Interpretable Domain Shift Diagnosis In Industrial Process Data With Variational Autoencoder

Brendan L'Ollivier,[1] **Sonia Tabti,** [1] **Julien Budynek** [1]

[1] Fieldbox, Quai Armand Lalande, 33300, Bordeaux, France
blollivier@fieldbox.ai, stabti@fieldbox.ai, jbudynek@fieldbox.ai

Abstract

This paper explores the use of sparse Variational Autoencoders (VAE) in order to build an easily interpretable shift diagnosis framework applied to industrial datasets. To this end, several models are compared, in particular, we introduce the LassoVAE, a sparse model with a computationally efficient training. Comparisons are obtained thanks to an experimental protocol we designed that allows to generate synthetic data and different types of shifts with various parameters. New metrics are also introduced to evaluate the models' ability to retrieve the sources of shifts. Results show that sparse models are highly more efficient at recovering the true interactions between variables than a VAE with a dense decoder.

Introduction

Industrial Context

Industrial actors have long been focused on continuous improvement of their processes. To this end, as many other economic agents, industries have engaged in a digital transformation that aims at rationalizing, optimizing and automating production processes through data collection, modeling and transformation into exploitable knowledge (Eifert et al. 2020; Lasi et al. 2014). From a technical point of view, the path towards complete automation will have to go through a certain number of milestones. The first one that is currently ongoing is data collection. An increasing number of industrial actors realize the value of the data generated by their activities and thus invest into modern data architecture storage. The next milestone, which will benefit from the maturity of data collection activities, is the development of artificial intelligence models that enable the transformation of raw sensors data into valuable insights. One limitation of this modeling phase is the transgression of the "independent and identically distributed" (i.i.d.) assumption or, in other words, the assumption that the data used during the different stages of the model life cycle (training, testing, production) share the same underlying distribution. In reality it is often impossible to maintain complete control over the data generation process since some variables may inevitably vary from sample to sample; e.g. ambient temperature or pressure in a physical process, or several machines operating in different contexts (Liu et al. 2019; Yang et al. 2020; Lu et al. 2017). Such discrepancy in distribution of sensor data is referred to in the literature as domain shift (Lemberger and Panico 2020), and is one major concern that should be monitored when deploying machine learning models.

Related work in Statistical Process Monitoring

This article relates to the field of Statistical Process Monitoring (SPM), which essentially refers to the set of machine learning and statistical models that aim at monitoring the health of an industrial process (Joe Qin 2003). Since its emergence as a research topic, one of the most challenging aspects of SPM has been the multivariate nature of industrial data, leading to complex correlations structure between process variables. Successfully learning these interactions is a critical requirement for any SPM model. To that end, latent factors models, which learn, from unsupervised training, to represent the observable data as combinations of a smaller number of independent latent factors, have been widely used to decorrelate the process variables (Qin et al. 2020). They are well suited for modeling multivariate processes since they decompose the variability of the data into two sources: the latent factors space captures the main process variability, while the residual space captures variability that causes the breakdown of normal interactions between observable variables.

The most popular unsupervised model that implements these concepts is the Principal Component Analysis (PCA), which sequentially extracts the factors that explain the most variability in the data. PCA have received a lot of attention for anomaly detection in industrial process (Teppola et al. 1998; Yin et al. 2012; Qin and Chiang 2019). In particular two anomaly scores are commonly computed : the Hoteling's T2 from the latent space and the squared prediction error (SPE) from the residual space.

More recently, Variational Autoencoders (Kingma and Welling 2014) have been used to palliate the limitations of PCA and its non-linear extension, the kernel PCA (Schölkopf, Smola, and Müller 1997), by providing an architecture that handles the non-linearities of industrial processes and which scales easily to large volumes of data. Also, the high flexibility of their neural architecture allows to model a wide range of data structures. Currently the most widely used VAE architecture for anomaly detection is the

one with encoder and decoder made of fully connected layers, also called vanilla VAE, (Lee et al. 2019; Wang et al. 2019; Zhu, Jiang, and Liu 2022). While these studies show that the VAE framework is a promising extension of PCA, they face one important drawback of vanilla VAE: the lack of explicit interpretability of the anomalies detected in the latent space, regarding the underlying physics of the industrial process.

If linear interactions are assumed, the interpretability of latent anomalies is ensured by the coefficients of the linear combinations learned by the model, known as the loading matrix (Cadima and Jolliffe 1995). In the case of PCA, this method is known as the contribution plot. However, the loading matrix learned by PCA is not sparse, hence, contribution plots are affected by residual sources of variations and thus rely on a thresholding operation to isolate the true cause of an anomaly. As pointed out by (Greco and Farcomeni 2016), the strategy to ignore features associated with loading coefficients that are small in absolute value may be misleading. This phenomenon, referred to as the smearing-out effect, has adverse impacts on the isolation of the faults (Van den Kerkhof et al. 2013). As a workaround of this interpretability issue, studies have been conducted with SparsePCA, an extension of PCA, that learns to reconstruct the data from sparse combinations of principal components for explicit interpretability of detected anomalies (Luo et al. 2017; Theisen et al. 2021; Yu, Khan, and Garaniya 2016).

Related work in Domain Shift Analysis

In spite of being a closely related subject, the measure and analysis of the shift in industrial process have received marginal attention. It is nevertheless a well known concern for model monitoring. Domain shift (Lemberger and Panico 2020) can have harmful impact on model performance and should be properly monitored if detected in a production environment. While techniques for measuring shift in univariate data are numerous and well documented, they are not designed to scale up to multivariate data. In that case, reducing the dimension of the input space appears to be a good practice as promoted in (Rabanser, Günnemann, and Lipton 2019). They expose the efficiency of measuring the shift as a statistical distance computed in a reduced dimension space.

Contributions

This paper extends the previous studies on VAEs for process monitoring by introducing a domain shift diagnosis framework. It is based on the decomposition of the shift into a covariate shift, measured in the latent space and a concept shift measured in the residual space. It also promotes sparsity inducing mechanisms in the decoder of the VAE in order to provide explicit interpretability of the covariate shift in terms of contribution of the observable features. The article compares the performance of sparse and non-sparse variational autoencoder models for detecting and providing interpretable insights about the sources of shift in the data. In particular, we introduce the LassoVAE as a computationally effective option for inducing sparsity in the decoder weights. The SparsePCA (Mairal et al. 2009) and Probabilistic PCA (Tipping and Bishop 1999) algorithms are also used as baselines. We designed a data generation process that simulates the normal behavior of an industrial process, and a unidirectional shift generator to evaluate the capacity of each trained model to isolate the true sources of shifts. The generator is a latent factors model with sparse linear interactions between the latent factors and the observable features. It is shown that the sparse models better recover the true underlying interactions between the latent factors and the observable features, leading to better isolation of the sources of the shifts. In order to quantify these two aspects, two metrics are introduced: the Mapping Recovery Score (MRS) and the Shift Dispersion Score (SDS).

VAEs applied to shift analysis

For this section and the rest of the article, we introduce the following notation conventions. Let be $\boldsymbol{x} = (x_1, ..., x_D) \in \mathbb{R}^D$ a multivariate random variable composed of D features describing an industrial process, and $\boldsymbol{X} = (X_1, ..., X_D) \in \mathbb{R}^{N \times D}$ a dataset made of N samples drawn from $p(\boldsymbol{x})$, the probability distribution of $\boldsymbol{x}$.

In practice, an industrial process has multiple stages, leading to blocks of correlated sensors. A latent factors model learns these interactions by introducing a set of unobserved latent factors $\boldsymbol{z} = (z_1, ..., z_K) \in \mathbb{R}^K$, with $K \leq D$, and a mapping function $f : \mathbb{R}^K \mapsto \mathbb{R}^D$ that transforms $\boldsymbol{z}$ into $\boldsymbol{x}$.

Decomposition of the dataset shift into covariate and concept shifts

We suppose that we have access to a dataset $\boldsymbol{X}^S$, referred to as the source, made of samples recorded in nominal operating conditions. We aim at analyzing the various shifts between this dataset and another dataset $\boldsymbol{X}^T$, called the target, recorded in unknown conditions. The distributions of $\boldsymbol{x}$ in these domains are denoted by $p^S(\boldsymbol{x})$ and $p^T(\boldsymbol{x})$ respectively. The latent factors assumption applied to process variables $\boldsymbol{x}$ lead to the following joint distribution:

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

Equation (1) shows that the shift between $p^S(\boldsymbol{x}, \boldsymbol{z})$ and $p^T(\boldsymbol{x}, \boldsymbol{z})$ can be decomposed into:

- a covariate shift if $p^S(\boldsymbol{z}) \neq p^T(\boldsymbol{z})$ and $p^S(\boldsymbol{x}|\boldsymbol{z}) = p^T(\boldsymbol{x}|\boldsymbol{z})$,
- and a concept shift, if $p^S(\boldsymbol{x}|\boldsymbol{z}) \neq p^T(\boldsymbol{x}|\boldsymbol{z})$.

A detailed overview on these two types of shift is provided by (Lemberger and Panico 2020). In an industrial context, this decomposition of the shift is valuable since it discriminates between shifts that alter the variables interactions: $p^S(\boldsymbol{x}|\boldsymbol{z}) \neq p^T(\boldsymbol{x}|\boldsymbol{z})$ (e.g.: wear and sensor failures), and those that do not (e.g.: a different set point).

The Variational Autoencoder framework

The Variational Autoencoder (VAE), firstly introduced in (Kingma and Welling 2014), is a class of generative models that efficiently learn the distribution of high dimensional data. It learns to generate $\boldsymbol{x}$ from Gaussian latent factors $\boldsymbol{z} \in \mathbb{R}^K$, $K < D$, with an autoencoder architecture. From a

probabilistic point of view, the goal is to learn the posterior distribution:

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

One of the advantages of the VAE is to overcome the intractability of $p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$ with variational inference. It introduces $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, a distribution parameterized by a neural network, so that the Kullback–Leibler (KL) divergence $D_{KL}(p(\boldsymbol{z}|\boldsymbol{x})\|q_\phi(\boldsymbol{z}|\boldsymbol{x}))$ is minimized, which is equivalent to minimizing the following loss function:

$$\mathcal{L}_{\text{VAE}} = -\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] + D_{KL}\big(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z})\big)$$

Auto-pruning property

One fundamental property of the VAE is the auto-pruning of superfluous latent factors (Dai et al. 2019) leading to an estimation of the intrinsic dimension of the data. This behavior represents a significant advantage of VAE over latent factors models which require a fine-tuning of the number of latent factors.

Measuring the shift with a VAE

This section aims at analyzing the shift between $\boldsymbol{X}^S$ and $\boldsymbol{X}^T$ with a VAE model trained on $\boldsymbol{X}^S$. We refer to the learned encoder and decoder as g_ϕ^S and f_θ^S respectively. They are used to compute the latent spaces $\boldsymbol{Z}^S$, $\boldsymbol{Z}^T \in \mathbb{R}^{N\times K}$, and the residual spaces $\boldsymbol{E}^S, \boldsymbol{E}^T \in \mathbb{R}^{N\times D}$ defined below:

$$\boldsymbol{Z}^S = g_\phi(\boldsymbol{X}^S), \quad \boldsymbol{Z}^T = g_\phi(\boldsymbol{X}^T) \tag{2}$$

$$\boldsymbol{E}^S = f_\theta(\boldsymbol{Z}^S) - \boldsymbol{X}^S, \quad \boldsymbol{E}^T = f_\theta(\boldsymbol{Z}^T) - \boldsymbol{X}^T \tag{3}$$

We introduce $\Delta : \boldsymbol{X}, \boldsymbol{Y} \mapsto \Delta(\boldsymbol{X}, \boldsymbol{Y}) \in \mathbb{R}$, a distance function between two datasets $\boldsymbol{X}$ and $\boldsymbol{Y}$, defined as: $\Delta(\boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{D} \delta(X_i, Y_i)$, where, $\forall 1 \leq i \leq D$, $\delta(X_i, Y_i)$ represents the contribution of the i-th feature to the total distance. The distance δ can be any arbitrary distance between univariate distributions. We call $\delta(\boldsymbol{X}, \boldsymbol{Y}) = \big(\delta(X_i, Y_i)\big)_{1\leq i\leq D}$ the shift profile between $\boldsymbol{X}$ and $\boldsymbol{Y}$. We use Δ to evaluate jointly the covariate and concept shifts:

- Covariate shift: $\Delta(\boldsymbol{Z}^S, \boldsymbol{Z}^T) \gg 0 \wedge \Delta(\boldsymbol{E}^S, \boldsymbol{E}^T) \simeq 0$,
- Concept shift: $\Delta(\boldsymbol{E}^S, \boldsymbol{E}^T) \gg 0$,

where $\wedge$ is the logical "and" operator.

Affine Decoder

The combination of $\Delta(\boldsymbol{Z}^S, \boldsymbol{Z}^T) \gg 0$ and $\Delta(\boldsymbol{E}^S, \boldsymbol{E}^T) \simeq 0$ requires the decoder to generalize to unseen regions of the latent space. This property is usually not guaranteed by a deep non-linear decoder. That's why we assume linear interactions between variables. In that case, an affine decoder ensures the generalization of the decoder to any new region of the latent space, as long as the interactions (i.e. the weights of the loading matrix) are not altered.

Interpretability of the latent factors

One critical aspect of the shift diagnosis is the interpretability of both covariate shift and concept shift regarding the contribution of each observable feature. In the case of concept shift, such interpretability is straightforward since the shift profile $\big\{\delta(E_i^S, E_i^T)\big\}_{1\leq i\leq D}$ gives the contribution of each process variable to the overall shift. The interpretability of the covariate shift is less obvious since it heavily relies on the quality of the recovery of the true interactions between $\boldsymbol{x}$ and $\boldsymbol{z}$. This issue is directly linked to the smearing-out effect mentioned in the introduction: without specific regularization mechanism, an autoencoder-like model is likely to learn residual interactions that do not relate to the underlying physical process and thus corrupt the analysis of $\delta(Z_k^S, Z_k^T)$. This phenomenon is particularly problematic when dealing with industrial sensors data. The underlying physical interactions are sparse, and such sparsity should be reflected in the factors-to-features mapping. We assume that better recovering the true sparse mapping between the latent factors and the observable features is an efficient method to explicitly mitigate the smearing-out effect. In this section, we present VAE architectures with sparsity inducing regularization that can be used to recover the true mapping.

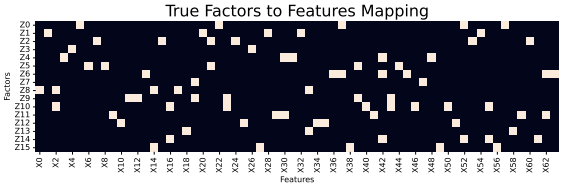


Figure 1: Example of a sparse interactions mapping between 16 latent factors and 64 observable variables. A white cell in position (i, j) shows that Z_i interacts with Xj.

Factors-to-features mapping matrix

The true interactions between $\boldsymbol{z}$ and $\boldsymbol{x}$ can be modeled by a binary mask W^{true} of size $K_{\text{true}} \times D$, with K_{true} the true dimension of the latent space, such that each entry of this matrix is defined as follows $\forall\ 1 \leq k \leq K_{\text{true}}$ and $1 \leq d \leq D$:

$$w_{k,d}^{\text{true}} = \begin{cases} 1 \text{ if } z_k \text{ interacts with } x_d \\ 0 \text{ elsewhere} \end{cases}$$

Figure 1 illustrates such a binary mapping. The interactions' mapping matrix informs whether a given factor contributes to the variation of a given feature. The Experiments section below shows that the recovery of the true interactions' mapping matrix is crucial for an accurate isolation of the true source of covariate shift.

Mapping recovery score

In order to measure the quality of the binary mapping learned by the model: W^{pred}, we introduce the Mapping Recovery Score (MRS), an aggregation of the pairwise Jaccard index between the rows of the true mapping and the rows of the mapping inferred from the model:

$$MRS(W^{\text{true}}, W^{\text{pred}}) = \frac{1}{K^{\text{true}}} \sum_{k_t=1}^{K^{\text{true}}} \max_{1 \leq k_p \leq K^{\text{pred}}} J(w_{k_t}^{\text{true}}, w_{k_p}^{\text{pred}}) \tag{4}$$

Where K^{pred} is the latent space dimension inferred by the model and $J(w_{k_t}^{\text{true}}, w_{k_p}^{\text{pred}})$ is the Jaccard similarity score between the k_t^{th} row of W^{true} and the k_p^{th} row of W^{pred}:

$$J(w_{k_t}^{\text{true}}, w_{k_p}^{\text{pred}}) = \frac{w_{k_t}^{\text{true}} \cap w_{k_p}^{\text{pred}}}{w_{k_t}^{\text{true}} \cup w_{k_p}^{\text{pred}}}$$

Since the Jaccard index ranges from 0 (no similarity) to 1 (perfect recovery of the mapping) the MRS also ranges from 0 to 1. Note the analysis of this score has to be paired with the accuracy of the latent space dimension estimation.

Sparse Decoder

This section presents two VAE architectures with sparsity inducing mechanisms. Sparsity is referred here as the property that each observable feature is a combination of a limited number of latent factors. This encourages the recovery of the true mapping between the latent factors and the observable features.

LassoVAE We propose a new architecture, the LassoVAE: a VAE with L1 regularization on the weights of the affine decoder, which, combined with the auto-pruning effect of the KL divergence, learns sparse interactions between the latent factors and the observable variables. Its loss function, in equation (5), directly derives from the VAE loss by adding a penalty term that corresponds to the sum of absolute values over the set Θ of weights and bias of the linear decoder:

$$\mathcal{L}_{\text{LassoVAE}} = \mathcal{L}_{\text{VAE}} + \alpha \sum_{\theta \in \Theta} |\theta| \tag{5}$$

The parameter α controls the intensity of the sparsity. Optimal value corresponds to a trade-off between the reconstruction loss and the L1 loss.

SparseVAE We also test the SparseVAE (Moran et al. 2022), that explicity learns the binary mask W during the training phase by setting a Spike and Slab Prior (Ročková and George 2016) on the weights of a $K \times D$ parameters matrix. The mask W is then used as an input selection mask for the decoding of each feature. However, the SparseVAE's training procedure has a higher computational burden than the LassoVAE. This is due to the several passes through the decoder that are required to reconstruct all features.

Both architectures benefit from the auto-pruning mechanism, enabling the recovery of the true latent dimension K_{pred}.

Experiments

This section presents the results of a series of experiments that test the capacity of a model to isolate the true sources of shifts. Our objective is to quantify the advantages that could provide the learning of the true sparse interactions between process variables, on the isolation of the source of shifts. We first synthesized a source dataset representing the normal behavior of an industrial process with sparse linear interactions. The source data is used to train the models. The interactions mapping between $\boldsymbol{z}$ and $\boldsymbol{x}$ is inferred from the weights of the decoder. A set of unidirectional shifts is then applied to both of the latent space and the features space to simulate sources of covariate and concept shift respectively. The Shift Dispersion Score is finally introduced as a measure of the quality of the isolation of the sources of shift.

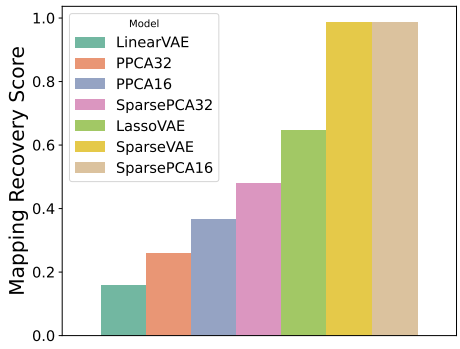


Figure 2: Mapping Recovery Scores of all models tested in this study. We see the superiority of of sparse models LassoVAE, SparseVAE and SparcePCA16 for this task. Note that without the correct number of latent factors the Sparse PCA do not recover the true mapping, as shown by the poorer performace of SparsePCA32.

Beta-Bernouilli prior on the factors-to-features mapping

This paragraph describes how the binary interaction matrix W of size $K \times D$ is synthesized. Its sparsity can be randomly generated with a Beta-Bernoulli distribution such that $\forall k \in [\![1 : K]\!]$ and $d \in [\![1 : D]\!]$:

$$\eta_k \sim \text{Beta}(a, b), \quad w_{k,d} \sim \text{Bernoulli}(\eta_k) \tag{6}$$

where η_k controls the proportion of features that depend on the k-th factor, $w_{k,d}$ controls whether the d^{th} feature x_d, depends on the k^{th} factor Z_k. The parameters of the Beta distribution are $a, b > 0$. When b is set to 1, $a \in]0, 1]$ controls the intensity of the sparsity. The closer a is to zero, the sparser the mapping matrix is (and the closer to 1 is a, the denser the mapping matrices are).

Once a mask W is generated, the overall latent factors process can be expressed as:

$$\begin{aligned} z_k &\sim \mathcal{N}(0, 1), k = 1, ..., K \\ x_d &\sim \mathcal{N}\big(f(w_d \odot z_k), \sigma_d^2\big), d = 1, ..., D \end{aligned} \tag{7}$$

where σ_d^2 is the per-feature noise variance and $\odot$ the element-wise multiplication. Generating the latent factors

from a Normal distribution is a common assumption already used in PCA for instance.

Source dataset

Let's assume that the Beta-Bernoulli prior described in equation (6) was used to generate the true mapping matrix $W^{\text{true}} \in \mathbb{R}^{K^{\text{true}} \times D}$ and that $\boldsymbol{X}^{\text{true}} \in \mathbb{R}^{N \times D}$ and $\boldsymbol{Z}^{\text{true}} \in \mathbb{R}^{N \times K^{\text{true}}}$ were synthesized thanks to equation (7). Since linear interactions are assumed, the mapping function f can be simplified to a matrix product with the loading matrix $C^{\text{true}} \in \mathbb{R}^{K^{\text{true}} \times D}$ as follows: $\boldsymbol{X}^{\text{true}} = \boldsymbol{Z}^{\text{true}} C^{\text{true}}$. The positions of the non-zero entries of C^{true} are the same as the ones of the binary matrix W^{true}. The absolute values of the non-zero entries are drawn from a uniform distribution $\mathcal{U}_{[0.1,5]}$ and the sign of those values as well.

To make the data more realistic, a small intensity (relatively to the features variances) Gaussian noise, of standard deviation σ^{noise}, is added to the true distribution (where $\mathbf{0}$ is a vector with all entries equal to zero):

$$\boldsymbol{X}^{\text{obs}} = \boldsymbol{X}^{\text{true}} + \mathcal{N}(\mathbf{0}, (\sigma^{\text{noise}})^2 I_D) \tag{8}$$

Target dataset

In order to test the capacity of the models to isolate the true sources of shifts, we developed a generator of unidirectional shifts, for which the ground truth shift profile is known.
The target data is generated by batch. Each batch, follows the same shifted distribution. A batch is initialized by generating N_{batch} samples from the source distribution, with a level of Gaussian noise of standard deviation $\sigma^{\text{noise}}_{\text{batch}}$:

$$\boldsymbol{Z}^{\text{true}}_{\text{batch}} \sim \mathcal{N}(\mathbf{0}, I_K), \quad \boldsymbol{X}^{\text{true}}_{\text{batch}} = \boldsymbol{Z}^{\text{true}}_{\text{batch}} C^{\text{true}} \tag{9}$$

Unidirectional shifts are generated from that batch by applying unidirectional linear translations to $\boldsymbol{Z}^{\text{true}}_{\text{batch}}$ and $\boldsymbol{X}^{\text{true}}_{\text{batch}}$. The interest of using latent factors is here emphasized: a shift that would affect a group of correlated sensors is simply modeled by a unidirectional translation in the latent space.

- **Unidirectional covariate shift** Unidirectional covariate shift is generated by applying the linear translation to one of the latent factors.

  $$\boldsymbol{Z}^{\text{cov}} = u^{\text{cov}}_k(\boldsymbol{Z}^{\text{true}})$$
  $$u^{\text{cov}}_k : \boldsymbol{z} = (z_1, ..., z_K) \mapsto (z_1, ..., z_k + \sigma_{z_k}, ..., z_K)$$

- **Unidirectional concept shift** Unidirectional concept shift is generated by applying the linear translation to one of the observable features.

  $$\boldsymbol{X}^{\text{con}} = u^{\text{con}}_d(\boldsymbol{X}^{true}) + \mathcal{N}(\mathbf{0}, (\sigma^{\text{noise}}_{\text{batch}})^2 I_D)$$
  $$u^{\text{con}}_d : \boldsymbol{x} = (x_1, ..., x_D) \mapsto (x_1, ..., x_d + \sigma_{x_d}, ..., x_D)$$

The amplitude of each translation is set to the standard deviation of the variable affected by the shift. The standard deviations of z_k and x_d are referred to as σ_{z_k} and σ_{x_d} respectively.
By looping over all K latent factors and D observable features, we get K batches of covariate shift samples and D batches of concept shift samples. In addition, the generation process is repeated for various values of N_{batch} and $\sigma^{\text{noise}}_{\text{batch}}$. The Results section below provides visual aggregation of all configurations. We refer as $\boldsymbol{Z}^{\text{cov}} = \{Z^{\text{cov}}_n\}_{1 \leq n \leq N^{\text{cov}}}$ the set of all batches made of N^{cov} covariate shift samples and $\boldsymbol{X}^{\text{con}} = \{X^{\text{con}}_n\}_{1 \leq n \leq N^{\text{con}}}$ the one with all batches made of N^{con} concept shift samples.

Shift Dispersion Score

We aim at comparing the performance of each model in the task of shift isolation for various values of N_{batch} and $\sigma^{\text{noise}}_{\text{batch}}$. The models are scored according to their capacity to retrieve the true direction of each unidirectional shift. This property is measured by comparing the shift profiles in the latent space and the residual space, computed with the VAE model trained on the source data, to the expected shift profiles. Similarly to a classification task: the shift profile can be seen as a distribution of probabilities over the potential source of shifts. That's why we propose to compute the Shift Dispersion Score (SDS) of an estimated shift profile noted $\hat{\delta} = (\hat{\delta}_1, ..., \hat{\delta}_D)$ as the crossentropy between the descending ordered estimated shift profile vector: $(\max_{1 \leq d \leq D} \hat{\delta}_d, ..., \min_{1 \leq d \leq D} \hat{\delta}_d)$, and the corresponding true shift profile, noted $\delta^{\text{true}} = (1, 0, ..., 0)$, as follows:

$$SDS(\hat{\delta}, \delta^{\text{true}}) = -\log\big(\max_{1 \leq d \leq D} \hat{\delta}_d\big) \tag{10}$$

Note that eq.(10) has been simplified due to the binary nature of δ^{true}.

Results

Synthetic data We have trained three VAE models, each one with a different decoder architecture: LinearVAE, Lasso-VAE and the SparseVAE, on a source dataset, whose sparse interactions can be visualized in figure 1. The interactions mapping inferred by each model is evaluated with the Mapping Recovery Score defined in equation (4). The target dataset is used to evaluate the models at the task of isolating the source of unidirectional covariate shifts. The batches of shifted data have been generated with various values of $N_{\text{batch}} \in [20, 40, 80, 160, 320]$ and $\sigma^{\text{noise}}_{\text{batch}} \in [0.0, 0.1, 0.2, 0.5, 1]$.

In total, the source data is made of 5000 samples, 64 observable features and 16 latent factors. The target data is made of 2000 batches of shifted data, with 20 to 320 samples per batch. The parameters used for the different models (α, β and the architectures) can be consulted in the code repository linked to the paper. The Probabilistic PCA (PPCA) and the SparsePCA algorithms from the scikit-learn library are also trained and tested as baselines. Note that these two models do not prune automatically the unnecessary factors. The number of factors is a parameter requiring fine tuning. Both of PPCA and SparsePCA are trained with the optimal number of factors: 16, and with a non optimal number: 32. The respective models have a "16" or "32" suffix added to their base name (e.g. "SparsePCA16").

As distance δ, the 1D Wasserstein distance was selected because it is relevant to compare distributions and robust against outliers and disjoint support. Note that the distance

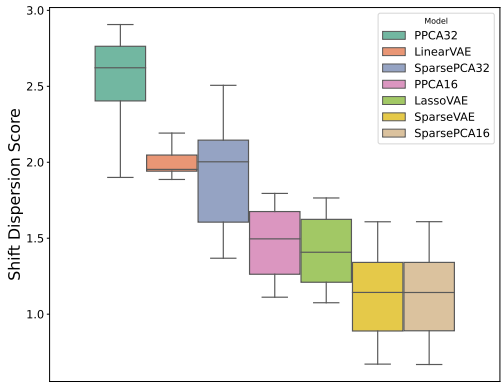


Figure 3: Aggregated Shift Dispersion Scores. Lowest values mean a better isolation of the true source of shift. Each boxplot shows the distribution of the SDS for various values of N_{batch} and $\sigma^{\text{noise}}_{\text{batch}}$.

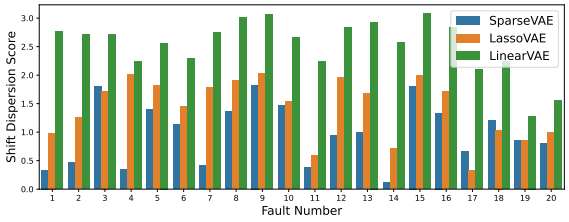


Figure 4: Shift Dispersion Scores (SDS) computed from the shift profile of each of the twenty different faults. The lower SDS values show the superior performance of sparse models for isolating the sources of shift in the latent space.

vector was normalized before computing the SDS. Figure 5 exposes the link between the quality of the recovery of the true sparse interactions (columns 1 and 2) and the improvement of the shift isolation scores. The pairwise Jaccard index matrices inform about the quality of the recovery of the true mapping matrix. The closer that matrix is from the ideal matrix: one activated cell per row, the better the recovery is. Figures 2 and 3 show an aggregated version of these results.

The most salient observations are the following:

- When the latent dimension is correctly inferred, the sparse algorithms (LassoVAE, SparseVAE and SparsePCA16) outmatch the others (LinearVAE and PPCA) at the task of isolating unidirectional shifts, as measured by the Shift Dispersion Score. This property is insured by the sparsity inducing regularization that encourages the recovery of the true mapping between the latent factors and the observable variables.
- The SparseVAE is robust to the choice of β, the intensity of the KL-Divergence term in the VAE loss function. Whereas the LassoVAE requires a fine tuning of the β parameter in order to correctly estimate the intrinsic dimension of the data. In return, the LassoVAE training is up to three times more computationally efficient than the SparseVAE. Making it a good candidate for contexts of regular re-training that do not involve adjusting β.
- The auto-pruning property of the SparseVAE and the LassoVAE make them competitive against a Sparce PCA model that requires prior knowledge on the true dimension of the latent space. In particular, the SparseVAE and SparsePCA have comparable MRS and SDS.
- As shown in fig. 5, high noise levels lead to higher SDS values meaning that it is more difficult to retrieve true sources of shifts in very noisy conditions.

Tennessee Eastman Process data The Tennessee Eastman Process dataset (Chen 2019) comes from the numerical simulation of a real chemical industrial process. This dataset is consistently used for comparing and benchmarking anomaly detection algorithms. The data-set is divided in two parts, the first part contains "fault-free" simulation runs, and the second part contains 20 different types of "faulty" runs. As pointed out by (Yin et al. 2012), each fault is caused by the abnormal behavior of only one of the process variables. We have compared the performance of each VAE model to isolate the sources of shift in "faulty" data, after training on the "fault-free" data. For each type of fault, a shift profile is obtained by computing the Wasserstein distances on all latent factors between "fault-free" and "fault" data. The resulting shift profiles are compared according to their Shift Dispersion Score. Figures 6 and 4 show the results of this experiment. The barplot clearly exposes the superior performance of sparse models for isolating unidirectional shifts in the latent space (lower SDS). Here as well, the LassoVAE improves isolation over the LinearVAE and is faster to train than the SparseVAE.

In addition, the sparsity of the loading matrices insures the interpretability of the detected shifts in terms of physics of the process.

Conclusion and perspectives

This article tackles the problem of interpretability of shifts measured in the latent space of latent factors models in the context of industrial process monitoring. We show that introducing sparsity in the modelization of the structure of the correlation matrix between sensor variables is a promising solution. We introduced the LassoVAE, a VAE with sparse linear decoder that manages to recover the true interactions between process variables and with the advantage of a computationally efficient training. It also improves the isolation of the sources of shifts in the latent space when compared to traditional VAEs. A complete shift diagnosis framework and an experimental protocol to test it have also been proposed. Hence, two new metrics have been defined: the Shift Dispersion Score and the Mapping Recovery Score. Possible future contributions would consist in generalizing this framework to spatio-temporal interactions. Similarly to what have be done with Dynamic PCA, the LassoVAE framework can be extended to the learning of sparse temporal interactions in the data. Also, since the domain shift diagnosis presented in the paper could be easily extended to multiple target domains as long as a reference domain is defined, more experiments on real world datasets relevant to this case could be conducted.

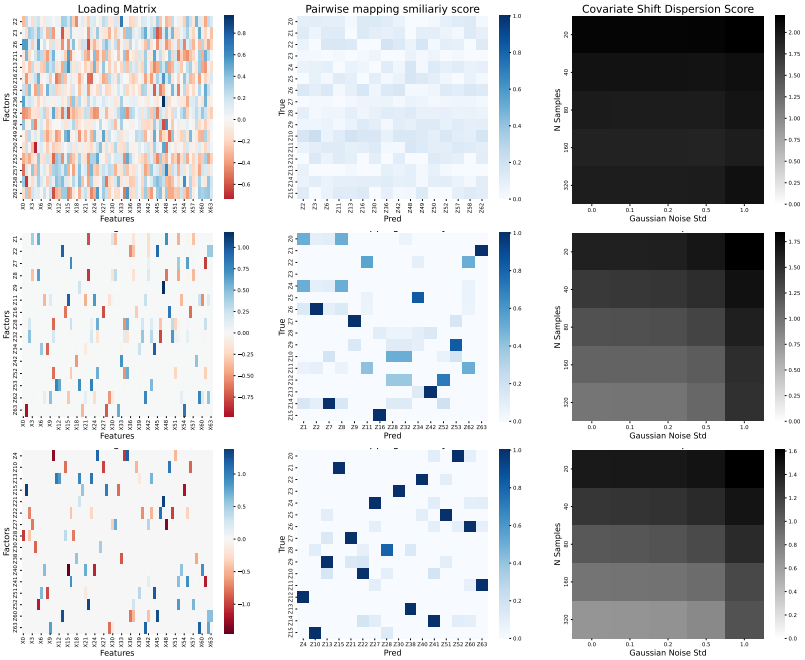
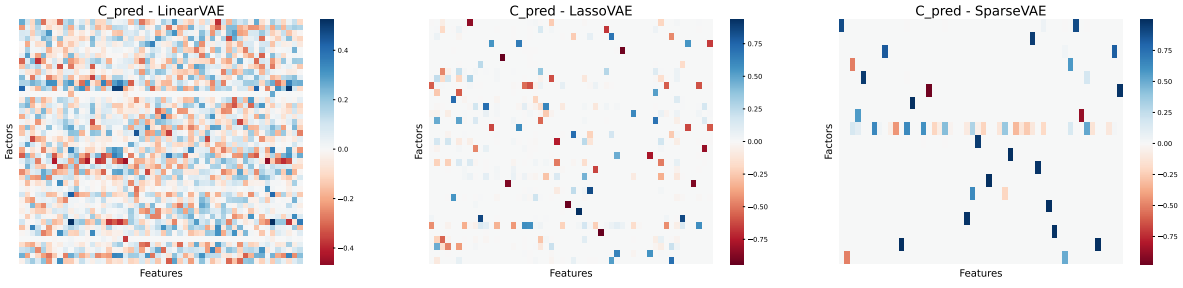


Figure 5: Results of experiments with the Linear VAE (top row), the Lasso VAE (second row) and the Sparse VAE (last row) respectively. For each model: the first column shows the loading matrix computed during training, the second column shows the pairwise Jaccard index matrix between the true mapping and the one inferred by the model, and the last column shows covariate shift dispersion scores for various combinations of shift parameters (number of samples and noise level). We see the link between the recovery of the true mapping (a single activated cell per row in the pairwise Jaccard index matrix), and the shift isolation performance.



(a) LinearVAE (b) LassoVAE (c) SparseVAE

Figure 6: Loading matrices of VAE models after training on the ”fault-free” Tennessee Eastman Process data.

References

Cadima, J.; and Jolliffe, I. T. 1995. Loading and correlations in the interpretation of principle compenents. *Journal of Applied Statistics*, 22(2): 203–214. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/757584614.

Chen, X. 2019. Tennessee Eastman simulation dataset. Publisher: IEEE Type: dataset.

Dai, B.; Wang, Y.; Aston, J.; Hua, G.; and Wipf, D. 2019. Hidden Talents of the Variational Autoencoder. *arXiv:1706.05148 [cs]*. ArXiv: 1706.05148.

Eifert, T.; Eisen, K.; Maiwald, M.; and Herwig, C. 2020. Current and future requirements to industrial analytical infrastructure—part 2: smart sensors. *Analytical and Bioanalytical Chemistry*, 412(9): 2037–2045.

Greco, L.; and Farcomeni, A. 2016. A plug-in approach to sparse and robust principal component analysis. *TEST*, 25(3): 449–481.

Joe Qin, S. 2003. Statistical process monitoring: basics and beyond. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 17(8-9): 480–502.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*. ArXiv: 1312.6114.

Lasi, H.; Fettke, P.; Kemper, H.-G.; Feld, T.; and Hoffmann, M. 2014. Industry 4.0. *Business & Information Systems Engineering*, 6(4): 239–242.

Lee, S.; Kwak, M.; Tsui, K.-L.; and Kim, S. B. 2019. Process monitoring using variational autoencoder for high-dimensional nonlinear processes. *Engineering Applications of Artificial Intelligence*, 83(C): 13–27.

Lemberger, P.; and Panico, I. 2020. A Primer on Domain Adaptation. *arXiv:2001.09994 [cs, stat]*. ArXiv: 2001.09994.

Liu, Y.; Yang, C.; Liu, K.; Chen, B.; and Yao, Y. 2019. Domain adaptation transfer learning soft sensor for product quality prediction. *Chemometrics and Intelligent Laboratory Systems*, 192: 103813.

Lu, W.; Liang, B.; Cheng, Y.; Meng, D.; Yang, J.; and Zhang, T. 2017. Deep Model Based Domain Adaptation for Fault Diagnosis. *IEEE Transactions on Industrial Electronics*, 64(3): 2296–2305. Conference Name: IEEE Transactions on Industrial Electronics.

Luo, L.; Bao, S.; Mao, J.; and Tang, D. 2017. Fault Detection and Diagnosis Based on Sparse PCA and Two-Level Contribution Plots. *Industrial & Engineering Chemistry Research*, 56(1): 225–240. Publisher: American Chemical Society.

Mairal, J.; Bach, F.; Ponce, J.; and Sapiro, G. 2009. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, 1–8. Montreal, Quebec, Canada: ACM Press. ISBN 978-1-60558-516-1.

Moran, G. E.; Sridhar, D.; Wang, Y.; and Blei, D. M. 2022. Identifiable Deep Generative Models via Sparse Decoding. *arXiv:2110.10804 [cs, stat]*. ArXiv: 2110.10804.

Qin, S. J.; and Chiang, L. H. 2019. Advances and opportunities in machine learning for process data analytics. *Computers & Chemical Engineering*, 126: 465–473.

Qin, S. J.; Dong, Y.; Zhu, Q.; Wang, J.; and Liu, Q. 2020. Bridging systems theory and data science: A unifying review of dynamic latent variable analytics and process monitoring. *Annual Reviews in Control*, 50: 29–48.

Rabanser, S.; Günnemann, S.; and Lipton, Z. C. 2019. Failing Loudly: An Empirical Study of Methods for Detecting Dataset Shift. *arXiv:1810.11953 [cs, stat]*. ArXiv: 1810.11953.

Ročková, V.; and George, E. 2016. The Spike-and-Slab LASSO. *Journal of the American Statistical Association*, 113(521): 431–444.

Schölkopf, B.; Smola, A.; and Müller, K.-R. 1997. Kernel principal component analysis. In Gerstner, W.; Germond, A.; Hasler, M.; and Nicoud, J.-D., eds., *Artificial Neural Networks — ICANN'97*, Lecture Notes in Computer Science, 583–588. Berlin, Heidelberg: Springer. ISBN 978-3-540-69620-9.

Teppola, P.; Mujunen, S.-P.; Minkkinen, P.; Puijola, T.; and Pursiheimo, P. 1998. Principal component analysis, contribution plots and feature weights in the monitoring of sequential process data from a paper machine's wet end. *Chemometrics and Intelligent Laboratory Systems*, 44(1): 307–317.

Theisen, M.; Dörgő, G.; Abonyi, J.; and Palazoglu, A. 2021. Sparse PCA Support Exploration of Process Structures for Decentralized Fault Detection. *Industrial & Engineering Chemistry Research*. Publisher: American Chemical Society.

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

Van den Kerkhof, P.; Vanlaer, J.; Gins, G.; and Van Impe, J. F. M. 2013. Analysis of smearing-out in contribution plot based fault isolation for Statistical Process Control. *Chemical Engineering Science*, 104: 285–293.

Wang, K.; Forbes, M. G.; Gopaluni, B.; Chen, J.; and Song, Z. 2019. Systematic Development of a New Variational Autoencoder Model Based on Uncertain Data for Monitoring Nonlinear Processes. *IEEE Access*, 7: 22554–22565. Conference Name: IEEE Access.

Yang, B.; Li, Q.; Chen, L.; and Shen, C. 2020. Bearing Fault Diagnosis Based on Multilayer Domain Adaptation. *Shock and Vibration*, 2020: e8873960. Publisher: Hindawi.

Yin, S.; Ding, S. X.; Haghani, A.; Hao, H.; and Zhang, P. 2012. A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. *Journal of Process Control*, 22(9): 1567–1581.

Yu, H.; Khan, F.; and Garaniya, V. 2016. A sparse PCA for nonlinear fault diagnosis and robust feature discovery of industrial processes. *AIChE Journal*, 62(5): 1494–1513. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/aic.15136.

Zhu, J.; Jiang, M.; and Liu, Z. 2022. Fault Detection and Diagnosis in Industrial Processes with Variational Autoencoder: A Comprehensive Study. *Sensors*, 22(1): 227. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.