FieldBox.ai
Artificial Intelligence for Industry

CAp 2021

Manifold exploration of industrial processes with Variational AutoEncoders

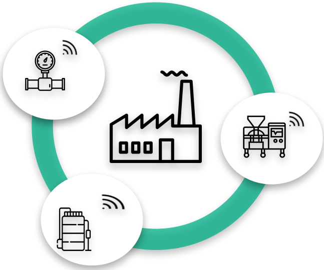

Brendan L'Ollivier


Sonia Tabti


Julien Budynek

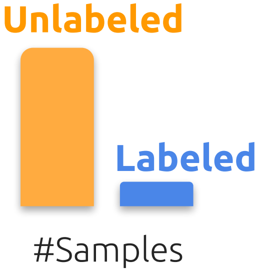
FieldBox.ai
Artificial Intelligence for Industry

Digitalization of the production lines



Increase of:

- the amount of data
- the diversity of sensors



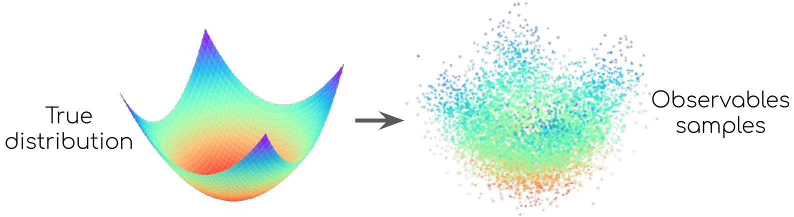
High cost of data labeling

⇒ Highly imbalanced ratio

between the two populations.

→ Unlabeled samples remains a poorly exploited source of information.

The manifold learning assumption



Observables data are sampled from of a low dimensional manifold that is embedded inside of a higher-dimensional vector space.

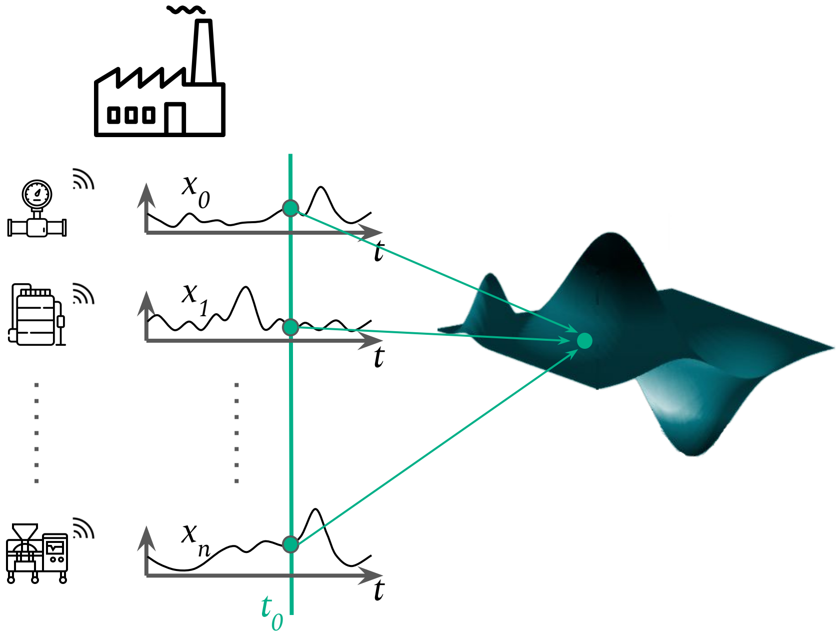
State of the art

Linear: PCA, MDS (Multidimensional Scaling)

Non-linear: Kernel PCA, ISOMAP, LLE (Locally Linear Embedding) and Laplacian Eigenmaps

Deep learning: Auto-Encoder, Variational Auto-Encoder

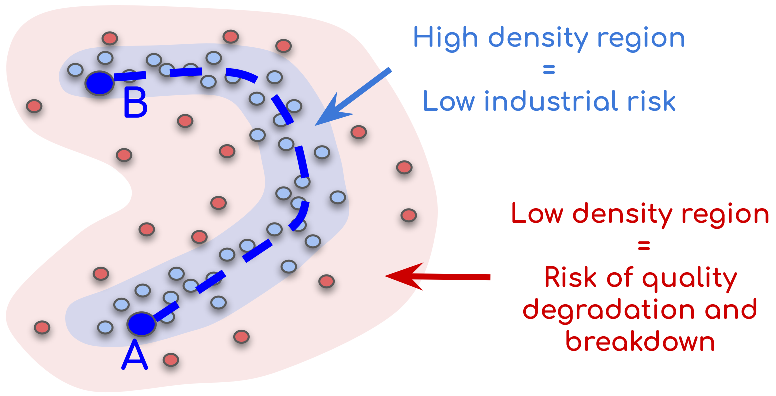
Manifold hypothesis on industrial data



For any date t_0, the interactions between the sensor variables $x_0, x_1, \ldots, x_n$ are smoothed enough to be modeled by a lower dimensional manifold.

Industrial problematic

Shifting a process from one setpoint A to another setpoint B raises the risk of applying inappropriate set of parameters.



Objectives of this work:

- Designing a scalable manifold learning model.
- Using the model to safely explore the manifold of process variables.

Contribution: Combining neighbor graph and VAE

Neighbor graph common issues:

- Lack of robustness to noise and high curvature of the space.
- Has an exponential time complexity in the dimension of the data.

How a VAE can mitigate these issues ?:

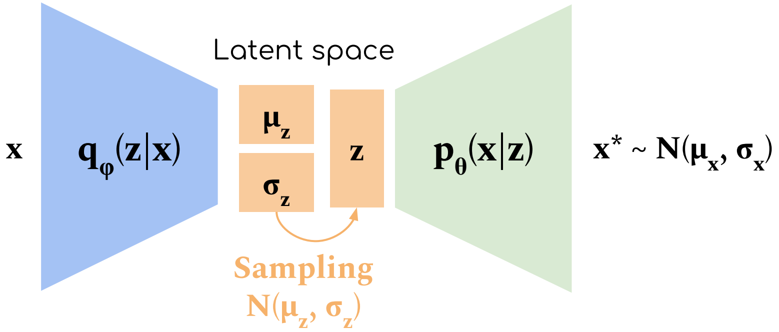
- The low dimensional latent space reduces the computational time significantly.
- The reconstruction loss can be used to filter the input samples, acting like a smoothing function.

Advantages of the graph framework:

- Provides fast path search algorithms.
- Makes the incorporation of business metrics very convenient.

Learning a statistical representation of non linear manifolds with VAE

Architecture of a Variational AutoEncoder (VAE)



$$(\boldsymbol{\mu_z}, \boldsymbol{\sigma_z}) = Encoder_\phi(\boldsymbol{x}) \qquad (\boldsymbol{\mu_x}, \boldsymbol{\sigma_x}) = Decoder_\theta(\boldsymbol{z})$$

$$q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu_z}, \boldsymbol{\sigma_z}) \qquad p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu_x}, \boldsymbol{\sigma_x})$$

With: $\mathbf{dim}(\mathbf{z}) << \mathbf{dim}(\mathbf{x}^*)$

Loss function

Loss = Reconstruction_Loss + Regularization_loss

$$\underbrace{\text{Reconstruction\_Loss}}_{\text{MSE}(\mathbf{x}, \boldsymbol{\mu}_\mathbf{x})} + \underbrace{\text{Regularization\_loss}}_{\text{KL Divergence}(\mathbf{z}, \mathbf{N}(0, 1))}$$

Smoothing noisy data with a VAE

For any input sample $\boldsymbol{x} \in \mathbb{R}^N$, the VAE computes its averaged version $\boldsymbol{\mu_x} = VAE(\boldsymbol{x})$.

And the quadratic error $\mathcal{E}(\boldsymbol{x})$ between $\boldsymbol{x}$ and $\boldsymbol{\mu_x}$ informs about the likelihood of $\boldsymbol{x}$.

$$\mathcal{E}(\boldsymbol{x}) = \sum_{n=1}^{N} \frac{1}{2} (x_n - \mu_{x_n})^2$$

$$\mathcal{L}(\boldsymbol{x}) \propto -\exp(\mathcal{E}(\boldsymbol{x}))$$

Smoothing: Building the nearest neighbor graph on the sublevel set of $\mathcal{E}$ with level e:

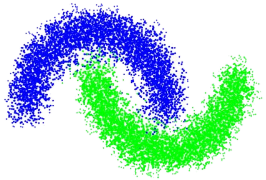
$$L_e^- = \{\boldsymbol{x} \in \mathbf{R}^N,\ \mathcal{E}(\boldsymbol{x}) \leq e\}$$

Experiment on the two moons dataset

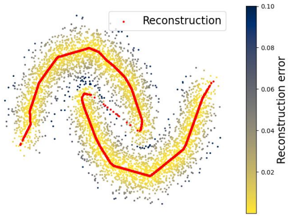
Filtering out points according to their reconstruction error:

- Removes the outliers from the scope of the graph computation,
- Reduces the likelihood of connecting the two clusters.

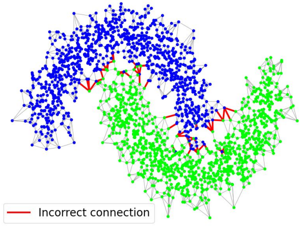
Noisy two moons



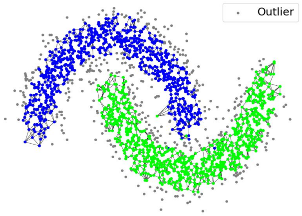
1D manifold learned by a VAE



Graph on original data

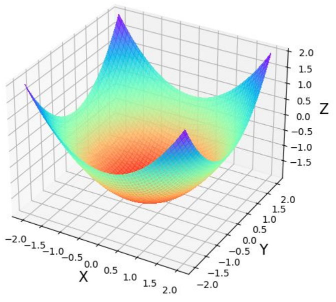


Graph on smoothed data

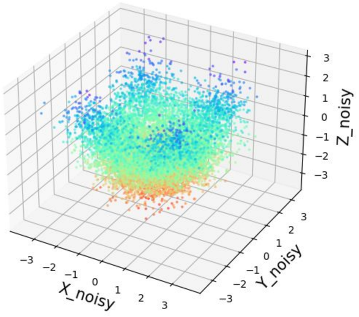


Experiment on a synthetic 3D parabola

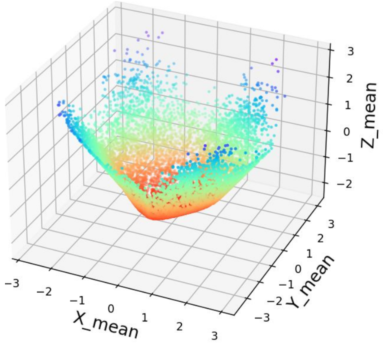
Training a VAE on random samples:



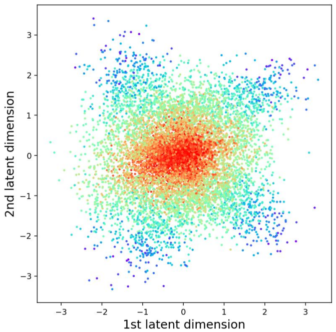
True manifold



Random samples



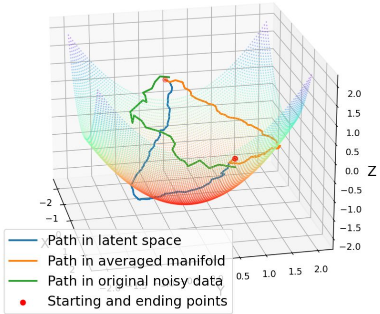
Learned manifold



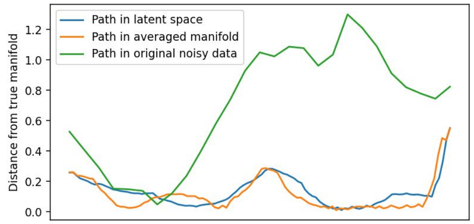
2D latent space

Comparison of trajectories between two diametrically opposite samples:

- Shortest paths (with A* algorithm) in the latent space and in the averaged data $\boldsymbol{\mu_x} = VAE(\boldsymbol{x})$ follow the underlying manifold.
- The path built on the original data doesn't respect the curvature



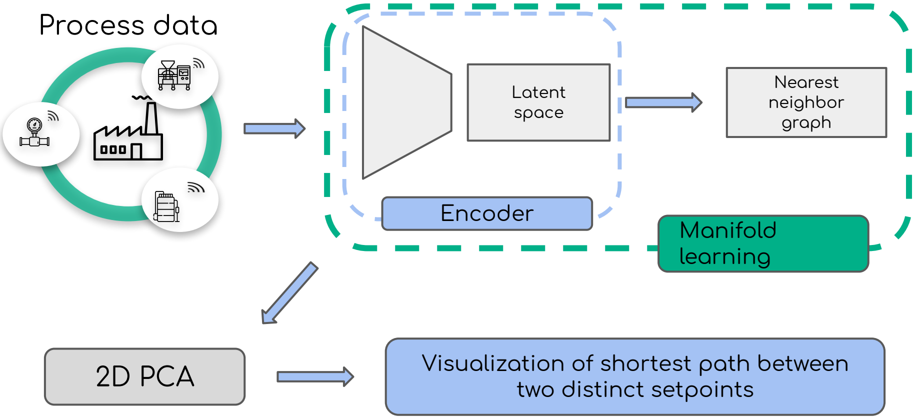
Comparison of trajectories embedded in the real space.



Comparison of distances between each point of the trajectories and the nearest point from the true manifold

Application on industrial data

Experimental pipeline:

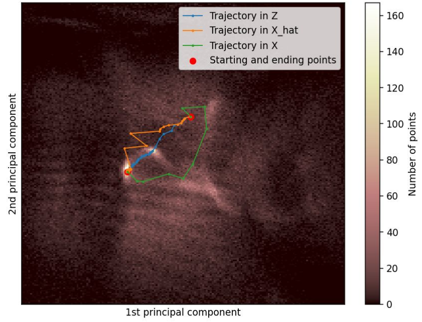


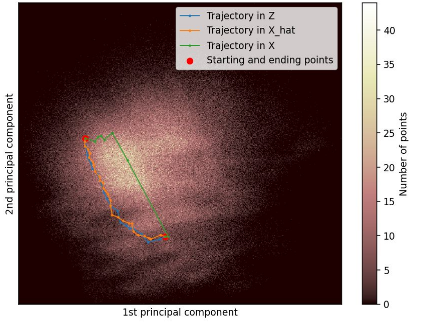
Three industrial datasets taken from Kaggle:

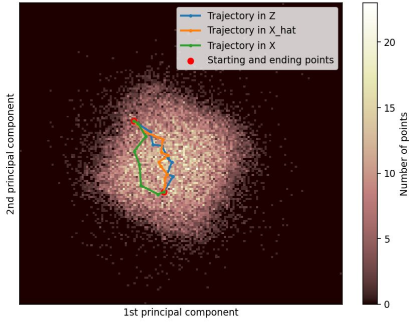
Source	#Features	#Samples	#Latent dimensions
Roasting Machine	17	2M	4
Bosch production line station L0S0	11	670K	6
Bosch production line station L1S24	169	180K	16

Results

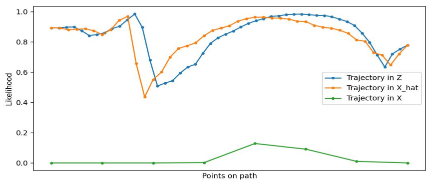
- The paths computed in the real space are sensitively different from the ones computed on the latent space and on the averaged data.
- The values of the likelihoods along each path show that points on the path computed with the real data tend to be less likely in the sense of the distribution learned by the VAE.

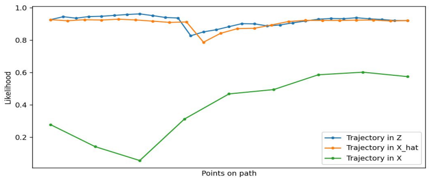


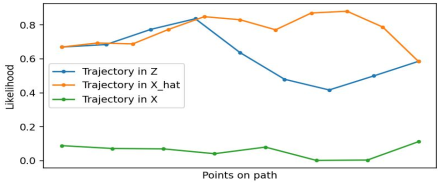




Shortest paths projected on the two first PCA components







Likelihood of intermediary points along the path

Incorporating business metrics into the weights of the graph:

- Any business metric can be incorporated into the weights in order to add an additional constraint on the shortest path search.
- Smoothing the distribution of the business metric with a regression model helps regularizing the path.

30k samples labelled with the output quality — Roastingmachine

↓

Weighted nearest neighbor graph$(w_{i,j})$

$$\text{target}_i = \left|\frac{\text{quality}_i - \text{quality}_i^{\text{ref}}}{\text{quality}_i^{\text{ref}}}\right|$$

$$w_{i,j} = d_{i,j} \times \left(\frac{\text{target}_i + \text{target}_j}{2}\right)^n$$

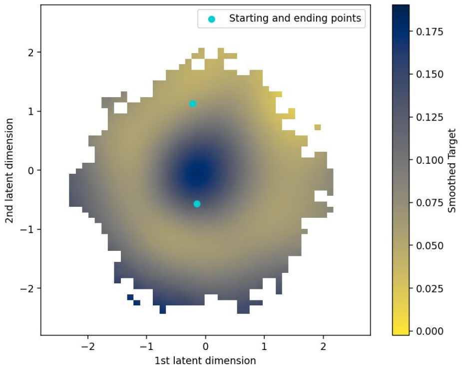
↓

Smoothing of the target with a Multi Layer Perceptron

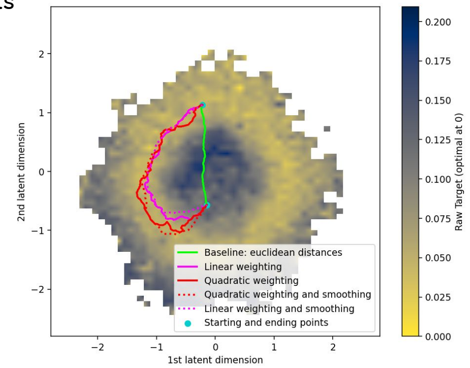
$$\text{smoothed\_target}_i = MLP(\text{target}_i)$$

Results

Smoothed target



Shortest path in a neighbor graph with custom weights



Main contributions

- Experimentation of the usage of a VAE as a smoothing step before the construction of a nearest neighbor graph.
- Exploration of the manifold of industrial processes with the resulting graph.
- Weights customization with a business metric

Limitations of the VAE

- The latent space doesn't preserve distances.
- Doesn't handle disconnected manifolds.

Perspectives

- Using the learned variance as metric for dynamically adapting the connectivity criterion in the graph.
- Extending the approach to categorical variables.
- Experimenting more sophisticated weighting formulas.

References

Cayton, Lawrence. "Algorithms for manifold learning." *Univ. of California at San Diego Tech. Rep* 12.1-17 (2005): 1.

Carreira-Perpinán, Miguel A., and Richard S. Zemel. "Proximity graphs for clustering and manifold learning." Advances in neural information processing systems 17 (2005): 225-232.

Carreira-Perpinán, Miguel A., and Richard S. Zemel. "Proximity graphs for clustering and manifold learning." *Advances in neural information processing systems* 17 (2005): 225-232.

Wolf, F. Alexander, et al. "PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells." *Genome biology* 20.1 (2019): 1-9.

McInnes, Leland, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction." *arXiv preprint arXiv:1802.03426* (2018).

Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." *arXiv preprint arXiv:1312.6114* (2013).

Thank you for your attention

blollivier@fieldbox.ai

Loss function of VAE

For any

$$\boldsymbol{x} = (x_1, x_2, ..., x_N) \in \mathbb{R}^N \text{, and}$$

$$\boldsymbol{z} = (z_1, z_2, ..., z_N) = \text{Encoder}(\boldsymbol{x}) \in \mathbb{R}^L$$

$$\log p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \sum_{n=1}^{N} \left(-\frac{1}{2}\log(2\pi\sigma_{x_n}^2) - \frac{1}{2\sigma_{x_n}^2}(x_n - \mu_{x_n})^2 \right)$$

$$\mathbb{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) = \sum_{l=1}^{L} -\frac{1}{2}\left(1 + \log\sigma_{z_l}^2 - \sigma_{z_l}^2 - \mu_{z_l}^2\right)$$

Architecture of the VAE network

```
vae = VAE(
    input_dim=input_dim,
    latent_dims=[laten_dim],
    enc_mlp_hidden_units_list=[
        [input_dim, int(input_dim/2)]
    ],
    activation='tanh',
    reconstruct_var=False
)
```

Task	Computational time* (Order of magnitude)
VAE training	5 - 10 min
5-nearest neighbors graph construction	10 sec
Finding shortest path in the graph	1 sec

* With Intel® Core™ i7-8850H CPU @ 2.60GHz × 12