Machine Learning
https://doi.org/10.1007/s10994-023-06340-x

A moment-matching metric for latent variable generative models

Cédric Beaulac[1]

Received: 20 September 2022 / Revised: 7 March 2023 / Accepted: 9 April 2023

Abstract
It is difficult to assess the quality of a fitted model in unsupervised learning problems. Latent variable models, such as variational autoencoders and Gaussian mixture models, are often trained with likelihood-based approaches. In the scope of Goodhart's law, when a metric becomes a target it ceases to be a good metric and therefore we should not use likelihood to assess the quality of the fit of these models. The solution we propose is a new metric for model comparison or regularization that relies on moments. The key idea is to study the difference between the data moments and the model moments using a matrix norm, such as the Frobenius norm. We first show how to use this new metric for model comparison and then for regularization. We show that our proposed metric is faster to compute and has a smaller variance than the commonly used procedure of drawing samples from the fitted distribution. We conclude this article with a demonstration of both applications and we discuss our findings and future work.

Keywords Moment estimators · Latent variable models · Goodness-of-fit · Frobenius norm

1 Introduction

When fitting supervised models, statisticians and computer scientists alike have come up with a variety of metrics in order to evaluate the quality of their predictions, from the simple mean-squared error to more general loss functions. However, in the context of unsupervised learning there is no direct measure of success and it can be difficult to assess the validity of the fitted model (Hastie et al., 2009).

Assume an unsupervised learning context where we have a data set $S = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and the abstract goal of capturing the distribution $p(\mathbf{x})$. One way to train such model is to assume a distribution and then maximize the likelihood of the data set with respect to the parameters of the distribution. If multiple models are trained, they are usually compared using the

Editors: Nathalie Japkowicz.

Cédric Beaulac
beaulac.cedric@gmail.com

[1] Département de mathématiques, Université du Québec à Montréal, Montréal, QC, Canada

Published online: 12 June 2023

likelihood as well. Goodhart's law (Goodhart, 1984; Strathern, 1997) states that *when a measure becomes a target, it ceases to be a good measure* and thus we should not strictly rely on the likelihood to evaluate models that were trained with the likelihood.

In this article we propose a new way to assess the quality of the fit of a large family of unsupervised models with respect to our abstract goal of capturing the distribution $p(\mathbf{x})$. In other words, we propose a new way to measure if an estimated distribution $\hat{p}(\mathbf{x})$ resembles the observed distribution $p_S(\mathbf{x})$. More precisely, we offer a diagnostic technique for parametric latent variable models such as Variational AutoEncoders (VAEs) (Kingma & Welling, 2014; Kingma, 2017) and Gaussian Mixture Models (GMMs). Our proposed metric evaluates the quality of the *fitted model*; we compare the learned parameters of the unsupervised model with the observed data distribution the model is trying to capture. We do so by building distinct moment estimators and comparing them. The main purpose of such metric is to provide a new way to compare the fit of multiple models from different families by assessing how well these models captured the first two moments of the data. Though capturing the first and second moment of a data set is not a sufficient condition to claim the trained model has perfectly captured the data distribution, it certainly is a necessary condition under some assumption we discuss later.

In statistics and machine learning, Goodhart's law is often compared with the concept of overfitting. One popular way to circumvent model overfitting has been regularization. Consequently, we offer a second perspective on our new metric; it can be used for regularization. Our metric favours simple models and thus it can be easily integrated in the optimization procedure as a regularizer.

The technique we propose is fast to compute, works for a wide range of models and is built upon a rigorous mathematical formulation. It provides a new way to compare multiple models or regularize them and behaves similarly to previously used heuristic techniques.

In the next section, we establish the family of latent variable models that this metric was designed for. We then discuss related work in Sect. 3 and we introduced the moment estimators used in Sect. 4. In Sect. 5, we present our metric and its implementation and next we demonstrate how it performs for model evaluation on simple examples in Sect. 6. We then introduce the framework for the application of our metric for regularization in Sect. 7 and we demonstrate how it behaves as regularizer in Sect. 8. Finally, we discuss the limitations of our approach in Sect. 9 before some concluding remarks in Sect. 10.

2 Latent variable generative models

Let us first define what we refer to as latent variable generative models (LVGMs). Assume we have a data set $S = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ consisting of N observations of a D-dimensional variable $\mathbf{x}$ where $\mathbf{x} \in \mathcal{X}$ which is D-dimensional. We want to estimate the distribution of the random variable $\mathbf{x}$ but it is *too complicated* to be captured by a simple distribution. Latent variable models suppose there exist an unobserved latent variable, say $\mathbf{z}$, that has a direct influence on the distribution of $\mathbf{x}$

$$p(\mathbf{x}) = \int_z p(\mathbf{x}||\mathbf{z})p(\mathbf{z})dz, \tag{1}$$

where we assume $\mathbf{z} \in \mathcal{Z}$ which is M-dimensional. The model proposed by Eq. (1) is quite general but allows relatively complex marginal distributions over observed variables $\mathbf{x}$ to be expressed in terms of more tractable conditional distributions $p(\mathbf{x}\|\mathbf{z})$ (Bishop, 2007). Similarly, it leads to a tractable joint distribution as well

$$p(\mathbf{x},\mathbf{z}) = p(\mathbf{z})p(\mathbf{x}\|\mathbf{z}), \tag{2}$$

and this is quite often represented using a simple graph as seen in Fig. 1.

These models are *generative models* because learning $p(\mathbf{x},\mathbf{z}) = p(\mathbf{z})p(\mathbf{x}\|\mathbf{z})$ allows us to generate new samples of $\mathbf{x}$ using ancestral sampling. What makes this model probabilistic is that the mapping from $\mathbf{z}$ to $\mathbf{x}$ is not a deterministic function $f : \mathcal{Z} \rightarrow \mathcal{X}$ but instead a *probabilistic mapping* from $\mathcal{Z}$ to Θ, where Θ is the parameter space of $p_\theta(\mathbf{x}\|\mathbf{z})$; $\theta \in \Theta$. We call $p_\theta(\mathbf{x}\|\mathbf{z})$ the emission distribution or observation distribution interchangeably.

When training or fitting such models, we train the function $f : \mathcal{Z} \rightarrow \Theta$ to maximize the likelihood of the data set S under the model of Eq. (1). This mapping f explains the effect of $\mathbf{z}$ on $\mathbf{x}$ and is at the centre of latent variable models. Therefore learning this function f is the main challenge of training latent variable models and the Expectation-Maximization algorithm (EM) or variational inference are common solutions to this problem. In most cases, $p(\mathbf{z})$ is assumed to be known and fixed but in some cases the parameters of $p(\mathbf{z})$ are estimated as well.

Usually $p_\theta(\mathbf{x}\|\mathbf{z})$ is a simple parametric distribution and the latent variable increases the complexity of $p_\theta(\mathbf{x})$. Additionally, the function f can take many forms, from simple linear combination to neural network functions. We use $f(\mathbf{z})$ interchangeably with f or the *distribution parameters* it outputs directly.

Let us begin with a simple example. Assume the emission distribution is Poisson: $p_\theta(\mathbf{x}\|\mathbf{z}) = \text{Poisson}(\lambda)$ then $f : \mathcal{Z} \rightarrow \mathbb{R}^+$ because $\theta = \lambda \in \mathbb{R}^+$ and we use $f(\mathbf{z})$ and $\lambda(\mathbf{z})$ interchangeably. If there exist a simple mapping from the parameters of the distribution to its expectation and its variance, we also use them interchangeably. For the Poisson example, $\mathbf{E}_x[\mathbf{x}\|\mathbf{z}] = \lambda(\mathbf{z})$ and $\mathbf{Var}_x[\mathbf{x}\|\mathbf{z}] = \lambda(\mathbf{z})$.

One important detail to bring up is that the moments are only meaningful for a certain family of emission distributions. For the application of our metric, we will consider the family of emission distribution for which the moment generating function exists.

2.1 Probabilistic principal component analysis

The Probabilistic Principal Component Analysis (pPCA) (Tipping & Bishop, 1999; Bishop, 2007) is a member of the LVGM family we just described where $p(\mathbf{z})$ is assumed to be a normal distribution $N(0, I)$. The emission distribution is also assumed to be Normal: $p(\mathbf{x}\|\mathbf{z}) = N(W\mathbf{z} + b, I\sigma^2)$. In this formulation, we see that $f : \mathbb{R}_M \rightarrow \mathbb{R}_D$ is a linear function that maps the latent variable $\mathbf{z}$ to $\mathbf{E}_x[\mathbf{x}\|\mathbf{z}]$: $\mathbf{E}_x[\mathbf{x}\|\mathbf{z}] = \mu(\mathbf{z}) = W\mathbf{z} + b$. Outside of estimating W and b as part of f, the model also estimates the parameter σ^2 though it is not a function of $\mathbf{z}$. However it is a function of D and M, the dimension of $\mathcal{X}$ and $\mathcal{Z}$.

The parameters of pPCA can be obtained analytically as the solution of a direct maximization of the likelihood or with the EM algorithm.

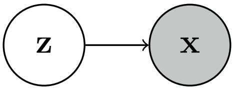


Fig. 1 Graphical representation of latent variables models with joint distribution $p(\mathbf{x},\mathbf{z}) = p(\mathbf{z})p(\mathbf{x}\|\mathbf{z})$

2.2 Variational autoencoders

The VAE is also a member of the LVGM family. It is assumed that $\mathbf{z}$ is a continuous variable where $p(\mathbf{z})$ is assumed to be $N(0, I)$ in the introductory papers (Kingma & Welling, 2014; Kingma, 2017). $p(\mathbf{x}\|\mathbf{z})$ can be any parametric distribution where $f(\mathbf{z})$ outputs the parameters of this distribution. For instance, if $p(\mathbf{x}\|\mathbf{z})$ is normal then $f(\mathbf{z})$ will output a mean and a variance parameter, $f : \mathbb{R}_M \rightarrow \mathbb{R}_D \times \mathbb{R}^+_{(D,D)}$.

One novelty of VAEs is that the function f proposed is much more flexible than a linear combination; it is a neural network. In turn, this makes the posterior distribution $p(\mathbf{z}\|\mathbf{x})$ intractable and prevents the model from being fitted by the EM algorithm. The solution proposed is to assume a variational distribution $q(\mathbf{z}\|\mathbf{x})$ and optimize the likelihood by maximizing the Evidence Lower BOund (ELBO) (Bishop, 2007; Kingma & Welling, 2014), a lower bound of the observed-data log-likelihood.

2.3 Gaussian mixture models

For a GMM with K-components we define $\mathbf{z}$ as a K-class categorical variable with $\mathbf{z} \in \{1, \ldots, K\} = \mathcal{Z}$ and $p(\mathbf{z})$, a categorical distribution where $\pi_j = p(\mathbf{z} = j)$ and $\sum_{j=1}^{K} \pi_j = 1$. Finally, setting $p(\mathbf{x}\|\mathbf{z} = j) = N(\mu_j, \Sigma_j)$ leads to a GMM:

$$p(\mathbf{x}) = \sum_{j=1}^{K} \pi_j p(\mathbf{x}\|\mathbf{z} = j). \tag{3}$$

In this situation, f maps the latent variable to a pair of *distribution parameters*, μ and Σ, $f : \{1, \ldots, K\} \rightarrow \mathbb{R}_D \times \mathbb{R}^+_{D\times D}$. In this particular case $\mathbf{E}_x[\mathbf{x}\|\mathbf{z}] = f_1(\mathbf{z})$ where f_1 is the *first* output of $f(\mathbf{z})$ or simply $\mu(\mathbf{z})$ and $\mathbf{Var}_x[\mathbf{x}\|\mathbf{z}] = f_2(\mathbf{z}) = \Sigma(\mathbf{z})$.

The GMM is a special case of LVGM where we also estimate the parameters $\{\pi_j : j \in 1, \ldots K\}$ of $p(\mathbf{z})$, identifiable up to a permutation. A GMM is usually trained with the EM algorithm.

3 Related works

In this article we propose a new metric to evaluate the goodness-of-fit for the family of latent variable models defined in Sect. 2 and in this section we discuss some common alternatives. When proposing new LVGMs researchers rely either on the likelihood of the data under the fitted model or a heuristic analysis of generated data points (Kingma & Welling, 2014; Li et al., 2015; Higgins et al., 2017; Zhao et al., 2019; Vahdat & Kautz, 2020). To evaluate the performance of the models, both those techniques have their fair share of problems which we discuss in this section. The metric we propose is an alternative to those techniques.

A problem with evaluating models with the likelihood is that a high likelihood does not necessarily mean that the proposed model captured the distribution of the observed data. For instance, Bishop (2007) demonstrate that for GMMs it is possible to have the likelihood be infinite (∞) by setting the mean of one component to be exactly one of the observed points, say $\mathbf{x}_i$, and then pushing the variance of that component to 0, thus the likelihood of $\mathbf{x}_i$ under that particular component will be infinite. Similarly, Zhao et al. (2019) built a toy example where the ELBO (a lower bound of the log-likelihood) would converge to infinity.

Another commonly employed strategy is to generate new observations and try to determine if they look like real data with a simple visual inspection, this is a technique used by many authors (Kingma & Welling, 2014; Li et al., 2015; Higgins et al., 2017; Kingma, 2017; Zhao et al., 2019; Vahdat & Kautz, 2020). A weakness of visual inspection is that it is subjective, it incentivizes cherry-picking of results and it is not a rigorous means of comparing models.

Outside of papers that introduced new LVGMs, some techniques have been developed to compared samples from different distribution and to assess goodness-of-fit using kernels. The Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) is a measure of similarities between two samples which the authors used to construct statistical tests to determine if the two samples are drawn from different distributions. Unfortunately, this requires to sample from the fitted distribution, which we later demonstrate to be a slow procedure and the original implementation proposed by the authors is no longer available online. The Kernel Stein Discrepancy test (Liu et al., 2016) is a goodness-of-fit test that was later extended specifically for latent variable models (Kanagawa et al., 2019; Jitkrittum et al., 2020). Both of these Kernel Stein Discrepancy goodness-of-fit tests have strong theoretical guarantees. Unfortunately, these guarantees only hold under strict assumptions that can be complicated to verify. Additionally, the magnitude of this discrepancy metric is not only affected by the evaluated models by also the choice of reproducing kernel (Kanagawa et al., 2019). On the opposite, our proposed metric works for any LVGMs that fit the definition of Sect. 2 and is not affected by arbitrary choices.

Comparing data moments with model moments have been proposed in the past but only for training purposes (Anandkumar et al., 2012, 2014; Chaganty & Liang, 2014; Li et al., 2015; Podosinnikova, 2016). Anandkumar (Anandkumar et al., 2012, 2014) proposes efficient ways to code and optimize latent variable models by comparing the true data set with a sample generated from the model. Podosinnikova (Podosinnikova, 2016) approaches this topic very thoroughly in their thesis where they also discuss the use of moment-generating function estimators. What we propose here is different; we propose a metric. Even though we discuss the possibility of using our metric for optimization in later sections we believe there already exists a rich literature that discusses new ways of optimizing latent variable models but few publications that address the lack of evaluation metrics.

4 Moment estimators

In this section we define two different moment estimators for the first and the second moment. The goal is to build different estimators containing different information. To begin, we define moment estimators of the data set, we call those Data Estimators (DE). Then we define another set of moment estimators that represent the distribution captured by the LVGM which we call Forward Model Estimators (FME).

4.1 Second moment

We build two different estimators of the same quantity, the second moment. One uses observed data while the other uses the proposed generative model. What makes the proposed FME different is that we do not sample new data points from the LVGM but instead rely on a simple probability identity to build the FME.

To do so, let us introduce the well-know Law of Total Variance

$$\mathbf{Var}_x(\mathbf{x}) = \mathbf{E}_z[\mathbf{Var}_x(\mathbf{x}\|\mathbf{z})] + \mathbf{Var}_z[\mathbf{E}_x(\mathbf{x}\|\mathbf{z})], \tag{4}$$

and notice that

$$\begin{aligned}\mathbf{Var}_z[\mathbf{E}_x(\mathbf{x}\|\mathbf{z})] &= \mathbf{E}_z[\mathbf{E}_x(\mathbf{x}\|\mathbf{z})^2] - (\mathbf{E}_z[\mathbf{E}_x(\mathbf{x}\|\mathbf{z})])^2 \\ &= \mathbf{E}_z[\mathbf{E}_x(\mathbf{x}\|\mathbf{z})^2] - (\mathbf{E}_x[\mathbf{x}])^2.\end{aligned} \tag{5}$$

We combine and reorganize both equations

$$\mathbf{Var}_x(\mathbf{x}) + (\mathbf{E}_x[\mathbf{x}])^2 = \mathbf{E}_z[\mathbf{Var}_x(\mathbf{x}\|\mathbf{z})] + \mathbf{E}_z[\mathbf{E}_x(\mathbf{x}\|\mathbf{z})^2]. \tag{6}$$

We have reorganized both terms in this particular was so that the left-hand side of Eq. (6) is independent of the latent variables and can be estimated from S independently from the choice of model while the right-hand side contains information about both the expectation and the variance of the generative model. Additionally, notice the left-hand side is actually $\mathbf{E}_x[\mathbf{x}^2]$, the second moment of $\mathbf{x}$ and thus our work here consists of comparing two different estimators of $\mathbf{E}_x[\mathbf{x}^2]$ which we introduce next. The left-hand side of Eq. (6) can be estimated using the observed data

$$\mathbf{Var}_x(\mathbf{x}) + (\mathbf{E}_x[\mathbf{x}])^2 \approx \frac{\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})^T(\mathbf{x}_i - \bar{x})}{n-1} + \bar{\mathbf{x}}^T\bar{\mathbf{x}} := \mathrm{DE}, \tag{7}$$

where $\bar{\mathbf{x}}$ is the mean vector. The right-hand side of Eq. (6) can be estimated using the proposed generative model with a Monte Carlo sample from $p_\theta(\mathbf{z})$ and using both $\mathbf{Var}_x(\mathbf{x}\|\mathbf{z})$ and $\mathbf{E}_x(\mathbf{x}\|\mathbf{z})$.

$$\begin{aligned}\mathbf{E}_z[\mathbf{Var}_x(\mathbf{x}\|\mathbf{z}) + \mathbf{E}_x(\mathbf{x}\|\mathbf{z})^2] &= \int_z (\mathbf{Var}_x(\mathbf{x}\|\mathbf{z}=z) + \mathbf{E}_x(\mathbf{x}\|\mathbf{z}=z)^2)p(z)dz \\ &\approx \frac{1}{m}\sum_{i=1}^{m}\left[\mathbf{Var}_x(\mathbf{x}\|\mathbf{z}=z_i) + \mathbf{E}_x(\mathbf{x}\|\mathbf{z}=z_i)^T\mathbf{E}_x(\mathbf{x}\|\mathbf{z}=z_i)\right] \\ &:= \mathrm{FME},\end{aligned} \tag{8}$$

where $z_i \sim p(\mathbf{z})$, and $\mathbf{E}_x(\mathbf{x}\|\mathbf{z}=z_i)$ and $\mathbf{Var}_x(\mathbf{x}\|\mathbf{z}=z_i)$ are expressed as functions f. Consequently this estimator relies on both components of the fitted LVGM: $p(\mathbf{z})$ and $f(\mathbf{z})$. This is the forward model estimate (FME). Notice that this estimator does not require we sample from $p_\theta(\mathbf{x}\|\mathbf{z})$ and directly uses the estimated parameters of the emission distribution. It is faster to sample a large amount of $\mathbf{z}$ than sample a large amount of $\mathbf{x}$ because traditionally $M << D$. Additionally, this is a simple Monte Carlo sample and it is unbiased(Rosenthal, 2019).

Given a sample of $\mathbf{E}_x(\mathbf{x}\|\mathbf{z}=z_i)$ and $\mathbf{Var}_x(\mathbf{x}\|\mathbf{z}=z_i)$, it takes D^2+m operations to compute the FME. Based on Shalev-Shwartz and Ben-David's definition of efficiency (Shalev-Shwartz & Ben-David, 2014), computing the FME would be considered efficient with respect to D and m as the number of operations required is a polynomial function of those parameters, opposed to an exponential function which would be considered inefficient.

Thus it follows from Eq. (6) that

$$(\mathbf{Var}_x(\mathbf{x}) + (\mathbf{E}_x[\mathbf{x}])^2) - (\mathbf{E}_z[\mathbf{Var}_x(\mathbf{x}\|\mathbf{z})] + \mathbf{E}_z[\mathbf{E}_x(\mathbf{x}\|\mathbf{z})^2]) = 0, \tag{9}$$

and since both the DE and the FME are unbiased estimators of the second moment then consequently, the gap between those two estimators reflects how the LVGM captured the second moment of the data set; the bigger the gap is, the poorer the fit is. Thus, we propose to analyse the following moment estimator gap

$$\text{DE} - \text{FME} = 2\text{MEGA}, \tag{10}$$

which is a matrix of dimension $D \times D$.

4.2 First moment

Similarly, for the first moment we have

$$\mathbf{E}_x[\mathbf{x}] = \mathbf{E}_z[\mathbf{E}_x(\mathbf{x}\|\mathbf{z})], \tag{11}$$

where we estimate the left-hand side with $\bar{\mathbf{x}}$ (DE) and the right-hand side with $\frac{1}{m}\sum_{i=1}^{m}\mathbf{E}_x(\mathbf{x}\|\mathbf{z} = z_i)$ where $z_i \sim p_\theta(\mathbf{z})$ (FME). At this moment, we proposed looking at the gap between moment estimators for both the first and the second moment. However, with machine learning models being mostly optimized towards point estimation, the first moment estimator gap is usually less interesting than its second moment counterpart.

4.3 Additional justifications

For any generative models, it is always possible to generate a new sample of G points say $S_{LVGM} = \{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_G\}$ and a simple model estimator of the first and second moment would be to simply compute $\frac{1}{G}\sum_{i=1}^{G}(\tilde{\mathbf{x}}_i)$ for the first moment and similarly $\frac{1}{G}\sum_{i=1}^{G}(\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i^T)$ for the second moment. Let us call these *sample estimators* (SE). These estimators could replace both FMEs defined previously as they also reflect the distribution learned by the LVGM. Conceptually, the SEs are simple and easy to use and metrics such as the MMD relies on additional samples; thus we want to justify why our estimators (FMEs) are *better*.

To begin, FMEs are faster to compute. In order to draw the same sample size for the Monte Carlo estimates, say a sample of size m, we need to sample m times for the FMEs. However, for the SEs have to draw twice as many samples ($2m$), since we must sample from $p(\mathbf{z})$ m times and then from $p(\mathbf{x}\|\mathbf{z})$ an additional m times. More importantly, because $M << D$ it means that not only FMEs require half as many samples, but these samples are from a much lower dimension distribution which further increases the difference in computational cost between the FMEs and the SEs.

Another reason why we prefer FMEs over SEs is that FMEs have a smaller variance for both the first and the second moment. Both FMEs and SEs are unbiased estimators of the same value, but the lower variance of FMEs is a huge benefit. A complete proof is located in the appendices.

Finally, we demonstrated with a simulation that over multiple Monte Carlo samples sizes, the FMEs are closer to the true LVGM moments than SEs. For simple models, such as GMMs, we can compute analytically $\mathbf{E}_z[\mathbf{Var}_x(\mathbf{x}\|\mathbf{z}) + \mathbf{E}_x(\mathbf{x}\|\mathbf{z})^2]$, the LVGM second moment. We compared the gap between the true LVGM second moment and the FME to the gap between the true LVGM second moment and the SE and plotted both against m the number of Monte Carlo

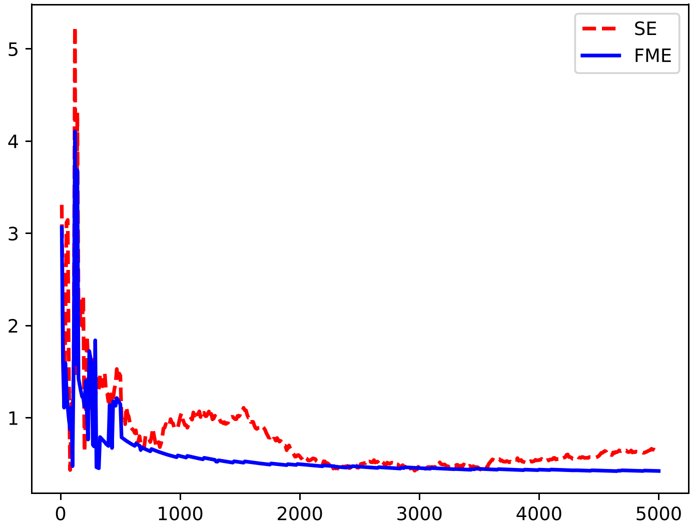


Fig. 2 The evolution of both gaps plotted against m. These gaps are computed using the Frobenius norm as justified in Sect. 5

samples. We can see in Fig. 2 that the estimator we proposed (FME) is overall closer to the true LVGM second moment than the SE while being faster to compute.

5 MEGA: a new metric for comparing LVGMs

For latent variable models as defined in Sect. 2 we assess the ability of the model to capture the moment of the data set S by comparing the DE with the FME. Since we are looking at the difference between two different moment estimators of the same value, we named this metric the Moment Estimators Gap (MEGA). We study the difference between our two second moment estimators and we refer to this matrix as 2MEGA. Similarly, we refer to the vector representing the first moment estimator gap as 1MEGA.

5.1 Selecting a matrix norm

In order to make the MEGA tangible and comparable, we propose to use a matrix norm of the MEGA as our metric. There exist a wide range of possible candidates. Let us introduce a few and justify our finale choice.

We want to use a norm that looks at the global properties of the matrix and fortunately Rigollet (Rigollet & Hütter, 2015) introduces and studies the behaviour of well-established matrix norms. To begin, we use norms inspired by vector norms. Given the vector v, assume $||v||_q$ is the following vector norm

$$||v||_q = \left(\sum_i ||v_i||^q \right)^{(1/q)}, \tag{12}$$

and given the matrix M, its matrix equivalent is $||M||_q$

$$||M||_q = \left(\sum_{ij} ||M_{ij}||^q\right)^{(1/q)}. \tag{13}$$

When $q = 2$, this is a special case call the Frobenius norm

$$||M||_2 = ||M||_F = \left(\sum_{ij} ||M_{ij}||^2\right)^{(1/2)} = \sqrt{\mathrm{Tr}(M^T M)}, \tag{14}$$

where Tr is the trace operator that sums the elements of the diagonal of the input matrix.

This norm is also a member of the Schatten q-norms (for $q = 2$) which is a family of matrix norms defined as the vector norm of Eq. 12 for the singular values of the matrix. Since we work with second moment estimators, our matrix M is a squared matrix (dimension $m \times m$) and symmetric and thus the singular values of M are equal to its eigenvalues. We identify the vector of eigenvalues as λ. Consequently, the Schatten q-norm for matrix M is $|| \, ||M|| \, ||_q = ||\lambda||_q$.

Another member of this family is considered, when $q = \infty$ we define $|| \, ||M|| \, ||_\infty = \lambda_{max} = || \, ||M|| \, ||_{op}$ and this is referred to as the operator norm.

In random matrix theory and in matrix estimation, these norms appear frequently and are consequently well known in the statistical research community. For instance, in covariance matrix estimation it is possible, under *mild* assumptions, to bound the operator norm of the difference between the true covariance matrix and simple estimators (Rigollet & Hütter, 2015). Because of the popularity and the known properties of both the Frobenius norm and the operator norm, they are both legitimate options to measure the MEGA.

The bigger the 2MEGA is, the further away our model second moment is from the data second moment. However, one can perceive the Frobenius norm as the length of the hypotenuse of a multidimensional triangle whose sides are given by the eigenvectors of the matrix while the operator norm is the length of the longest cathetus of the same multidimensional triangle. For computational reasons expressed in the next section, we selected the Frobenius norm and this will serve as our metric in the sections to come. In other words, the metric we propose to evaluate the quality of the fit for the second moment is

$$\text{2MEGA-F} := ||\text{2MEGA}||_F. \tag{15}$$

Additionally, if we also consider 1MEGA, we can then again simply use the Frobenius norm (the vector q-norm with $q = 2$) on the vector 1MEGA

$$\text{1MEGA-F} := ||\text{1MEGA}||_2. \tag{16}$$

5.2 Implementation of the selected norm

Implementing the Frobenius norm is quite straight forward. To compute to Frobenius norm, we need to square the 2MEGA, which involves m^2 operations, and then sum its diagonal which results in $m^2 + m$ operations. The largest eigenvalue of a matrix can be estimated with the von Mises iteration (Mises & Pollaczek-Geiringer, 1929), where each of the p iterations will require m^2 operations resulting in pm^2 operations in total. Once again, based on Shalev-Shwartz and Ben-David's definition of efficiency (Shalev-Shwartz & Ben-David, 2014), computing the Frobenius norm of the MEGA would be considered efficient with respect to m. Because we can implement an exact computation of the Frobenius norm and because it is computationally faster than the operator norm (as soon as $p \geq 2$) we strictly consider the Frobenius norm for the rest of this article. However, there could be some merit to exploring the operator norm in future work.

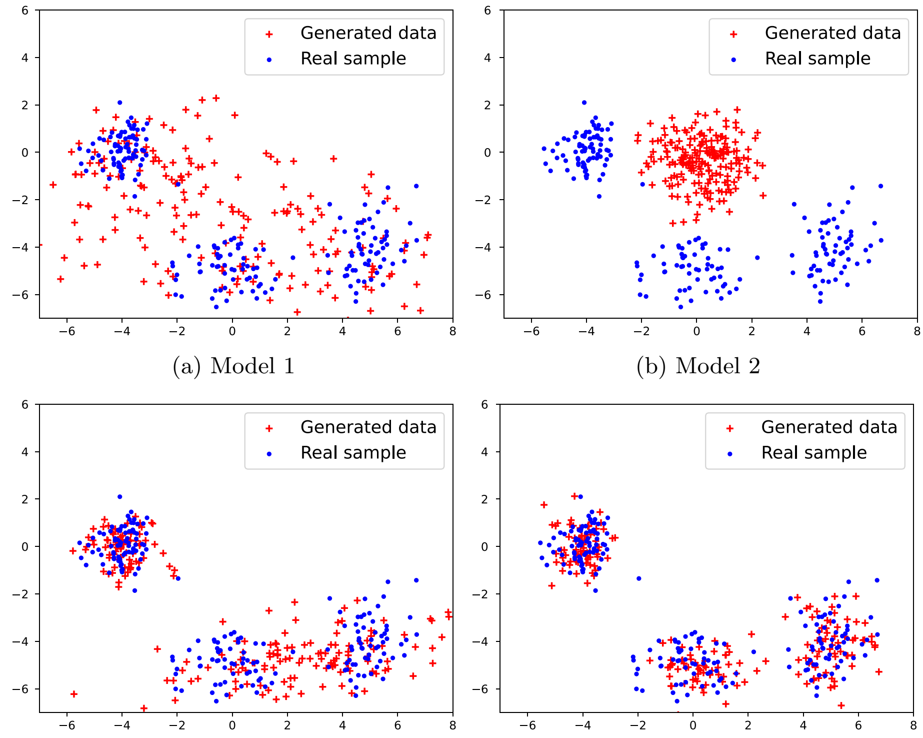
We implemented the vector Frobenius norm and the matrix Frobenius norm in Python using the NumPy library (Harris et al., 2020) and using the Pytoch library (Paszke et al., 2019). We also implemented a MEGA function that takes as input a sample of $\mathbf{Var}_x(\mathbf{x}\|\mathbf{z})$ and $\mathbf{E}_x(\mathbf{x}\|\mathbf{z})$ alongside the data set S and returns 1MEGA-F and 2MEGA-F. These implementations are publicly available on the author's GitHub (Beaulac, 2021).

6 Experiments: comparing models

Now that we established the metric, let us use it to compare distributions $\hat{p}(\mathbf{x})$ learned from different models.

For this demonstration we use simple 2-dimensional observations. We demonstrate that large MEGAs are associated with poor fit when visualizing generated data from LVGMs. We also show that it is concordant with the MMD in most cases. In other words, our metric is concordant with the currently used techniques while providing a new perspective.

Figure 3 contains both the training observations (in blue) and generated data points (in red) from four LVGMs. Table 1 lists the 1MEGA-F, the 2MEGA-F and the MMD for all



(a) Model 1

(b) Model 2

(c) Model 3

(d) Model 4

Fig. 3 Generated data from four different LVGM models trained on the three clusters data set

Table 1 MEGA and MMF for the four LVGM models trained on the three clusters data set

	1MEGA-F	2MEGA-F	MMD
Model 1	0.2704	2.0392	0.0812
Model 2	2.4222	4.5510	1.6717
Model 3	3.1166	3.1712	0.0069
Model 4	0.2974	1.7014	0.0012

four models. We notice first that the 2MEGA-F and the MMD agree on the worst performing model being model 2. A visual inspection of Fig. 3 does support that model 2 fits the data rather poorly. Similarly we notice that 2MEGA-F and MMD agree on the best performing model being model 4 which is also supported by a visual inspection of Fig. 3. We can also use both MEGA metrics to better understand specific problems with fitted models. For instance, even though a visual inspection of model 1 reveals a poor fit, a small 1MEGA-F tells us the problem is not coming from a poor estimation of the expectation of $\mathbf{x}$. As a matter of fact, the 2MEGA-F and the MMD are also rather small, indicating the bad fit of model 1 is not due to a poor estimation of the variance either. It is most likely caused by model 1 being unimodal, something that is not measured in the metric we currently propose.

Next, we propose another small demonstration with a second data set as illustrated in Fig. 4.

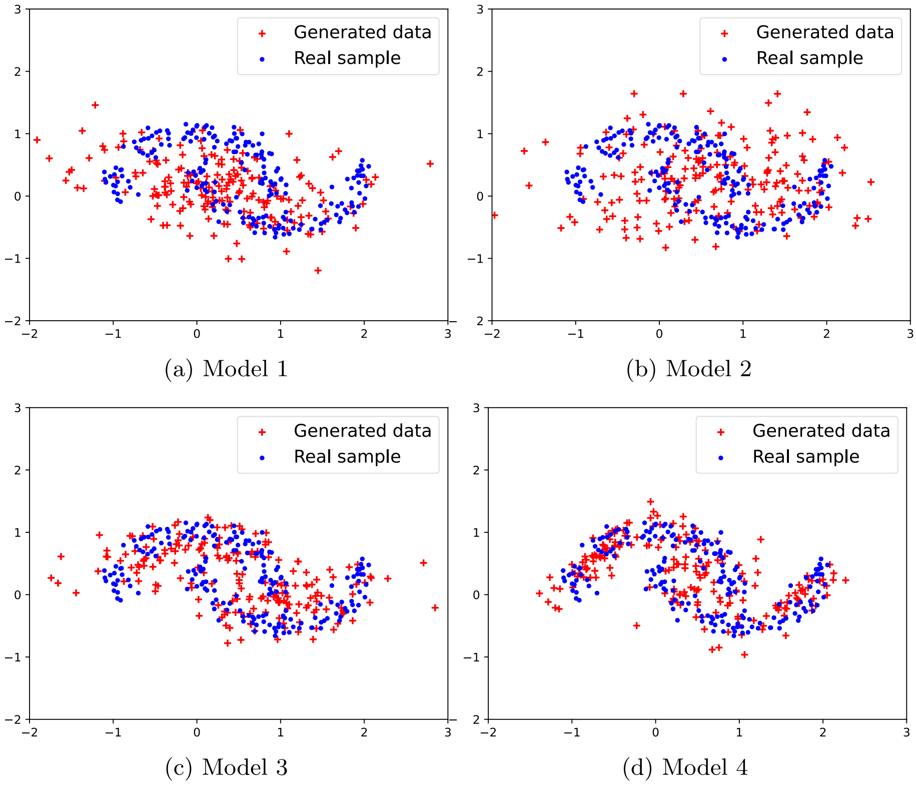
By visually inspecting Fig. 4, model 4 seems like a good contender for best fit. In Table 2, Model 4 has indeed the lowest 2MEGA-F, which demonstrates again that our proposed metric behaves somewhat intuitively. Additionally, this is concordant with the MMD for which model 4 is also the best model. Visually comparing model 1 and model 2 would be a difficult task since they both seem equally good or equally bad. However, the proposed metric and the MMD are in agreement that model 1 is worse than model 2. Finally, model 3 requires a bit more attention. Even though a visual inspection seems to be positive towards that model it has a rather large first moment estimator gap. Indeed, a closer examination of Fig. 4c reveals that the centre of mass of the fitted model is at the right end tip of the top half moon rather than in the centre of both half-moons. A careful examination of the metric we propose can help us identify good models, but also reveal aspects of fitted models that require more attention.

Finally, we provide an additional experiment using real data this time. In this last demonstration, we analyse the well-known MNIST data set (Lecun et al., 1998) in an unsupervised way. A sequence of GMMs was trained with $K = \{2, 4, 6, 8, 10, 12, 14\}$. We expect intuitively $K = 10$ to be a good choice, since the data contains 10 different digits.

In Table 3, we see that the MEGAs and the MMD agree that the models with 10 and 14 components provide the two best fits of the data. Moreover, our proposed metric favours the model with 10 components which is intuitively a good choice. A natural consequence of using a metric for model comparison is to use it for model selection. In this experiment, we would choose the model with 10 components. In Sect. 7, we explore that concept further by discussing the use of the proposed metrics for regularization.

7 MEGA for regularization

As we previously mentioned, based on the principle of Goodhart's Law, we proposed to use the proposed metrics 1MEGA-F and 2MEGA-F to compare complex LVGMs such as IAF-VAEs (Kingma et al., 2016) or NVAEs(Vahdat & Kautz, 2020). In Sect. 3 we discussed



(a) Model 1

(b) Model 2

(c) Model 3

(d) Model 4

Fig. 4 A closer examination of figure on the half-moon data set

Table 2 MEGA and MMD for the four LVGMs trained on the half-moon data set

	1MEGA-F	2MEGA-F	MMD
Model 1	0.2565	0.5633	0.0141
Model 2	0.0023	0.5248	0.0008
Model 3	0.7616	0.8126	7.4341e−05
Model 4	0.1072	0.4851	1.2092e−05

Table 3 MEGA and MMD for the sequence of GMM models trained on the MNIST data set

K	1MEGA-F	2MEGA-F	MMD
2	0.24997318	2.92840915	0.0131
4	0.26690068	3.03135908	0.0111
6	0.2578028	2.925411	0.0121
8	0.26219767	3.00054297	0.0108
10	**0.23850984**	**2.82369332**	0.0089
12	0.26362526	3.09359567	0.0104
14	0.24450328	2.92585433	**0.0084**

The number of components suggested by the metrics are in bold some issues with the likelihood as an evaluation metric in some special cases. Additionally, it is unfair to use likelihood as a metric to compare two models when one is trained using a likelihood approach and the other is not.

However, if this metric has some merit when comparing models, can we use it for model selection and can we directly integrate it in the optimization process ? We answer these questions in this section.

Because the likelihood and the moments reflect different aspects of the distribution $p(\mathbf{x})$ then incorporating MEGA as part of the optimization process for models trained by maximum likelihood acts as regularization.

This is easy to see for models with Gaussian emission distribution such as GMMs and VAEs. When fitting a single Gaussian distribution with maximum likelihood, we set its parameters μ and σ to the data mean and the data standard error and thus a single Gaussian performs very well according to the MEGAs. As we increase the number of components K in a GMM, we increase its likelihood but we also increase its MEGA. This means that we can use the MEGA as a model selection (or regularization metric) to fit GMM in place of alternatives such as AIC and BIC in order to get a model that balances likelihood and moment matching.

One way to do so is to select the model that maximizes:

$$\mathrm{ll}(S) - \alpha(\mathrm{1MEGA\text{-}F} + \sqrt{\mathrm{2MEGA\text{-}F}}), \tag{17}$$

where $\mathrm{ll}(S)$ is the log-likelihood of the data S under the fitted model and α is a hyper-parameter that interpolates between maximum-likelihood and moment estimator. Similar to what is produced when using Lasso (Tibshirani, 1996; Simon et al., 2011; Friedman et al., 2010) we can draw an entire regularization path (complexity selection path) by evaluating the regularized likelihood of Eq. 17 over different values of α.

Similarly, we can use a MEGA term in the objective function when training a VAE. Our proposed metric again serves as a regularizer in order to ensure the VAE model captures those moments by maximizing:

$$\mathbf{E}_q[\ln p(\mathbf{x}\|\mathbf{z})] - KL(q(\mathbf{z}\|\mathbf{x})\|p(\mathbf{z})) - \alpha(\mathrm{1MEGA\text{-}F} + \sqrt{\mathrm{2MEGA\text{-}F}}). \tag{18}$$

Arguments have been made in the past (Kingma, 2017; Higgins et al., 2017) that the KL divergence term serves as a regularizer and that adding a β parameter can provide a way to adjust the strength of the regularization. Adding a MEGA term to the ELBO should provide a different type of regularization. The KL divergence term provides regularization for the distribution $q(\mathbf{z}\|\mathbf{x})$ and the MEGA term provides regularization for $p(\mathbf{x})$:

$$\underset{\text{Reconstruction error}}{\mathbf{E}_q[\ln p(\mathbf{x}\|\mathbf{z})]} - \underset{\text{Regularization for } q(\mathbf{z}\|\mathbf{x})}{\beta KL(q(\mathbf{z}\|\mathbf{x})\|p(\mathbf{z}))} - \underset{\text{Regularization for } p(\mathbf{x})}{\alpha(\mathrm{1MEGA\text{-}F} + \sqrt{\mathrm{2MEGA\text{-}F}})}. \tag{19}$$

As previously demonstrated (Beaulac, 2021), tunning a VAE to obtain both good reconstruction and generative performances is difficult. For instance, when using the β-VAE (Higgins et al., 2017, 2018) we can make the hyper-parameter β arbitrary small and obtain close to perfect reconstruction but this leads to poor generative performance because the distribution $q(\mathbf{z})$ used for training would be significantly different from $p(\mathbf{z})$ used for generation. The additional term we add to the VAE objective function should help with generative perspectives of VAEs.

Thus, to understand the benefits of integrating this additional term to the objective function we have to understand how different it is from the first term. The first term measures the reconstruction error because the likelihood $p(\mathbf{x}\|\mathbf{z})$ is computed on latent variable $\mathbf{z}$ sampled from $q(\mathbf{z}\|\mathbf{x})$. In contrast, the MEGA penalty term evaluates the first and second moment of $p(\mathbf{x})$ independently from $q(\mathbf{z}\|\mathbf{x})$ but rather based on the generative model where $p(\mathbf{x}) = \int_z p(\mathbf{x}\|\mathbf{z})p(\mathbf{z})dz$. We believe this additional constraint in the objective function should lead to samples that matches the true distribution of $p(\mathbf{x})$ more closely.

8 Experiments: regularization

8.1 Regularizer for GMMs

For GMMs, increasing the number of components increases the likelihood and thus it is necessary to regularize this model; we cannot simply select the ideal number of components based on the likelihood alone. We have generated a simple data set from a 3-component GMM. Figure 5 is a scatter plot of the simulated data, by looking at the figure we would like the model selection procedure to settle on three components.

The Akaike Information Criteria (AIC) (Akaike, 1974, 1998) is a well-establish penalized likelihood function that can be used to select the number of components in a GMM. The AIC can be expressed as

$$\text{AIC} = 2p - 2\text{ll}(S), \tag{20}$$

where $\text{ll}(S)$ is the log-likelihood of the data S under the model, and p is the number of parameters ($p = 2K$ in the GMM case). In Eq. 20 we see a natural tradeoff; $-2\text{ll}(S)$ goes down as we increase the number of components but $2p$ goes up. To select the right GMM model, we can fit multiple GMM with various number of components and we select the model that minimizes the AIC.

Based on Fig. 6, the AIC is minimized at $K = 3$ which is concordant the model used to generate the data.

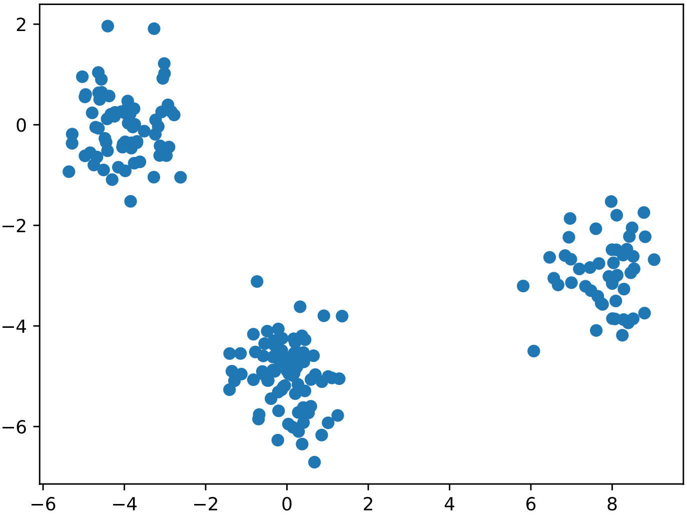


Fig. 5 Scatter plot of the simulated data set

Next, we use our MEGA-penalized likelihood function described in Eq. 17 to select the number of components.

We have computed the MEGA-regularized log-likelihood as expressed in Eq. 17 for different values of α. In Fig. 7, we plot the number of components of the model that maximizes the MEGA-regularized log-likelihood for different values of α. With small α, there is no regularization and the selected model has $K = 20$ components which is the highest complexity model fitted. With large values of α, the model with the smallest complexity (the model with $K = 1$ component) is selected. In between those two extremes may lie a sequence of nested models and the best one can be selected with the use of a validation set. In the demonstration, only one model is proposed between the two extreme cases, a model with 3 components which is concordant with both the AIC and the true generative model.

Compared to the AIC, the flexibility given by the hyper-parameter α is both a benefit and a drawback. Using the hyper-parameter, we can easily adjust how strong we want the penalty to be but on the flip side there are no automated ways to adjust it.

8.2 Regularized GMMs for anomaly detection

Because GMMs are popular models for unsupervised tasks, they have been used for anomaly and outlier detection for quite a while now (Zimek et al., 2012) and are still at the centre of the development of new models such as the deep autoencoding Gaussian mixture model (Zong et al., 2018). For outlier detection, we rely on the following procedure. First we fit a GMM to a data set and then point with likelihood lower than a predetermined threshold are considered outliers.

For this demonstration, we use the ionosphere data set (Sigillito et al., 1989). This data set consists of 33 variables and 351 observations. This data set has been previously used to test anomaly detection techniques (Liu et al., 2008; Keller et al., 2012) and 126 outliers have already been identified.

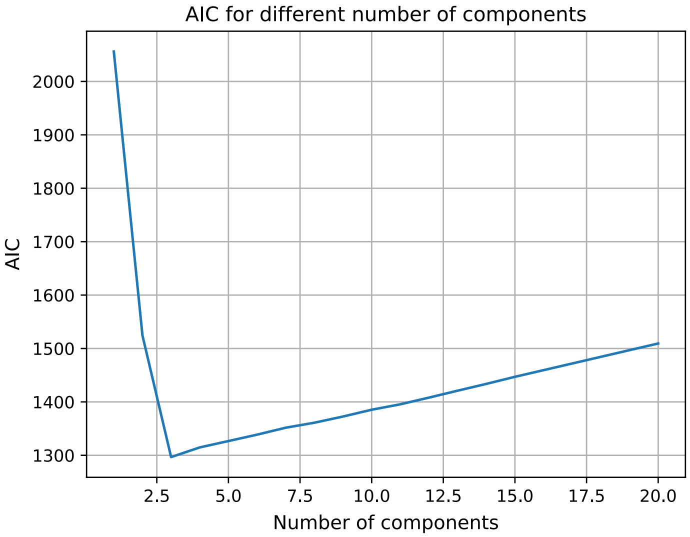


Fig. 6 AIC plotted against the number of components for trained GMM models (lower is better)

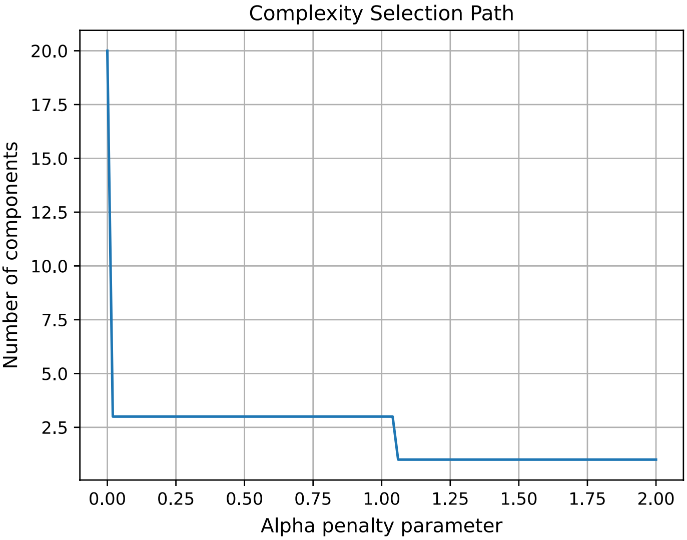


Fig. 7 Number of components of the selected model plotted against different values of α

In this experiment, we select the GMM that maximizes the MEGA-regularized likelihood of Eq. 17 and we then use this model for anomaly detection. For simplicity, we assumed we know there are about 35% of outliers, this allows us to easily select a threshold that labels the 35% lowest likelihood data to be outliers. This results in 123 points considered outliers.

Based on Figs. 8 and 9, the AIC suggests a GMM with 20 components and the MEGA-regularized likelihood suggests a model with 3 components. The model suggested by the AIC correctly identifies 67 outliers, thus missing 59 outliers and miss-labelling 56 observations as outliers. In contrast, the model selected using MEGA correctly identifies 103 outliers while miss-labelling 20 observations as outliers. This also means that this model missed 23 outliers.

8.3 Regularizer for VAEs

Finally we experiment with using MEGA as a regularization for VAEs. Now that we used MEGA as part of the objective function we can no longer use it to assess the quality of the samples so we will visually inspect samples and their parameters as well as rely on the Fréchet Inception Distance (FID) (Heusel et al., 2017).

We run this demonstration on a subset of the MNIST data set (Lecun et al., 1998) that contains only the digit four. We have experimented with a wide range of parameters α and β, as defined in Eq. 19 and fixed the hyper-parameters to the values leading to the more realistic-looking images.

Figures 10 and 11 illustrate our results. The images on in the left column were produced by a VAE trained without MEGA and the right one with MEGA. We have included images of a sample and its mean.

The images themselves are very difficult to analyse which is why we have incorporated Table 4 that contains the FID for both models. Even with the MEGA regularization, the samples are still grainy and this is due to the pixel-independence assumption

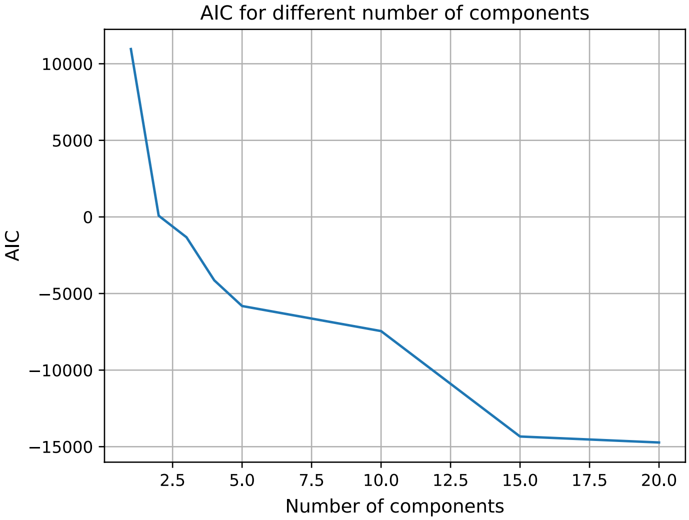


Fig. 8 AIC plotted against the number of components for trained GMM models (lower is better)

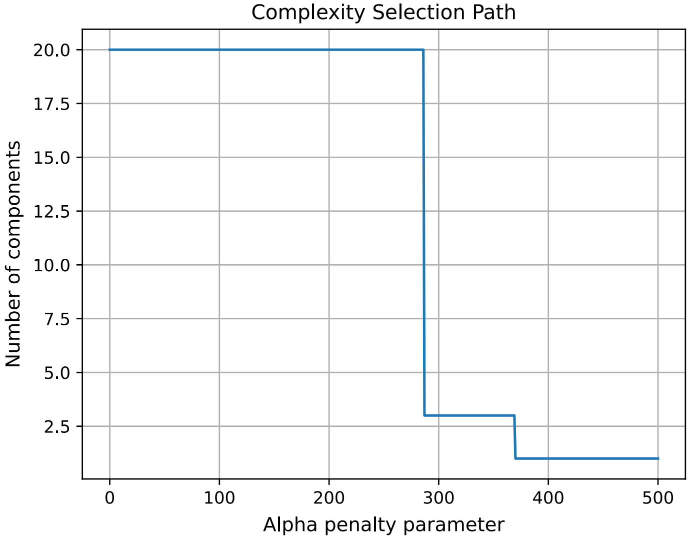
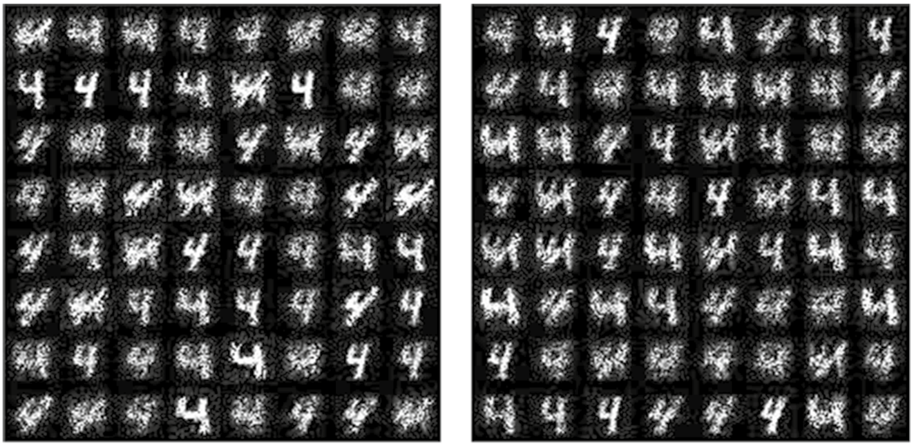


Fig. 9 Number of components of the selected model plotted against different values of α

of the simple VAE which only learns individual pixel variance and not a full-covariance matrix. This is why when data are generated with VAEs, it is common to simply sample $\mu(\mathbf{z})$.

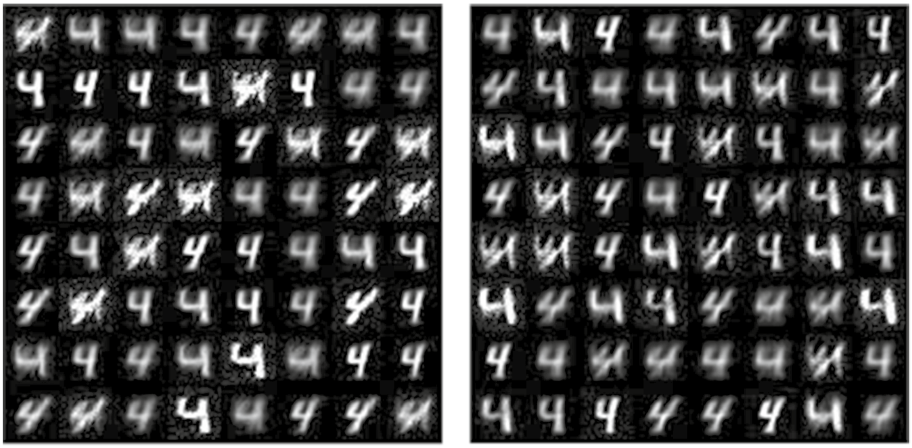
Based on the FID, the images produced with the MEGA-regularized VAE are more realistic. To better understand why that is, we also calculated the FID for the sample of means $\mu(\mathbf{z})$. Since those are pretty equivalent for both model, we conclude that the improvement in the image generated with the regularized VAE comes from the captured variance.



(a) Model train without MEGA

(b) Model train with MEGA

Fig. 10 A sample of 64 images from $p_\theta(\mathbf{x}\|\mathbf{z}) = N(\mu(\mathbf{z}), \sigma(\mathbf{z}))$ where $\mathbf{z} \sim N(0, 1)$



(a) Model train without MEGA

(b) Model train with MEGA

Fig. 11 The 64 sampled mean of the images: $\mu(\mathbf{z})$ where $\mathbf{z} \sim N(0, 1)$

Table 4 FID for VAE and regularized-VAE samples

	FID for $\mathbf{x}$	FID for $\mu(\mathbf{z})$
Without MEGA	94.9096	46.2973
With MEGA	58.3258	47.5723

In all of the other experiments, the MEGA was computed only a few times, which is extremely fast even on a single CPU. However, for this task, the MEGAs have to be computed for every mini-batch during training. Computing the MEGA over 25 epochs of 20 mini-batch each, for a $D = 784$ dimensional observation $\mathbf{x}$ using a Mont Carlo sample size of $m = 5000$ took 1246.0781 s on a single CPU (Ryzen 5 5600X). Because the computational time is a polynomial function of these 4 parameters, we believe MEGA can be used as a regularizer on more complex data sets and problems especially if those are trained on clusters of powerful computers.

9 Discussion

As a comparative metric, MEGA is fast to compute and easy to interpret. The larger 1MEGA is, the larger the gap is between the first moment of the trained distribution $\hat{p}(\mathbf{x})$ and the empirical first moment of the data set S. Similarly, the larger 2MEGA is, the larger the gap is for the second moment. This can give us a quick and easy way to compare and evaluate the quality of fitted models.

However, there are still some limitations to this approach. The most obvious is that the formulation we propose only allow us to quickly evaluate the gap for the first two moments. This leads to an incomplete comparison of the learned and empirical distribution which can create some problems in niche cases. A example of this problem is the high performance of the simple Gaussian distribution. Usually, when fitting a Gaussian distribution to a data set we set the parameters μ and σ to be the empirical mean and the empirical standard deviation, thus the Gaussian mixture with a single component shows very good results (low MEGA).

Fortunately, we have designed this metric for complex latent variable model and there are no reasons to use it when assessing the fit of a single Gaussian. Additionally, other comparative strategies discussed in Sect. 3 can still be used in parallel of our proposed one. For instance, the MMD agree with MEGA in most experiments we have done but when they don't agree, they provide different information and can be complementary.

If a trained distribution has low MEGA but bad-looking generated samples, this still provides us with insightful information. It indicates that the problem is in the LVGM distribution's higher moments and our metric was able to provide us with that information very quickly.

For regularization applications, we were able to successfully used MEGA with two different LVGMs, GMMs and VAEs. Though in both these cases we managed to achieve good results, selecting the appropriate constraint using the hyper-parameters is not an easy task. We recommend plotting the selected number of components against various values of α and to consider all of the models between the most and the least complex model. The use of a validation set can also help select the right value for α.

We are currently working on a generalization of MEGA. We want to extend our metric not only to higher moments of $\mathbf{x}$ but to any functions $g(\mathbf{x})$. This would not only allow us to compare the skewness of the trained model compared to the data but also more complicated properties of distributions such as multimodality. This generalized MEGA would provide a more complete evaluation of models than the currently proposed metric.

10 Conclusion

In this article we introduced a fairly simple and computationally efficient way to check the generative model's distribution of a large class of LVGMs. This metric, MEGA, evaluates the gap between the LVGM distribution's first and second moment and the training (or validation) data set's first and second moment.

The premise of the proposed metric is theoretically simple and quite intuitive. Both the DE and the FME are unbiased estimators of both the first and the second moment and if a gap exists between them then the LVGM distribution does not match key aspects of the data distribution.

To support our theoretical arguments, we have demonstrated how to use this metric for two different purposes. First, as an evaluation metric that can replace the more heuristics approaches that rely on eyeballing generated samples. Second, since this metric is currently available for the first two moments, it favours a simple model, such a single Gaussian, and thus can be used as regularization for models such as GMMs and VAEs.

However, we believe we have only scratched the surface of all of the applications and ways to incorporate these moment-gap-based metrics in model fitting and model selection pipelines. We hope to make further progress in this direction in future work. Another future work direction is to extend these moment gap estimators to sequential LVGMs, such as hidden Markov models and state-space models. Finally, the biggest improvement we could work on is to extend the moment estimators to higher moments; this would make the evaluation metrics much more valuable.

Nonetheless, we believe this work is a first step in data-driven automated model selection based on moments and we hope it inspires similar contributions.

Appendix: Proof that FMEs have a small variance than SEs

First moment

For the SE, we first sample $z_1, \dots z_m$, a set of m latent variables where $\mathbf{z} \sim p(\mathbf{z})$ then sample $x_1, \dots, x_m$ with $\mathbf{x} \sim p(\mathbf{x} \| \mathbf{z}_i)$; this implies that $\mathbf{x} \sim p(\mathbf{x} \| \mathbf{z}) p(\mathbf{z}) = p(\mathbf{x})$. Then, the SE for the first moment is $\frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i$.

$$\mathbf{Var}_x(\text{SE}) = \mathbf{Var}_x\left(\frac{\sum_{i=1}^{m} x_i}{m}\right) = \frac{1}{m^2} \sum_{i=1}^{m} \mathbf{Var}_x(x_i) = \frac{\mathbf{Var}_x(\mathbf{x})}{m}. \tag{21}$$

When building our FME, we sample $z_1, \dots z_m$, a set of m latent variables where $\mathbf{z} \sim p(\mathbf{z})$ and then compute $\frac{1}{m} \sum_{i=1}^{m} \mathbf{E}_x(\mathbf{x} \| \mathbf{z} = z_i)$.

$$\mathbf{Var}_z(\text{FME}) = \mathbf{Var}_z\left(\frac{\sum_{i=1}^{m} \mathbf{E}_x(\mathbf{x} \| z_i)}{m}\right) = \frac{1}{m^2} \sum_{i=1}^{m} \mathbf{Var}_z(\mathbf{E}_x(\mathbf{x} \| z_i)) = \frac{\mathbf{Var}_z[\mathbf{E}_x(\mathbf{x} \| z_i)]}{m}. \tag{22}$$

Then using the Law of Total Variance we have that:

$$\begin{aligned} \mathbf{Var}_x(\mathbf{x}) &= \mathbf{E}_z[\mathbf{Var}_x(\mathbf{x} \| \mathbf{z})] + \mathbf{Var}_z[\mathbf{E}_x(\mathbf{x} \| \mathbf{z})] \\ &\geq \mathbf{Var}_z[\mathbf{E}_x(\mathbf{x} \| \mathbf{z})] \\ \Rightarrow \frac{\mathbf{Var}_x(\mathbf{x})}{m} &\geq \frac{\mathbf{Var}_z[\mathbf{E}_x(\mathbf{x} \| z_i)]}{m} \\ \Rightarrow \mathbf{Var}(\text{SE}) &\geq \mathbf{Var}(\text{FME}), \end{aligned} \tag{23}$$

and thus our FME has lower variance than the commonly used alternative.

Second moment

For the SE, we first sample $z_1, \dots z_m$, a set of m latent variables where $\mathbf{z} \sim p(\mathbf{z})$ then sample $x_1, \dots, x_m$ with $\mathbf{x} \sim p(\mathbf{x}\|\mathbf{z}_i)$; this implies that $\mathbf{x} \sim p(\mathbf{x}\|\mathbf{z})p(\mathbf{z}) = p(\mathbf{x})$. Then, the SE for the second moment is $\frac{1}{m}\sum_{i=1}^{m} \mathbf{x}_i^2$.

$$\mathbf{Var}_x(\mathrm{SE}) = \mathbf{Var}_x\left(\frac{\sum_{i=1}^{m} x_i^2}{m}\right) = \frac{\mathbf{Var}_x(\mathbf{x}^2)}{m}. \tag{24}$$

When building our FME, we sample $z_1, \dots z_m$, a set of m latent variables where $\mathbf{z} \sim p(\mathbf{z})$ and then compute $\frac{1}{m}\sum_{i=1}^{m}\left(\mathbf{Var}_x(\mathbf{x}\|\mathbf{z} = z_i) + [\mathbf{E}_x(\mathbf{x}\|\mathbf{z} = z_i)]^2\right)$.

$$\begin{aligned}\mathbf{Var}_z(\mathrm{FME}) &= \mathbf{Var}_z\left(\frac{\sum_{i=1}^{m}\left(\mathbf{Var}_x(\mathbf{x}\|\mathbf{z} = z_i) + [\mathbf{E}_x(\mathbf{x}\|\mathbf{z} = z_i)]^2\right)}{m}\right)\\ &= \frac{\mathbf{Var}_z\left(\mathbf{Var}_x(\mathbf{x}\|\mathbf{z} = z_i) + [\mathbf{E}_x(\mathbf{x}\|\mathbf{z} = z_i)]^2\right)}{m}.\end{aligned} \tag{25}$$

Now let's take a closer look at the numerator of Eq. (25).

$$\begin{aligned}\mathbf{Var}_z\left(\mathbf{Var}_x(\mathbf{x}\|\mathbf{z} = z_i) + [\mathbf{E}_x(\mathbf{x}\|\mathbf{z} = z_i)]^2\right) &= \mathbf{Var}_z\left(\mathbf{E}_x[\mathbf{x}^2\|\mathbf{z}] - \mathbf{E}_x[\mathbf{x}\|\mathbf{z}]^2 + \mathbf{E}_x[\mathbf{x}\|\mathbf{z}]^2\right)\\ &= \mathbf{Var}_z\left(\mathbf{E}_x[\mathbf{x}^2\|\mathbf{z}] - \mathbf{E}_x[\mathbf{x}\|\mathbf{z}]^2 + \mathbf{E}_x[\mathbf{x}\|\mathbf{z}]^2\right)\\ &= \mathbf{Var}_z\left(\mathbf{E}_x[\mathbf{x}^2\|\mathbf{z}]\right)\\ &= \mathbf{E}_z\left(\mathbf{E}_x[\mathbf{x}^2\|\mathbf{z}]^2\right) - \mathbf{E}_z\left(\mathbf{E}_x[\mathbf{x}^2\|\mathbf{z}]\right)^2\\ &= \mathbf{E}_z\left(\mathbf{E}_x[\mathbf{x}^2\|\mathbf{z}]^2\right) - \mathbf{E}_x\left(\mathbf{x}^2\right)^2\\ \Rightarrow \mathbf{Var}(\mathrm{FEM}) &= \frac{\mathbf{E}_z\left(\mathbf{E}_x[\mathbf{x}^2\|\mathbf{z}]^2\right) - \mathbf{E}_x\left(\mathbf{x}^2\right)^2}{m}\end{aligned} \tag{26}$$

Finally, let us apply the Law of Total Variance to $\mathbf{x}^2$:

$$\begin{aligned}\mathbf{Var}_x(\mathbf{x}^2) &= \mathbf{E}_z(\mathbf{Var}_x[\mathbf{x}^2\|\mathbf{z}]) + \mathbf{Var}_z(\mathbf{E}_x[\mathbf{x}^2\|\mathbf{z}])\\ &= \mathbf{E}_z(\mathbf{Var}_x[\mathbf{x}^2\|\mathbf{z}]) + \mathbf{E}_z\left(\mathbf{E}_x[\mathbf{x}^2\|\mathbf{z}]^2\right) - \mathbf{E}_z(\mathbf{E}_x[\mathbf{x}^2\|\mathbf{z}])^2\\ &= \mathbf{E}_z(\mathbf{Var}_x[\mathbf{x}^2\|\mathbf{z}]) + \mathbf{E}_z\left(\mathbf{E}_x[\mathbf{x}^2\|\mathbf{z}]^2\right) - \mathbf{E}_x[\mathbf{x}^2]^2\\ \Rightarrow \mathbf{Var}_x(\mathbf{x}^2) &\geq \mathbf{E}_z\left(\mathbf{E}_x[\mathbf{x}^2\|\mathbf{z}]^2\right) - \mathbf{E}_x[\mathbf{x}^2]^2\\ \Rightarrow \frac{\mathbf{Var}_x(\mathbf{x}^2)}{m} &\geq \frac{\mathbf{E}_z\left(\mathbf{E}_x[\mathbf{x}^2\|\mathbf{z}]^2\right) - \mathbf{E}_x[\mathbf{x}^2]^2}{m}\\ \Rightarrow \mathbf{Var}(\mathrm{SE}) &\geq \mathbf{Var}(\mathrm{FME})\end{aligned} \tag{27}$$

Acknowledgements I want to thank Yanbo Tang for his valuable help with selecting the matrix norm and for providing sketches for the demonstration that FMEs have smaller variances than SEs. I also want to thank David Duvenaud, Michael Lalancette, Jeffrey S. Rosenthal, Anthony Coache and Renaud Alie for their insightful comments.

Author's contribution Not applicable (only one author).

Funding The author was funded by the Ontario Graduate Scholarship (OGS) program and the University of Toronto School of Graduate Studies during the early parts of this project and was funded by the Canadian Statistical Sciences Institute (CANSSI) during the later parts.

Data availability The simulated data was generated using publicly available packages, the code used to generate the data is provided. The real data application uses MNIST, a widely used and publicly available data set.

Code availability The code used in preparation of this manuscript is available on the author's GitHub page: https://github.com/CedricBeaulac/LVGM_Analysis. The URL is referenced in the article. The files containing the proposed metric are clearly commented and easy to import and use.

Declarations

Conflict of interest The author has no conflict of interest to declare.

Consent for publication Not applicable.

Consent to participate Not applicable.

Ethics approval All ethical responsibilities were considered and respected.

References

Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. Selected papers of hirotugu akaike, 199–213

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19*(6), 716–723.

Anandkumar, A., Foster, D. P., Hsu, D. J., Kakade, S. M., & Liu, Y.-K. (2012). A spectral algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, vol. 25. Curran Associates, Inc., Red Hook, NY

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., & Telgarsky, M. (2014). Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research, 15*(80), 2773–2832.

Beaulac, C. (2021). MEGA for LVGM. GitHub. https://github.com/CedricBeaulac/LVM_Analysis

Beaulac, C. (2021). Performance and accessibility of statistical learning algorithms for applied data analysis. PhD thesis, University of Toronto

Bishop, C. M. (2007). *Pattern recognition and machine learning*. New York: Springer.

Chaganty, A. T., & Liang, P. (2014). Estimating latent-variable graphical models using moments and likelihoods. In *International Conference on Machine Learning* (pp 1872–1880). PMLR

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software, 33*(1), 1–22.

Goodhart, C. A. (1984). Problems of monetary management: The UK experience. *Monetary theory and practice* (pp. 91–121). Palgrave, London. https://doi.org/10.1007/978-1-349-17295-5_4

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research, 13*(25), 723–773.

Harris, C. R., Millman, K. J., der Walt, S. J. V., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del R'ıo, J. F., Wiebe, M., Peterson, P., Oliphant, T. E. (2020). Array programming with NumPy. *Nature, 585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (2nd ed., p. 100). New York: Springer.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., Red Hook, NY.

Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., & Lerchner, A. (2018). Towards a definition of disentangled representations. arXiv:1812.02230

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., & Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR, 2*(5), 6.

Jitkrittum, W., Kanagawa, H., & Schölkopf, B. (2020). Testing goodness of fit of conditional density models with kernels. In *Conference on Uncertainty in Artificial Intelligence* (221–230). PMLR, New York

Kanagawa, H., Jitkrittum, W., Mackey, L., Fukumizu, K., & Gretton, A. (2019). A kernel stein test for comparing latent variable models. arXiv:1907.00586

Keller, F., Muller, E., & Bohm, K. (2012). HiCS: High contrast subspaces for density-based outlier ranking. In 2012 *IEEE 28th international conference on data engineering* (pp. 1037–1048). IEEE.

Kingma, D. P. (2017). Variational inference & deep learning: A new synthesis. PhD thesis, Universiteit van Armsterdam

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, Conference Track Proceedings

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., & Welling, M. (2016). *Improved variational inference with inverse autoregressive flow* (pp. 4743–4751). Curran Associates Inc.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324.

Li, Y., Swersky, K., & Zemel, R. (2015). Generative moment matching networks. In *International Conference on Machine Learning* (pp. 1718–1727). PMLR

Liu, Q., Lee, J., & Jordan, M. (2016). A kernelized stein discrepancy for goodness-of-fit tests. In *Proceedings of The 33rd International Conference on Machine Learning*, vol. 48, (pp 276–284). PMLR, New York

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining, Piscataway, New Jersey, pp. 413–422. IEEE

Mises, R., & Pollaczek-Geiringer, H. (1929). Praktische verfahren der gleichungsauflösung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, 9*(1), 58–77.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Red Hook: Curran Associates Inc.

Podosinnikova, A. (2016). On the method of moments for estimation in latent linear models. PhD thesis, PSL Research University

Rigollet, P., & Hütter, J.-C. (2015). High dimensional statistics. Lecture notes for course 18S997 813, 814

Rosenthal, J. S. (2019). Monte Carlo methods. Lecture notes for course STA2431

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

Sigillito, V. G., Wing, S. P., Hutton, L. V., & Baker, K. B. (1989). Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest, 10*(3), 262–266.

Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2011). Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software, 39*(5), 1–13.

Strathern, M. (1997). 'Improving ratings': Audit in the British university system. *European Review, 5*(3), 305–321.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological), 58*(1), 267–288.

Tipping, M. E., & Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 61*(3), 611–622.

Vahdat, A., & Kautz, J. (2020). Nvae: A deep hierarchical variational autoencoder. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA

Zhao, S., Song, J., & Ermon, S. (2019). Infovae: Balancing learning and inference in variational autoencoders. In *Proceedings of the Aaai Conference on Artificial Intelligence*, vol. 33, pp. 5885–5892

Zimek, A., Schubert, E., & Kriegel, H.-P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal, 5*(5), 363–387.

Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., & Chen, H. (2018). Deep autoencoding gaussian mixture model for unsupervised anomaly detection. *International Conference on Learning Representations*.