy is merely one side of the coin:- to evaluate the learnt representation quality, it is necessary to look inside the

(a) feature 0 ($8 \rightarrow 9$)

(b) feature 1 ($9 \rightarrow 2$)

(c) feature 2 ($3 \rightarrow 4$)

(d) feature 3 ($0 \rightarrow 8$)

(e) feature 4 ($8 \rightarrow 0$)

(f) feature 5 ($3 \rightarrow 9$)

(g) feature 6 ($8 \rightarrow 9$)

(h) feature 7 ($5 \rightarrow 7$)

Figure 4.9 Traversal associated with the VAE representation, trained with MNIST and $\beta = 10$. All the samples are blurry, and not all the digit classes are visible, suggesting that each data-point is generated as a combination of more than one feature.

network. Moreover, we expect that the disentanglement analysis will underline the role of the latent entropy within the WAE model; differences that are not easy to observe by means of the generative comparison made above.

In accordance on what we have done in the ELBO comparison, we start the disentanglement analysis with the MIG scores of the Dsprites dataset. From the MIG comparison illustrated in figure 4.11, we see that the $\sigma = 1$ WAE representation is disentangled as is the ELBO one, confirming that the optimal VAE is coincident with the WAE. Moreover, we see that bounding the latent entropy is a way to bound the encoding information. Indeed, in agreement with the behaviour seen in table 4.5, the most informative representations (respectively small β or high σ) have smaller variance in terms of MIG score and, in contrast, the smallest informative representations are associated with the highest MIG variance.

(a) feature 0 ($8 \rightarrow 9 \rightarrow 4$)

(b) feature 1 ($9 \rightarrow 7 \rightarrow 8$)

(c) feature 2 ($2 \rightarrow 7 \rightarrow 0$)

(d) feature 3 ($0 \rightarrow 1$)

(e) feature 4 ($4 \rightarrow 9 \rightarrow 3$)

(f) feature 5 ($7 \rightarrow 9 \rightarrow 6$)

(g) feature 6($9 \rightarrow 0 \rightarrow 1$)

(h) feature 7 ($2 \rightarrow 5$)

Figure 4.10 Traversal associated to the WAE representation, trained with MNIST and $\sigma = 1$. Each digit is associated with at least one feature. Excluding the "9" and "7" that are the two digits associated with the null space, each digit appears in just two transversal, confirming that a disentangled representation is sparse.

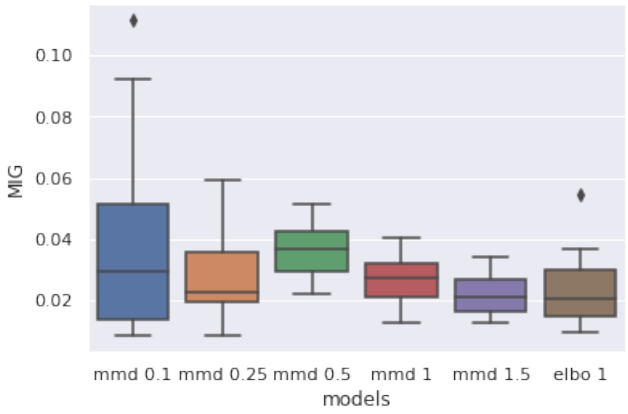


Figure 4.11 Comparison Dsprites MIG scores obtained by the WAE with different priors $p(z) \sim \mathcal{N}(0, \sigma^2 I)$, for different standard deviation σ, with the optimal VAE with $\sigma = 1$

The differences arising between the VAE and WAE representations are also visible when looking at the generated transversal data, which are plotted in figures 4.9, 4.10, 4.12 and 4.13. The data generated by a single feature of each representation, maintaining the others fixed to 0. In particular, by this qualitative comparison, we notice that, although the VAE and WAE representations are almost equally informative about the data (the sample quality is comparable), the WAE representations look more disentangled than the VAE ones.

Let us also notice that the WAE traversal have all the same pattern: blank data (around $z = 0$) and data associated with two different classes. The reason of such a behaviour would be explained in the next chapter, but for now let us observe that is the same pattern sketched in figure 4.1, where the inference distribution of each data-class is distributed along a subspace and the center of the distribution is empty.

DECOMPOSITION In the classic setting in which the generative factors are unknown, it is necessary to resort on the indirect metrics, namely the Hoyer evaluating the sparseness and the MMD distance which is used to evaluate the fitting of the latent with the prior. Since in the Wass, the MMD distance is directly minimised, the latter quantity is negligible for all the models, thus in table 4.5 are listed only the Hoyer scores.

From the Hoyer scores presented in table 4.5, we see that the WAE learnt representations are, in general, at least as decomposed as those for the optimal VAE latent variables. Indeed, the WAE representations are equally or more sparse than the VAE counterpart, as in the MNIST and Omniglot setting, or else are slightly less sparse, but nonetheless fitting the prior as in the CelebA. The only WAE representations that are less decomposed than the VAE ones are the DSprites representations. This result is in agreement with the low Log-Likelihood measures, but not with the MIG results, where the WAE representations are the most disentangled. These results highlight how difficult it is to find a metric that satisfactorily evaluates the optimal representation.

The effect of σ. Both the Hoyer and the MMD metrics are evaluating the geometric properties of the learnt features and, as observed in table 4.5, they are unable to discriminate

(a) feature 0

(b) feature 1

(c) feature 2

(d) feature 3

(e) feature 4

(f) feature 5

(g) feature 6

(h) feature 7

(i) feature 8

(j) feature 9

Figure 4.12 Dsprites generated by each representation of the optimal VAE, $\beta = 0.1$

(a) feature 0

(b) feature 1 (unused)

(c) feature 2 (shape)

(d) feature 3 (unused)

(e) feature 4

(f) feature 5

(g) feature 6 (orientation)

(h) feature 7 (position y)

(i) feature 8 (unused)

(j) feature 9 (position x)

Figure 4.13 Dsprites generated by each representation of the optimal WAE, $\sigma = 0.25$

Table 4.5 Comparison of the decomposable quality of the representation learnt by WAE, with different prior $p(z)$. Since the MMD distance arising between the prior and inference is ever risible, in the table the decomposability is described only by the Hoyer metric (smaller is better)

Dataset	σ (Wass)					ELBO
	0.1	0.25	0.5	1	1.5	β^*
Dsprites	0.63	0.59	0.58	0.63	0.57	**0.42**
MNIST	0.73	0.75	**0.71**	0.73	0.72	0.71
Omniglot	0.75	0.64	0.69	**0.62**	0.74	0,71
CelebA	0.78	0.79	0.78	0.77	0.78	**0.67**

Table 4.6 L_2 norm of the difference between the visible and the reconstructed corrupted data, $\|x - WAE(\hat{x})\|$, where $\hat{x} = x + \nu$, with $\nu \sim \mathcal{N}(0, 0.4I)$, (smaller is better)

Dataset	σ (Wass)					ELBO
	0.1	0.25	0.5	1	1.5	β^*
Dsprites	35	32.7	30.4	**28.6**	30.2	31.3
MNIST	36.3	34.2	34.1	32.3	**30.4**	35.2
Omniglot	48.5	47.2	46.4	41.5	46.7	**41.2**
CelebA	99.6	98.3	93.2	**90.4**	93.7	480

in a clear way the differences between the learnt representations as achieved by the MIG score. In light of those difficulties, to evaluate the differences arising between the learnt representations, we decided to consider, as a decomposable metric, the robustness of the network. Indeed, as seen in the theoretical section, the invariance to input nuisance is a property that is strictly related to the proper overlapping and then to the disentanglement.

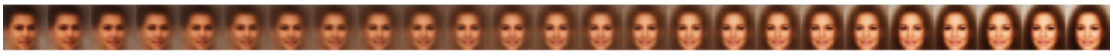
Given a data-point x, its representation $z = z(x)$ is invariant to input noise if $z(x) \approx z(x + \nu)$, for any small noise ν. For this reason, to evaluate the robustness of the latent representation, we consider the reconstruction robustness by computing the reconstruction accuracy given noisy input data.

From the results in table 4.6 wherein are listed the L_2 reconstruction losses obtained by the two AE models with input data obtained adding a small noise $\nu \sim \mathcal{N}(0, 0.4 \cdot I)$, it is possible to appreciate the role of the prior in that a too small representation has no room for error and a large representation $\sigma > 1$, as expected, is, in general, not very robust. Moreover, confirming what was observed until this point, the unitary variance WAE is learning an equal or better representation than the ones learnt by the VAE.

(a) unused feature

(b) orientation

(c) haircut



(d) gender

(e) hair color

Figure 4.14 CelebA transversal generated by ELBO ($\beta = 1$). The sample quality is quite good, but just four generative factors are clearly discovered.

(a) haircut

(b) orientation

(c) gender

(d) hair color

(e) smile

Figure 4.15 CelebA transversal generated by Wass ($\sigma = 1$). The sample quality is not particularly good, but it is possible to recognise at least five generative factors.

In this chapter we related the most common objectives for Variational AutoEncoder: ELBO and Wass, under a common principle, specifically the Constrained InfoMax (CIM). The unitary information-theoretic description of the two objectives shows us that it is necessary to find a compromise between encoding and decoding information and that information has to be controlled by the prior entropy and not by the encoding information as achieved by the ELBO.

We arrived at this conclusion by associating the optimal representation with the Minimum Description Length solution: learning a representation storing *only* that information necessary to generate the visible data. The theoretical argument was followed by computational experiments. In the first part, we observed the effect of the parameter β in terms of the ELBO trained VAEs. In the second part we showed that, for the optimal β, ELBO and Wass objectives coincide, and that an accurate choice of the variance of the prior leads to more disentangled representations.

5

WAE FOR MIXTURE MODELS

In the previous chapter we observed that the two main objectives, ELBO and Wass, both optimise the Constrained InfoMax objective. Thus we observed that the best VAE is the one learning the most informative representation of the data, fitting the prior $p(z)$. In the previous chapter we also observed that, although ELBO and Wass are informative equivalent, the performances of the model trained with the two objectives are not coincident. Aiming to explain the reason of such unexpected behaviour, in the following we restrict our attention to the *linear decoder VAE*.

The reason to consider that setting is two-fold: (i) it is possible to connect the VAE representations with the ones learnt by a linear Gaussian Process Latent Variable Model, and (ii) by what was illustrated in Lucas et al. [55], we have a formal proof that the learnt representations by the ELBO trained VAE are the PCA ones, independently by the encoder.

In this chapter we will observe that the *Wass trained representations are encoder-dependent* and *the learnt representations are equivalent to the ones learnt by a kernel PCA* having features described by the encoder itself. To show this result, we have first show the (geometric) connection between kernel PCA and unsupervised Mixture PCA, and highlight that the posterior inference is closer to a Gaussian Mixture Model (GMM) than to a multivariate Gaussian.

Summarising, the main contribution of this chapter can be listed by these points:

- Mixture PCA is a special Kernel PCA
- the posterior inference distribution learnt by the Wass trained VAE with Gaussian Prior is closer to a Mixture of Gaussian than to the Prior
- provide a geometric (non-informative) reason why bound the latent entropy is necessary to have good representations
- show the Wass representations are equivalent to the Kernel PCA ones, with kernel defined by the encoding map.

5.1 RELATED WORK: FROM PCA TO VAE

The main goal of this chapter is to connect the learnt representation by the VAE with a linear Gaussian Process Latent Variable Model (GPLVM). In chapter 3 we introduced already one of these: the Probabilistic Principal Component Analysis (PCA), here we introduce other two classic GPLVMs that will be utilised to describe the Wass trained VAE representations: - Kernel PCA and Mixture PCA.

5.1.1 *Mixture PCA*

The PCA is based on the assumption that the visible data are sampled from a uni-modal distribution (i.e. there is only a single class defining all the data-points), and then the data-points are identically independently distributed (iid). As it is possible to observe in real contexts, the iid assumption is quite restrictive. For instance, a two classes dataset or a Markov Chain time series are not iid. Indeed, in the Markov Chain setting, each data-point is dependent on the past and, within a multi-class dataset, the data-points are sampled from different distributions, and in the two class data-set it is reasonable to assume the data-points belonging to two different classes are sampled from different distributions.

In the case the data are not iid, and we know the distribution from which each data-point is sampled (e.g. the data-class) the Mixture PCA idea is to apply a "classwise" PCA, i.e. to apply a PCA for each distribution.

GEOMETRIC INTERPRETATION A set of iid distributed data can be thought as a sample of points lying into a plane, and the PCA idea is to project such a plane into a smaller one. As most of the time the data lie in more than one plan (each plan is the data-class) the Mixture PCA idea is to project each plan into a different sub-plane. Thus if the PCA representations is a *single* sub-plane, the Mixture PCA representation is a collection of planes.

A PROBABILISTIC DESCRIPTION From a probabilistic perspective, assuming that the visible data lies in K different tangent planes $\mathcal{Z}_k$, and each data-point x_i belongs to just one plane $\mathcal{Z}_k$, the generative distribution is given by

$$p(x_i|z_i, q_i = k, \theta) = \mathcal{N}(\mu_k + W_k z_i, \sigma^2 I) \tag{5.1}$$

where the latent prior is $p(z_k) = \mathcal{N}(0, I_k)$ for any k, where I_k is the $\dim(\mathcal{Z}_k) \times \dim(\mathcal{Z}_k)$ identity matrix associated to the space $\mathcal{Z}_k$, and the probability to sample a point in the space k is defined by the categorical distribution: $p(k) = Cat(\pi)$.

Combining all these elements together, we obtain the (probabilistic) *Mixture PCA* defined as

$$\sum_{k}^{K} p(x|z, k)p(z|k)p(k) \tag{5.2}$$

or, in vectorial form:

$$p(x) = p(x|\vec{z})p(\vec{z}|\vec{k})p(\vec{k}), \tag{5.3}$$

where $\vec{z} = [z_1, \ldots, z_K]^T \sim \mathcal{N}(0, I)$, $\vec{k} = [0, \ldots, 1_k, \ldots, 0] \sim Cat(\pi)$ and $p(x|\vec{z}) = \mathcal{N}(\mu + W\vec{z}, \sigma^2 I_K)$with $W = [W_1, \ldots, W_K]^T$.

5.1.2 *Kernel PCA*

Often the distributions (classes) from which each data-point is sampled are not known in advance, and it is more reasonable to know some general property of the data. One of these possible information is that the visible data-points are not iid sampled in the visible space where we see them, i.e. in *the visible space* $\mathcal{X}$, but they are iid sampled in a different space: the Feature (*latent*) space $\mathcal{F}$. Assuming to know the map $\phi : \mathcal{X} \to \mathcal{F}$, a possible solution is to apply the PCA on the new space, or equivalently find the eigenvalues and eigenvectors of the *Kernel* $K = FF^T$ with $F = [\phi(x_1), \dots, \phi(x_n)]$. For this reason this approach that has firstly proposed in [60], it is called *Kernel PCA*.

GEOMETRIC INTERPRETATION It is possible that we do not know the local information about the many plans composing a manifold, but we can know the general shape of the manifold. In that setting we know that a (smooth enough) differential manifold can be parametrised as a plane in a different space (e.g. a parabola $y = x^2$, is a manifold (curve) in the canonical 2D space $R^2 = \{(x, y) : x, y \in \mathbb{R}\}$) but it is a straight line in the quadratic feature space $\mathcal{F} = \{(x^2, x) : x \in \mathbb{R}\}$. Thus, the geometric idea behind the Kernel PCA is to map a manifold into a plane, and then apply the linear projection on the feature space. Let us observe, that the projection is linear only on the feature space but not with respect to the visible space.

GENERATIVE DESCRIPTION As the Kernel PCA is simply a PCA on a different space it has not a proper probabilistic interpretation. Indeed, given z the latent representation of the visible data-point x, the "natural" generative distribution $p(x_l|z) = \mathcal{N}(x_l|Wz, \sigma^2 I)$ is defined with respect the latent $x_l = \phi(x) \in \mathcal{F}$. Thus, the description with respect to the visible latent variable can be done only when the latent space has the same dimensionality of the visible one. Indeed, in that extreme case (never used in practice), it is possible to apply the change of variable formula, and thus we have that the distribution with respect the visible space is defined as

$$p(x|z) = p(x_l|z)|J_\phi(x)|, \tag{5.4}$$

where $J_\phi(x) = (\partial_i \phi(x_i))_{i,j}$ is the Jacobian associated to the map ϕ. By this Jacobian value, the generative map is not more Gaussian distributed, it is Gaussian only in the special setting the kernel is linear, and so the Kernel PCA coincides with the PCA.

For the sake of completeness we conclude this section providing the sketch derivation of the equivalence between pPCA representations and the linear ELBO trained VAE, that was firstly illustrated in Lucas et al, [55]

5.1.3 *Linear VAE recovers pPCA*

A linear Variational AutoEncoder is a model with decoder $p(x|z)$ and encoder $q(z|x)$ having the form

$$\begin{aligned} p(x|z) &= \mathcal{N}(Wz, \sigma^2 I) \\ q(z|x) &= \mathcal{N}(Vx, D), \end{aligned} \tag{5.5}$$

where D is a diagonal covariance matrix, used globally for all the data-points.

To show equivalence with the pPCA, it is first necessary to show that the ELBO objective has a unique solution, and it is the same of the Maximum Likelihood optimising the pPCA. In this thesis, we limit ourself to showing that the ELBO optimum is the same of the log-likelihood optimum of pPCA (lemma 1 in [55]), and so, for the second part of the proof we refer to [55].

As we seen in the third chapter, the pPCA is the model parameterising the inference and generative distributions as follows,

$$\begin{aligned} p(x) &= \mathcal{N}(\mu, WW^T + \sigma^2 I), \\ p(z|x) &= \mathcal{N}(M^{-1}W(x-\mu), \sigma^2 M^{-1}), \end{aligned} \tag{5.6}$$

with $M = WW^T + \sigma^2 I$. For this model the Maximum Likelihood Estimation (MLE) parameters are:

$$\begin{aligned} \sigma^2_{MLE} &= \frac{1}{n-K} \sum_{j=K+1}^{m} \lambda_j \\ W_{MLE} &= U_k(\Lambda_k - \sigma^2_{MLE} I)^{1/2} R, \end{aligned} \tag{5.7}$$

with U_k, the matrix of the first K principal components, $\Lambda_K = diag(\lambda_1, \ldots, \lambda_K)$ the diagonal matrix of the eigenvalues associated to the principal component, and R an arbitrary rotation matrix.

Assuming the matrix $R = I$, the matrix M is the eigenvalues diagonal matrix Λ. Thus, setting $V = M^{-1}W^T_{MLE}$ and $D = \sigma^2_{MLE}M^{-1}$, the inference distribution is exact the posterior $p(z|x)$, and the ELBO is equivalent to the likelihood, indeed

$$\mathbb{E}[\log p_\theta(x)] = ELBO + D_{KL}(q(z|x)||p(z|x)). \tag{5.8}$$

In this way the optimal ELBO recovers the MLE parameters.

For a complete proof of the connection between the linear VAE and the pPCA we refer to the original paper [55]. To convince ourself of the truth of the statement, let us remember that the PCA solution can be recovered via the EM algorithm, an iterative algorithm which is almost equivalent to the ELBO.

It is relevant to notice that the analysis is done considering the standard ELBO, and not the general β-ELBO. That choice could be considered restrictive, but actually, it is not; since as observed in [55], within the Gaussian decoder setting, wherein the distortion term has the form $\|x - Wz\|/\sigma^2$, the β can be seen as the decoder variance σ^2. In this way, in the Gaussian decoder context, the β term is a measure of the decoder entropy: the larger is β, the more blurred are the sampled data.

NON-LINEAR VAE In agreement with the intuition, in the case the encoder network is not being more linear, the equivalence with the PCA is lost. Indeed, in this setting, the ELBO has many optima, but the proposition above still holds, since it is layer independent. Then, one of these optima is the PCA solution. Other possible optima are those where the posterior collapse is occurring.

Rather more strangely, as observed by empirical results [55], a non-linear VAE is not learning a *better* inference than the linear encoder VAE. To better understand the reason for that phenomenon, let us examine the geometric description of the KL divergence.

A GEOMETRIC DESCRIPTION In the PCA assumption we have established that the data $X = \{x_i\}_{i\in[1,N]}$ are living around an hyper-plane univocally defined by some tangent vectors $\{z_k\}_{k\in[1,K]}$, wherein the goal of PCA is to find the linear projection map W and the plane Z such that the scalar product $Wx_i \cdot Z$ is minimised for any x_i, or equivalently the L_2 norm $\|Wx_i - z\|_2^2$ is minimal for any $z \in Z$, indeed

$$\|Wx_i - Z\|_2^2 = \|Z\|_2^2 - \|Wx_i\|_2^2 - 2Wx_i \cdot (Z - Wx_i). \tag{5.9}$$

Let us notice that it is not necessary to put any condition on the projection map W, it in principle could be any function, but as we constrain that has to project into a plane, that is a sub-space of the visible one, that turns out to be a linear projection.

By the connection in (5.9) we can say that, in the deterministic scenario, the PCA is looking for the matrix W minimising the Bregman divergence,

$$B_F(x, y) = F(x) - F(y) - \nabla F(y) \cdot (x - y), \tag{5.10}$$

between the projected data-points WX and the latent plane Z with respect the natural norm in $\mathbb{R}^n$, specifically the L_2 norm.

As the Bregman metric can be defined for any function F, the natural way to define the pPCA via the Bregman divergence, is to consider as F the L_2 norm in the random variables (rv) space. In the rv space the L_2 norm of a rv X is defined as

$$\|X\|_2^2 = \mathbb{E}_{p(x)}[|X|^2], \tag{5.11}$$

which, for the Gaussian distribution, is equal (up to constant) to the latent representation entropy $H(Z)$. Indeed, $H(Z) = \mathbb{E}[-\log p(z)] \propto \mathbb{E}_z[(Z-\mu)^2/|\Sigma|]$. Thus, as we are considering only Gaussian distributions, we expect that the pPCA projection map is the one minimising the Bregman divergence with respect to the entropy,

$$\begin{aligned} B_H(q(z|x), p(z)) &= H(q(z|x)) - H(p(z)) - \sum_i (\log p(z) + 1) \cdot (q(z|x) - p(z)) \\ &= H(q(z|x)) - H(p(z)) - \sum_i \log p(z_i) q(z_i|x) - \log p(z_i) p(z_i)(q(z_i|x) - p(z_i)) \\ &= H(q(z|x)) - \sum_i \log p(z_i) q(z_i|x)(q(z_i|x) - p(z_i) \\ &= D_{KL}(q(z|x) \| p(z)), \end{aligned}$$

(5.12)

i.e. the Rate term.

The connection between the Rate term and the Bregman divergence reveals that the KL divergence operator, is minimised by a posterior inference distribution $q(z|x)$ that is linear in the Gaussian distribution space, i.e. $q(z|x) = \mathcal{N}(z|Wx, S)$

5.2 MIXTURE PCA AS AN APPROXIMATED KERNEL PCA

By the geometric description of the the two PCA generalizations, Mixture PCA finding the different planes where the data-points lie, and Kernel PCA finding the Manifold where all the data-points lie, it seems that the two approaches are strictly connected, and the former can be described as an approximation of the latter. Indeed, for any manifold M and any small ε there exits a piecewise linear approximation of M, M_L s.t. $\|x_M - x_L\|$ for any $x_M \in M$ and $x_L \in M_L$.

As the formal connection between the two principle is not so trivial, here we limit ourselves to show that any Mixture PCA can be described as a special Kernel PCA, and we try to give a geometric intuition why any Kernel PCA can be approximated by a Mixture PCA. We will leave, for future work, to show the formal connection.

5.2.1 *Mixture PCA: a special Kernel PCA*

As we have seen above the main idea of Kernel PCA is to apply the (standard PCA) on the Kernel Matrix $K = FF^T$, where F is the matrix having columns the feature vectors, $F = [\phi(x_i)^T]^T$, with ϕ a generic feature map.

Let us notice that a possible feature map could be the following one

$$\phi(x) = \sum_i^K \mathbb{1}_{\mathcal{S}_i} \cdot [0, \dots, x, \dots, 0], \tag{5.13}$$

with $\mathcal{S}$ a partition subset of the data-set X s.t. $\cup \mathcal{S}_i = X$ and $[0, \dots, x, \dots, 0]$ is a $K \times n$ vector zeros everywhere but on the i-th slot that is x. In words, the feature map in (5.13) is mapping each data-point in some subspace of a larger space. If $x \in \mathbb{R}$, the feature space is $\mathbb{R}^K$ and on the new space each data-point can leave in only one of the axes.

The kernel map associated to the feature map in (5.13) has the following form

$$S = \begin{bmatrix} \boxed{S_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boxed{S_K} \end{bmatrix} \tag{5.14}$$

where S_i is the covariance (linear kernel) $S_i = X_i X_i^T$, with $X_i = [x_i]_{i \in \mathcal{S}_i}$ the matrix with column the data-points in $\mathcal{S}_i$.

It is straightforward to see that for block diagonal matrix like this one the the associated generative matrix W of the Kenel PCA is diagonal as well, i.e.

$$W = \begin{bmatrix} \boxed{W_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boxed{W_K} \end{bmatrix}, \tag{5.15}$$

with W_i orthonormal $n \times n$ matrices, obtained applying PCA on each block S_i.

So, for each data-point x_i, the generative distribution of the associated feature $\phi(x_i) = [0, \dots, x_i, \dots, 0]^T$ is the following:

$$\begin{aligned} p(\phi(x_i)) &= \int \mathcal{N}(\phi(x_i)|Wz, \sigma^2 I p(z))dz \\ &= \sum_k^K \int \mathcal{N}(\phi(x_i)|W_k z_k, \sigma^2 I_k) p(z_k) dz_k, \end{aligned} \tag{5.16}$$

with $p(z_k) = \int p(z)dz_{-k}$, i.e. the marginal over the "unused" space ($z_{-k} = (z_i 1, \dots, z_{k-1}, z_{k+1}, \dots z_K)$), indeed by the shape of the generative matrix W, each row is all zeros but n entries associated with the k-th space, and so they are invariant to the z values outside the associated space.

In this way we have derived the kernel of a general Mixture PCA model. In the next section we provide an argument to show (intuitively) why any Kernel PCA can be approximated by a Mixture PCA (or by Kernel PCA with Kernel as above).

5.2.2 *Approximate Kernel PCA with Mixture PCA*

From the definition of Mixture PCA and Kernel PCA, we have seen that the two models are designed for different tasks. The former is more suitable when the data is subdivided into clusters with each cluster distributed around μ_i, instead the latter is more suitable in the case the data is distributed around a manifold, and the data is linearly separable with respect to the metric induced by the manifold itself.

We observed that the two approaches coincide only in two special cases: the data has a standard geometric structure, i.e. it is distributed on a plane, i.e. the optimal Kernel PCA and optimal Mixture PCA are actually the standard PCA, and in the case the manifold where the data live is actually a composition of planes.

The latter special case, actually is not so rare; indeed, any geometric structure can be described as a cluster collection (e.g. a smooth curve can be approximated via a collection of segments). In this way assuming to have a large number of factors a Mixture PCA can separate the main factors of the data.

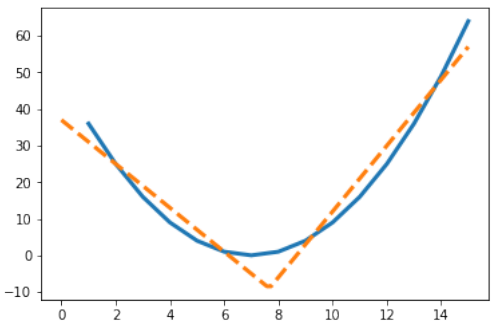


Figure 5.1 Piecewise linear approximation of a parabola

Let us notice, that by sub-dividing the geometric structure into small pieces,

the learnt representation by Mixture
PCA and (an optimal) Kernel PCA will not coincide, because the Kernel PCA representations will be an aggregate version of the Mixture PCA ones. If for instance one data-class is distributed in a square-like shape, the kernel PCA is learning a single representation (the square perimeter) and the Mixture PCA is learning four representations (one for each edge).

In the case more than one data-class is visible, a possible way to relieve the Kernel PCA representations from the Mixture PCA ones, is to apply a standard PCA on the Mixture Representation space. Indeed, in this way all the many representations in the Mixture space that have similar properties, because belonging on the same geometric structure, are clustered all together and separate from the ones with different properties.

5.3 WASSERSTEIN FOR MIXTURE PCA

In light of the relationship highlighted above between the Mixture PCA and Kernel PCA, the goal of this section is to show that the Wasserstein loss is pushing the associated VAE to learn a Mixture PCA-like representation in its latent variables.

To show the connection between the Wasserstein trained representations and the Mixture PCA ones, we -first show that the Wass loss can be read as a variational approximation of the Mixture Likelihood, then -following an informative argument we show that from the possible local minima of the Wass distance, the global minimum is the Mixture PCA one.

5.3.1 *ELBO for Mixture PCA*

FROM MIXTURE PCA TO NON-LINEAR VAE As sdescribed above, under the assumption to know the k factors, the Mixture PCA it is trained maximising the visible log-likelihood:

$$\begin{aligned}\sum_{i}^{N} \log p(x_i) &= \sum_{i}^{N} \log \Big(\sum_{k}^{K} \pi_k p(x_i|k) \Big) \\ &= \sum_{i}^{N} \log \Big(\sum_{k}^{K} \pi_k \int_{z_k} p(x_i|z_k) p(z_k) dz_k \Big),\end{aligned} \tag{5.17}$$

where π is the discrete mixing distribution, that for the sake of simplicity we will assume in this section uniformly distributed (i.e. $\pi_k = K^{-1}$) and $p(x_i|z_k)$ the generative distribution of the data-point x_i, given the latent variable z_k, multivariate normally distributed.

Let us notice that by Mixture PCA definition each factor z_k is associated with a single data-class, thus the vector $[p(x_i|z_1), \dots, p(x_i|z_K)]$ is a sparse vector wherein most of the mass is concentrated onto a single value $p(x_i|z_{k(i)}) = p(x_i|z_{k*})$. By the sparsity of the distribution

vector, and the LogSumExp property we can approximate the equation in (5.17) with the following one:

$$\begin{aligned}\sum_{i}^{N} \log p(x_i) &\approx \frac{1}{K} \sum_{i}^{N} \log p(x_i|k^*) \\ &= \frac{1}{K} \sum_{i}^{N} \log \Big(\int_{\mathcal{Z}_{k*}} p(x_i|z_k) p(z_{k^*}) dz_{k^*} \Big).\end{aligned} \tag{5.18}$$

In agreement on what has been proved in [55], and in light of the approximation above, we can maximise the likelihood via the following variational objective

$$\begin{aligned}\sum_{i}^{N} \log p(x_i) &\geqslant \sum_{i}^{N} \frac{1}{K} \text{ELBO}(q(z_{k(i)}|x_i), p(x_i|z_{k(i)})) \\ &= \frac{1}{K} \sum_{i}^{N} \log \mathbb{E}_{q(z_{k^*}|x_i)}[p(x_i|z_{k^*})] - D_{KL}(q(z_{k^*})||p(z)).\end{aligned} \tag{5.19}$$

Indeed, the objective in (5.19) is the composition of many ELBOs, objective that, in the linear setting (the same of the Mixture PCA) has the same minima of the PCA.

Unfortunately, the (5.19) expression is useless in practice because it assumes to employ K different VAEs, one for each class-data, indeed we do not have knowledge of the different classes, and we do not know to which class each data-point belongs to. A possible way to overcome that issue is to consider a cumulative approach: -select a batch B of K data-points, one for each class (let us assume for now we have this extra information), and rewrite the Variational objective in (5.19).

Observing that for any two independent distribution pairs (p_1, p_2), (q_1, q_2), such that $\vec{p} = p(x_1, x_2) = p_1(x_1)p_2(x_2)$, the following equation holds

$$D_{KL}(q_1||p_1) + D_{KL}(q_2||p_2) = D_{KL}(\vec{q}||\vec{p}), \tag{5.20}$$

for any batch B the Mixture ELBO in (5.19) can be rewritten as follows:

$$\begin{aligned}\sum_{k}^{K} \underbrace{\frac{1}{K} \log \mathbb{E}_{q(z_{k^*}|x_i)}[p(x_i|z_{k^*})]}_{D_k} - D_{KL}(q(z_{k^*}|x_k)||\mathcal{N}(0, I) &= \sum_{k} D_k - D_{KL}(\prod_{k} q(\vec{z}|x_k)||\mathcal{N}(0, I)) \\ &= D - D_{KL}(\vec{q}(\vec{z}|[x_k]_1^K)||\mathcal{N}(0, I)) \\ &= D - D_{KL}(\vec{q}(\vec{z}|B)||\mathcal{N}(0, I)),\end{aligned} \tag{5.21}$$

where the first equation follows by independence of each x_k.

By the relationship exploited in (5.21), we see that (i) in the multimodal setting, i.e. if the input is a batch of data-points, the ELBO is suitable to learn a representation close to the Mixture PCA one, and (ii) that for any x the associated inference distribution is distributed as

$$\vec{q}(\vec{z}|x) = \mathcal{N}(0, diag(\varepsilon_1, \dots, I, \dots \varepsilon_{K-1})) \tag{5.22}$$

with $0 < \varepsilon_i \ll 1$.

WHY IN WASS TRAINED MODEL IS PREFERABLE TO BOUND THE ENTROPY By the description done above, we observed from a different perspective why a deep encoder is not able to learn a latent distribution that differs from the linear encoder, and thus why the ELBO trained VAE in linear setting are learning VAE-representations.

Let us notice, moreover, that the analysis above does not say that the Wass is not suitable to learn Mixture-like representation. Indeed, on average $q(z)$ is still Gaussian distributed with $q(z) = \mathcal{N}(0, \mathtt{diag}(1 + \varepsilon(N-1)/N))$. Thus, we deduce that (i) *the optimal Wass solution satisfies the Mixture ELBO properties*, and (ii) *for a fixed number of non-linear units* $\hat{K}$, *the selected variance* σ *(controlling the entropy) is inversely associated with the number of factors describing the mixture* , i.e. $\sigma \geqslant 1$ implies one common factor for all the data (PCA setting) $q(z) = \prod \mathcal{N}(0, 1)$, otherwise if $\sigma < 1$ each latent unit is associated with a Mixture factor, $q(z) = 1/\sqrt{\hat{K}} \prod \mathcal{N}(0, 1)$.

In this way we have obtained (i) a pure probabilistic interpretation of why bounding the entropy is necessary, and in particular (ii) a lower bound for the latent variance, i.e. $\sigma > 1/\sqrt{\hat{K}}$.

INFOMAX IMPROVES THE CLUSTERING By the argument above we have seen that the Mixture PCA solution has to maximise the Wass objective. However, we have not shown any property that guarantees that the optimal Wass solution is close to the Mixture PCA one. In the following, we relate the two solutions showing that the Mixture PCA representations, as the Wass solutions, are the most informative about the data.

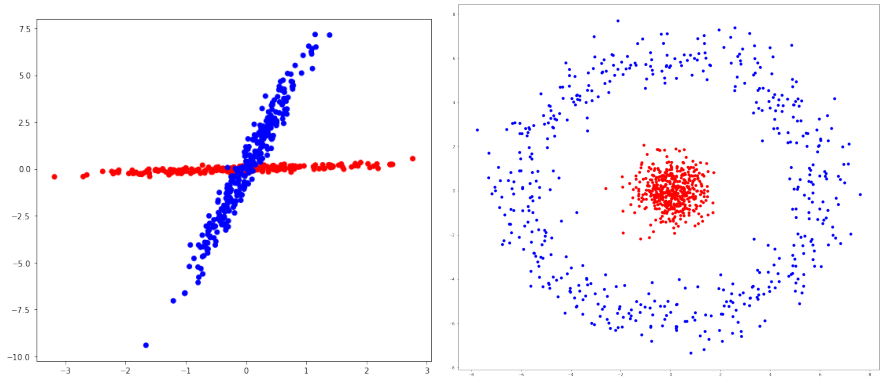
For the sake of simplicity, let us assume that, without loss of generality, that each Principal Component (PC) is associated with a different subspace, with generative model $p_k(x)$. Then, defining z_k the k-th feature of $\vec{z}$, $\vec{z}^{(k)} = z_k$, the Mixture PCA goal is to associate each data-point x_i to the correct cluster, $\vec{z} = [0, \dots, z_k, \dots, 0] = f(x)$.

As a clustering problem, the optimum mixture PCA representation is the one maximising the information with the visible data, [7].

To determine this point, it is useful to write the mutual information in terms of the sum of two entropies, $I(Z;X) = H(Z) - H(Z|X)$ and consider them separately. The entropy of Z, $H(Z)$, is a quantity measuring the information contained within the representations, which is proportional to the number of clusters K, and the data is maximally entropic if the representation is sparse: each data-point falls in just one cluster. Instead, the conditional entropy $H(Z|X)$ is a measure of the information lost in the channel. It is minimal if each data-point is associated (deterministically) to the respective cluster, leading to hard assignments of cluster labels to equiprobable data regions.

Combining the two terms, we derive that the Mixture PCA representation is the one maximising the MI $I(Z;X)$, where the entropy is bounded in order to avoid that each data-point is associated to a single feature.

Remembering that, by definition, the Wass-trained network is the one optimising the encoding information, we see that the Wass solution is strictly associated with the one optimising the Maximum Likelihood in (5.17).



(a) cross data-set (b) 2 sphere data-set

Figure 5.2 Sample from the two synthetic data-sets used for these experiments. Both are not suited for PCA, as in the cross generated data, the Principal Factors are not orthogonal, and in the two sphere data, the two factors are not linearly related.

5.4 EXPERIMENTS

The goal of this section is to show that the representations learnt by the Wass trained VAE (WAE) are comparable to the Kernel PCA ones (a generalisation of the unsupervised Mixture PCA), in contraposition to the PCA-like representations learnt by the ELBO trained model. To achieve this, we first consider some simple toy examples, wherein it is possible to compare the two objectives with the associated PCA solution directly; and then to consider the classical benchmarks dataset: MNIST, Fashion-MNIST and DSprites.

With the latter dataset, we compare the MIG obtained by the two different trained VAEs, wherein we observe, confirming the theoretical argument, that the latent representations learnt by WAE are more disentangled than those associated to the classic VAE.

5.4.1 *Synthetic data-sets*

To better understand what the VAE models are learning and how they are connected with classical models, we first decided to illustrate the behaviour of the two objectives within some simpler contexts. Thus, the learnt representation can be plotted easily and compared directly with the known generative factors.

CROSS DATASET In the theoretical section we have shown that, if the ELBO-trained model is equivalent to the PCA, then the Wass-trained VAE is learning representations that are closer to the Mixture PCA ones. Thus, the decoder of the WAE is a matrix where each row is a spanning a different space, and then we do not expect that is orthogonal.

To verify that assertion, we start considering the cross data-set illustrated in fig. 5.2a, which is formally sampled by

$$x \sim cv_1 + (1-c)v_2 + n, \tag{5.23}$$

where $c \sim \mathtt{Unif}(\{0,1\})$ defines the probability of falling into of the two classes, $v_i \sim \mathtt{Unif}(d_i)$ the position in the direction d_i, where the data are distributed and $n \sim \mathcal{N}(0, I)$ the extra noise to add in the data.

In this setting, the generative factors are the two directions d_i and the class c.

GENERATED SPACES According to the theoretical description given above, in a Mixture PCA, we expect to find two separate PCs, one for each class, with generative matrices W_1 and W_2 spanning the directions d_1 and d_2 respectively. Conversely, in a standard PCA setting, the directions of the generative matrix W has to be orthogonal and so also the mapped direction. To see if these deductions are true and to analyse the properties of the learnt matrices, we will observe the generated spaces for each column of the generative matrix W.

In all the experiments that we will examine whitin a non-linear encoder VAE model of form: $x \xrightarrow{\phi} z \xrightarrow{W} x$. The function ϕ is a potentially non-linear function described by a multi-layer encoder and W is the generative map, trained to maximise one of the two objectives,

$$\mathrm{ELBO} = \sum_i -\|W\phi(x_i) - x_i\|_2^2 - \beta D_{KL}(q(z|x_i)\|p(z)) \tag{5.24}$$

and

$$Wass = -\sum_i \|W\phi(x_i) - x_i\|_2^2 - \lambda D_{KL}(q(\bar{z})\|p(z)), \tag{5.25}$$

where with abuse of notation, since we are using the same network for both the models, we decide to remove the vector notation from the Wass objective.

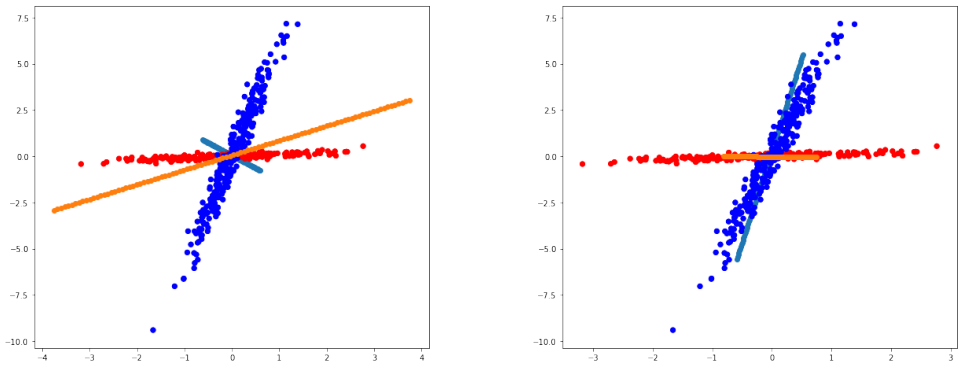
Both the KL divergences are computed with the approximation illustrated in the previous chapter.

From empirical observations we observed that the hyper-parameters λ and β and the variance factor σ^2 do not have a relevant role, indeed changing them does not change radically the results. In particular, for all those synthetic experiments we consider the parameters that better optimise the respective objective: $\beta = 0.2$, $\lambda = 1$, $\sigma = 1$. All the networks, in this chapter, i.e. both the Fully Connected Network and the Convolutional ones, are trained with a standard Stochastic Gradient Descent with learning rate $lr = 1e-4$.

In the specific case of the Cross dataset, we consider an encoder of shape $x \to h_1 \in \mathbb{R}^{256} + \mathrm{ReLU} \to z \in \mathbb{R}^2$, with the representation z chosen to be two-dimensional since we know that the generative factors are two, and the data are obtained via linear combination of the generative factors.

As we see from figure 5.3, wherein we compared the generated data associated to the two features (i.e. the transversal) for both ELBO and Wass-trained VAE, the WAE features generate the two vectors that lie in correspondence of the principal components of each direction. Instead, the ELBO generated factors are almost perpendicular, and lie in the proximity of the PCs of the whole data; in agreement with the theoretical deduction asserting that ELBO-trained models recover the PCA solution and the Wass the Mixture PCA one.

From a geometric perspective, the cross dataset is the composition of two manifolds, wherein the derivative is constant in each cylinder and zero outside. The Wass representations define



(a) ELBO (b) Wass

Figure 5.3 Generated directions (yellow and blue lines) of the same VAE trained with ELBO (left) and Wass (right), over the visible data (red and blue dots). The Wass generated data overlap the visible data, suggesting that the representation has discovered the principal factors of each class. In the ELBO instead the generated data are orthogonal following the Principal components of the whole dataset.

the directions where the tangent vectors to the manifold are constant (i.e. the two vectors describing a manifold). Instead, the first principal component PCA (yellow line) is the one associated with the direction in which the derivative of the plane embedding the two cylinders is maximal. That property, as we will see better in the next experiment, is equivalent to the property necessary to discover the hidden manifold wherein the data-points lie.

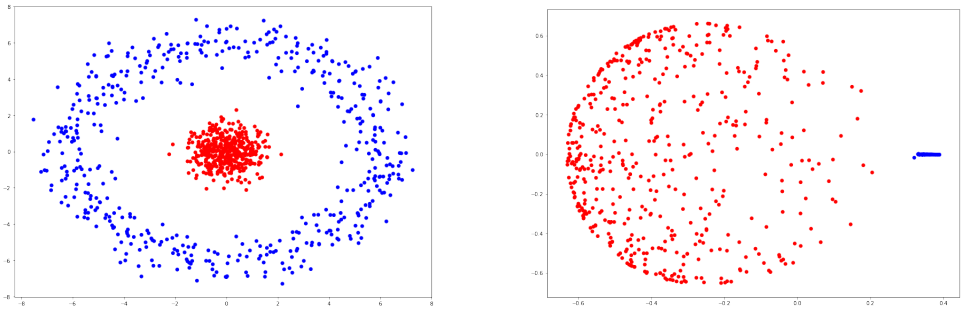
TWO SPHERE DATASET Thanks to the cross data-set we have seen that the Wass is learning representations that are close to the ones learnt by a Mixture PCA. However, in the theoretical section we observed that within more complex settings, it is better to describe the Wass representations in a manner akin to the ones associated with a Kernel PCA. To illustrate the connection arising between WAE and the Kernel PCA representation, we consider the two concentric sphere dataset, figure 5.2b: which serves as a standard benchmark of non-linear separable data and is useful to explain the differences arising between Kernel PCA and PCA. Indeed, in this setting, the Kernel PCA can discover a linear separable representation, while the standard PCA is learning solely the identity map, as we can see illustrated in figure 5.4.

The two sphere data-set is a set of points in $\mathbb{R}^2$, generated by the following function:

$$x \sim c \cdot r(\sin(\theta), \cos(\theta)) + \varepsilon, \tag{5.26}$$

where $c \sim \mathrm{Unif}(0, 1)$ is the probability of being in one of the two sphere, r is the radius of the largest sphere (the smallest one has radius 0); $\theta \sim \mathrm{Unif}([0, 2\pi))$ is the angle defining the point position; and $\varepsilon \sim \mathcal{N}(0, \sigma I)$ is the additive noise.

From the description in (5.26) we observe that the generative factors for the data are: the class factor c (or in a more general setting the radius) and the angle θ. Since the generative function is non-linear, $x \neq W(c, \theta)$, we cannot expect that the PCA is learning the two factors. To apply the PCA, it is first necessary to map the datapoints X living in the Cartesian space



(a) PCA decomposition

(b) KPCA decomposition (RBF kernel)

Figure 5.4 2d projection of the 2 sphere dataset for PCA and Kernel PCA. The first one is not learning the factors as it is only projecting the identity; the second one is learning the two classes, wherein the red data-points are sparse in the Kernel PCA projection, since they do not have any structure, or at least their relationship is different from the one connecting the blue-points.

$\mathbb{R}^2$ into the torus $\mathbb{T}^2(\mathbb{R} \times 2\pi)$, $(t_1, t_2) \in \mathbb{T}^2$ if $(r, \theta) = (r, \theta + 2k\pi)$ for any $k \in \mathbb{Z}$, wherein the generative factors are disentangled. Formally, it is necessary to apply the PCA in the feature space which is obtained via the map $\phi : \mathbb{R}^2 \to \mathbb{T}^2$ defined as

$$\begin{aligned} r &= \phi_1(x, y) = \sqrt{x^2 + y^2} \\ \theta &= \phi_2(x, y) = tg^{-1}\left(\frac{x}{y}\right). \end{aligned} \tag{5.27}$$

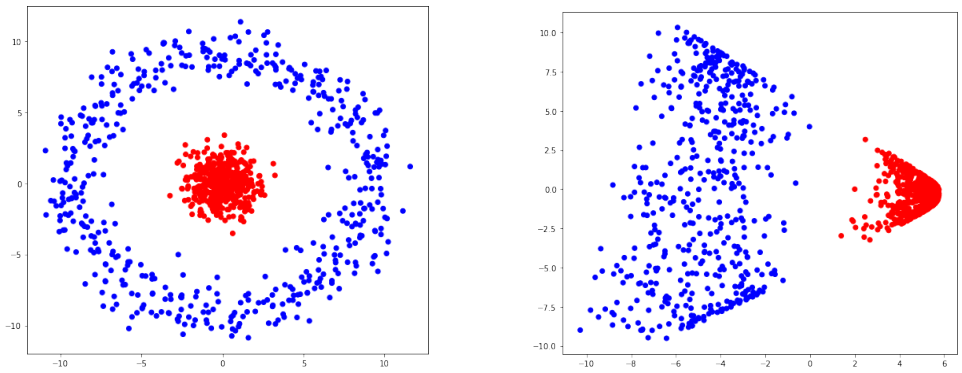
That approach is exactly what we defined to be a Kernel PCA method, where the kernel is defined by the feature map ϕ. Obviously, as we generally do not know what the optimal feature map ϕ is it has to be defined a priori, or better learnt via the non-linear encoder VAE.

In this setting in which the goal is to compare the two objectives, ELBO and Wass, the decoder of the linear encoder VAE has the form:

$$x \in \mathbb{R}^2 \to h_1 \in \mathbb{R}^{256} + \text{ReLU} \to z \in \mathbb{R}^{128}. \tag{5.28}$$

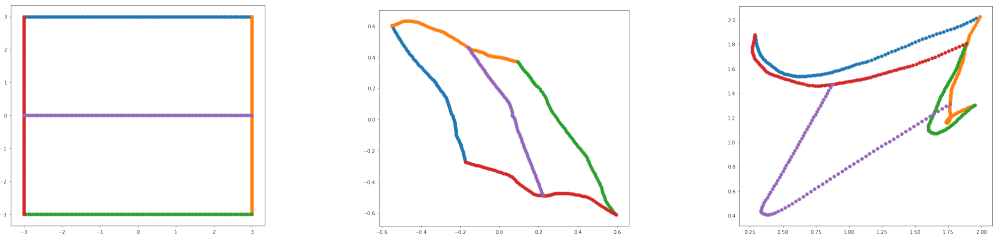
Let us note that, in this setting, although the generative factors are still two-fold - specifically the radius and the angle - we have a latent representation that has more than two hidden units. This choice is necessary to have redundancy within the latent variable and thus the opportunity to generate data that are not simply a linear combination of the two factors. Indeed, as we observed in the Piecewise Continuous function setting, any non-linear function such as $\cos(\theta)$, can be described as a linear combination of linear functions.

To be consistent with the results summarised in figure 5.4, and have a visible comparison, we compare the 2d projection of the two latent representations. These projections can be obtained interchangeable applying the PCA to the learnt features or simply considering the two most used features (the two with the highest magnitude), indeed as described above all the learnt features are essentially copies (up to dilatation and rotation) of the two main features.



(a) ELBO (b) Wass

Figure 5.5 2d projection of the learnt representation by the VAE as trained with ELBO (left) and Wass (right). The ELBO is learning the identity map, like the PCA; and the Wass representations are separated, but the structure is not the same as for the Kernel PCA above, suggesting that the kernel learnt in the encoder is not the RBF one.



(a) Data (b) ELBO representation (c) Wass representation

Figure 5.6 2d projection of the learnt representation by the VAE as trained with ELBO (middle) and Wass (right), of the data presented on the left. The ELBO encoder is performing a rotation of the data, as performed by the PCA or any linear encoder, whereas the Wass is clearly performing a non-linear mapping, wherein the middle points (purple line) are well separated by the counter data.

As we see in figure 5.5, in agreement with the theoretical argument, the ELBO model is learning a PCA-like solution; instead, the Wass one is learning a representation that discovers the generative factors.

To better understand the differences arising between the encoders learnt by the two objectives, it could be useful to take a look at figure 5.6, where the latent space of the two most used features are plotted, giving as input a square dataset. From this figure, it is possible to see that the VAE encoder is simply performing a rotation (i.e. a linear operation); instead the WAE encoder is performing a non-linear map wherein the center data (purple line) is well separated from the outer square lines.

THE LEARNT REPRESENTATION BY WAE We observed that the WAE encoder is a non-linear map, but the map is not equivalent to the RBF one.

In the theoretical section and via the previous example we suggested that the learnt representations are associated to those directions wherein the hidden manifold, the space where the data lie, is changing the most (the derivative is the highest and possibly constant).

This deduction is also confirmed in the 2-sphere experiment. Indeed, according to the representation plotted in figure 5.5, we can determine that the (hidden) manifold describing the data is a cone, whereupon the red dots are concentrate around the vertex and the blue dots around the basis (the visible data can be interpreted as a projection from the above). In this setting, the orthogonal tangent vectors defining the manifold are the tangent vector to the basis and the one tangent to the lateral surface.

Moreover, this experiment is useful in that it highlights that the hidden manifold does not live any time in a subspace of the visible one; indeed, in general, it lives in a larger space, and the visible data are just corrupted projection of the hidden data.

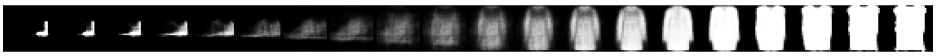
5.4.2 *Classical Benchmarks*

By virtue of the experiments with the synthetic datasets, we have seen that the Wass objective helps to discover the latent structure of the hidden manifold.

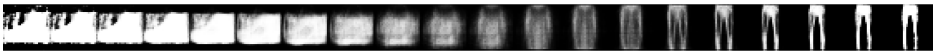
The differences arising within this special context are evident, but what are the differences between Wass and the ELBO solutions in the *real* setting? To answer this question, we compare the representations and the samples obtained by the VAE models as trained with three standard datasets: namely MNIST, FashionMNIST and DSprites. The FashionMNIST [87] is a collection of 70k 28×28 grey-scale Zalando's article images, subdivided in ten classes *(specifically T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag and Ankle boot)*.

We choose these datasets for two main reasons: they are relatively complex (i.e. the data-points have to be generated by a linear operator and thus it is not possible to consider more challenging datasets); and because they are associated with a ground-truth generative factor (this is the label set for MNIST datasets). The knowledge of the ground-truth generative factors is not just important to evaluate the disentanglement of the learnt representation, but it also allows us to guess the manifold structure wherein the data is distributed. In particular, following the standard intuition, we firstly assume that, within the MNIST data-sets, the data are distributed into a star-shape manifold which has as many vertices as generative factors; something akin to the cross considered above, wherein each edge of the cross is a generative factor.

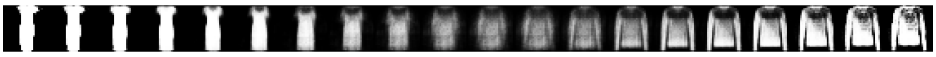
To evaluate the differences arising between the two objectives, we divide the experiments into two parts. First, we provide a qualitative analysis of the learnt representation, describing the generated space for each row of the generative map. In the second part, we provide a computational analysis of the learnt representation (in the classic way as performed in chapter 4): computing the Negative Log-Likelihood of the generated samples in order to evaluate the informativeness of the representation, and the Hoyer and the MMD distances to estimate the degree of disentanglement. This form of description is useful to allow us to better understand



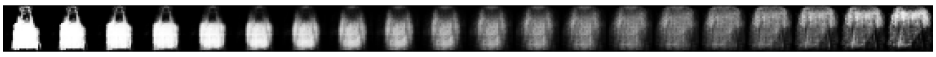
(a) 1st feature (sandals → sneaker → coat → dress)



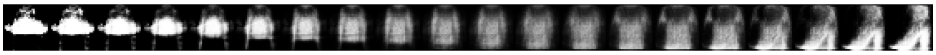
(b) 2nd feature (ankle boot → pullover → trouser)



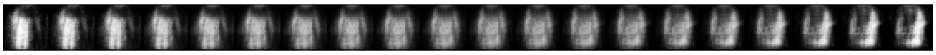
(c) 3rd feature (dress → pullover → shirt)



(d) 4th feature (bag → pullover → t-shirt)



(e) 5th feature (bag (?) → pullover → ankle boot)



(f) 6th feature (coat/pullover)

Figure 5.7 Generated Fashion-MNIST transversal spanned by the hidden features of the WAE. All ten different classes appear within the transversal data and if the class appears more than one time, (e.g. sneaker or bag), they appear with two different prototypes (i.e. the representation is disentangled). Here we show just six transversals, since the others are unused such as the sixth one plotted here.

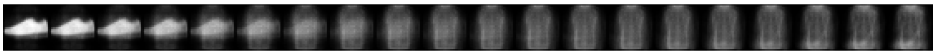
the process of disentanglement, and to understand whether the disentanglement metrics are discovering those differences.

THE GENERATED SPACE

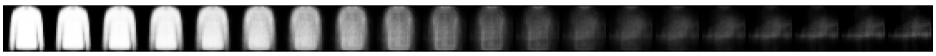
SPANNED SPACE All the experiments are performed by a VAE with 4-convolutional layer encoder, linear decoder and 10 latent features within the hidden layer. The hyper-parameters β and λ chosen for the experiments are, respectively, $\beta = 0.1$ and $\lambda = 10$, wherein the parameter β in ELBO is chosen so as to have balanced encoding and decoding performances. For both models, the prior $p(z)$ is chosen normally distributed with unitary variance.

Figures 5.7, 5.8, 5.9, 5.10 illustrate the traversals of the respective models: the data-points x generated by each feature $z^{(i)}$, $x = Wz^{(i)} + b$, for the MNISTs trained models.

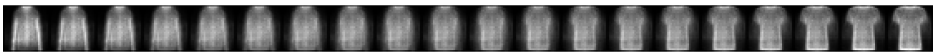
From these figures, we see that the solutions associated with the two objectives are different and they are not equivalent. The (ELBO) VAE is using all the possible features since each one is generating different data-points, but all these data-points are blurred and some features seem redundant (i.e. they are generating the same space). Differently, the WAE model is using



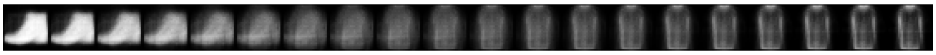
(a) 1st feature (sneaker → coat)



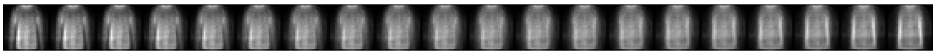
(b) 2nd feature (shirt → coat → sandals)



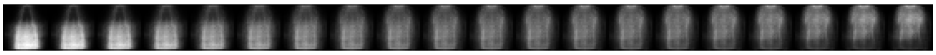
(c) 3rd feature (pullover → t-shirt)



(d) 4th feature (ankle boot → coat)



(e) 5th feature (t-shirt → pullover)



(f) 6th feature (bag → pullover)

Figure 5.8 Generated Fashion-MNIST transversal spanned by the hidden features of the VAE. Some classes are not represented (such as trousers and dress), whereas other classes appear in more than one traversal (e.g. t-shirt), suggesting that the representation is not fully disentangled since more than one feature is used for some data-points arising. The other transversals are not represented as, since the respective features are unused, they are generating the same space for the 5th feature.

(a) 1st feature ($9 \rightarrow 3$)

(b) 2nd feature ($9 \rightarrow 8$)

(c) 3rd feature ($3 \rightarrow 5$)

(d) 4th feature ($9 \rightarrow 8 \rightarrow 2$)

(e) 5th feature ($1 \rightarrow 9 \rightarrow 0$)

(f) 6th feature ($5 \rightarrow 0$)

(g) 7th feature (blurriness)

(h) 8th feature ($8 \rightarrow 9$)

(i) 9th feature ($0 \rightarrow 9$)

(j) 10th feature ($6 \rightarrow 3$)

Figure 5.9 Generated MNIST transversal spanned by the hidden features of the VAE. The generated data is not sharp, and not all the classes are represented (e.g. the 4 and the 7), instead some classes appear in more than one traversal. That means that the representation is neither sparse, nor disentangled.

(a) 1st feature ($7 \rightarrow 3 \rightarrow 5 \rightarrow 0$)

(b) 2nd feature ($6 \rightarrow 5 \rightarrow 8$)

(c) 3rd feature ($9 \rightarrow 5$)

(d) 4th feature (rotation)

(e) 5th feature ($5 \rightarrow 9$)

(f) 6th feature (thickness)

(g) 7th feature ($4 \rightarrow 5 \rightarrow 3 \rightarrow 2$)

(h) 8th feature (unused)

(i) 9th feature ($1 \rightarrow 9 \rightarrow 5 \rightarrow 0$)

(j) 10th feature ($5 \rightarrow 3$)

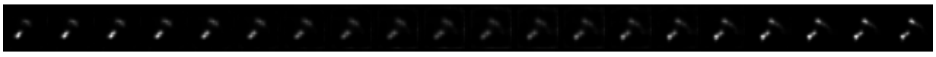
Figure 5.10 Generated MNIST data spanned by the hidden features of the WAE. All the used features generate sharp samples associated to the different classes. The redundant features, like the fourth and the sixth, are describing hidden properties common for all the data-sets.

(a) 1st feature (unused)

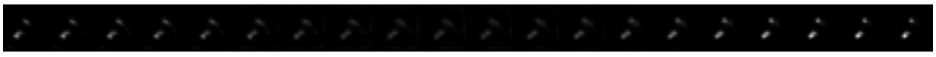
(b) 2nd feature (unused)

(c) 3rd feature (size)

(d) 4th feature (vertical position)



(e) 5th feature



(f) 6th feature

(g) 7th feature

(h) 8th feature

(i) 9th feature

(j) 10th feature

(k) 11th feature

(l) 12th feature

Figure 5.11 Generated DSprites data spanned by the hidden features of the VAE. Just two features are unused, but for some of the used features it is not clear if we understand to which generative factor(s) they are related.

(a) 1st feature (shape)

(b) 2nd feature (unused)

(c) 3rd feature (rotation)

(d) 4th feature

(e) 5th feature (square)

(f) 6th feature (y-position)

(g) 7th feature (unused)

(h) 8th feature (size)

(i) 9th feature (unused)

(j) 10th feature (close the)

(k) 11th feature

(l) 12th feature (unused)

Figure 5.12 Generated DSprites data spanned by the hidden features of the WAE. Four features are unused, and in most of the other eight it is possible to recognise a generative factor.

a smaller number of features, since most of them are simply generating the null class (digit five and coat respectively). The null class which is visible in all the other transversal associated with the null feature $z = 0$, i.e. the digit 5 is the o$W0 + b = b$. Moreover, the samples, in general, appear to be of better quality. Let us remember that since we are considering just one feature and a linear generator, we cannot expect to have particularly good samples.

Such behaviour is in agreement with what observed in the cross dataset, wherein the WAE is able to discover the hidden manifold, a task where it outperforms the VAE. Indeed, the data-points appear to be distributed in a cross with a centre class data (the digit 5 and coat), and any other classes appear on one side of the cross.

This is different from the behaviour observed in the DSprites setting, wherein both the ELBO and Wass trained VAEs are learning different, but not interpretable representations. We think that the issue lies in the complexity of the dataset. Indeed, the DSprites set is more complex than the two MNIST sets, since the relationship between the generative factors is not linear. That conclusion is confirmed by the learnt representation for $z = 0$, which yields an all-black picture.

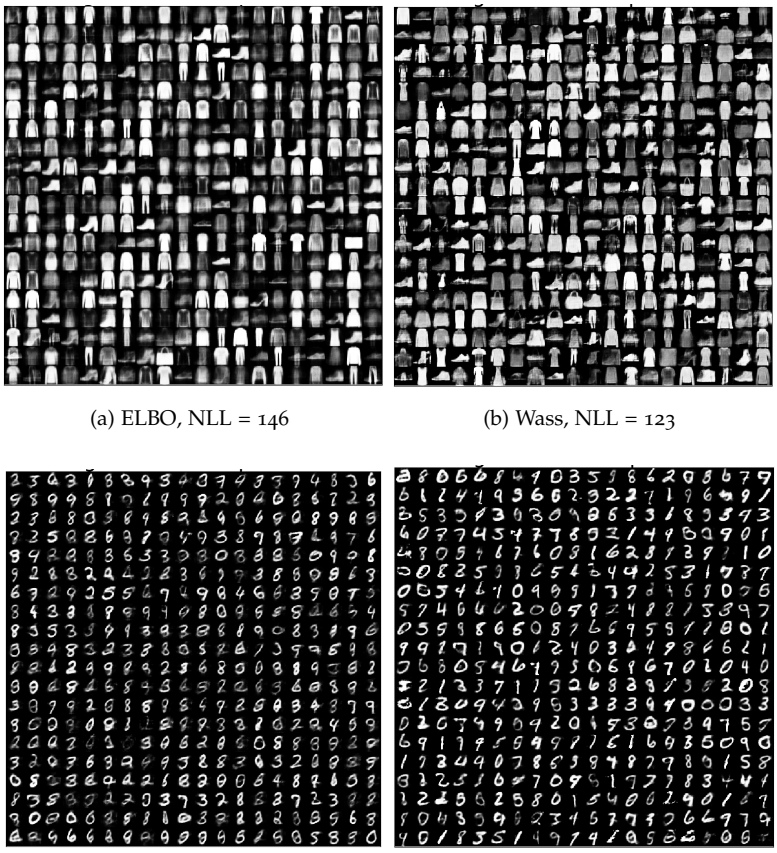
Finally, we note that the ELBO results are not associated with a no-informative representation. Indeed, as we observe in figure 5.13 the generated samples of the data for latent variable sampled by $p(z)$ are similar for all the models tested, and this is confirmed by the Negative Log-Likelihood (NLL) metric. However, the Wass-trained model generated slightly better samples.

REPRESENTATION ANALYSIS Through the experiments described above, we have observed that, although from an information-theoretic description that two objectives, namely ELBO and Wass, should be equivalent, the associated models are learning different kinds of representations. Those learnt by WAE are intuitively more disentangled. In the following experiments, we seek to establish if the intuitive idea of good representation is confirmed by the metrics introduced in Chapter 4.

The first metric that we consider is the Mutual Information Gap, a quantitative metric of the disentanglement, in that it measures the gap arising between the first two most informative features for the same generative factor.

For the MNISTs datasets, as it is not possible to compute the MIG, or any other disentangled metric, following what was established in the previous chapter, we evaluate the decomposition property of each variable. A representation is said to decompose if the encoding fits the prior and there is a correct overlapping in the latent space. To evaluate the encoding fitting, we consider the MMD distance between the encoding $q(z)$ and the prior $p(z)$; and to evaluate the proper overlapping as we consider the Hoyer metric introduced in chapter 4.

As we see from the table 5.1 the differences arising between representations associated with the two learning objectives are minimal, although the ELBO representations appear slightly more entangled than those for Wass.

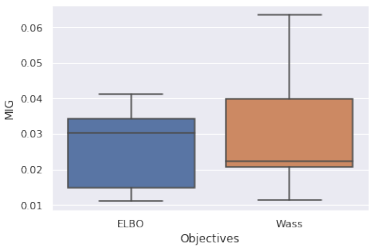


(a) ELBO, NLL = 146 (b) Wass, NLL = 123

(c) ELBO, NLL = 90 (d) Wass, NLL = 77

Figure 5.13 Generated data, Fashion-MNIST (above) and MNIST (below), obtained sampling from the latent variable prior, where $x = Wz + b$ with $z \sim p(z)$. Both networks are generating similar samples, highlighting the conclusion that the sample quality does not say everything about the degree of disentanglement.

Table 5.1 Disentanglement comparison: MIG for Dsprites (left), and Decomposition metrics (Right) (Hoyer, MMD), the more decomposed is, the closer to the origin (0,0). Both the objectives have similar metrics, in conflict with the qualitative analysis.



(a) MIG, Dsprites

Dataset	Objective	
	ELBO	Wass
MNIST	(0.69, 0.07)	(0.65, 0.06)
F-MNIST	(0.78, 0.09)	(0.62, 0.05)

(b) MMD and Hoyer for MNISTs

In particular, the differences are more visible within the most complex settings, DSprites and FashionMNIST, than in the standard MNIST where the two objectives seem to be learning equivalent representations. This result is not in agreement with the qualitative comparison where, also within the MNIST setting, the differences arising between the objectives are clear.

In this way, we observe that, as already noted in Chapter 4, that the decomposition metrics are merely describing some of the properties of the optimal representation, but do not serve, in general, as a precise metric for the disentanglement.

5.5 CONCLUSION

In this chapter, we compared from a geometric perspective the ELBO and Wass objectives in the special case in which the Variational AutoEncoder decoder is linear. We observed that the two objectives, although equivalent from an information theoretic perspective – i.e. both of them are maximising the Constrained InfoMax – they are learning different forms of representations. Indeed, the representations learnt by ELBO are equivalent to the ones learnt by PCA, independently by the depth of the Encoder network. In contrast, the Wass is learning a representation that is strongly related to the Mixture PCA one.

The differences in the learnt representations are evident in the experiments performed both with the synthetic data, where it is shown that the WAE representations are describing the hidden manifold; and with the MNIST data-sets, wherein each data-class is associated to just one latent feature.

The description provided illustrates that a disentangled representation does not have to generate orthogonal factors, since that condition holds only in the special case the data are iid.

Possible future work includes the extension of this analysis to the general VAE network. Indeed, as in the linear setting we have seen that both ELBO and Wass linear VAE are associated to two *linear* Gaussian Process Latent Variable Model, we expect that, in the non-linear encoder case, the associated VAEs can be related to general GPLVMs as well.

That connection arising between Gaussian Processes and variational methods will be strengthened by the analysis performed in the following chapter wherein we consider the Constrained InfoMax within the supervised setting.

6

INFORMATION BOTTLENECK REVISITED

In the previous chapters we focused on the problem definition with a view to learning the optimal representations in the unsupervised scenario where no tasks were provided. Although most of the learning is by its very nature unsupervised, an important setting is the *supervised* one in which each visible data-point x is associated with a task (label) y.

In such a scenario the optimal representation, or at least *a priori*, does not coincide with the optimal learned one in the unsupervised setting, since optimally depends on the task at hand. In this chapter, we will discover that, in practical terms, the two optimal definitions in the supervised and unsupervised scenarios are strictly related and the optimal supervised definition is also an optimal unsupervised definition.

Following the same argument as for the previous chapters, we will describe supervised representation learning from the information-theoretic and Gaussian Process perspectives. That description will lead us to observe, in total agreement with previous chapters, that an optimal representation is the one optimising the Constrained InfoMax objective and a generalisation of the objective derived for the unsupervised setting. The description of the optimal network as being the one optimising the CIM, allows us to propose a novel variational objective for the stochastic network, the Variational InfoMax. A variational objective that outperforms, both in terms of accuracy and robustness, the Deep Variational Information Bottleneck, the most popular variational objective used to learn optimal representation within the supervised setting.

6.1 OPTIMAL REPRESENTATION AND INFORMATION BOTTLENECK

FRAMEWORK In this chapter we are working on a complete supervised setting, and in particular under the classification framework. Differently from the previous chapters, the data-points are defined as the pair (x, y), where $x \in \mathbb{R}^D$ represents the visible data, and $y \in [0, 1]^K$ the one-hot vector describing the class where the associated data-point belongs to. From a probabilistic perspective, assuming the data-distribution $p(x)$ is a mixture distribution, $p(x) = \sum_k p(x|k)\pi_k$ we can describe y as the factor defining the distribution from which the associated data is sampled; i.e. $x \sim p(x|k)$ with $y[k] = 1$. A model M trained for this task is the one minimising the distance between the output prediction $M(x) = \hat{y}$ and the correct label y.

6.1.1 *Optimal representation*

The quality of the representation is often associated with the quality of the network itself. For this reason, it is a common assumption to assert that the optimal network learns optimal representations, wherein the optimal network is the one that can *generalise*: i.e. it is able to predict well using unseen data. In light of that definition, a straightforward conclusion is that the network learning the optimal representation is the one having a minimal test error.

However, this definition is not entirely correct. Indeed, within general settings, such as continual or multi-task learning, the unseen data are sampled from another distribution or from another dataset [3, 34, 63] and, for this reason, a network able to generalise is considered as a network learning an interpretable representation, fitting the prior with correct overlapping as per the unsupervised scenario.

Intuitively a network is able to generalise if, in its latent layers and representations, it contains all the relevant information about the data: where each layer has to store the necessary knowledge to predict both the training and the test task correctly, but nothing more than that. Indeed, a representation storing unnecessary information is prone to over-fitting.

A representation satisfying the properties described above can be formally defined in information-theoretic terms as a *minimal sufficient* statistic of the data with respect to the task.

MINIMAL SUFFICIENT STATISTIC A *statistic* Z of the visible data X, is the output of any encoding (possibly stochastic) function f of the data (i.e. $Z = f(X)$.) We say that a statistic Z is *sufficient* for the label (task) Y, if Z contains all the knowledge of X that is shared with Y, i.e.

$$I(Z;Y) = I(X;Y). \tag{6.1}$$

We say that the sufficient representation is *minimal*, if Z is the least informative about X, i.e.

$$\min_Z I(X;Z) \qquad \text{s.t.} \quad I(Z;Y) = I(X;Y). \tag{6.2}$$

INFORMATION BOTTLENECK In light of the given definition, the optimal representation which is able to generalise seems to be the minimal sufficient statistic of the data. Therefore, the idea is to learn the network parameters optimising the objective in (6.2). However, that objective is not operative and it is necessary to consider a more relaxed version, since the sufficiency condition is too restrictive.

To effectively optimise the objective in (6.2), a possible approach is to consider its Lagrangian description, by Lagrange multiplier theorem we know that a minimisation problem with an equal constrain as the one in (6.2) has the same global minimum of the following relaxed objective:

$$\min_Z I(X;Z) - \lambda(I(Z;Y) - I(X;Y)) \tag{6.3}$$

with λ the Lagrange Multiplier, a real number associated with the problem itself. By Data Process Inequality (DPI) $I(Y;X) \geqslant I(Y;Z)$, and so the second term is negative and it is null

only when $I(Z;Y)$ is maximal and so the original, unfeasible to compute objective in (6.2) can be relaxed as

$$\max_{Z} I(Y;Z) - \beta I(X;Z). \tag{6.4}$$

By virtue of the concavity of the objective, the optima of the objective (6.4) exist for any $\beta > 0$ and these are sufficient solutions (e.g. [17, 23]). Moreover, if β is the exact Lagrange multiplier, then the representation is also minimal.

The objective in (6.4), it is called the*Information Bottleneck* (IB) and it was originally proposed in Tishby et al. [82], and after in [81] as a principle by which to describe the optimal Deep Neural Network (DNN). The basic idea is that an optimal DNN, modelled as per the following Markov Chain $X \to Z_1 \to \cdots \to Z_n \to \hat{Y}$, is trained to minimise the supervised loss $D(\hat{Y}, Y)$, in learning in each layer L_i a representation where Z_i is as close as possible to the minimal sufficient one.

The assumption was showed not to be true within general contexts, (e.g. [69]), since the behaviour described in [81] works only for a special structure of the network (which is fully connected with sigmoidal activation). However, since the Minimal Sufficient properties are those defining the optimal representation, many learning objectives were derived to optimise the IB (6.4) directly, see, e.g. [68, 41, 76, 4].

Before proceeding with a brief review of these approaches, let us describe some essential properties of the MinSuf representations. These properties serve to explain why MinSuf is the most desirable for the representation.

6.1.2 *Properties of Minimal Sufficient representation*

In the unsupervised setting, we defined optimal as being the representation informative about the data X and disentangled (each factor associated with just one latent feature). In the following section, we show that the two properties are satisfied by the MinSuf representations.

INFORMATIVENESS Defining $\hat{Y}$ as the predicted and Y as the real task, the neural net can be described by the following Markov Chain: $Y \to X \to Z \to \hat{Y}$. By means of the sufficiency condition, and by DPI, the information shared between the representation and the visible data $I(Z;X)$ is at least $I(Y;X)$. Moreover, by minimality we have, assuming that the channel $Y \to X$ is noiseless, the equality $I(Z;X) = I(Y;X)$. Under such conditions the representation is informative about X, as the label is informative about the data.

DISENTANGLEMENT We have understood that a disentangled representation is one which learns within each feature a generative factor of the data, where each generative factor cannot be shared with other features *(i.e. the disentangled representation is sparse)*. By sufficiency, we know that the representations contain all the information required about the data necessary for the task, wherein all the generative factors are stored within the latent variable. Instead,

the minimality ensures that the generative factors are stored just once within a single feature variable.

Moreover, in agreement with what we observed in the unsupervised setting, in the supervised case it is also possible to prove (e.g. [1]), that a MinSuf representation is maximally insensitive (*invariant*) to noise within the input.

THE MOST INVARIANT REPRESENTATION Until now we have discussed a theoretical representation in the network, but in a DNN all the layers are a different representation *(otherwise some layers are unused)*. For practical reasons, we want to choose a specific representation (layer) and we want to choose the layer that is learning the representation closest to the MinSuf statistic. That representation, in a standard-setting *(i.e. the ones that we take into account)* are the ones associated with the deepest hidden layer. This conclusion follows from the following theorem (see e.g. [1], Corollary 3.4)

Theorem Suppose we have the Markov Chain of layers:

$$X \to Z_1 \to Z_2 \to \hat{Y}, \tag{6.5}$$

and suppose that there is a communication or computation bottleneck arising between Z_1 and Z_2, such that $I(Z_1; Z_2) < I(Z_1, X)$. Then, if Z_2 is still sufficient, it is more invariant to nuisance than Z_1. In other words, the more minimal the more robust to nuisance is the representation.

The bottleneck condition it is a standard assumption since it occurs if $|Z_2| < |Z_1|$, or a pooling is applied after a convolution, and both conditions are often true.

From that property, consistent with what we observed through empirical observations, the idea is to apply the IB with respect the representation learned in the last layer, the one that is the most invariant, and the closest to the MinSuf one.

6.2 RELATED WORK

The first algorithms proposed to optimise the IB principle were limited to the tabular case, wherein X, Y and Z are all categorical variables. In that setting the IB objective (6.4) can be optimised via the Blahut-Arimoto algorithm [82], deterministic annealing [72], or via bottom-up greedy agglomerative clustering [80].

The first Neural Net objective explicitly derived to optimise the IB objective was proposed in Alemi et al. [4]: known as the Deep Variational Information Bottleneck (VIB). The idea, similar to the one considered in the variational AutoEncoder, is to divide the neural network into two subsections, namely the encoder $p(z|x)$, inferring the representation of the data, and the decoder $q(y|z)$ performing the supervised task (regression or classification) on the learnt representation.

Going into finer detail, following the same argument as delineated in [4], we will describe the two mutual information sets composing the IB (6.4) separately.

The decoding information $I(Y;Z)$,

$$I(Y;Z) = \int p(y,z) \log \frac{p(y|z)}{p(y)} dy\,dz, \tag{6.6}$$

is an intractable quantity, since both the joint $p(y,z)$ and the marginal $p(y|z)$ distribution are unfeasible to compute. To overcome the latter issue, the idea is to consider a variational approximation $q(y|z)$, in that setting by means of the KL divergence non-negativity property:

$$D_{KL}(p(y|z)||q(y|z)) \geqslant 0 \rightarrow \int p(y|z) \log p(y|z) \geqslant \int p(y|z) \log q(y|z) \tag{6.7}$$

we obtain a lower bound of the correct (6.6):

$$\begin{aligned} I(Y;Z) &\geqslant \int p(y,z) \log \frac{q(y|z)}{p(y)} \\ &= \int p(y,z) \log q(y|z) dy\,dz - \int p(y) \log p(y) dy \\ &= \int p(y,z) \log q(y|z) dy\,dz + H(Y), \end{aligned} \tag{6.8}$$

with $H(Y)$ being a constant of the system. Unfortunately, the unknown joint distribution persists within the provided lower bound and so, to eliminate this term, it is sufficient to rewrite the joint distribution in the alternative form:

$$\begin{aligned} p(y,z) = p(y|z)p(z) &= p(z) \int p(x,y|z) dx \\ &= p(z) \int p(y|x)p(x|z) dx \\ &= \int p(y|x)p(z|x)p(x) dx. \end{aligned} \tag{6.9}$$

Substituting the joint distribution with the integral expression above, we obtain the following lower bound of the decoding information:

$$I(Y;Z) \geqslant \int p(y|x)p(z|x)p(x) \log q(y|z) dy\,dz\,dx + H(Y). \tag{6.10}$$

The proposed lower bound is feasible to compute since both $p(y|x)$ and $p(x)$ are defined by the data *(if we assume that the empirical distribution is the real one)*, and $p(z|x)$ and $q(y|z)$ are described by the encoder and decoder network, respectively.

In agreement with what was done with the VAE, assuming that we know the prior $p(z)$, we estimate the encoding information,

$$I(Z;X) = \int p(x,z) \log \frac{p(z|x)}{p(z)}. \tag{6.11}$$

with its upper bound: the rate term

$$I(Z;X) \leqslant \int D_{KL}(p(z|x)||p(z)) dx. \tag{6.12}$$

Combining the two terms, we obtain the Deep Variational InfoBottleneck (VIB) objective:

$$\text{VIB} = \int \mathbb{E}_{p(y|x)}[\mathbb{E}_{p(z|x)}[\log q(y|z)]]p(x)dx - \int D_{KL}(p(z|x)||p(z))p(x)dx, \tag{6.13}$$

a variational lower bound of (6.4), wherein the entropy term $H(Y)$ was not included since it is a constant.

Under the assumption the distribution $p(x, y)$ is the empirical one, $p(x, y) = \sum_i \delta_{x_i}(x)\delta_{y_i}(y)$, we can rewrite the VIB as follows,

$$\text{VIB} = \frac{1}{N}\sum_i^N \mathbb{E}_{p(z|x_i)}[\log q(y_i|z)]] - \beta D_{KL}(p(z|x_i)||p(z)). \tag{6.14}$$

CONNECTION WITH THE ELBO If the task y_i is to reconstruct the data-point, i.e. $y_i = x_i$, the VIB in (6.14) is mathematically equivalent to the ELBO objective. The only conceptual difference is that the unsupervised VIB is not defined as a generative model, but rather as an inferential one. Indeed the variational distribution in the VIB is the decoder rather than the encoder. This difference highlights from an alternative perspective that the ELBO is also an optimal inference objective.

Moreover, the VIB connection with the ELBO provides a further information description, one that was not completely clear from the description made in the previous chapter. Indeed, the ELBO can be read, see [4] appendix B, as the variational lower bound of the special Information Bottleneck

$$\max_Z I(Z; X) - \beta I(Z; i), \tag{6.15}$$

where i denotes the identity of each data-point, with

$$p(z, i) = p(z|i)p(i) = \frac{1}{N}p(z|x_i). \tag{6.16}$$

The principle idea is that the ELBO is minimising the information arising between each single data-point x_i and the global representations Z, and it thereby maximises the information within the data X.

To better understand the objective given in (6.15), we can consider the CIM equivalence we exploited in the fourth chapter. Remembering that, in the setting of optimal β, the rate term can be substituted with its "average" $D_{KL}(N^{-1}\sum_i q(z|x_i)||p(z))$. we have that the unsupervised objective in (6.15) can be rewritten as:

$$\max_Z I(Z; X) - \beta I(Z; \hat{X}), \tag{6.17}$$

where $\hat{X}$ denotes the average value of the data.

In other words, the unsupervised IB is performing the InfoMax between the latent and the visible data, eliminating the average information arising in terms of the data. That behaviour is in agreement with the intuitive idea of learning which we experience in daily life, where we tend to remember the rare events instead than the average ones.

STOCHASTIC NETWORK AND REPARAMETRISATION TRICK In a stochastic network, to describe the encoder distribution $p(z|x)$ there is a commonly used approach termed the reparametrisation trick [38]. Assuming that the network is normally distributed, the idea is to consider two separate encoder functions $\phi_1(x)$ and $\phi_2(x)$, describing the mean and standard deviation of the network, respectively (i.e. $p(z|x) = \mathcal{N}(z|\phi_1(x), \phi_2(x)^2 I)$), and then to sample the latent variable z as $\phi_1(x) + \varepsilon\phi_2(x)$ with $\varepsilon \sim \mathcal{N}(0, I)$. Instead, in this instance, the decoder distribution depends on the task and the chosen accuracy loss. If, for instance, the task is the regression, it is usually considered as the L_2 loss which is associated with the normal encoder $q(y|z) = \mathcal{N}(y|Wz, \sigma_y^2 I)$ with σ_y being a possible term to learn.

Summarising all these elements, the Stochastic Neural network f describes the distribution $q(y|z)p(z|x)$ and is written as

$$f(x) = W(\phi_1(x) + \phi_2(x)\varepsilon) = Wz_0 + W(z_1\varepsilon). \tag{6.18}$$

The general assumption is that the two-components of $\phi = [\phi_1, \phi_2]$, are obtained via the same network and they differ just in respect of just the last layer (i.e. z_0 and z_1) and can be written, respectively, as $A_0\varphi$ and $A_1\varphi$, where φ is the common encoder function, and A_0 and A_1 are the matrices associated to the respective layers.

By that characterisation, the additive noise in (6.18) can be written as a multiplicative term, and indeed defined as ε_1 , a normally distributed noise such that $\varepsilon A_0 = \varepsilon_1 A_1$, the stochastic network, is written equivalently as:

$$\begin{aligned} f(x) &= W(A_0 b + \varepsilon A_1 b) \\ &= W(A_0 b + \varepsilon_1 A_0 b) \\ &= W(1 + \varepsilon_1) z_0 \\ &= W(\xi z_0). \end{aligned} \tag{6.19}$$

The equivalent description of the stochastic network with multiplicative noise in (6.19), one that can be considered both as a noise in the latent variable or in the weights, leads us to rethink the encoder and decoder distribution of the network. First of all, this approach allows us to see the dual connection between noise within the variable and within the weights, obsevation equivalent to the one highlighted in the unsuepervised setting between VAE and PCA. Indeed, as the Probabilistic PCA was first introduced as an inference problem on the weights and then revisited as a GPLVM in [46], the supervised network can be described as a Gaussian Process wherein the inference on the latent variables is equivalent to the inference on the weights.

Second, it allows us to rewrite the network as being a deterministic encoder of form $p(z|x) = \delta(z = \phi_0(z))$ in which the decoder distribution is normally distributed $p(y|z) = \mathcal{N}(Wz, \alpha(Wz))$ with α representing the variance of ξ. This means that the DIB approximation is not a loose approximation of VIB, but rather an alternative one. Let us describe these two implications separately.

6.3.1 *The variational objective*

By the equivalent description in (6.19), we have seen that the encoder network can be considered to be deterministic, and thus it follows that the Information Bottleneck can be approximated by the simpler Deterministic IB. Moreover, by means of the DPI we observed that the encoding information $I(Z;X)$ has to be at least $I(X;Y)$.

Combining these two observations, we can ascertain that the optimal network is the one optimising the following Constrained InfoMax (CIM) objective:

$$\max I(Z;Y) - \lambda H(Z), \qquad \text{s.t } .H(Z) \geqslant I(X;Y). \tag{6.20}$$

The CIM objective above is however unfeasible to optimise, since the lower bound condition is not easy to satisfy and, in the setting that we are interested in considering, the prior $p(z)$ is chosen in advance.

A more feasible objective to compute is the following:

$$\max I(Z;Y), \qquad \text{s.t.} \quad H(Z) = I(X;Y), \tag{6.21}$$

where assuming that we know the optimal encoding information, it can be optimised via the following variational objective, the Variational InfoMax (VIM):

$$VIM = \sum_i \mathbb{E}_{q(z|x_i)}[p(y_i|z)] - \lambda D(q(z)||p(z)), \tag{6.22}$$

where the prior $p(z)$ is chosen in such a way that $H(Z) = I(Z;X)$, and D denotes any divergence between the data and the .

Rewriting the VIB objective as follows:

$$VIB = \sum_i \mathbb{E}_{q(z|x_i)}[p(y_i|z)] - \lambda D_{KL}(q(z)||p(z)) - \lambda I(X;Z), \tag{6.23}$$

we see that in the case we choose $D = D_{KL}$ on (6.22), the only difference between the two is the absence of the encoding bound in the VIM objective. In light of the observation made above, that the optimal encoder can be described as deterministic one, the extra term in VIB penalising the encoding information can be read as adding further noise, to the one already introduced during training, into the channel; leading to a less decomposed representation.

Let us finally observe, that in practical cases we do not know the optimal $I(X;Z)$ and, for this reason, the optimal prior $p(z)$ has to be chosen as being the one with minimal entropy such that the decoding information is maximal or, in computational terms, the test error is minimal.

CIM VS IB From a purely theoretical perspective, the IB (6.4) is preferable to the Constrained InfoMax (6.21) and, since the IB is more general, it works for both the stochastic and deterministic encoders as it finds the optimal encoding information by itself. Conversely, the CIM works only with the deterministic model and it needs to know the optimal encoding information in advance.

If, theoretically, the IB is preferable to the CIM objective, the same cannot be said for their variational lower bounds:- the VIB (6.14) and the VIM (6.22). Indeed, in the variational scenario, the prior $p(z)$ is chosen in advance for both the objectives and then the VIB can control the encoding information simply by adding some noise in the channel, thereby obtaining a more entangled representation.

Conversely, in the VIM setting, although it is still not easy to set the correct prior to having the optimal representation, the representation is, in any case, the most informative and the quantity of noise in the channel is minimal.

6.4 EXPERIMENTS

Following what was performed in the theoretical sections, in this section, with the help of both synthetic and standard datasets, we compare the VIB and VIM objectives. We will observe that both the objectives are associated with networks obtaining the best test accuracy performance, but with different learnt representations, since the VIM representations are, in general, more disentangled than the VIB ones. Moreover, we analyse the effect of the latent entropy, noting that the choice of that parameter does not influence the test accuracy or the geometric properties of the representation, yet the robustness of the network then overlaps the properties of the features.

TWO-SPHERE DATASET To better understand the differences arising between the two learning objectives, VIB and VIM, we start to describe their behaviour in the synthetic setting of the two spheres dataset, which we already considered in the previous chapter. For this dataset, we consider a fully connected network with a linear decoder and one hidden layer encoder $\phi : x \to h \in \mathbb{R}^{128} + \mathrm{ReLu} \to z \in \mathbb{R}^2$, where the latent variable z is chosen to be two-dimensional since the classes describing the data are just two-fold *(and so that we may provide a clear plot of the representations).* The decoder distribution is chosen to be a spherical Gaussian with accuracy loss defined by the L_2 loss $\|y - \hat{y}\|_2^2$. After fine-tuning the hyper-parameters associated to the two regularisers, the values are, respectively, $\beta = 0.2$ for VIB and $\lambda = 5$ for VIM, all the modesl are trained with SGD with learning rate $lr = 1e-4$.

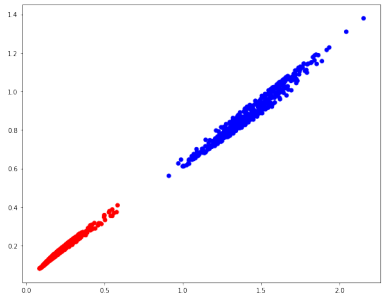
Accuracy. The main goal of any objective is to learn a network that maximises the accuracy. The task, in this special case, is particularly simple, and thus the results are not really relevant but, as we see in table 6.1, the regularised models have slightly better accuracy performances, suggesting that this encoding regularisation is a good choice.

Learnt representation In the theoretical analysis a disentangled minimal sufficient representation is the one learning the hidden dynamics of the Gaussian Process, or more concretely the hidden structure of the data, i.e. the optimal features are associated to the directions where the hidden manifold has higher derivative.

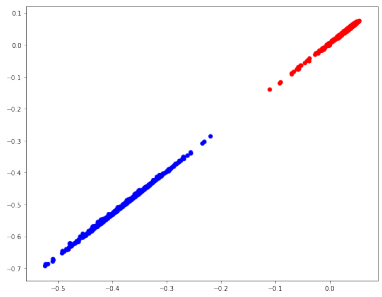
As we see from figure 6.1, confirming the optimal test error results, all the models are able to separate out the two different classes, yet only the VIM solution is learning, in agreement

Table 6.1 Accuracy for the two-sphere dataset. The task is particularly simple and the difference arising is minimal between the models.

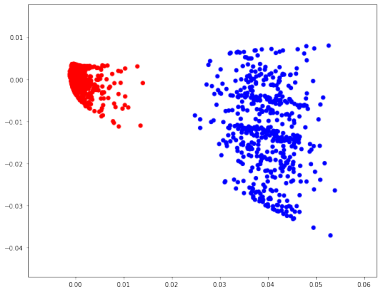
Models	Test-error, $\|y - \hat{y}\|_2^2$
Deterministic	2.07
VIB	2.04
VIM	2.03



(a) Deterministic



(b) VIB

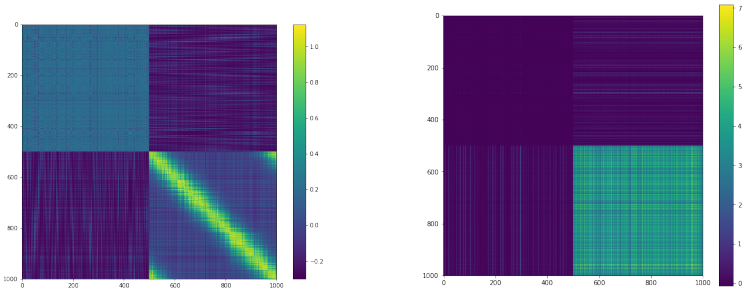


(c) VIM

Figure 6.1 Learnt representations by the three objectives. Both VIB and the Deterministic model are learning a representation that separates the two classes; but it is not learning the hidden structure of the data as the VIM is.

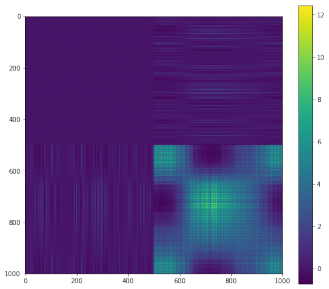
with the theoretical deductions, the hidden manifold:- the cone where the red points are depicted *(top)*, and the blue ones *(in the basis).*

That difference in the learnt representation is also made clearly visible by the learnt kernel $K = ZZ^T$, as we see in figure 6.2, where the kernel learnt by VIM and VIB are compared with the RBF one, a kernel constructed by the feature map that better describes the dataset structure. In particular, as we see from this comparison, the VIB kernel has the same structure as the one learnt by the representations of the deterministic model *(i.e. the kernel is describing the labels* Y*, and not the data* X*).* A more complex kernel is learned by VIM, where the points around the diagonal have higher correlation values. These observations suggest that the VIM network is learning a better representation, since it is discovering the geometric relationship in the data-space that goes beyond the information given by the labels.



(a) RBF

(b) VIB



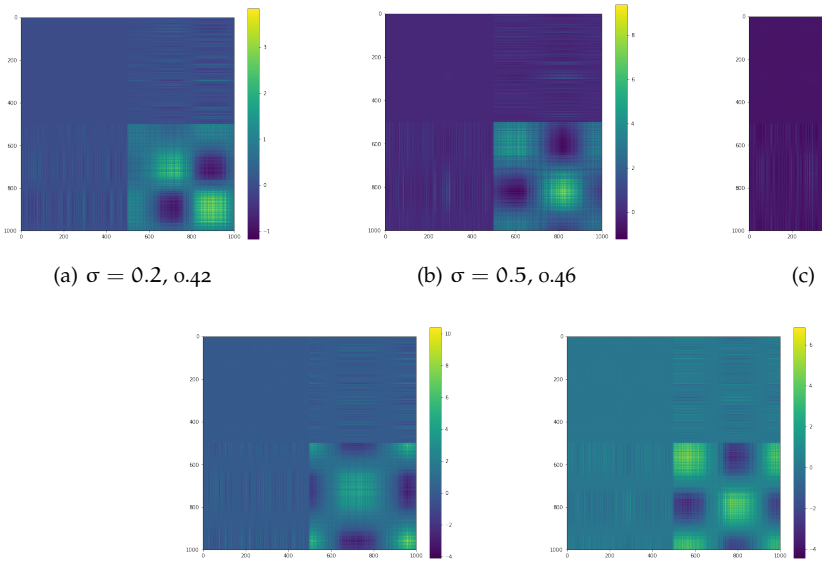
(c) VIM

Figure 6.2 Comparison of the latent kernels associated to the representation. Both VIB and VIM solutions are not learning the RBF, assumed to be the correct one, but if the VIB is learning a kernel describing the labels, the VIM kernel is describing the hidden structure of the data.

The optimal prior The Kernel description is useful to show how the choice of the prior affects the learning performances. As we see from figure 6.3, the geometric structure of the latent data is similar for all the representations, since the kernel structure is the same for all the variables. The only difference arising between the kernels is in the absolute value of the matrix entries; where a higher latent entropy is associated with higher covariance values. This phenomenon can be explained by the fact that a higher entropy network is storing more information about the data and then relating more "close" points. Assuming that the learnt feature is the RBF one, it is possible to state that the variance on the prior is proportional to the radius of the feature map, and that the higher entropy is storing more context about each data-point.

Unfortunately, a qualitative description of the kernel does not implicitly state which one is the best. In this case, since we are considering a synthetic dataset, we can assume that we know the correct kernel. In this instance, let us say it is the RBF one presented in fig.6.2. In this way we can evaluate the best-learnt kernel as being the closest to the correct one.

As we see from the results in table 6.2, in which we list the alignments between the correct and the learnt kernels *(if we consider the alignment measure proposed in* [19]), optimal results are obtained for $\sigma = 1$, but more importantly we see that, in agreement with the theoretical observations, the quality of the representation is proportional to the accuracy, where a higher alignment corresponds to a greater accuracy.



(a) $\sigma = 0.2$, 0.42 (b) $\sigma = 0.5$, 0.46 (c) $\sigma = 1$, 0.46

(d) $\sigma = 1.2$, 0.48 (e) $\sigma = 1.5$, 0.42

Figure 6.3 Comparison of latent kernels as a function of σ^2. The difference lies not in the shape, but rather in the absolute values of the kernel matrix.

Table 6.2 Accuracy for the two-sphere dataset, for the different VIM trained models.

σ^2	Accuracy	Alignment
0.2	2.12	0.41
0.5	2.09	0.43
1.0	2.03	0.46
1.2	2.08	0.45
1.5	2.12	0.43

Table 6.3 Comparison test-error on MNIST (smaller is better), with $Z \in \mathcal{N}(0, I)$, $I \in \mathbb{R}^{K \times K}$, $K = 256$

Model	test error (%)
Baseline	1.38
Dropout [4]	1.34
Label Smoothing [64]	1.23
Confidence Penalty [64]	1.17
VIB ($\beta = 10^{-3}, \sigma = 1$) [4]	1.13
VIM($\lambda = 10^{-3}, \sigma = 1$)	**1.10**

6.4.1 *MNIST*

With the help of the two-sphere synthetic dataset, we were able to describe the qualitative behaviour of the representations learnt by the network as trained with the two variational objectives. To proceed with a quantitative comparison, it is first necessary to consider some more challenging datasets. The first that we consider is the MNIST, a classic benchmark dataset, for which it is possible to compare the different objectives described above fairly.

To be consistent with the literature, the listed results are obtained from training the same network as considered in [4], using the same hyper-parameters. In particular, we consider a fully interconnected neural network with a linear decoder, followed by a softmax non-linearity, and an encoder ϕ having the following shape:

$$\phi : x \in \mathbb{R}^{728} \to h_1 \in \mathbb{R}^{1024} \to h_2 \in \mathbb{R}^{1024} \to [\mu_z, \sigma_z] \in \mathbb{R}^{20},$$

where each hidden layer h is followed by a ReLU activation, trained with a Stochastic Gradient Descent, and learning rate $lr = 1e-4$.

By virtue of the choice of the decoder, the Accuracy loss is the cross-entropy: $H(q(y|z)||\delta(y|x)) = \sum_i \log q(y_i^*|x)$.

Accuracy Following the same procedure considered for the 2 sphere dataset, we start comparing the accuracy of the network. In table 6.3 are listed the test errors obtained by the same network trained with the principal objectives described above. By these results we see that the two variational objectives are the optimal ones, the ones with the smallest test error, confirming that the MinSuf representation is associated with the optimal network. Moreover, although the difference is small, the VIM objective yields better accuracy, in agreement with the theoretical observation that a VIM encoder is more informative and less noisy than the VIB one.

REPRESENTATION QUALITY By virtue of these accuracy experiments, we have seen that the VIM and VIB objectives are the ones learning optimal representations. However, as seen in the previous chapter, although the task accuracy is informative about the representation

Table 6.4 Adjusted Rand and Hoyer index of the learned representation. For the Rand index, a higher value is better.

Model	adjR	Hoyer
Baseline	0.938	0.63
VIB ($\beta = 10^{-3}, \sigma = 1$)	0.948	0.69
VIB ($\beta = 10^{-4}, \sigma = 1$)	0.951	0.67
VIM($\lambda = 10^{-3}, \sigma = 1$)	**0.954**	**0.59**

quality, it does not say everything about it. For this reason, it is necessary to look inside the network to understand which objective is learning the most disentangled representations.

Decomposability As observed previously, the disentangled metrics are biased, and they work only when the ground-truth factors are given. For this reason, following what was done in [58], we considered evaluating the decomposability of the representations via two associated metrics:- the sparseness, measuring the degree of overlap in the latent space; and the clustering, measuring the capacity to learn the hidden structure of the data. To measure the sparseness, we consider the Hoyer metric, which we introduced previously and defined as:

$$\mathrm{Hoyer}(y) = \frac{\|y\|_1/\|y\|_2 - 1}{\sqrt{d} - 1}, \tag{6.24}$$

with d the dimension of the latent space.

To evaluate the clustering, we evaluate the adjusted Rand index $adjR$ between the sets C_i, individuated by a classic K-means trained with 10 clusters and the set of representations associated by labels L_i. In detail, defining $a_i = C_i \cap L_i$ the $adjR$-index is defined as:

$$adjR = \frac{\sum_i a_i}{\sum C_i} \in [0, 1],$$

yielding 1 for a complete overlapping between the clusters and the correct set; and a 0 value if no point lies in the intersection between the two sets.

As we see from table 6.4, wherein are listed the disentangled performances for the different objectives, the most decomposed representations are learnt by the VIM, in agreement with the task results, but perhaps most surprisingly, those most decomposed for VIB are not those for which the test error is minimal, where $\beta = 10^{-3}$, but for the VIB with $\beta = 10^{-4}$, which exhibits a slightly larger test error of 1.15%. This behaviour is actually similar to what we already have seen within the unsupervised setting (e.g. the fourth chapter) where the compromise between the optimal task and optimal representation was not easy to find.

The role of σ From the experiments described above, we have seen that the VIM-trained model is learning better representations than the VIB one. In the theoretical section, we previously asserted that the entropy of the prior has an effect on the network performance and, in particular, on the quality of the latent representation. For this reason, in the following section, we describe the behaviour of the network that is trained to optimise the VIM objective with different σ^2 values.

Table 6.5 Adjusted Rand and Hoyer index of the learned representation as a function of σ^2. There are no real differences arising between the VIM solutions.

Metrics	σ^2				
	0.1	0.25	0.5	1	2
Test error (%)	1.08	1.07	1.07	1.10	1.14
adjR	0.955	0.956	0.953	0.956	0.954
Hoyer	0.591	0.592	0.595	0.594	0.596

As we see by the results listed in table 6.5, the effect of the prior is not completely visible, either by the test error, where all the values are optimal *(all of them are smaller than the VIB error)*; or by the two decomposable metrics which are essentially the same for all the priors. Confirming what was seen for the synthetic datasets: geometrically the VIM representations are the same for all the priors.

Robustness to noise In light of the poor informative decomposable metrics, we decided to evaluate the representation quality indirectly. In particular, we consider the network robustness as being a proxy for the decomposability. Indeed, a MinSuf representation is the most robust to noise. To evaluate the robustness of the network, we measure how much noise we have to add to the input data in order to get an incorrect output, and then we can state that the network A is more *robust* than network B if, on average, we have to add *more* noise to the A input than the B input to generate misleading outputs.

Formally, given a network M and an input x with label C_i , such that $M(x) = C_i$, the successful adversary $A(x) = x + v$ of $x \in C_i$, of a (targeted) attack with target C_j with $i \neq j$ is the closest element x', with respect to a prescribed measure, such that $M(A(x)) = C_j$. Defining x' as the successful adversary of x, the robustness of the network is defined as the average distance $||x - x'||_n$, with $n \in \{1, 2, \infty\}$.

In our experiments, consistent with the choices made in [4], the robustness metric is evaluated with respect to the first ten zero digits within the test set with the adversary target being the label one, i.e. $M(A(x)) = C_1$ with $x \in C_0$, using the adversary method proposed in [12] optimised according to the L_2 distance.

As we see from figure 6.4, where the relative magnitude of the adversarial perturbation, as measured using the L_1, L_2 and the L_∞ norms, for the first 0 digits in the test data, as a function of σ^2: a small σ^2 network is more robust to noise than a network with large σ^2 prior.

Such behaviour is in total agreement with the Minimal Sufficient theory, where was observed that, given an equally informative representation about the task *(for all the VIM models the test error is almost equal)*, the most robust to noise is the representation sharing less information with the visible data *(i.e. the one with the smallest entropy)*.

The norm values in figure 6.4 are all normalised by the corresponding norm of the perturbation against the optimal VIB model. That choice is made to observe that, for $\sigma = 1$, the

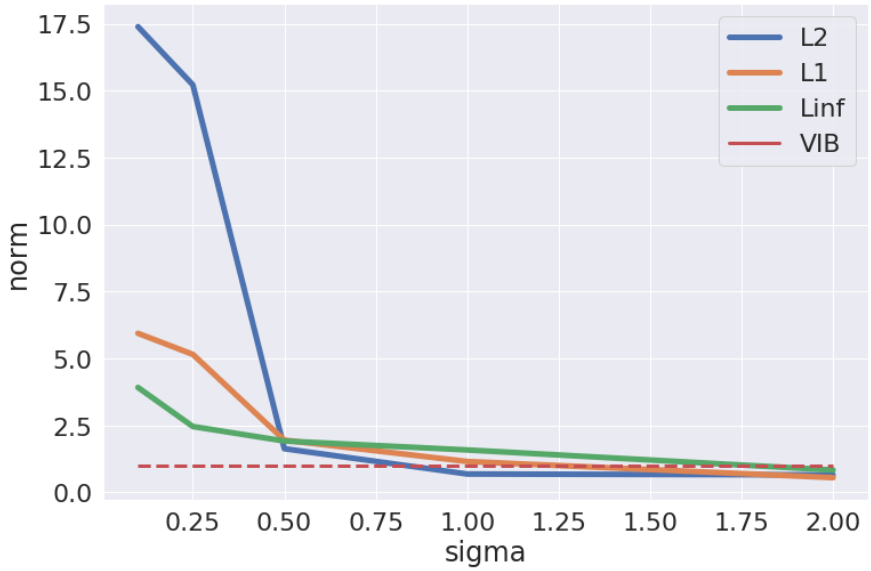


Figure 6.4 Relative magnitude of the adversarial perturbation, measuring using L_1, L_2, L_∞ norms, of the first ten 0 MNIST digits for VIM as a function of σ^2, with respect the VIB solution *(dashed line)*. The effect of σ^2 is clear, as the smaller the value of σ^2, the higher the network robustness.

representations learnt by VIM and VIB are almost identical, confirming the theoretical analysis asserting that, where an optimal choice of β the VIM and VIB are equivalent, it is the least informative.

6.4.2 *CIFAR 10*

After starting our experiments with a toy dataset to describe the qualitative behaviour of the latent variables; we moved to the MNIST which was a slightly more challenging dataset to understand in terms of the differences arising between VIB and VIM from a computational perspective and to see which metrics we should consider. Then we conclude the chapter with the CIFAR10 experiments.

The CIFAR10 [29] is a set of 60k 32×32 natural colour images which has been divided into ten classes *(e.g. aeroplane, car and cat)*. We choose to consider this dataset for two reasons:- it is more challenging than the MNIST dataset and it allows us to determine if the VIM objective can be applied to a complex network such as the Convolutional Neural Network, an architecture that is necessary to yield good performances.

In particular, we performed these experiments while considering an encoder network of four convolutional layers with a filter of size 4×4 and an increasing kernel size, followed by a Batch Normalisation, as illustrated in table 6.6, and, using as a decoder, a classic logistic as for the MNIST setting. The structure of the network is similar to that considered in [2, 88] and, as already observed in [2], the batch normalisation is added only to have the more stable computation without really affecting the final results. The network is trained using Adam with a learning rate starting from 10^{-3} and decreasing after 30 epochs by a factor of 2.

Table 6.7 Accuracy, adjusted Rand and Hoyer index of the learned representation, as a function of σ^2

Metrics	σ^2					VIB (β)
	0.1	0.25	0.5	1	2	10^{-3}
Test error (%)	33.37	33.24	33.19	32.9	32.7	33.29
adjR	0.32	0.34	0.38	0.39	0.39	0.42
Hoyer	0.22	0.22	0.22	0.22	0.22	0.24

In accordance with what we observed in the MNIST setting, as we can appreciate from table 6.7 and also in the CIFAR10 experiments, the task accuracy of the VIM and VIB-trained models are similar, but the VIM is slightly better than for the optimal VIB trained model ($\beta = 10^{-3}$).

Table 6.6 CNN architecture of the encoder network used for the CIFAR experiments

Input ($32 \times 32 \times 3$)
Conv(4×4, 128)
BN + ReLu
Conv(4×4, 256)
BN + ReLu
Conv(4×4, 512)
BN + ReLu
Conv(4×4, 1024)
BN + ReLu
Fully connected $2K$, $K = 64$

Moreover, from table 6.7, we see that the decomposable metrics, adjusted Rand index and Hoyer, do not say very much about the disentanglement degree of the representation. Indeed, all the values are really similar, only the Rand index differs slightly along with the different representations indicating that, in a small variance representation, it is difficult to find the correct cluster for the data.

In light of that behaviour, as performed above, to evaluate the quality of the learnt representations we must consider the robustness to noise. The robustness metric is evaluated with respect to the first ten data-points in the test set where the adversary target is set to be successive to the correct label (i.e,. if $x \in C_i$, x' is a successful attack if $M(x') = C_{i+1}$).

By means of these robustness results, as summarised in figure 6.5, we see how the robustness *(high error magnitude*) is inversely related to the latent entropy, highlighting that the best way to bound the encoding information is by bounding the latent entropy and not the encoding information itself.

Moreover, by virtue of the comparison with the optimal VIB solution *(the dashed red line)*, we see that the $\sigma^2 = 1$ VIM is almost equivalent to the optimal VIB and the $\sigma^2 = 2$ is learning less efficiently *(i.e. providing less robust representation)* than the VIB, suggesting that, in the latter case, the encoder channel is transferring too much information and that the representation is also storing the noise of the visible data.

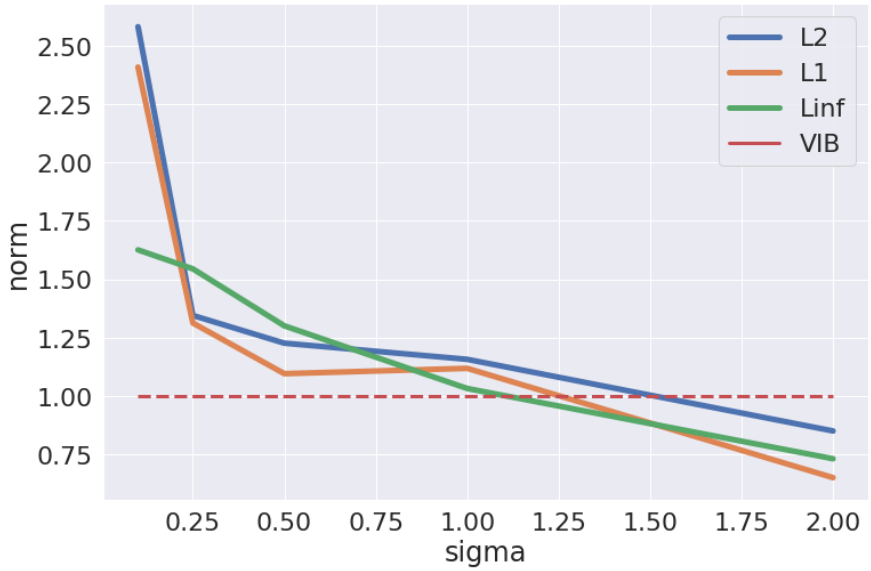


Figure 6.5 Relative magnitude of the adversarial perturbation, measured using L_1, L_2, L_∞ norms, of the first ten test CIFAR data-points for VIM as a function of σ^2, with respect the VIB solution *(dashed line)*. The effect of σ^2 is clear; the smaller σ^2, the higher the network robustness. In the case of $\sigma = 1$, the solution is equivalent to the VIB one. However, for $\sigma^2 = 2$, the network is less robust than the optimal VIB, suggesting that the amount of shared information between the input and the representation is unnecessary high.

6.5 CONCLUSION

In this chapter, after an informative description of the supervised network, we provided a principle by which to learn the optimal – Decomposable and Minimal Sufficient – representations: the Constrained InfoMax. A supervised generalisation of the objective was introduced for the unsupervised problem, wherein the generative task was substituted with a generic (supervised) task.

The CIM objective is informatively equivalent to the Information Bottleneck, but the proposed equivalent description has both theoretical and computational advantages over the IB description.

From a theoretical perspective, the CIM provides a unique principle that works both within the unsupervised and supervised settings, yielding a Gaussian Process description of the stochastic network and also an alternative definition of the optimal representation *(i.e. the one describing the data dynamics)*. Instead, from a computational side, the introduction of CIM leads to an alternative variational approximation of the IB, the Variational InfoMax, wherein the encoding information is bounded *(by the latent entropy)* but not penalised as in the VIB setting.

The derived objective, as observed from these experiments, performs better than the VIB with the two main structures of DNN, namely the fully connected and convolutional networks. Future work might include extending the VIM objective to the autoregressive network, wherein the computation of the latent inference is not trivial.

7

CONCLUSIONS

In this thesis we introduced the Constrained InfoMax (CIM), a principle by which we may learn optimal representations - informative and disentangled - within both supervised and unsupervised settings. The CIM principle generalises the Information Bottleneck to the unsupervised setting, wherein the task with respect to optimising the objective is the generative one. The advantage of introducing the CIM principle is not solely theoretical, constraining the information of the latent representation rather than the encoding information, but also computational. Indeed, the associated variational objectives; the Variational Wasserstein distance (Wass) and Variational InfoMax (VIM) respectively for unsupervised and supervised tasks; are both learning better representations than were achieved for the informatively theoretical equivalent variational objectives, ELBO and VIB within the unsupervised and supervised settings, respectively. Moreover, the CIM description highlights the effect of the latent entropy in determining the representation quality:- a small entropy latent variable is, in general, more robust than an high entropy one.

To understand the reasons why the CIM variational objectives, within the unsupervised scenario, are learning better representations than the ELBO *(the two objectives are informatively equivalent)*, Chapter Five provides a geometric informative description of the two variational objectives. By virtue of this analysis we have shown that the ELBO representations are associated with the linear projection of the visible data. Instead, the Wass representations are obtained, in general, via a non-linear projection. Alternatively, from a Gaussian Process perspective, the ELBO representations are describing a linear kernel of the data.

This issue is associated with the rate term present within the ELBO and in the VIB; indeed, also in the supervised case, we have seen that the VIB learnt representations are associated to the label kernel while the VIM kernel is describing the hidden structure of the data.

Unfortunately, although qualitatively visible, the differences arising between the learnt representations of the different families of models are not clear from the disentanglement metrics considered in the manuscript. This unexpected result motivates us to investigate, in future work, the properties of the disentangled metric so we can provide a possibly better metric highlighting the qualitative differences observed in these experiments.

7.1 FUTURE WORK

The research on new metrics to evaluate the degree of disentanglement in the latent representation is just one possible future avenue for exploration. This thesis, as presented, opens many different potential paths for future research that can be divided into two main categories:-

computational improvement of the variational objectives and seeking a generalisation theory for the CIM.

COMPUTATIONAL IMPROVEMENTS As asserted throughout this thesis, the entropy of the prior plays a fundamental role in representation learning, but in respect of the standard variational objectives considered here, the prior has to be fixed before the training and the optimal solution has to be chosen at the validation stage. One possible improvement of the actual work is to consider a variant of the Wass, wherein the optimal variance of the distribution is learnt simultaneously with the weights. The idea is to learn the σ via an annealing process in a manner akin to [32], wherein the annealed term was the β parameter.

In a similar vein, a further potential improvement of the Wass is to consider a free-inference objective, wherein the prior distribution is learnt during the training itself. Indeed, throughout this thesis, we have implicitly assumed that a Gaussian distribution could describe the latent representation but, in particular scenarios, that assumption could prove restrictive, and thus it is necessary to consider different distributions. One possible solution is provided in [39] where, using Inverse Autoregressive Flow techniques, a general prior distribution was learnt without fixing any shape in advance.

THEORY GENERALISATION In Chapter Five, we analysed the representations learnt by a linear decoder VAE and connected them to linear Gaussian Process Latent Variable Models (GPLVMs). In future work, we will consider the more general setting in which the decoder is a general non-linear map. In this setting, we expect that the learnt representation will still be associated with the GPLVM, wherein the inference and generative kernels are described by the encoder and decoder maps, respectively.

If that generalisation looks straightforward, the second objective - to generalise the CIM for continual learning - is more challenging.

In this thesis, we have considered a static learning scenario, wherein the data-set and the associated task are the same for all the training stages. As we experience in our daily life, the static scenario is unrealistic as natural learning is *continual*:- therefore, the representation of a data-set has to evolve continuously during the training period since the associated task changes with the context.

Let us note that the continual learning scenario is challenging since it cannot be performed as a simple collection of static steps:- training one task each time. Indeed, in the latter case, the model suffers from the infamous *catastrophic forgetting* issue, [40] as, once the network starts to learn the new task Y_t, the previous one Y_{t-1} is *forgotten*.

In light of the demonstrated ability of CIM-trained networks to learn informative representations about the data *(independently of the task)*, in future research we will extend the Constrained InfoMax to deal with the continual learning scenario. The main idea is to increase the prior entropy during the training with the number of tasks to store all the necessary information for any task and not to forget anything.

In particular, we have seen that, in the supervised setting, the CIM objective for the single task Y_1 is:

$$\max_Z I(Z;Y_1) \text{ s.t. } H(Z) = I(X;Y_1), \tag{7.1}$$

then, the natural generalisation to the continual learning scenario, with task $Y = \{Y_i\}_i^t$, is to consider the following objective:

$$\max_Z I(Z;Y) \text{ s.t. } H(Z) = I(X;Y). \tag{7.2}$$

By chain rule of the mutual information $I(X;Y_1,Y_2) = I(X;Y_1) + I(X;Y_2|Y_1)$, this objective can be continually optimised as:

$$\max_Z I(Z;Y_n) \text{ s.t. } H(Z_n) - H(Z_{n-1}) = I(X;Y_n|Y_{n-1}), \tag{7.3}$$

where Z_n is storing all the information about Z_{n-1}, and $I(X;Y_n|Y_{n-1})$ is the amount of information shared by X with Y_n knowing all the previous terms Y_{n-1}.

We expect, thanks to the generalisation provided in (7.3), that it would be possible to perform continual learning as a collection of single-task learning events, thereby avoiding the catastrophic forgetting scenario. Indeed, in the proposed objective, the latent variable changes with the task and the previous information should thereby remain untouched, and the representation is simply adding new knowledge.

BIBLIOGRAPHY

[1] Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19(50):1–34, 2018. URL `http://jmlr.org/papers/v19/17-646.html`.

[2] Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence*, 40(12): 2897–2905, 2018.

[3] Alessandro Achille, Tom Eccles, Loic Matthey, Chris Burgess, Nicholas Watters, Alexander Lerchner, and Irina Higgins. Life-long disentangled representation learning with cross-domain latent homologies. In *Advances in Neural Information Processing Systems*, pages 9873–9883, 2018.

[4] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[5] Alexander A Alemi, Ben Poole, Ian Fischer, Joshua V Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken elbo. *arXiv preprint arXiv:1711.00464*, 2017.

[6] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 37–49, 2012.

[7] David Barber and Felix V Agakov. Kernelized infomax clustering. In *Advances in neural information processing systems*, pages 17–24, 2006.

[8] Horace B Barlow et al. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1:217–234, 1961.

[9] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[10] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[11] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[12] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[13] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620, 2018.

[14] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pages 2980–2988, 2015.

[15] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[16] D. Coon and J.O. Mitterer. *Introduction to Psychology: Gateways to Mind and Behavior*. Available Titles CengageNOW Series. Cengage Learning, 2008. ISBN 9780495599111. URL https://books.google.it/books?id=vw20LEaJe10C.

[17] Costas Courcoubetis and Richard Weber. Pricing and communications networks. *Wiley-Interscience series in systems and optimization*, page 3, 2003.

[18] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[19] Nello Cristianini, John Shawe-Taylor, Andre Elisseeff, and Jaz S Kandola. On kernel-target alignment. In *Advances in neural information processing systems*, pages 367–373, 2002.

[20] Cian Eastwood and Christopher K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=By-7dz-AZ.

[21] Mohammad Emtiyaz Khan, Alexander Immer, Ehsan Abedi, and Maciej Korzepa. Approximate inference turns deep networks into gaussian processes. *arXiv*, pages arXiv–1906, 2019.

[22] Karl Friston. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2): 127–138, 2010.

[23] Ran Gilad-Bachrach, Amir Navot, and Naftali Tishby. An information theoretic tradeoff between complexity and accuracy. In *Learning Theory and Kernel Machines*, pages 595–609. Springer, 2003.

[24] Peter W Glynn and Donald L Iglehart. Importance sampling for stochastic simulations. *Management science*, 35(11):1367–1392, 1989.

[25] Aditya Grover and Stefano Ermon. Uncertainty autoencoders: Learning compressed representations via variational information maximization. *arXiv preprint arXiv:1812.10539*, 2018.

[26] Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taiga, Francesco Visin, David Vazquez, and Aaron Courville. Pixelvae: A latent variable model for natural images. *arXiv preprint arXiv:1611.05013*, 2016.

[27] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[28] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, volume 3, 2017.

[29] Tien Ho-Phuoc. Cifar10 to compare visual recognition performance between deep neural networks and humans. *CoRR*, abs/1811.07270, 2018. URL http://dblp.uni-trier.de/db/journals/corr/corr1811.html#abs-1811-07270.

[30] A. Honkela and H. Valpola. Variational learning and bits-back coding: an information-theoretic view to bayesian learning. *IEEE Transactions on Neural Networks*, 15(4):800–810, 2004.

[31] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359 – 366, 1989. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(89)90020-8. URL http://www.sciencedirect.com/science/article/pii/0893608089900208.

[32] Chin-Wei Huang, Shawn Tan, Alexandre Lacoste, and Aaron C Courville. Improving explorability in variational inference with annealed variational objectives. In *Advances in Neural Information Processing Systems*, pages 9701–9711, 2018.

[33] Niall Hurley and Scott Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, 2009.

[34] Maximilian Igl, Kamil Ciosek, Yingzhen Li, Sebastian Tschiatschek, Cheng Zhang, Sam Devlin, and Katja Hofmann. Generalization in reinforcement learning with selective noise injection and information bottleneck. In *Advances in Neural Information Processing Systems*, pages 13978–13990, 2019.

[35] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

[36] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[37] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[38] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in neural information processing systems*, pages 2575–2583, 2015.

[39] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016.

[40] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526, 2017.

[41] Andreas Kirsch, Clare Lyle, and Yarin Gal. Unpacking information bottlenecks: Unifying information-theoretic objectives in deep learning. *arXiv preprint arXiv:2003.12537*, 2020.

[42] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[43] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017.

[44] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. ISSN 0036-8075. doi: 10.1126/science.aab3050. URL https://science.sciencemag.org/content/350/6266/1332.

[45] Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, pages 536–543, 2008.

[46] Neil Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of machine learning research*, 6(Nov):1783–1816, 2005.

[47] Neil D Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in neural information processing systems*, pages 329–336, 2004.

[48] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

[49] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2007.

[50] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[51] Ralph Linsker. An application of the principle of maximum information preservation to linear systems. In *Advances in neural information processing systems*, pages 186–194, 1989.

[52] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[53] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124, 2019.

[54] John Locke. *An Essay Concerning Human Understanding*, volume 2. Oxford University Press, 1689.

[55] James Lucas, George Tucker, Roger B Grosse, and Mohammad Norouzi. Don' t blame the elbo! a linear vae perspective on posterior collapse. In *Advances in Neural Information Processing Systems*, pages 9403–9413, 2019.

[56] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[57] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[58] Emile Mathieu, Tom Rainforth, Siddharth Narayanaswamy, and Yee Whye Teh. Disentangling disentanglement. *arXiv preprint arXiv:1812.02833*, 2018.

[59] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[60] Sebastian Mika, Bernhard Schölkopf, Alex J Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel pca and de-noising in feature spaces. In *Advances in neural information processing systems*, pages 536–542, 1999.

[61] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[62] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[63] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[64] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

[65] Marc'Aurelio Ranzato, Christopher Poultney, Sumit Chopra, and Yann L Cun. Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems*, pages 1137–1144, 2007.

[66] Salah Rifai, Grégoire Mesnil, Pascal Vincent, Xavier Muller, Yoshua Bengio, Yann Dauphin, and Xavier Glorot. Higher order contractive auto-encoder. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 645–660. Springer, 2011.

[67] D. E. Rumelhart and J. L. McClelland. *Learning Internal Representations by Error Propagation*, pages 318–362. 1987.

[68] Andrew Michael Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan Daniel Tracey, and David Daniel Cox. On the information bottleneck theory of deep learning. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=ry_WPG-A-`.

[69] Andrew Michael Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan Daniel Tracey, and David Daniel Cox. On the information bottleneck theory of deep learning. 2018.

[70] Anna Sepliarskaia, Julia Kiseleva, and Maarten de Rijkee. Evaluating disentangled representations. *arXiv preprint arXiv:1910.05587*, 2020.

[71] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[72] Noam Slonim and Naftali Tishby. Agglomerative information bottleneck. In *Advances in neural information processing systems*, pages 617–623, 2000.

[73] Alexander J Smola and Risi Kondor. Kernels and regularization on graphs. In *Learning theory and kernel machines*, pages 144–158. Springer, 2003.

[74] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Advances in neural information processing systems*, pages 3738–3746, 2016.

[75] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[76] DJ Strouse and David J Schwab. The deterministic information bottleneck. *Neural computation*, 29 (6):1611–1630, 2017.

[77] Michael E Tipping. Bayesian inference: An introduction to principles and practice in machine learning. In *Summer School on Machine Learning*, pages 41–62. Springer, 2003.

[78] Michael E Tipping and Christopher M Bishop. Mixtures of probabilistic principal component analysers. 1998.

[79] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[80] Naftali Tishby and Noam Slonim. Data clustering by markovian relaxation and the information bottleneck method. In *Advances in neural information processing systems*, pages 640–646, 2001.

[81] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.

[82] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[83] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[84] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[85] Sida Wang and Christopher Manning. Fast dropout training. In *international conference on machine learning*, pages 118–126, 2013.

[86] Xin Wang, Shinji Takaki, Junichi Yamagishi, Simon King, and Keiichi Tokuda. A vector quantized variational autoencoder (vq-vae) autoregressive neural f_0 model for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:157–170, 2019.

[87] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[88] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[89] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.