Unsupervised Learning of Data Representations and Cluster Structures

Applications to Large-scale Health Monitoring of Turbofan Aircraft Engines

---

Apprentissage non supervisé de représentations de données et structures de partitionnement

Applications à la surveillance à grande échelle de turbofans

Florent Forest

PhD defense

Université Sorbonne Paris Nord
LIPN — CNRS UMR 7030
Safran Aircraft Engines

March 22, 2021

Introduction

Statistical learning

Aircraft engine health monitoring

Monitoring the **condition** of an engine to increase availability and safety, while reducing maintenance costs (e.g. *condition-based* or *predictive* maintenance). Knowledge of a machine's condition can be extracted from data.

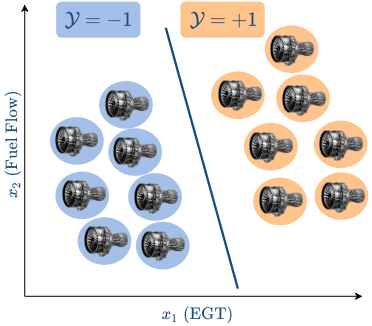
Aircraft engine health monitoring

Monitoring the **condition** of an engine to increase availability and safety, while reducing maintenance costs (e.g. *condition-based* or *predictive* maintenance). Knowledge of a machine's condition can be extracted from data.

Statistical learning

Supervised learning: $\{\mathbf{x}, y\}_1^N \in (\mathcal{X} \times \mathcal{Y})^N$

- Classification ($\mathcal{Y}$ is discrete, e.g. $\mathcal{Y} = \{-1, +1\}$)



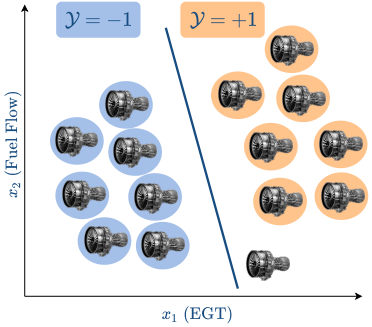
Aircraft engine health monitoring

Monitoring the **condition** of an engine to increase availability and safety, while reducing maintenance costs (e.g. *condition-based* or *predictive* maintenance). Knowledge of a machine's condition can be extracted from data.

Statistical learning

Supervised learning: $\{\mathbf{x}, y\}_1^N \in (\mathcal{X} \times \mathcal{Y})^N$

- Classification ($\mathcal{Y}$ is discrete, e.g. $\mathcal{Y} = \{-1, +1\}$)



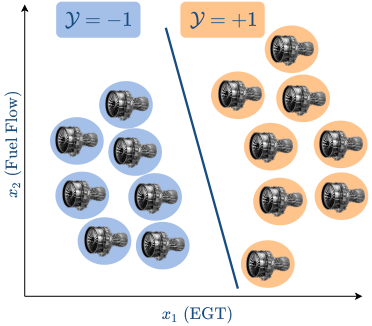
Aircraft engine health monitoring

Monitoring the **condition** of an engine to increase availability and safety, while reducing maintenance costs (e.g. *condition-based* or *predictive* maintenance). Knowledge of a machine's condition can be extracted from data.

Statistical learning

Supervised learning: $\{\mathbf{x}, y\}_1^N \in (\mathcal{X} \times \mathcal{Y})^N$

- Classification ($\mathcal{Y}$ is discrete, e.g. $\mathcal{Y} = \{-1, +1\}$)



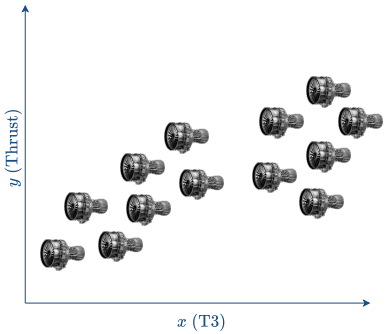
Aircraft engine health monitoring

Monitoring the **condition** of an engine to increase availability and safety, while reducing maintenance costs (e.g. *condition-based* or *predictive* maintenance). Knowledge of a machine's condition can be extracted from data.

Statistical learning

Supervised learning: $\{\mathbf{x}, y\}_1^N \in (\mathcal{X} \times \mathcal{Y})^N$

- Classification ($\mathcal{Y}$ is discrete, e.g. $\mathcal{Y} = \{-1, +1\}$)
- Regression ($\mathcal{Y}$ is continuous, e.g. $\mathcal{Y} = \mathbb{R}$)



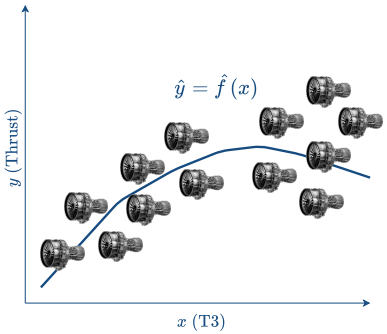
Aircraft engine health monitoring

Monitoring the **condition** of an engine to increase availability and safety, while reducing maintenance costs (e.g. *condition-based* or *predictive* maintenance). Knowledge of a machine's condition can be extracted from data.

Statistical learning

Supervised learning: $\{\mathbf{x}, y\}_1^N \in (\mathcal{X} \times \mathcal{Y})^N$

- Classification ($\mathcal{Y}$ is discrete, e.g. $\mathcal{Y} = \{-1, +1\}$)
- Regression ($\mathcal{Y}$ is continuous, e.g. $\mathcal{Y} = \mathbb{R}$)



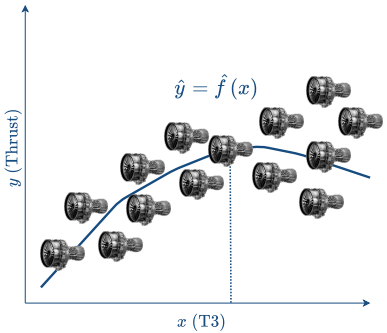
Aircraft engine health monitoring

Monitoring the **condition** of an engine to increase availability and safety, while reducing maintenance costs (e.g. *condition-based* or *predictive* maintenance). Knowledge of a machine's condition can be extracted from data.

Statistical learning

Supervised learning: $\{\mathbf{x}, y\}_1^N \in (\mathcal{X} \times \mathcal{Y})^N$

- Classification ($\mathcal{Y}$ is discrete, e.g. $\mathcal{Y} = \{-1, +1\}$)
- Regression ($\mathcal{Y}$ is continuous, e.g. $\mathcal{Y} = \mathbb{R}$)



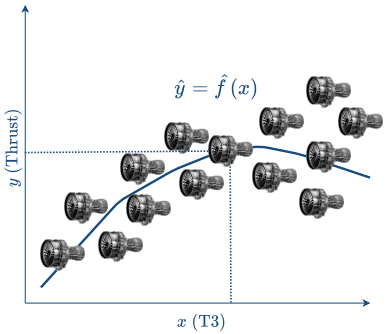
Aircraft engine health monitoring

Monitoring the **condition** of an engine to increase availability and safety, while reducing maintenance costs (e.g. *condition-based* or *predictive* maintenance). Knowledge of a machine's condition can be extracted from data.

Statistical learning

Supervised learning: $\{\mathbf{x}, y\}_1^N \in (\mathcal{X} \times \mathcal{Y})^N$

- Classification ($\mathcal{Y}$ is discrete, e.g. $\mathcal{Y} = \{-1, +1\}$)
- Regression ($\mathcal{Y}$ is continuous, e.g. $\mathcal{Y} = \mathbb{R}$)

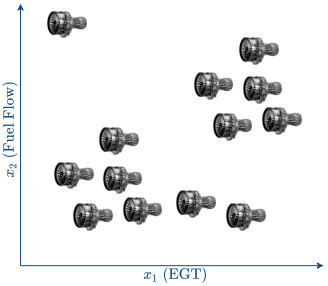


Unsupervised learning

Aircraft engine health monitoring

Monitoring the **condition** of an engine to increase availability and safety, while reducing maintenance costs (e.g. *condition-based* or *predictive* maintenance). Knowledge of a machine's condition can be extracted from data.

No labels (rare events) $\rightarrow$ **Unsupervised learning setting**



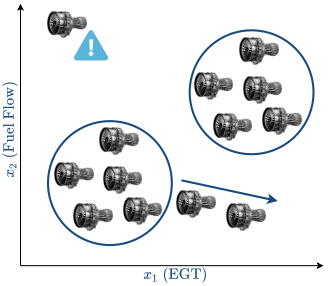
Unsupervised learning: $\{\mathbf{x}\}_1^N \in \mathcal{X}^N$

- **Visualization, interpretability**

Aircraft engine health monitoring

Monitoring the **condition** of an engine to increase availability and safety, while reducing maintenance costs (e.g. *condition-based* or *predictive* maintenance). Knowledge of a machine's condition can be extracted from data.

No labels (rare events) → **Unsupervised learning setting**



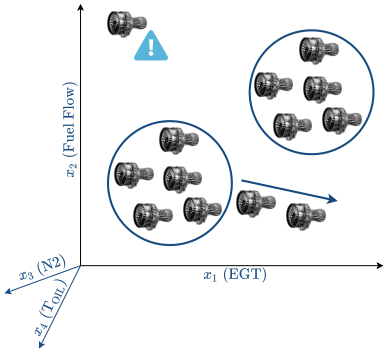
Unsupervised learning: $\{\mathbf{x}\}_1^N \in \mathcal{X}^N$

- **Visualization, interpretability**
- **Clustering**, trend monitoring, anomaly detection

Aircraft engine health monitoring

Monitoring the **condition** of an engine to increase availability and safety, while reducing maintenance costs (e.g. *condition-based* or *predictive* maintenance). Knowledge of a machine's condition can be extracted from data.

No labels (rare events) $\rightarrow$ **Unsupervised learning setting**



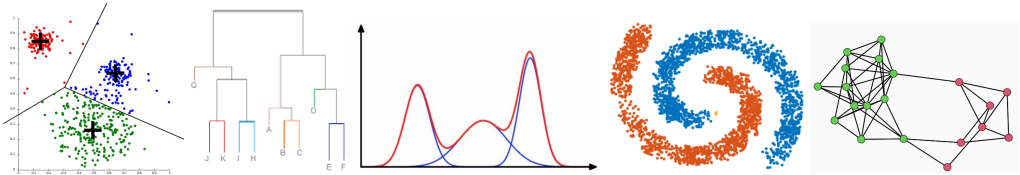
Unsupervised learning: $\{\mathbf{x}\}_1^N \in \mathcal{X}^N$

- **Visualization, interpretability**
- **Clustering**, trend monitoring, anomaly detection
- **Dimensionality reduction and Representation learning**

Overview of clustering

Clustering

The goal of data clustering is to discover the natural grouping of a set of [...] objects [Jain, 2010]. Partitioning of data into groups so that similar elements share the same cluster and dissimilar elements are separated into different clusters [Ben-David, 2018].



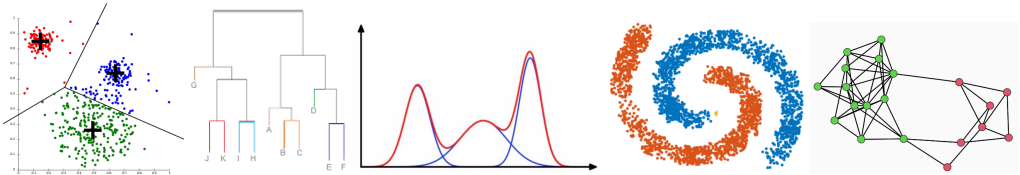
↑ Partitive, **prototype**-based methods (e.g. K-means)

$$\{\mathbf{x}_i\}_1^N, \mathbf{x}_i \in \mathbb{R}^P, K \longrightarrow \{\mathbf{m}_k\}_1^K, \mathbf{m}_k \in \mathbb{R}^P, \mathcal{C}_K = \{C_k\}_1^K, \forall \mathbf{x} \in \mathbb{R}^P \; \mathbf{x} \in C_k \iff \text{argmin}_j \; ||\mathbf{x} - \mathbf{m}_j||_2 = k$$

Overview of clustering

Clustering

The goal of data clustering is to discover the natural grouping of a set of [...] objects [Jain, 2010]. Partitioning of data into groups so that similar elements share the same cluster and dissimilar elements are separated into different clusters [Ben-David, 2018].

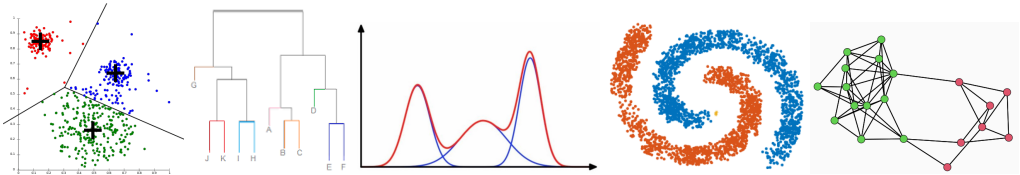


↑ Hierarchical methods (e.g. agglomerative HC)

Overview of clustering

Clustering

The goal of data clustering is to discover the natural grouping of a set of [...] objects [Jain, 2010]. Partitioning of data into groups so that similar elements share the same cluster and dissimilar elements are separated into different clusters [Ben-David, 2018].

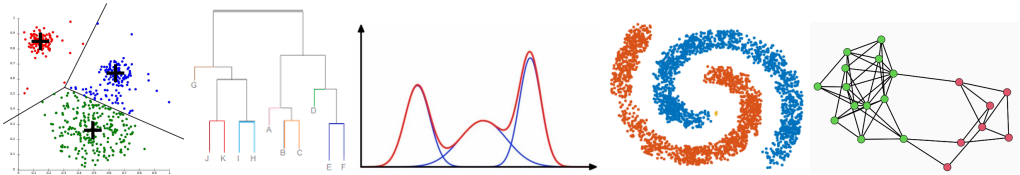


↑ Model-based methods

Overview of clustering

Clustering

The goal of data clustering is to discover the natural grouping of a set of [...] objects [Jain, 2010]. Partitioning of data into groups so that similar elements share the same cluster and dissimilar elements are separated into different clusters [Ben-David, 2018].

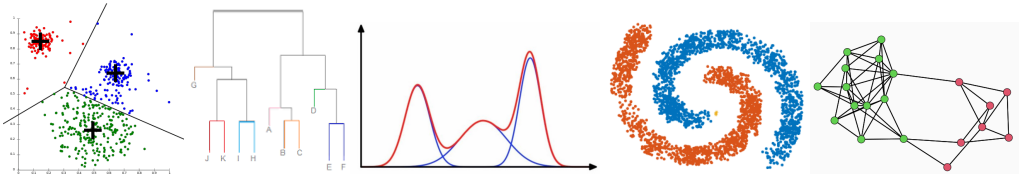


↑ Density-based methods

Overview of clustering

Clustering

The goal of data clustering is to discover the natural grouping of a set of [...] objects [Jain, 2010]. Partitioning of data into groups so that similar elements share the same cluster and dissimilar elements are separated into different clusters [Ben-David, 2018].



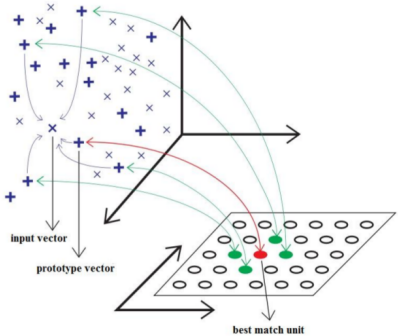
↑ Graph clustering

Topology-preserving clustering algorithms

Topology-preserving maps

A topology-preserving algorithm is a transformation that preserves similarities, or similarity orderings, of the points in the input space when they are mapped into the output space [Martinetz and Schulten, 1994].

Prototype-based methods inspired from biological cells:

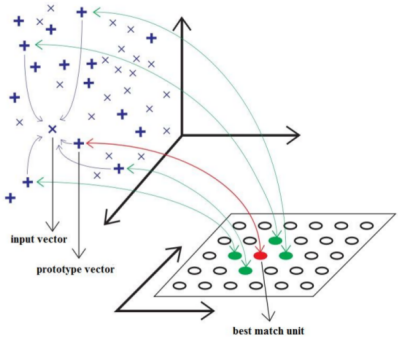


Topology-preserving maps

A topology-preserving algorithm is a transformation that preserves similarities, or similarity orderings, of the points in the input space when they are mapped into the output space [Martinetz and Schulten, 1994].

Prototype-based methods inspired from biological cells:

- **Self-Organizing Maps (SOM)** [Kohonen, 1982]

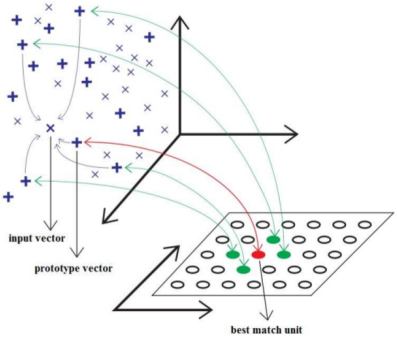


Topology-preserving maps

A topology-preserving algorithm is a transformation that preserves similarities, or similarity orderings, of the points in the input space when they are mapped into the output space [Martinetz and Schulten, 1994].

Prototype-based methods inspired from biological cells:

- **Self-Organizing Maps (SOM)** [Kohonen, 1982]
- Neural Gas and growing cell structures
  [Martinetz and Schulten, 1991, Fritzke, 1995]

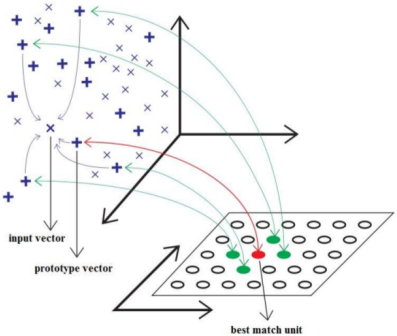


Topology-preserving maps

A topology-preserving algorithm is a transformation that preserves similarities, or similarity orderings, of the points in the input space when they are mapped into the output space [Martinetz and Schulten, 1994].

Prototype-based methods inspired from biological cells:

- **Self-Organizing Maps (SOM)** [Kohonen, 1982]
- Neural Gas and growing cell structures [Martinetz and Schulten, 1991, Fritzke, 1995]
- Probabilistic maps and Generative Topographic Mapping [Anouar et al., 1998, Bishop et al., 1998]

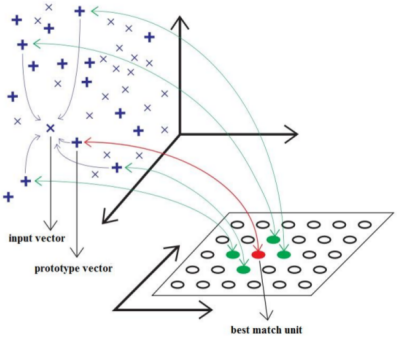


Topology-preserving maps

A topology-preserving algorithm is a transformation that preserves similarities, or similarity orderings, of the points in the input space when they are mapped into the output space [Martinetz and Schulten, 1994].

Prototype-based methods inspired from biological cells:

- **Self-Organizing Maps (SOM)** [Kohonen, 1982]
- Neural Gas and growing cell structures [Martinetz and Schulten, 1991, Fritzke, 1995]
- Probabilistic maps and Generative Topographic Mapping [Anouar et al., 1998, Bishop et al., 1998]
- and many variants.



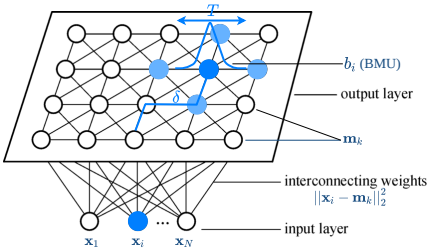
Distortion loss

$$\mathcal{L}_{\text{SOM}}(\{\mathbf{m}_k\}_1^K, \{b_i\}_1^N) := \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\mathcal{K}^T\left(\delta(b_i, k)\right)||\mathbf{x}_i - \mathbf{m}_k||_2^2$$

Kohonen algorithm:

1. Find best-matching unit $b_i = \text{argmin}_k ||\mathbf{x}_i - \mathbf{m}_k||_2^2$
2. Update each prototype vector $\mathbf{m}_k \leftarrow \mathbf{m}_k + \alpha\mathcal{K}^T\left(\delta(b_i, k)\right)(\mathbf{x}_i - \mathbf{m}_k)$
3. Decrease temperature (neighborhood radius) e.g. $T(t) = T_{max}\left(\frac{T_{min}}{T_{max}}\right)^{t/\text{iterations}}$

   where $\mathcal{K}^T(d) = e^{-d^2/T^2}$ and $\delta(\cdot,\cdot)$ is the distance on the lattice (ℓ_1)



- **Neighborhood relationship** between prototypes, organized as a low-dimensional lattice.
- **Self-organization**: competitive learning process inspired from biological cells.
- Visualization capabilities, interpretability and computational efficiency.
- Applied to aircraft engine health monitoring
  [Cottrell et al., 2009, Côme et al., 2010a, Côme et al., 2010b, Côme et al., 2011, Faure et al., 2017].

Research challenges

Subject: Unsupervised statistical learning methods and their applications to health monitoring of aircraft engines at an industrial scale.

Research challenges

Subject: Unsupervised statistical learning methods and their applications to health monitoring of aircraft engines at an industrial scale.

Theoretical challenges

1. How to learn representations to effectively cluster complex data?

Standard algorithms are ineffective on complex data sets (high-dimensional, noisy, redundant, correlated features) because **cluster structure is hidden in lower-dimensional subspaces** or **unobserved latent spaces**.

$\rightarrow$ Links between cluster structure and representation.

2. How to evaluate clustering algorithms?

$\rightarrow$ Model selection and the very definition of structure in clustering.

Research challenges

Subject: Unsupervised statistical learning methods and their applications to health monitoring of aircraft engines at an industrial scale.

Theoretical challenges

1. How to learn representations to effectively cluster complex data?

Standard algorithms are ineffective on complex data sets (high-dimensional, noisy, redundant, correlated features) because **cluster structure is hidden in lower-dimensional subspaces** or **unobserved latent spaces**.

→ Links between cluster structure and representation.

2. How to evaluate clustering algorithms?

→ Model selection and the very definition of structure in clustering.

Technical challenges

3. How to develop scalable engine health monitoring methodologies?

To be useful in industrial settings, methods must **scale to massive amounts of data** and be **flexible**.

→ **Engineering** challenges.

Table of contents

Unsupervised representation learning for self-organized clustering

High-dimensional data clustering

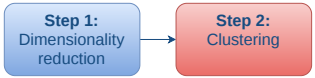
Dimensionality reduction (DR):

- Feature selection
- Feature transformation
  - **Linear DR**: approximation by a hyperplane, e.g. principal component analysis (PCA)
  - **Non-linear DR**: approximation by a lower-dimensional manifold [Lee and Verleysen, 2007]

High-dimensional data clustering

Dimensionality reduction (DR):

- Feature selection
- Feature transformation
  - **Linear DR**: approximation by a hyperplane, e.g. principal component analysis (PCA)
  - **Non-linear DR**: approximation by a lower-dimensional manifold [Lee and Verleysen, 2007]



High-dimensional data clustering

Dimensionality reduction (DR):

- Feature selection
- Feature transformation
  - **Linear DR**: approximation by a hyperplane, e.g. principal component analysis (PCA)
  - **Non-linear DR**: approximation by a lower-dimensional manifold [Lee and Verleysen, 2007]

Joint dimensionality reduction and clustering

Clustering with joint feature selection:

- Sparse subspace clustering [Elhamifar and Vidal, 2013]
- Sparse *K*-means [Witten and Tibshirani, 2010, Sun et al., 2012, Chavent et al., 2020], EM [Bouveyron et al., 2007]
- Subspace or weighted SOM [Kaly et al., 2004, Benabdeslem and Lebbah, 2007] and NG [Attaoui et al., 2020]

Clustering with joint linear DR:

- Projected and discriminative clustering (LDA)
  [De Soete and Carroll, 1994, De La Torre and Kanade, 2006, Ding and Li, 2007, Ye, 2007, Wang et al., 2019]

Representation learning and clustering

Non-linear DR using feature learning with neural networks (*deep learning*) [Bengio, 2012]

Representation learning and clustering

Non-linear DR using feature learning with neural networks (*deep learning*) [Bengio, 2012]

- **Supervised RL:**

$$\mathcal{X} \xrightarrow[\text{representation learning}]{} \mathcal{Z} \xrightarrow[\text{classification/regression}]{} \mathcal{Y}$$

Minimize error of predictor $\hat{f}$: $\mathcal{L}(\hat{f}(\mathbf{z}), y)$

$\mathcal{Y} = \{\text{cat}, \text{dog}\}$

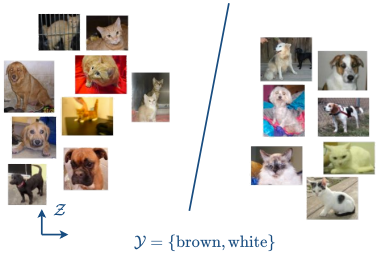
Representation learning and clustering

Non-linear DR using feature learning with neural networks (*deep learning*) [Bengio, 2012]

- **Supervised RL:**

$$\mathcal{X} \underset{\text{representation learning}}{\longrightarrow} \mathcal{Z} \underset{\text{classification/regression}}{\longrightarrow} \mathcal{Y}$$

Minimize error of predictor $\hat{f}$: $\mathcal{L}(\hat{f}(\mathbf{z}), y)$



Representation learning and clustering

Non-linear DR using feature learning with neural networks (*deep learning*) [Bengio, 2012]

- **Supervised RL:**

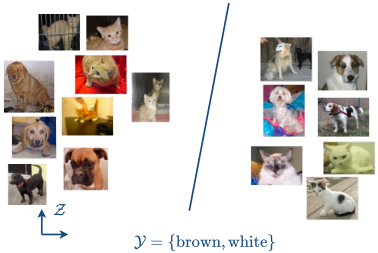
$$\mathcal{X} \underset{\text{representation learning}}{\longrightarrow} \mathcal{Z} \underset{\text{classification/regression}}{\longrightarrow} \mathcal{Y}$$

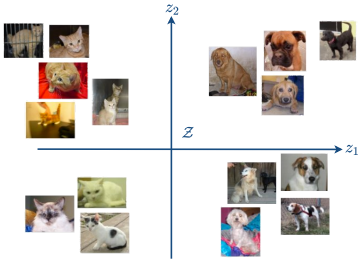
  Minimize error of predictor $\hat{f}$: $\mathcal{L}(\hat{f}(\mathbf{z}), y)$

- **Unsupervised RL:**

$$\mathcal{X} \underset{\text{representation learning}}{\longrightarrow} \mathcal{Z}$$

  Optimize some representation quality criterion: $\mathcal{L}_{\text{R}}(\mathbf{x}, \mathbf{z})$





Representation learning and clustering

Non-linear DR using feature learning with neural networks (*deep learning*) [Bengio, 2012]

- **Supervised RL:**

$$\mathcal{X} \underset{\text{representation learning}}{\longrightarrow} \mathcal{Z} \underset{\text{classification/regression}}{\longrightarrow} \mathcal{Y}$$

  Minimize error of predictor $\hat{f}$: $\mathcal{L}(\hat{f}(\mathbf{z}), y)$

- **Unsupervised RL:**

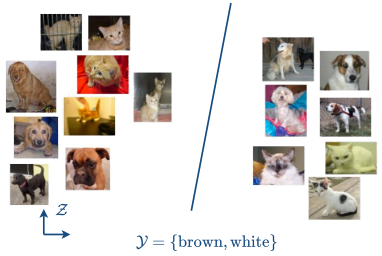
$$\mathcal{X} \underset{\text{representation learning}}{\longrightarrow} \mathcal{Z}$$

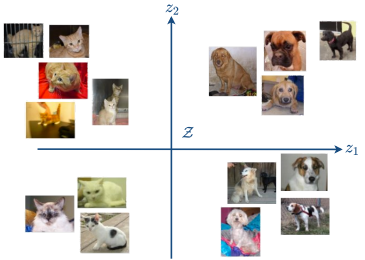
  Optimize some representation quality criterion: $\mathcal{L}_{\text{R}}(\mathbf{x}, \mathbf{z})$

- **Deep clustering:** consider RL and clustering as a **joint task** and learn clustering-friendly representations.

$$\mathcal{X} \underset{\text{representation learning}}{\longrightarrow} \mathcal{Z} \underset{\text{clustering}}{\longrightarrow} \mathcal{C}_K$$

$$\mathcal{L}(\mathbf{x}, \mathbf{z}, \mathcal{C}_K) := \underbrace{\mathcal{L}_{\text{R}}(\mathbf{x}, \mathbf{z})}_{\text{representation quality}} + \underbrace{\gamma}_{\text{hyperparameter}} \underbrace{\mathcal{L}_{\text{C}}(\mathbf{z}, \mathcal{C}_K)}_{\text{clustering loss}}$$

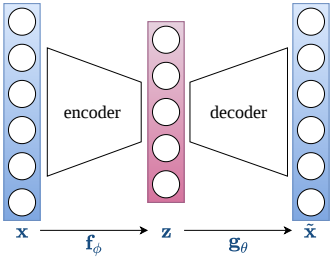




Autoencoder

Neural network trained to reconstruct its inputs in order to extract meaningful intermediate representations.

1. **encoder**, mapping the input to a latent representation (*code*): $\mathbf{z} = \mathbf{f}_\phi(\mathbf{x}) \in \mathbb{R}^L$
2. **decoder**, mapping the code back to the input space: $\tilde{\mathbf{x}} = \mathbf{g}_\theta(\mathbf{z}) \in \mathbb{R}^P$

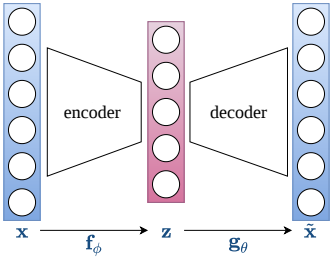


What is a *good* representation?

Autoencoder

Neural network trained to reconstruct its inputs in order to extract meaningful intermediate representations.

1. **encoder**, mapping the input to a latent representation (*code*): $\mathbf{z} = \mathbf{f}_\phi(\mathbf{x}) \in \mathbb{R}^L$
2. **decoder**, mapping the code back to the input space: $\tilde{\mathbf{x}} = \mathbf{g}_\theta(\mathbf{z}) \in \mathbb{R}^P$



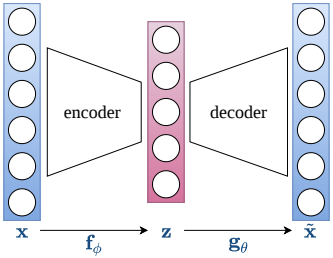
What is a *good* representation?

Maximize mutual information between $\mathbf{x}$ and $\mathbf{z}$

Autoencoder

Neural network trained to reconstruct its inputs in order to extract meaningful intermediate representations.

1. **encoder**, mapping the input to a latent representation (*code*): $\mathbf{z} = \mathbf{f}_{\phi}(\mathbf{x}) \in \mathbb{R}^{L}$
2. **decoder**, mapping the code back to the input space: $\tilde{\mathbf{x}} = \mathbf{g}_{\theta}(\mathbf{z}) \in \mathbb{R}^{P}$



What is a *good* representation?

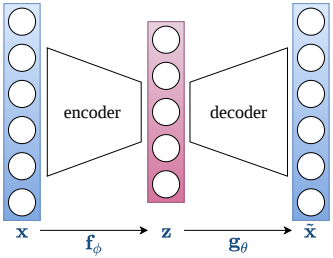
Maximize mutual information between $\mathbf{x}$ and $\mathbf{z}$

$$\max_{\phi,\theta} \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})]$$

Autoencoder

Neural network trained to reconstruct its inputs in order to extract meaningful intermediate representations.

1. **encoder**, mapping the input to a latent representation (*code*): $\mathbf{z} = \mathbf{f}_{\phi}(\mathbf{x}) \in \mathbb{R}^L$
2. **decoder**, mapping the code back to the input space: $\tilde{\mathbf{x}} = \mathbf{g}_{\theta}(\mathbf{z}) \in \mathbb{R}^P$



What is a *good* representation?

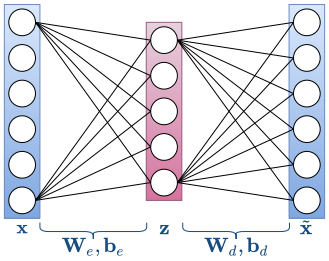
Maximize mutual information between $\mathbf{x}$ and $\mathbf{z}$

$$\max_{\phi,\theta} \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})] \rightarrow \min_{\phi,\theta} \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x}_i - \tilde{\mathbf{x}}_i||_2^2 \text{ (MSE)}$$

Autoencoder

Neural network trained to reconstruct its inputs in order to extract meaningful intermediate representations.

1. **encoder**, mapping the input to a latent representation (*code*): $\mathbf{z} = \mathbf{f}_{\phi}(\mathbf{x}) \in \mathbb{R}^L$
2. **decoder**, mapping the code back to the input space: $\tilde{\mathbf{x}} = \mathbf{g}_{\theta}(\mathbf{z}) \in \mathbb{R}^P$



What is a *good* representation?

Maximize mutual information between $\mathbf{x}$ and $\mathbf{z}$

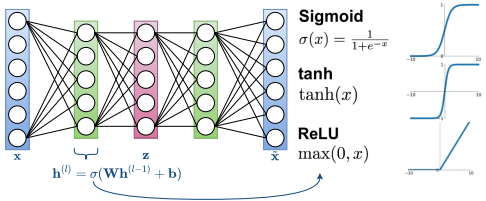
$\max_{\phi,\theta} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] \rightarrow \min_{\phi,\theta} \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x}_i - \tilde{\mathbf{x}}_i||_2^2$ (MSE)

- Linear: equivalent to PCA

Autoencoder

Neural network trained to reconstruct its inputs in order to extract meaningful intermediate representations.

1. **encoder**, mapping the input to a latent representation (*code*): $\mathbf{z} = \mathbf{f}_{\phi}(\mathbf{x}) \in \mathbb{R}^L$
2. **decoder**, mapping the code back to the input space: $\tilde{\mathbf{x}} = \mathbf{g}_{\theta}(\mathbf{z}) \in \mathbb{R}^P$



What is a *good* representation?

Maximize mutual information between $\mathbf{x}$ and $\mathbf{z}$

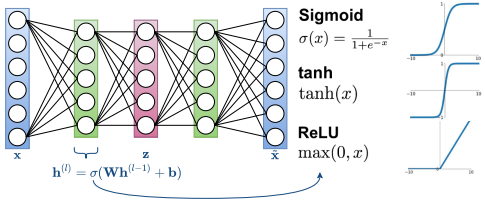
$$\max_{\phi,\theta} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] \rightarrow \min_{\phi,\theta} \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x}_i - \tilde{\mathbf{x}}_i||_2^2 \text{ (MSE)}$$

- Linear: equivalent to PCA
- **Non-linear deep AE**

Autoencoder

Neural network trained to reconstruct its inputs in order to extract meaningful intermediate representations.

1. **encoder**, mapping the input to a latent representation (*code*): $\mathbf{z} = \mathbf{f}_{\phi}(\mathbf{x}) \in \mathbb{R}^L$
2. **decoder**, mapping the code back to the input space: $\tilde{\mathbf{x}} = \mathbf{g}_{\theta}(\mathbf{z}) \in \mathbb{R}^P$



What is a *good* representation?

Maximize mutual information between $\mathbf{x}$ and $\mathbf{z}$

$\max_{\phi,\theta} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] \rightarrow \min_{\phi,\theta} \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x}_i - \tilde{\mathbf{x}}_i||_2^2$ (MSE)

- Linear: equivalent to PCA
- **Non-linear deep AE**

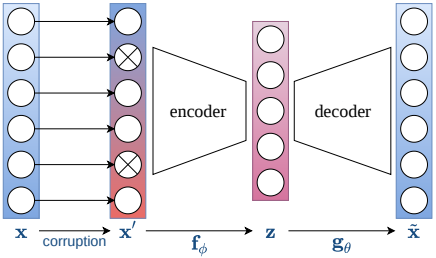
Regularization:

- **Undercomplete**

Autoencoder

Neural network trained to reconstruct its inputs in order to extract meaningful intermediate representations.

1. **encoder**, mapping the input to a latent representation (*code*): $\mathbf{z} = \mathbf{f}_\phi(\mathbf{x}) \in \mathbb{R}^L$
2. **decoder**, mapping the code back to the input space: $\tilde{\mathbf{x}} = \mathbf{g}_\theta(\mathbf{z}) \in \mathbb{R}^P$



What is a *good* representation?

Maximize mutual information between $\mathbf{x}$ and $\mathbf{z}$

$\max_{\phi,\theta} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] \rightarrow \min_{\phi,\theta} \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{x}_i - \tilde{\mathbf{x}}_i||_2^2$ (MSE)

- Linear: equivalent to PCA
- **Non-linear deep AE**

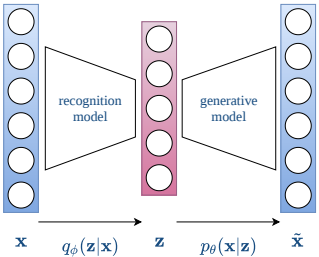
Regularization:

- **Undercomplete**
- Denoising

Autoencoder

Neural network trained to reconstruct its inputs in order to extract meaningful intermediate representations.

1. **encoder**, mapping the input to a latent representation (*code*): $\mathbf{z} = \mathbf{f}_{\phi}(\mathbf{x}) \in \mathbb{R}^L$
2. **decoder**, mapping the code back to the input space: $\tilde{\mathbf{x}} = \mathbf{g}_{\theta}(\mathbf{z}) \in \mathbb{R}^P$



What is a *good* representation?

Maximize mutual information between $\mathbf{x}$ and $\mathbf{z}$

$\max_{\phi,\theta} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] \rightarrow \min_{\phi,\theta} \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{x}_i - \tilde{\mathbf{x}}_i||_2^2$ (MSE)

- Linear: equivalent to PCA
- **Non-linear deep AE**

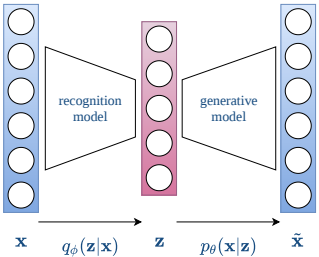
Regularization:

- **Undercomplete**
- Variational
- Denoising

Autoencoder

Neural network trained to reconstruct its inputs in order to extract meaningful intermediate representations.

1. **encoder**, mapping the input to a latent representation (*code*): $\mathbf{z} = \mathbf{f}_{\phi}(\mathbf{x}) \in \mathbb{R}^L$
2. **decoder**, mapping the code back to the input space: $\tilde{\mathbf{x}} = \mathbf{g}_{\theta}(\mathbf{z}) \in \mathbb{R}^P$



What is a *good* representation?

Maximize mutual information between $\mathbf{x}$ and $\mathbf{z}$

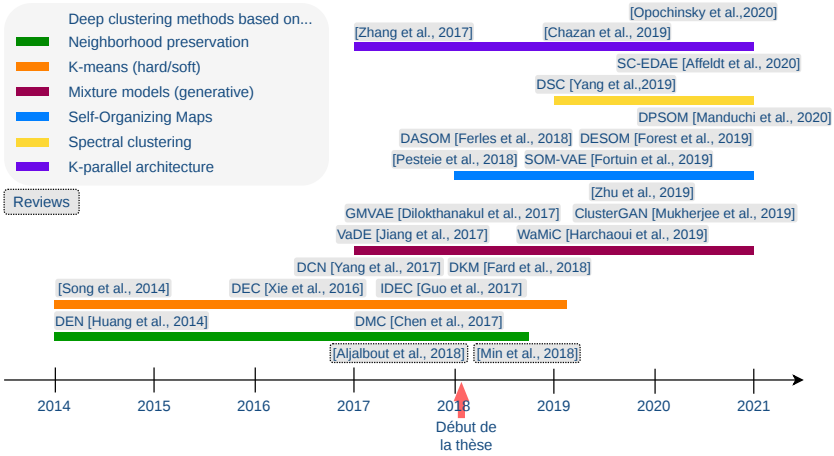
$$\max_{\phi,\theta} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] \rightarrow \min_{\phi,\theta} \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{x}_i - \tilde{\mathbf{x}}_i||_2^2 \text{ (MSE)}$$

- Linear: equivalent to PCA
- **Non-linear deep AE**

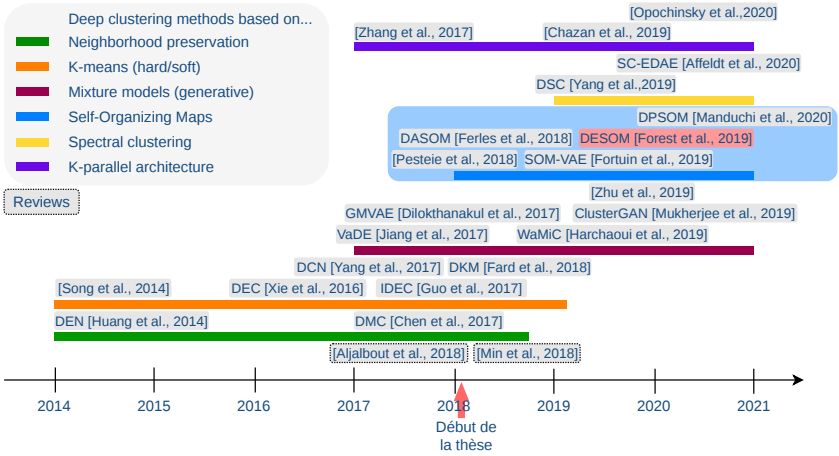
Regularization:

- **Undercomplete**
- Variational
- Denoising
- Sparsity, weight decay, etc.

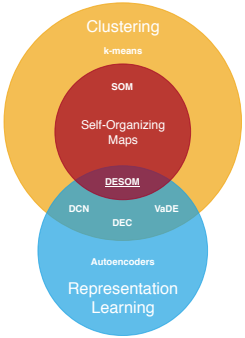
Literature timeline



Literature timeline



Deep Embedded SOM [Forest et al., 2019b, Forest et al., 2019a, Forest et al., 2020b]

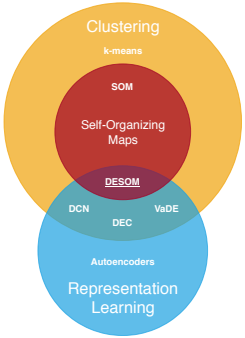


Proposition

The Deep Embedded SOM (DESOM) performs joint training of a **deep autoencoder** and a **SOM**.

- SOM prototypes are learned in **latent space**.
- Self-organization and representation learning are achieved as a **joint task**.
- The model is trained end-to-end using minibatch **stochastic gradient descent (SGD)**.
- The objective is to learn SOM-friendly representations.

Deep Embedded SOM [Forest et al., 2019b, Forest et al., 2019a, Forest et al., 2020b]



Proposition

The Deep Embedded SOM (DESOM) performs joint training of a **deep autoencoder** and a **SOM**.

- SOM prototypes are learned in **latent space**.
- Self-organization and representation learning are achieved as a **joint task**.
- The model is trained end-to-end using minibatch **stochastic gradient descent (SGD)**.
- The objective is to learn SOM-friendly representations.

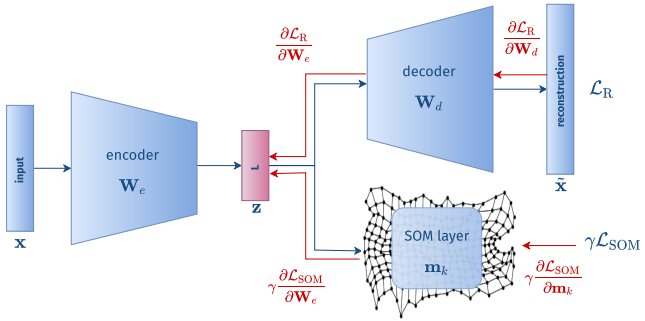
$$\mathcal{L}_{\text{DESOM}}(\mathbf{W_e}, \mathbf{W_d}, \{\mathbf{m}_k\}_1^K, \{b_i\}_1^N) := \mathcal{L}_{\text{R}}(\mathbf{W_e}, \mathbf{W_d}) + \gamma \mathcal{L}_{\text{SOM}}(\mathbf{W_e}, \{\mathbf{m}_k\}_1^K, \{b_i\}_1^N)$$

$$= \underbrace{\frac{1}{N}\sum_{i=1}^{N} ||\tilde{\mathbf{x}}_i - \mathbf{x}_i||_2^2}_{\text{MSE reconstruction loss}} + \gamma \underbrace{\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} \mathcal{K}^T\left(\delta(b_i, k)\right) ||\mathbf{z}_i - \mathbf{m}_k||_2^2}_{\text{SOM distortion loss}}$$

DESOM architecture and gradients flow

$$\mathcal{L}_{\text{DESOM}}(\mathbf{W_e}, \mathbf{W_d}, \{\mathbf{m}_k\}_1^K, \{b_i\}_1^N) = \mathcal{L}_{\text{R}}(\mathbf{W_e}, \mathbf{W_d}) + \gamma \mathcal{L}_{\text{SOM}}(\mathbf{W_e}, \{\mathbf{m}_k\}_1^K, \{b_i\}_1^N)$$

$$= \frac{1}{N} \sum_{i=1}^{N} ||\tilde{\mathbf{x}}_i - \mathbf{x}_i||_2^2 + \gamma \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \mathcal{K}^T(\delta(b_i, k)) \, ||\mathbf{z}_i - \mathbf{m}_k||_2^2$$



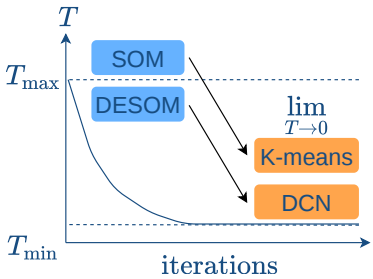
Training procedure

- Initialize $\mathbf{W_e}$, $\mathbf{W_d}$, $\{\mathbf{m}_k\}_1^K$
- Iterate following steps:
  1. Load next training batch
  2. Encode batch using encoder
  3. $T \leftarrow T_{max} \left(T_{min}/T_{max}\right)^{t/iterations}$
  4. Compute and fix the weight terms $w_{i,k} = \mathcal{K}^T(\delta(b_i, k))$
  5. Update parameters $\mathbf{W_e}$, $\mathbf{W_d}$, $\{\mathbf{m}_k\}$ by taking a SGD step

Convergence of the model

$$\lim_{T\to 0} \mathcal{L}_{\text{DESOM}}(\mathbf{W_e}, \mathbf{W_d}, \{\mathbf{m}_k\}_1^K, \{b_i\}_1^N) = \frac{1}{N}\sum_{i=1}^{N} ||\tilde{\mathbf{x}}_i - \mathbf{x}_i||_2^2 + \gamma \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{z}_i - \mathbf{m}_{b_i}||_2^2$$
$$= \mathcal{L}_{\text{R}}(\mathbf{W_e}, \mathbf{W_d}) + \gamma \mathcal{L}_{K\text{-means}}(\mathbf{W_e}, \{\mathbf{m}_k\}_1^K, \{b_i\}_1^N)$$

At the end of training, DESOM converges to the Deep Clustering Network (DCN) [Yang et al., 2017].



Data sets

4 benchmark data sets:

- **MNIST**: grayscale images of handwritten digits (28-by-28 pixels). $N_{\text{train}} = 60000 / N_{\text{test}} = 10000, P = 784, K^\star = 10$

Data sets

4 benchmark data sets:

- **MNIST**: grayscale images of handwritten digits (28-by-28 pixels).
  $N_{\text{train}} = 60000 / N_{\text{test}} = 10000, P = 784, K^\star = 10$
- **Fashion-MNIST**: grayscale images of clothing (28-by-28 pixels).
  $N_{\text{train}} = 60000 / N_{\text{test}} = 10000, P = 784, K^\star = 10$

Data sets

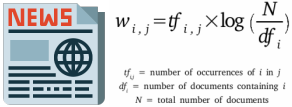
4 benchmark data sets:

- **MNIST**: grayscale images of handwritten digits (28-by-28 pixels).
  $N_{\text{train}} = 60000 / N_{\text{test}} = 10000, P = 784, K^\star = 10$
- **Fashion-MNIST**: grayscale images of clothing (28-by-28 pixels).
  $N_{\text{train}} = 60000 / N_{\text{test}} = 10000, P = 784, K^\star = 10$
- **USPS**: grayscale images of handwritten digits (16-by-16 pixels).
  $N_{\text{train}} = 7291 / N_{\text{test}} = 2007, P = 256, K^\star = 10$

Data sets

4 benchmark data sets:

- **MNIST**: grayscale images of handwritten digits (28-by-28 pixels). $N_{\text{train}} = 60000 / N_{\text{test}} = 10000, P = 784, K^\star = 10$
- **Fashion-MNIST**: grayscale images of clothing (28-by-28 pixels). $N_{\text{train}} = 60000 / N_{\text{test}} = 10000, P = 784, K^\star = 10$
- **USPS**: grayscale images of handwritten digits (16-by-16 pixels). $N_{\text{train}} = 7291 / N_{\text{test}} = 2007, P = 256, K^\star = 10$
- **Reuters-10k**: text data set built from the RCV1-v2 corpus (English news stories), classified in 4 categories, using 2000 TF-IDF features. $N_{\text{train}} = 7769 / N_{\text{test}} = 2231, P = 2000, K^\star = 4$



Example DESOM visualizations

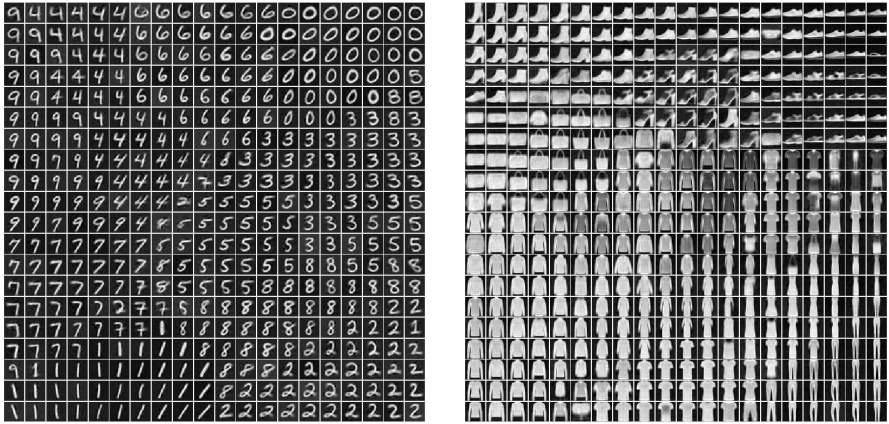


Figure 1: DESOM maps with decoded prototypes for MNIST and Fashion-MNIST data sets.

Clustering performance

Table 1: Purity and NMI. Best result underlined and results with no significant difference in bold.

Method	MNIST		Fashion-MNIST		USPS		Reuters-10k	
	Pur	NMI	Pur	NMI	Pur	NMI	Pur	NMI
K-means ($K = 64$)	0.845	0.581	0.718	0.514	0.858	0.598	0.895	0.439
AE+K-means ($K = 64$)	0.946	0.672	0.764	0.548	0.874	0.611	0.856	0.392
SOM (8×8)	0.832	0.576	0.712	0.513	**0.848**	0.595	0.554	0.225
AE+SOM (8×8)	**0.935**	**0.666**	**0.758**	**0.542**	**0.849**	**0.611**	0.782	0.323
DESOM-AE+SOM (8×8)	**0.933**	0.655	**0.756**	**0.543**	**0.852**	0.589	**0.799**	**0.355**
DESOM (8×8)	**0.934**	0.658	**0.751**	**0.541**	**0.857**	0.592	**0.808**	**0.364**
SOM-VAE [Fortuin et al., 2019] (8×8)	0.868	0.595	0.739	0.520	-	-	-	-
DPSOM [Manduchi et al., 2020] (8×8)	0.964	0.705	0.764	0.571	-	-	-	-

Table 2: Unsupervised clustering accuracy.

Method	MNIST	Fashion-MNIST	USPS	Reuters-10k
K-means ($K =$ #classes)	0.533	0.549	0.660	0.589
AE+K-means ($K =$ #classes)	0.801	0.489	0.680	0.538
SOM (8×8) + HC	0.598	0.491	0.666	0.439
AE+SOM (8×8) + HC	**0.791**	0.480	0.649	0.441
DESOM-AE+SOM (8×8) + HC	0.721	**0.553**	0.610	**0.467**
DESOM (8×8) + HC	**0.810**	**0.571**	**0.698**	**0.486**

Clustering performance

Table 1: Purity and NMI. Best result underlined and results with no significant difference in bold.

Method	MNIST		Fashion-MNIST		USPS		Reuters-10k	
	Pur	NMI	Pur	NMI	Pur	NMI	Pur	NMI
K-means ($K = 64$)	0.845	0.581	0.718	0.514	0.858	0.598	0.895	0.439
AE+K-means ($K = 64$)	0.946	0.672	0.764	0.548	0.874	0.611	0.856	0.392
SOM (8×8)	0.832	0.576	0.712	0.513	**0.848**	0.595	0.554	0.225
AE+SOM (8×8)	**0.935**	**0.666**	**0.758**	**0.542**	**0.849**	**0.611**	0.782	0.323
DESOM-AE+SOM (8×8)	**0.933**	0.655	**0.756**	**0.543**	**0.852**	0.589	**0.799**	**0.355**
DESOM (8×8)	**0.934**	0.658	**0.751**	**0.541**	**0.857**	0.592	**0.808**	**0.364**
SOM-VAE [Fortuin et al., 2019] (8×8)	0.868	0.595	0.739	0.520	-	-	-	-
DPSOM [Manduchi et al., 2020] (8×8)	0.964	0.705	0.764	0.571	-	-	-	-

Table 2: Unsupervised clustering accuracy.

Method	MNIST	Fashion-MNIST	USPS	Reuters-10k
K-means ($K =$ #classes)	0.533	0.549	0.660	0.589
AE+K-means ($K =$ #classes)	0.801	0.489	0.680	0.538
SOM (8×8) + HC	0.598	0.491	0.666	0.439
AE+SOM (8×8) + HC	**0.791**	0.480	0.649	0.441
DESOM-AE+SOM (8×8) + HC	0.721	**0.553**	0.610	**0.467**
DESOM (8×8) + HC	**0.810**	**0.571**	**0.698**	**0.486**

Clustering performance

Table 1: Purity and NMI. Best result underlined and results with no significant difference in bold.

Method	MNIST		Fashion-MNIST		USPS		Reuters-10k	
	Pur	NMI	Pur	NMI	Pur	NMI	Pur	NMI
K-means ($K = 64$)	0.845	0.581	0.718	0.514	0.858	0.598	0.895	0.439
AE+K-means ($K = 64$)	0.946	0.672	0.764	0.548	0.874	0.611	0.856	0.392
SOM (8×8)	0.832	0.576	0.712	0.513	**0.848**	0.595	0.554	0.225
AE+SOM (8×8)	**0.935**	**0.666**	**0.758**	**0.542**	**0.849**	**0.611**	0.782	0.323
DESOM-AE+SOM (8×8)	**0.933**	0.655	**0.756**	**0.543**	**0.852**	0.589	**0.799**	**0.355**
DESOM (8×8)	**0.934**	0.658	**0.751**	**0.541**	**0.857**	0.592	**0.808**	**0.364**
SOM-VAE [Fortuin et al., 2019] (8×8)	0.868	0.595	0.739	0.520	-	-	-	-
DPSOM [Manduchi et al., 2020] (8×8)	0.964	0.705	0.764	0.571	-	-	-	-

Table 2: Unsupervised clustering accuracy.

Method	MNIST	Fashion-MNIST	USPS	Reuters-10k
K-means ($K =$ #classes)	0.533	0.549	0.660	0.589
AE+K-means ($K =$ #classes)	0.801	0.489	0.680	0.538
SOM (8×8) + HC	0.598	0.491	0.666	0.439
AE+SOM (8×8) + HC	**0.791**	0.480	0.649	0.441
DESOM-AE+SOM (8×8) + HC	0.721	**0.553**	0.610	**0.467**
DESOM (8×8) + HC	**0.810**	**0.571**	**0.698**	**0.486**

Clustering performance

Table 1: Purity and NMI. Best result underlined and results with no significant difference in bold.

Method	MNIST		Fashion-MNIST		USPS		Reuters-10k	
	Pur	NMI	Pur	NMI	Pur	NMI	Pur	NMI
K-means ($K = 64$)	0.845	0.581	0.718	0.514	0.858	0.598	0.895	0.439
AE+K-means ($K = 64$)	0.946	0.672	0.764	0.548	0.874	0.611	0.856	0.392
SOM (8×8)	0.832	0.576	0.712	0.513	**0.848**	0.595	0.554	0.225
AE+SOM (8×8)	**0.935**	**0.666**	**0.758**	**0.542**	**0.849**	**0.611**	0.782	0.323
DESOM-AE+SOM (8×8)	**0.933**	0.655	**0.756**	**0.543**	**0.852**	0.589	**0.799**	**0.355**
DESOM (8×8)	**0.934**	0.658	**0.751**	**0.541**	**0.857**	0.592	**0.808**	**0.364**
SOM-VAE [Fortuin et al., 2019] (8×8)	0.868	0.595	0.739	0.520	-	-	-	-
DPSOM [Manduchi et al., 2020] (8×8)	0.964	0.705	0.764	0.571	-	-	-	-

Table 2: Unsupervised clustering accuracy.

Method	MNIST	Fashion-MNIST	USPS	Reuters-10k
K-means ($K =$ #classes)	0.533	0.549	0.660	0.589
AE+K-means ($K =$ #classes)	0.801	0.489	0.680	0.538
SOM (8×8) + HC	0.598	0.491	0.666	0.439
AE+SOM (8×8) + HC	**0.791**	0.480	0.649	0.441
DESOM-AE+SOM (8×8) + HC	0.721	**0.553**	0.610	**0.467**
DESOM (8×8) + HC	**0.810**	**0.571**	**0.698**	**0.486**

Clustering performance

Table 1: Purity and NMI. Best result underlined and results with no significant difference in bold.

Method	MNIST		Fashion-MNIST		USPS		Reuters-10k	
	Pur	NMI	Pur	NMI	Pur	NMI	Pur	NMI
K-means ($K = 64$)	0.845	0.581	0.718	0.514	0.858	0.598	0.895	0.439
AE+K-means ($K = 64$)	0.946	0.672	0.764	0.548	0.874	0.611	0.856	0.392
SOM (8×8)	0.832	0.576	0.712	0.513	**0.848**	0.595	0.554	0.225
AE+SOM (8×8)	**0.935**	**0.666**	**0.758**	**0.542**	**0.849**	**0.611**	0.782	0.323
DESOM-AE+SOM (8×8)	**0.933**	0.655	**0.756**	**0.543**	**0.852**	0.589	**0.799**	**0.355**
DESOM (8×8)	**0.934**	0.658	**0.751**	**0.541**	**0.857**	0.592	**0.808**	**0.364**
SOM-VAE [Fortuin et al., 2019] (8×8)	0.868	0.595	0.739	0.520	-	-	-	-
DPSOM [Manduchi et al., 2020] (8×8)	0.964	0.705	0.764	0.571	-	-	-	-

Table 2: Unsupervised clustering accuracy.

Method	MNIST	Fashion-MNIST	USPS	Reuters-10k
K-means ($K =$ #classes)	0.533	0.549	0.660	0.589
AE+K-means ($K =$ #classes)	0.801	0.489	0.680	0.538
SOM (8×8) + HC	0.598	0.491	0.666	0.439
AE+SOM (8×8) + HC	**0.791**	0.480	0.649	0.441
DESOM-AE+SOM (8×8) + HC	0.721	**0.553**	0.610	**0.467**
DESOM (8×8) + HC	**0.810**	**0.571**	**0.698**	**0.486**

Stability analysis for model selection in clustering

Model selection in clustering

Clustering validation

Evaluating results of cluster analysis in a *quantitative* and *objective* fashion [Roth et al., 2002], in order to **select the *right* number of clusters** K in a data set, or to tune any parameter of a clustering algorithm.

The objective of clustering is *ill-defined* [Kleinberg, 2003, von Luxburg et al., 2012, Shalev-Shwartz and Ben-David, 2013]

$\rightarrow$ **Challenging problem!**

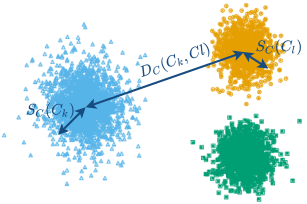
Clustering validation

Evaluating results of cluster analysis in a *quantitative* and *objective* fashion [Roth et al., 2002], in order to **select the *right* number of clusters** K in a data set, or to tune any parameter of a clustering algorithm.

The objective of clustering is *ill-defined* [Kleinberg, 2003, von Luxburg et al., 2012, Shalev-Shwartz and Ben-David, 2013]

→ **Challenging problem!**

- **Ground-truth labels**: External indices (e.g. ARI, NMI...)
- **No labels**: Internal indices (see [Arbelaitz et al., 2013, Desgraupes, 2013])
  1. Within/between-cluster distance ratios (compactness VS separateness)
     e.g. Calinski-Harabasz, Davies-Bouldin, Dunn, Silhouette...
     → Explicit geometrical prior, algorithm-specific
  2. Model-based likelihood criteria (AIC, BIC, ICL [Biernacki et al., 2000])
  3. Statistical robustness: **cluster stability analysis**



Cluster stability analysis

Stability principle

A good clustering is a **stable structure** in the data ($\neq$ algorithmic or sampling artifact).
A clustering algorithm $\mathcal{A}$ applied repeatedly (with the same parameter K) to perturbed versions of a data set should **find the same structure** and obtain similar results.

$$\text{Stab}(\mathcal{A}, K) := \mathbb{E}_{\mathbf{X}, \mathbf{X}' \sim \mathcal{P}^N} \left[s(\mathcal{C}_K, \mathcal{C}'_K) \right]$$

where s is a similarity measure between partitions.

Cluster stability analysis

Stability principle

A good clustering is a **stable structure** in the data ($\neq$ algorithmic or sampling artifact).
A clustering algorithm $\mathcal{A}$ applied repeatedly (with the same parameter K) to perturbed versions of a data set should **find the same structure** and obtain similar results.

$$\text{Stab}(\mathcal{A}, K) := \mathbb{E}_{\mathbf{X},\mathbf{X}' \sim \mathcal{P}^N} \left[s(\mathcal{C}_K, \mathcal{C}'_K) \right]$$

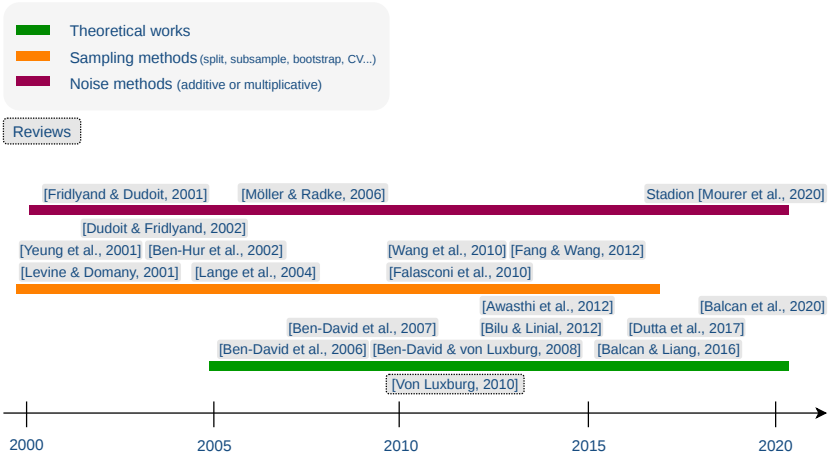
where s is a similarity measure between partitions.

How to estimate stability in practice?

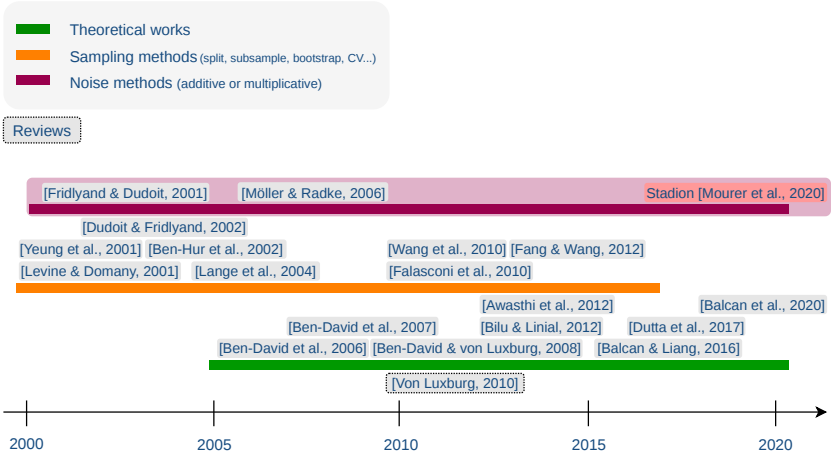
1. Generate several samples from the data set (sampling, noise).
2. Apply the clustering algorithm on each sample.
3. Measure similarities between the obtained partitions.
4. Aggregate these similarities into a stability score.
5. Select the best solution using a decision rule.

Numerous approaches and theoretical works. See [Von Luxburg, 2009] for a review.

Literature timeline



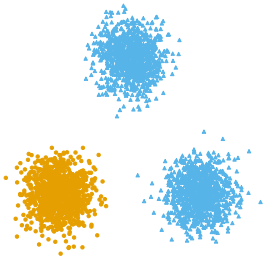
Literature timeline



Jumping VS jittering

Sources of instability

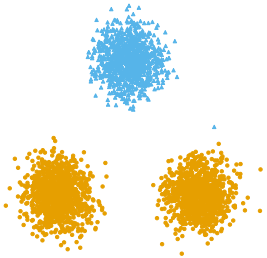
- **Jumping**: algorithm ends up in local minima, yielding very different solutions.



Jumping VS jittering

Sources of instability

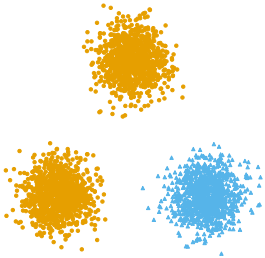
- **Jumping**: algorithm ends up in local minima, yielding very different solutions.



Jumping VS jittering

Sources of instability

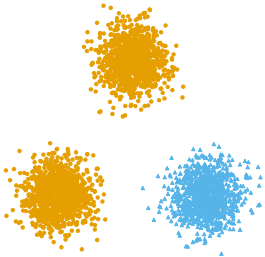
- **Jumping**: algorithm ends up in local minima, yielding very different solutions.

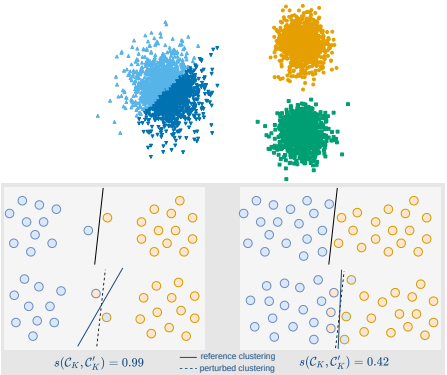


Jumping VS jittering

Sources of instability

- **Jumping**: algorithm ends up in local minima, yielding very different solutions.
- **Jittering**: cluster boundaries *jitter* in high-density regions, causing points to change clusters.





Stadion: a criterion based on a stability trade-off [Mourer et al., 2020]

Our setting

- No perfect symmetries in the distribution
- Effective initialization strategy (e.g. best of n runs)
- $N \gg K$

In this setting,

1. Jittering is the main source of instability.
2. Sampling perturbation is ineffective.
3. Stability is unable to detect $K < K^*$.

based on [Ben-David et al., 2006, Ben-David and Von Luxburg, 2008, Von Luxburg, 2009]

Stadion: a criterion based on a stability trade-off [Mourer et al., 2020]

Our setting

- No perfect symmetries in the distribution
- Effective initialization strategy (e.g. best of n runs)
- $N \gg K$

In this setting,

1. Jittering is the main source of instability.
2. Sampling perturbation is ineffective.
3. Stability is unable to detect $K < K^*$.

based on [Ben-David et al., 2006, Ben-David and Von Luxburg, 2008, Von Luxburg, 2009]

Proposition

A clustering is a partitioning of data into groups so that the partition is stable, and within each cluster, there exists no stable partition.

Stadion: a criterion based on a stability trade-off [Mourer et al., 2020]

Our setting

- No perfect symmetries in the distribution
- Effective initialization strategy (e.g. best of n runs)
- $N \gg K$

In this setting,

1. Jittering is the main source of instability.
2. Sampling perturbation is ineffective.
3. Stability is unable to detect $K < K^*$.

based on [Ben-David et al., 2006, Ben-David and Von Luxburg, 2008, Von Luxburg, 2009]

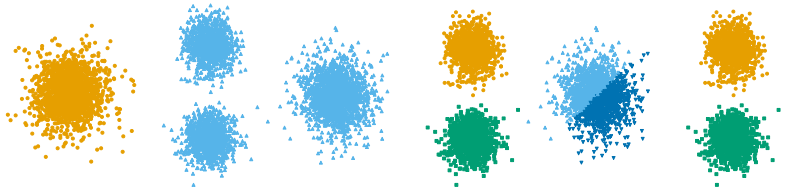
Proposition

A clustering is a partitioning of data into groups so that the partition is stable, and within each cluster, there exists no stable partition.

$$\underbrace{\text{Stadion}(\mathcal{A}, K, \mathcal{C}_K, \boldsymbol{\Omega})}_{\textbf{Sta}\text{bility }\textbf{di}\text{fference criter}\textbf{ion}} := \underbrace{\text{Stab}_\text{B}(\mathcal{A}, K, \mathcal{C}_K)}_{\text{Between-cluster stability}} - \underbrace{\text{Stab}_\text{W}(\mathcal{A}, K, \mathcal{C}_K, \boldsymbol{\Omega})}_{\text{Within-cluster stability}} \in [-1, 1]$$

- Only additive noise perturbation is reliable (uniform or Gaussian)
- Empirical methodology to estimate Stab_W and vary the level of noise $\varepsilon \rightarrow$ "stability paths"
- Extensive discussion and hyperparameter studies in [Mourer et al., 2020]

Usage example: stability paths



(a) $K = 2$ (stable) (b) $K = 3$ (stable) (c) $K = 4$ (unstable, jittering)

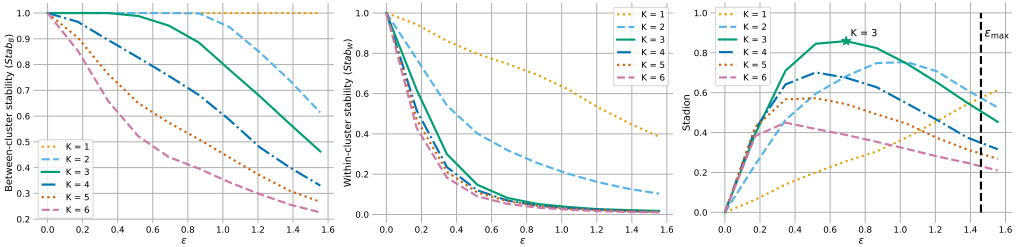


Figure 2: Between-cluster and within-cluster stability and Stadion paths for K-means, $K \in \{1 \dots 6\}$.

Is a data set clusterable?

Most indices are undefined for $K = 1$.

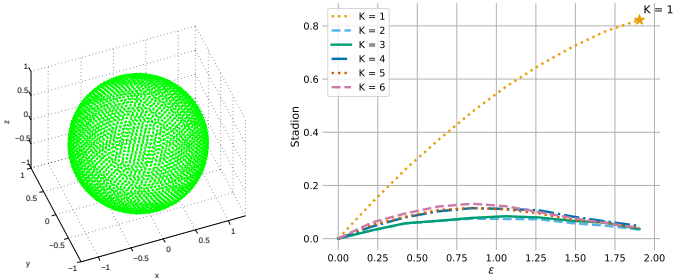


Figure 3: Golfball data set and Stadion paths for K-means, $K \in \{1 \dots 6\}$.

Benchmark results

Table 3: Benchmark results on 80 artificial and real data sets for K-means, Ward linkage and GMM. Average rank of ARI ($\overline{R_{\text{ARI}}}$) and number of correctly selected $K^\star$ (wins).

	Artificial data sets						Real data sets					
	K-means		Ward		GMM		K-means		Ward		GMM	
Method	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins
$K^\star$	6.47	73	4.77	73	5.05	73	4.50	7	3.36	7	3.93	7
Stadion-max	**6.02**	50	**5.25**	**54**	-	-	**4.93**	**5**	5.86	**4**	-	-
Stadion-mean	6.12	51	5.80	49	-	-	6.57	4	7.64	3	-	-
Stadion-max (extended)	6.13	**56**	-	-	**5.59**	**56**	6.29	3	-	-	**4.43**	**5**
Stadion-mean (extended)	6.42	48	-	-	6.79	43	6.29	3	-	-	5.50	3
BIC	-	-	-	-	6.45	48	-	-	-	-	7.29	2
Wemmert-Gançarski	6.62	53	5.40	**54**	5.77	52	6.00	**5**	**5.36**	**4**	5.79	4
Silhouette	7.51	46	6.47	45	7.01	45	7.21	4	5.86	**4**	6.50	4
[Lange et al., 2004]	7.93	45	6.53	51	6.99	48	8.64	3	5.86	**4**	7.14	3
Davies-Bouldin	8.11	40	6.45	41	7.29	34	8.29	4	7.29	3	8.57	3
Ray-Turi	8.19	37	6.97	40	7.68	33	8.29	4	6.29	3	7.36	4
Calinski-Harabasz	8.71	41	7.14	39	7.43	37	12.21	1	8.86	1	5.79	3
Dunn	10.11	26	7.77	33	7.92	34	10.57	1	7.79	2	9.07	2
Xie-Beni	10.27	22	7.61	34	8.19	28	11.50	1	7.57	2	9.93	2
Gap statistic	10.38	26	-	-	-	-	10.57	2	-	-	-	-
[Ben-Hur et al., 2002]	10.99	20	7.86	31	8.85	28	8.14	1	7.93	2	9.71	2

Benchmark results

Table 3: Benchmark results on 80 artificial and real data sets for K-means, Ward linkage and GMM. Average rank of ARI ($\overline{R_{\text{ARI}}}$) and number of correctly selected $K^\star$ (wins).

	Artificial data sets						Real data sets					
	K-means		Ward		GMM		K-means		Ward		GMM	
Method	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins
$K^\star$	6.47	73	4.77	73	5.05	73	4.50	7	3.36	7	3.93	7
Stadion-max	**6.02**	50	**5.25**	**54**	-	-	**4.93**	**5**	5.86	**4**	-	-
Stadion-mean	6.12	51	5.80	49	-	-	6.57	4	7.64	3	-	-
Stadion-max (extended)	6.13	**56**	-	-	**5.59**	**56**	6.29	3	-	-	**4.43**	**5**
Stadion-mean (extended)	6.42	48	-	-	6.79	43	6.29	3	-	-	5.50	3
BIC	-	-	-	-	6.45	48	-	-	-	-	7.29	2
Wemmert-Gançarski	6.62	53	5.40	**54**	5.77	52	6.00	**5**	**5.36**	**4**	5.79	4
Silhouette	7.51	46	6.47	45	7.01	45	7.21	4	5.86	**4**	6.50	4
[Lange et al., 2004]	7.93	45	6.53	51	6.99	48	8.64	3	5.86	**4**	7.14	3
Davies-Bouldin	8.11	40	6.45	41	7.29	34	8.29	4	7.29	3	8.57	3
Ray-Turi	8.19	37	6.97	40	7.68	33	8.29	4	6.29	3	7.36	4
Calinski-Harabasz	8.71	41	7.14	39	7.43	37	12.21	1	8.86	1	5.79	3
Dunn	10.11	26	7.77	33	7.92	34	10.57	1	7.79	2	9.07	2
Xie-Beni	10.27	22	7.61	34	8.19	28	11.50	1	7.57	2	9.93	2
Gap statistic	10.38	26	-	-	-	-	10.57	2	-	-	-	-
[Ben-Hur et al., 2002]	10.99	20	7.86	31	8.85	28	8.14	1	7.93	2	9.71	2

Benchmark results

Table 3: Benchmark results on 80 artificial and real data sets for K-means, Ward linkage and GMM. Average rank of ARI ($\overline{R_{\text{ARI}}}$) and number of correctly selected $K^\star$ (wins).

	Artificial data sets						Real data sets					
	K-means		Ward		GMM		K-means		Ward		GMM	
Method	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins
$K^\star$	6.47	73	4.77	73	5.05	73	4.50	7	3.36	7	3.93	7
Stadion-max	**6.02**	50	**5.25**	**54**	-	-	**4.93**	**5**	5.86	**4**	-	-
Stadion-mean	6.12	51	5.80	49	-	-	6.57	4	7.64	3	-	-
Stadion-max (extended)	6.13	**56**	-	-	**5.59**	**56**	6.29	3	-	-	**4.43**	**5**
Stadion-mean (extended)	6.42	48	-	-	6.79	43	6.29	3	-	-	5.50	3
BIC	-	-	-	-	6.45	48	-	-	-	-	7.29	2
Wemmert-Gançarski	6.62	53	5.40	**54**	5.77	52	6.00	**5**	**5.36**	**4**	5.79	4
Silhouette	7.51	46	6.47	45	7.01	45	7.21	4	5.86	**4**	6.50	4
[Lange et al., 2004]	7.93	45	6.53	51	6.99	48	8.64	3	5.86	**4**	7.14	3
Davies-Bouldin	8.11	40	6.45	41	7.29	34	8.29	4	7.29	3	8.57	3
Ray-Turi	8.19	37	6.97	40	7.68	33	8.29	4	6.29	3	7.36	4
Calinski-Harabasz	8.71	41	7.14	39	7.43	37	12.21	1	8.86	1	5.79	3
Dunn	10.11	26	7.77	33	7.92	34	10.57	1	7.79	2	9.07	2
Xie-Beni	10.27	22	7.61	34	8.19	28	11.50	1	7.57	2	9.93	2
Gap statistic	10.38	26	-	-	-	-	10.57	2	-	-	-	-
[Ben-Hur et al., 2002]	10.99	20	7.86	31	8.85	28	8.14	1	7.93	2	9.71	2

Benchmark results

Table 3: Benchmark results on 80 artificial and real data sets for K-means, Ward linkage and GMM. Average rank of ARI ($\overline{R_{\text{ARI}}}$) and number of correctly selected $K^\star$ (wins).

	Artificial data sets						Real data sets					
	K-means		Ward		GMM		K-means		Ward		GMM	
Method	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins
$K^\star$	6.47	73	4.77	73	5.05	73	4.50	7	3.36	7	3.93	7
Stadion-max	**6.02**	50	**5.25**	**54**	-	-	**4.93**	**5**	5.86	**4**	-	-
Stadion-mean	6.12	51	5.80	49	-	-	6.57	4	7.64	3	-	-
Stadion-max (extended)	6.13	**56**	-	-	**5.59**	**56**	6.29	3	-	-	**4.43**	**5**
Stadion-mean (extended)	6.42	48	-	-	6.79	43	6.29	3	-	-	5.50	3
BIC	-	-	-	-	6.45	48	-	-	-	-	7.29	2
Wemmert-Gançarski	6.62	53	5.40	**54**	5.77	52	6.00	**5**	**5.36**	**4**	5.79	4
Silhouette	7.51	46	6.47	45	7.01	45	7.21	4	5.86	**4**	6.50	4
[Lange et al., 2004]	7.93	45	6.53	51	6.99	48	8.64	3	5.86	**4**	7.14	3
Davies-Bouldin	8.11	40	6.45	41	7.29	34	8.29	4	7.29	3	8.57	3
Ray-Turi	8.19	37	6.97	40	7.68	33	8.29	4	6.29	3	7.36	4
Calinski-Harabasz	8.71	41	7.14	39	7.43	37	12.21	1	8.86	1	5.79	3
Dunn	10.11	26	7.77	33	7.92	34	10.57	1	7.79	2	9.07	2
Xie-Beni	10.27	22	7.61	34	8.19	28	11.50	1	7.57	2	9.93	2
Gap statistic	10.38	26	-	-	-	-	10.57	2	-	-	-	-
[Ben-Hur et al., 2002]	10.99	20	7.86	31	8.85	28	8.14	1	7.93	2	9.71	2

Benchmark results

Table 3: Benchmark results on 80 artificial and real data sets for K-means, Ward linkage and GMM. Average rank of ARI ($\overline{R_{\text{ARI}}}$) and number of correctly selected $K^\star$ (wins).

	Artificial data sets						Real data sets					
	K-means		Ward		GMM		K-means		Ward		GMM	
Method	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins
$K^\star$	6.47	73	4.77	73	5.05	73	4.50	7	3.36	7	3.93	7
Stadion-max	**6.02**	50	**5.25**	**54**	-	-	**4.93**	**5**	5.86	**4**	-	-
Stadion-mean	6.12	51	5.80	49	-	-	6.57	4	7.64	3	-	-
Stadion-max (extended)	6.13	**56**	-	-	**5.59**	**56**	6.29	3	-	-	**4.43**	**5**
Stadion-mean (extended)	6.42	48	-	-	6.79	43	6.29	3	-	-	5.50	3
BIC	-	-	-	-	6.45	48	-	-	-	-	7.29	2
Wemmert-Gançarski	6.62	53	5.40	**54**	5.77	52	6.00	**5**	**5.36**	**4**	5.79	4
Silhouette	7.51	46	6.47	45	7.01	45	7.21	4	5.86	**4**	6.50	4
[Lange et al., 2004]	7.93	45	6.53	51	6.99	48	8.64	3	5.86	**4**	7.14	3
Davies-Bouldin	8.11	40	6.45	41	7.29	34	8.29	4	7.29	3	8.57	3
Ray-Turi	8.19	37	6.97	40	7.68	33	8.29	4	6.29	3	7.36	4
Calinski-Harabasz	8.71	41	7.14	39	7.43	37	12.21	1	8.86	1	5.79	3
Dunn	10.11	26	7.77	33	7.92	34	10.57	1	7.79	2	9.07	2
Xie-Beni	10.27	22	7.61	34	8.19	28	11.50	1	7.57	2	9.93	2
Gap statistic	10.38	26	-	-	-	-	10.57	2	-	-	-	-
[Ben-Hur et al., 2002]	10.99	20	7.86	31	8.85	28	8.14	1	7.93	2	9.71	2

Benchmark results

Table 3: Benchmark results on 80 artificial and real data sets for K-means, Ward linkage and GMM. Average rank of ARI ($\overline{R_{\text{ARI}}}$) and number of correctly selected $K^\star$ (wins).

	Artificial data sets						Real data sets					
	K-means		Ward		GMM		K-means		Ward		GMM	
Method	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins	$\overline{R}_{\text{ARI}}$	wins
$K^\star$	6.47	73	4.77	73	5.05	73	4.50	7	3.36	7	3.93	7
Stadion-max	**6.02**	50	**5.25**	**54**	-	-	**4.93**	**5**	5.86	**4**	-	-
Stadion-mean	6.12	51	5.80	49	-	-	6.57	4	7.64	3	-	-
Stadion-max (extended)	6.13	**56**	-	-	**5.59**	**56**	6.29	3	-	-	**4.43**	**5**
Stadion-mean (extended)	6.42	48	-	-	6.79	43	6.29	3	-	-	5.50	3
BIC	-	-	-	-	6.45	48	-	-	-	-	7.29	2
Wemmert-Gançarski	6.62	53	5.40	**54**	5.77	52	6.00	**5**	**5.36**	**4**	5.79	4
Silhouette	7.51	46	6.47	45	7.01	45	7.21	4	5.86	**4**	6.50	4
[Lange et al., 2004]	7.93	45	6.53	51	6.99	48	8.64	3	5.86	**4**	7.14	3
Davies-Bouldin	8.11	40	6.45	41	7.29	34	8.29	4	7.29	3	8.57	3
Ray-Turi	8.19	37	6.97	40	7.68	33	8.29	4	6.29	3	7.36	4
Calinski-Harabasz	8.71	41	7.14	39	7.43	37	12.21	1	8.86	1	5.79	3
Dunn	10.11	26	7.77	33	7.92	34	10.57	1	7.79	2	9.07	2
Xie-Beni	10.27	22	7.61	34	8.19	28	11.50	1	7.57	2	9.93	2
Gap statistic	10.38	26	-	-	-	-	10.57	2	-	-	-	-
[Ben-Hur et al., 2002]	10.99	20	7.86	31	8.85	28	8.14	1	7.93	2	9.71	2

In this defense... (intermediate summary)

1. **How to learn representations to effectively cluster complex data?**

   Forest, F., Lebbah, M., Azzag, H., & Lacaille, J. (2019). **Deep Embedded SOM: Joint Representation Learning and Self-Organization**. In *ESANN 2019*.

   Forest, F., Lebbah, M., Azzag, H., & Lacaille, J. (2019). **Deep Architectures for Joint Clustering and Visualization with Self-Organizing Maps**. In *Workshop LDRC, PAKDD 2019*.

   Forest, F., Lebbah, M., Azzag, H., & Lacaille, J. (2020). **Carte SOM profonde : Apprentissage joint de représentations et auto-organisation**. In *CAp: Conférence d'Apprentissage 2020*.

   Journal submission (under review).

2. **How to evaluate clustering algorithms?**

   Mourer, A., Forest, F., Lebbah, M., Azzag, H., & Lacaille, J. (2020). **Selecting the Number of Clusters K with a Stability Trade-off: an Internal Validation Criterion**. arXiv:2006.08530 (under review).

   Forest, F., Mourer, A., Lebbah, M., Azzag, H., & Lacaille, J. (2021). **An Invariance-guided Stability Criterion for Time Series Clustering Validation**. In *ICPR 2020*.

Scalable aircraft engine health monitoring applications

The need to scale

Volume, Velocity:

- Growth of air traffic
- Aircraft equipped with more sensors, high-frequency temporal data

Variety:

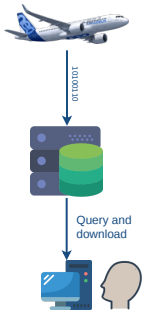
- Multiple data sources: production, tests, flights, maintenance, weather...
- Time series, geospatial, images (2D/3D), text...

Value:

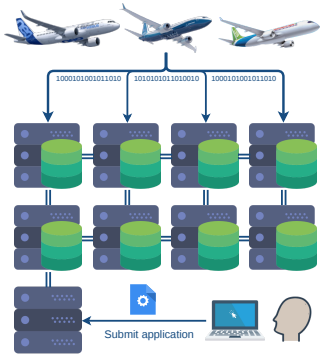
- Fly-by-the-hour leasing contracts $\rightarrow$ engine manufacturers responsible for maintenance

Scaling to Big Data with distributed computing

Then...



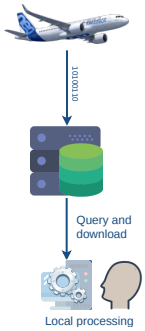
...and now



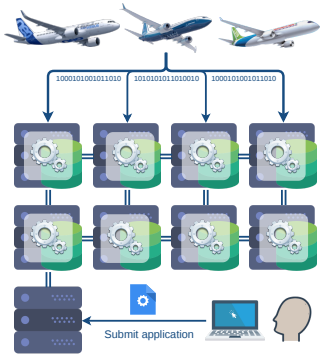
1. Send application to the data, avoid data transfer (locality)

Scaling to Big Data with distributed computing

Then...



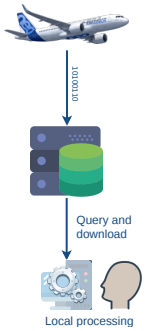
...and now



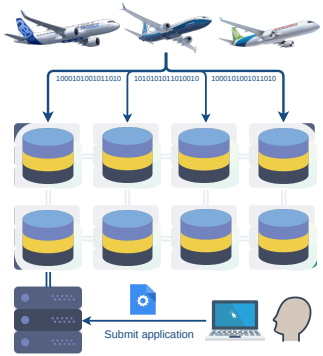
1. Send application to the data, avoid data transfer (locality)
2. Distributed, data-parallel processing (**Map-Reduce**, functional programming)

Scaling to Big Data with distributed computing

Then...



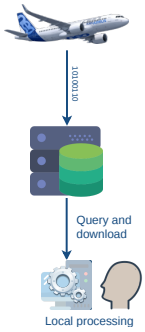
...and now



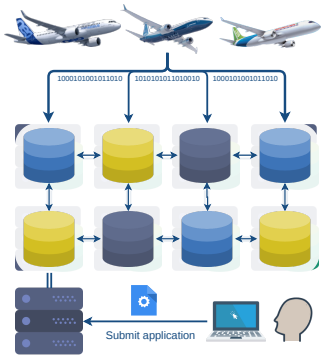
1. Send application to the data, avoid data transfer (locality)
2. Distributed, data-parallel processing (**Map-Reduce**, functional programming)
   2.1 Map

Scaling to Big Data with distributed computing

Then...



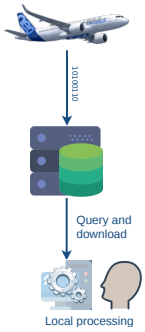
...and now



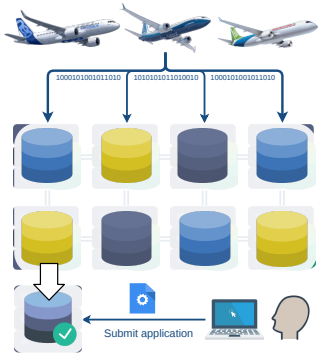
1. Send application to the data, avoid data transfer (locality)
2. Distributed, data-parallel processing (**Map-Reduce**, functional programming)
   - 2.1 Map
   - 2.2 Shuffle

Scaling to Big Data with distributed computing

Then...



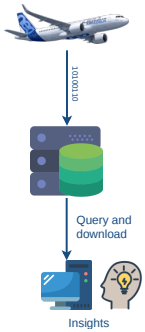
...and now



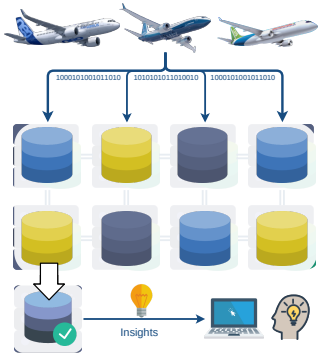
1. Send application to the data, avoid data transfer (locality)
2. Distributed, data-parallel processing (**Map-Reduce**, functional programming)
   - 2.1 Map
   - 2.2 Shuffle
   - 2.3 Reduce

Scaling to Big Data with distributed computing

Then...



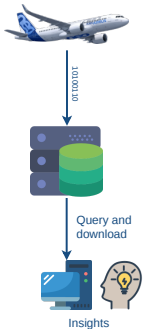
...and now



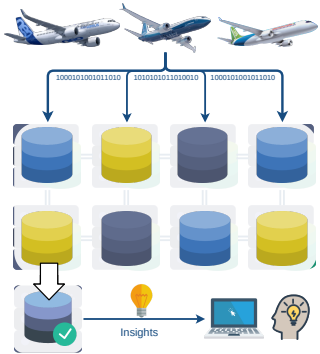
1. Send application to the data, avoid data transfer (locality)
2. Distributed, data-parallel processing (**Map-Reduce**, functional programming)
   - 2.1 Map
   - 2.2 Shuffle
   - 2.3 Reduce
3. Obtain results

Scaling to Big Data with distributed computing

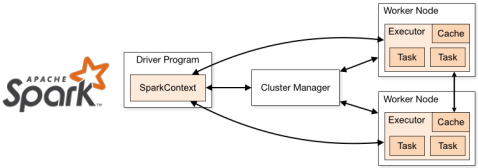
Then...



...and now

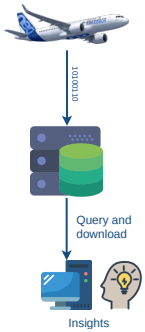


1. Send application to the data, avoid data transfer (locality)
2. Distributed, data-parallel processing (**Map-Reduce**, functional programming)
   - 2.1 Map
   - 2.2 Shuffle
   - 2.3 Reduce
3. Obtain results

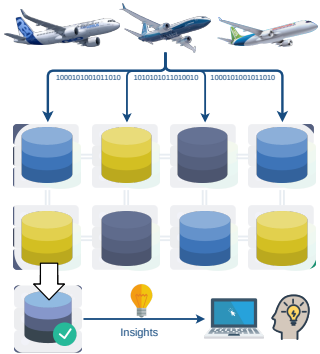


Scaling to Big Data with distributed computing

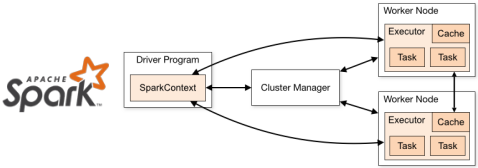
Then...



...and now



1. Send application to the data, avoid data transfer (locality)
2. Distributed, data-parallel processing (**Map-Reduce**, functional programming)
   - 2.1 Map
   - 2.2 Shuffle
   - 2.3 Reduce
3. Obtain results



▶ ML algorithms can often be expressed in this paradigm [Aggarwal and Reddy, 2013, Sarazin et al., 2014]

A generic and scalable analytics pipeline [Forest et al., 2018]

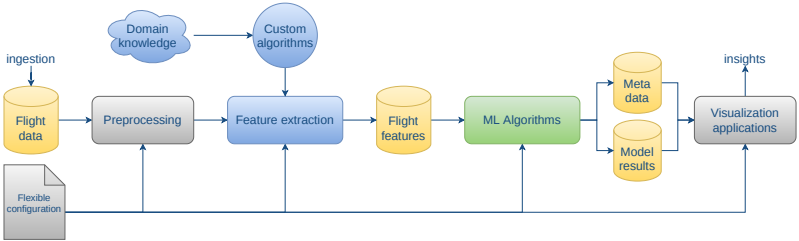
Objectives

- Provide **generic** tools enabling engineers to deploy domain-specific algorithms at scale across millions of flights stored on a cluster
- Scale health monitoring methodologies to produce value by leveraging Big Aircraft Data

A generic and scalable analytics pipeline [Forest et al., 2018]

Objectives

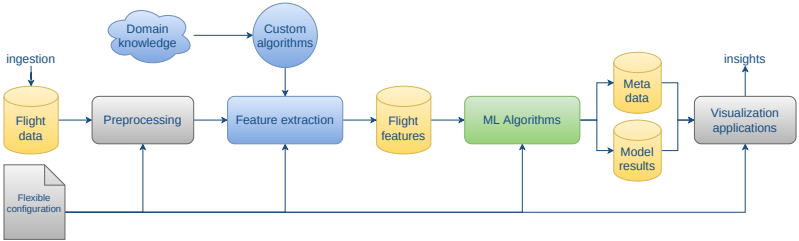
- Provide **generic** tools enabling engineers to deploy domain-specific algorithms at scale across millions of flights stored on a cluster
- Scale health monitoring methodologies to produce value by leveraging Big Aircraft Data



A generic and scalable analytics pipeline [Forest et al., 2018]

Objectives

- Provide **generic** tools enabling engineers to deploy domain-specific algorithms at scale across millions of flights stored on a cluster
- Scale health monitoring methodologies to produce value by leveraging Big Aircraft Data



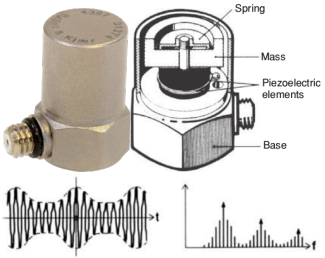
Implemented methodology: Fleet monitoring with Self-Organizing Maps

Monitoring fleets of engines using SOM based on indicators describing the *state* of an engine (or its subsystems) at each flight, following [Cottrell et al., 2009, Côme et al., 2010a, Côme et al., 2011].

Vibration monitoring use case [Forest et al., 2020a]

Vibration monitoring

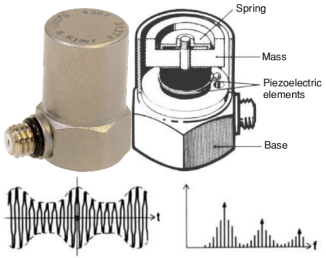
- Crucial part of condition monitoring for rotating equipment [Randall, 2011, Bastard et al., 2016]
- Detection of unbalance, misalignment due to wear (blades, bearings, gears), rotor/stator contact, etc.



Vibration monitoring use case [Forest et al., 2020a]

Vibration monitoring

- Crucial part of condition monitoring for rotating equipment [Randall, 2011, Bastard et al., 2016]
- Detection of unbalance, misalignment due to wear (blades, bearings, gears), rotor/stator contact, etc.



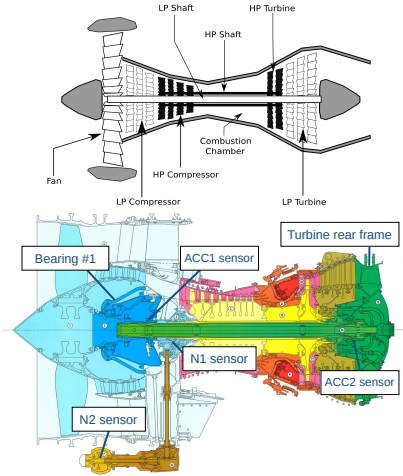
Proposition

Methodology for large-scale vibration monitoring of a fleet of aircraft engines using historical flight recorder data.

1. Massive extraction of **time-domain vibration signatures** using distributed processing.
2. Unsupervised learning with **self-organizing maps** for clustering and visualization.

✓ Monitoring, alerting, forecasting ✗ Diagnosis and prognosis are left to experts

Vibration sensors

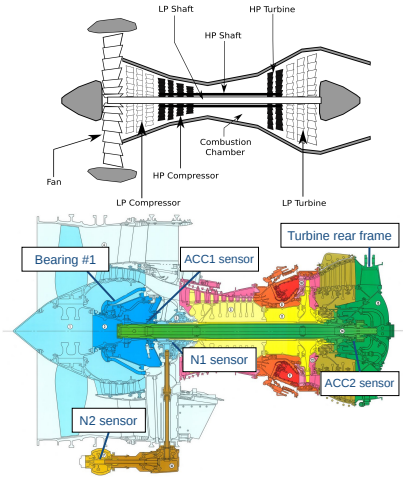


Sensors measure **rotation speeds** and **vibration peak amplitude** (displacement, speed or acceleration). Raw signals are filtered and downsampled by the onboard calculator.

Variables

- **N1**: LP shaft rotation speed @66Hz
- **N2**: HP shaft rotation speed @66Hz
- **LP-ACC1**, **LP-ACC2**: vibration amplitude at N1 speed @4Hz
- **HP-ACC1**, **HP-ACC2**: vibration amplitude at N2 speed @4Hz

Vibration sensors



Sensors measure **rotation speeds** and **vibration peak amplitude** (displacement, speed or acceleration). Raw signals are filtered and downsampled by the onboard calculator.

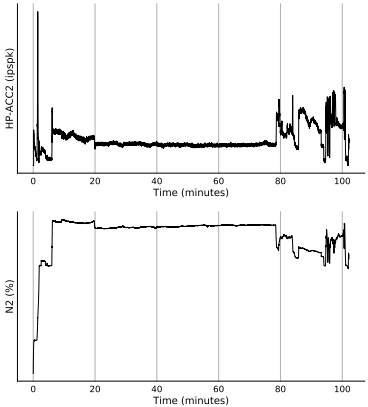
Variables

- **N1**: LP shaft rotation speed @66Hz
- **N2**: HP shaft rotation speed @66Hz
- **LP-ACC1**, **LP-ACC2**: vibration amplitude at N1 speed @4Hz
- **HP-ACC1**, **HP-ACC2**: vibration amplitude at N2 speed @4Hz

~400 engines, ~100k flights, ~1 TB of data

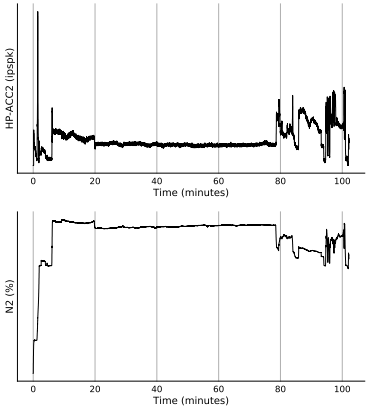
Vibration signatures

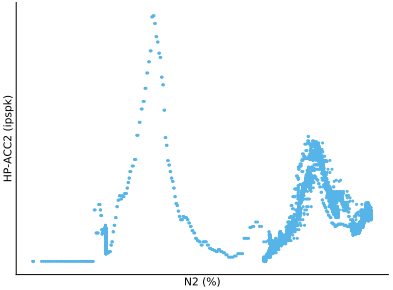
Vibratory response of the engine: vibration amplitude as a function of regime.



Vibration signatures

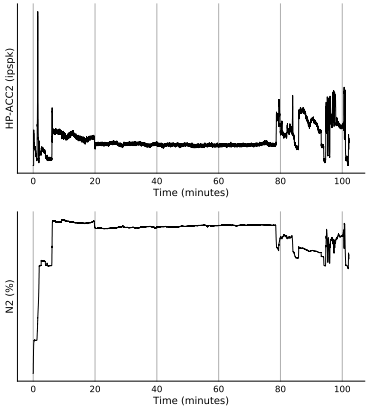
Vibratory response of the engine: vibration amplitude as a function of regime.

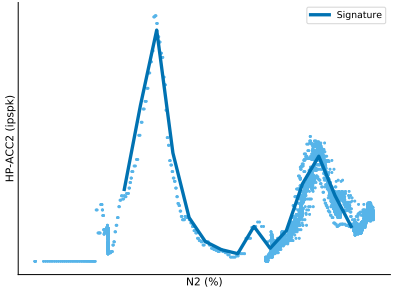




Vibration signatures

Vibratory response of the engine: vibration amplitude as a function of regime.





Modes at specific regimes $\rightarrow$ unbalance at specific locations of the engine

Vibration profiles SOM

- Spark-distributed implementation of batch SOM
- SOM units associated to prototype vibration signatures, representing **vibration profiles**
- Self-organization $\rightarrow$ smooth variations of factors of variations, **interpretability**
- Flight are clustered by projecting on the nearest vibration profile (Best-Matching Signature)

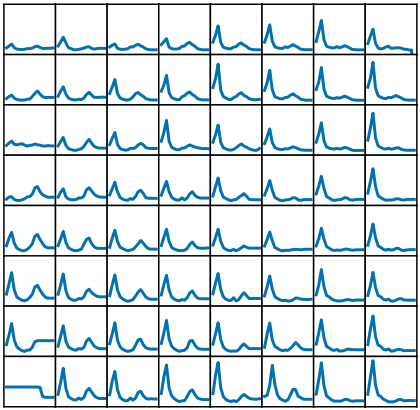


Figure 4: Signature map (HP-ACC2 vs N2).

Vibration profiles SOM — Results analysis

Vibration signatures describe **intrinsic properties** of an engine.

▶ Every engine is different!

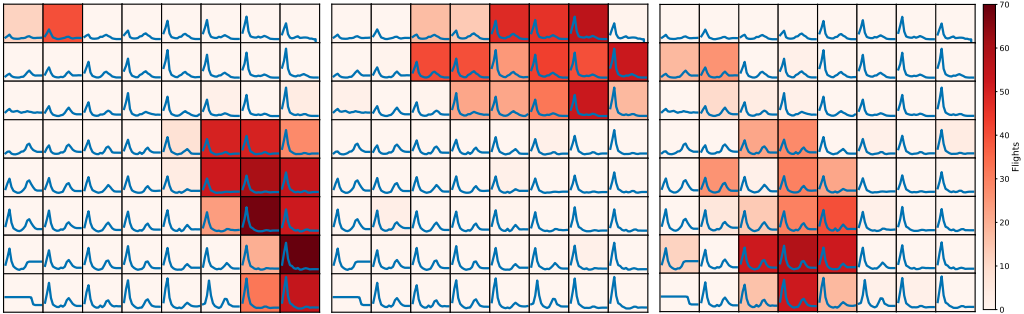
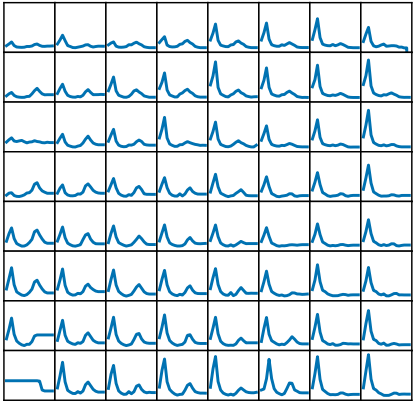


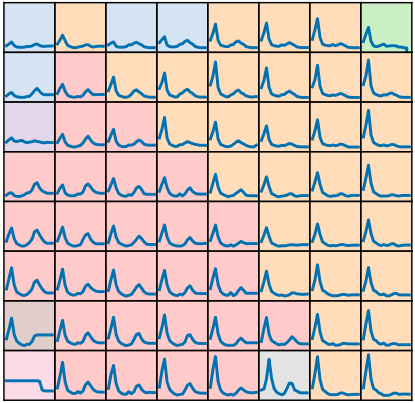
Figure 5: Heatmaps of projection counts for 3 different engines.

Vibration profiles SOM — Methodology



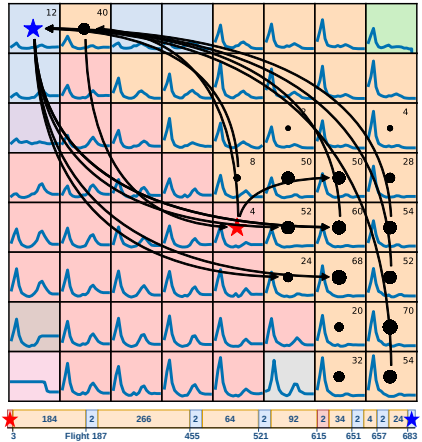
Vibration profiles SOM — Methodology

- Classification into **higher-level vibration profiles** by clustering the prototypes
- **Expert labelling** of map regions (e.g. well-balanced engines, unbalanced ones, switched off sensors, etc.)



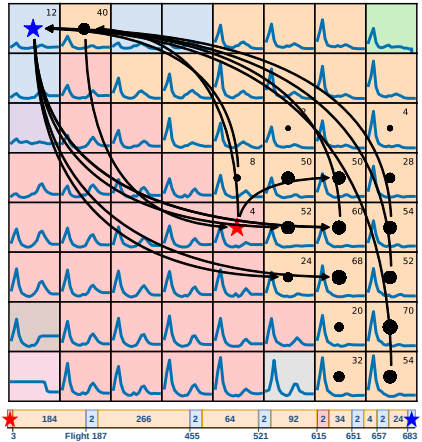
Vibration profiles SOM — Methodology

- Classification into **higher-level vibration profiles** by clustering the prototypes
- **Expert labelling** of map regions (e.g. well-balanced engines, unbalanced ones, switched off sensors, etc.)
- Distance to map $\rightarrow$ **Anomaly score**
- Analysis of the evolution of an engine flight after flight: **engine trajectory** (see [Côme et al., 2011])
  - Sudden jumps or progressive trends may detect abnormal wear
  - Find similar engines, forecast future trajectories (post-finding)



Vibration profiles SOM — Methodology

- Classification into **higher-level vibration profiles** by clustering the prototypes
- **Expert labelling** of map regions (e.g. well-balanced engines, unbalanced ones, switched off sensors, etc.)
- Distance to map → **Anomaly score**
- Analysis of the evolution of an engine flight after flight: **engine trajectory** (see [Côme et al., 2011])
  - Sudden jumps or progressive trends may detect abnormal wear
  - Find similar engines, forecast future trajectories (post-finding)
- Periodically re-train with up-to-date flight data, to account for new trends and aging of the fleet.



Conclusion and perspectives

In this defense...

1. **How to learn representations to effectively cluster complex data?**
2. **How to evaluate clustering algorithms?**
3. **How to develop scalable engine health monitoring methodologies?**

Forest, F., Lacaille, J., Lebbah, M., & Azzag, H. (2018). **A Generic and Scalable Pipeline for Large-Scale Analytics of Continuous Aircraft Engine Data**. In *IEEE International Conference on Big Data 2018*.

Forest, F., Cochard, Q., Noyer, C., Cabut, A., Joncour, M., Lacaille, J., Lebbah, M. & Azzag, H. (2020). **Large-scale Vibration Monitoring of Aircraft Engines from Operational Data using Self-organized Models**. In *Annual Conference of the PHM Society 2020*.

Lacaille, J. & Forest, F. (2020). Computer environment system for monitoring aircraft engines. *Patent FR3089501*.

Perspectives and future work

Deep Embedded SOM:

- Explore different AE architectures and more complex data sets
- Study interactions of neighborhood decay with learning
- Automatic hyperparameter tuning

Perspectives and future work

Deep Embedded SOM:

- Explore different AE architectures and more complex data sets
- Study interactions of neighborhood decay with learning
- Automatic hyperparameter tuning

Model selection & stability analysis:

- Speed up computation of Stadion (distance to boundaries)
- Extend estimation of within-cluster stability to all clustering algorithms
- Develop theoretical results, links with existing indices and adversarial attacks
- Confidence in the quality of unsupervised algorithms results

Perspectives and future work

Deep Embedded SOM:

- Explore different AE architectures and more complex data sets
- Study interactions of neighborhood decay with learning
- Automatic hyperparameter tuning

Model selection & stability analysis:

- Speed up computation of Stadion (distance to boundaries)
- Extend estimation of within-cluster stability to all clustering algorithms
- Develop theoretical results, links with existing indices and adversarial attacks
- Confidence in the quality of unsupervised algorithms results

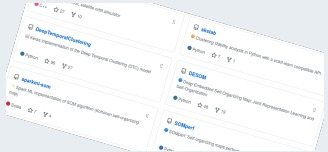
Industrial applications:

- Extract more variables from vibration signatures, add context parameters
- Put these tools into the hands of more people and extend methodology to other use cases
- Combine with other data sources (production, tests, maintenance, weather)
- Learn the ***actual* state** of an engine (integrated over its lifetime)
- Learn a **transition model** between states

Additional contributions

Open-source software to take home

- `DESOM` [github.com/FlorentF9/DESOM]
- `DeepTemporalClustering` [github.com/FlorentF9/DeepTemporalClustering]
- `SOMperf` [github.com/FlorentF9/SOMperf]
- `skstab` [github.com/FlorentF9/skstab]
- `Spark ML SOM` [github.com/FlorentF9/sparkml-som]



Teaching

- SupGalilée (2018, 2019, 2020)
- ISAE-Supaero (2019)

Scientific outreach

- Organization of 1st workshop on Large-Scale Industrial Time Series Analysis (LITSA) @ IEEE ICDM 2020

Thank you for your attention!

Huge thanks to:

- The jury members
- Safran and ANRT for supporting this project
- All my colleagues and collaborators
- My thesis supervisors

Appendix menu

1. Appendix 1 — Representation learning for self-organized clustering Go to
2. Appendix 2 — Model selection in clustering Go to
3. Appendix 3 — Scalable aircraft engine health monitoring applications Go to

Deep clustering baselines

Table 4: Deep clustering baselines on benchmark data sets (% clustering accuracy).

Method	MNIST	Fashion-MNIST	USPS	Reuters-10k
K-means	53.3	54.9	66.0	58.9
AE + *K*-means	80.1	48.9	68.0	53.8
GMVAE [Dilokthanakul et al., 2017]	82.3	-	-	-
DCN [Yang et al., 2017]	83.0	-	-	80.0
DKM [Fard et al., 2018]	84.0	-	75.7	58.3
DEC [Xie et al., 2016]	86.6	51.8	74.1	73.7
IDEC [Guo et al., 2017]	88.1	52.9	76.1	75.6
VaDE [Jiang et al., 2017]	94.5	-	56.6	79.8
ClusterGAN [Mukherjee et al., 2019]	95.0	63.0	-	-
JULE [Yang et al., 2016]	96.1	56.3	95.0	-
DEPICT [Dizaji et al., 2017]	96.3	39.2	89.9	-
WaMiC [Harchaoui et al., 2019]	97.3	-	-	79.8
Dual AE [Yang et al., 2019]	98.0	66.2	86.9	-
GAR [Kilinc and Uysal, 2018]	98.3	-	96.5	-
IMSAT [Hu et al., 2017]	98.4	-	71.0	

Back to menu

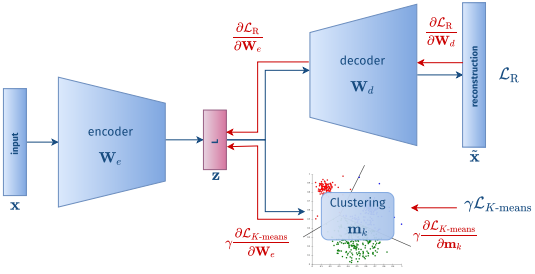
Deep Clustering Network (DCN)

Several methods based on (soft) K-means [Song et al., 2014, Xie et al., 2016, Guo et al., 2017, Fard et al., 2018].

DCN [Yang et al., 2017]

$$\mathcal{L}(\mathbf{W_e}, \mathbf{W_d}, \{\mathbf{m}_k\}_1^K, \{b_i\}_1^N) := \mathcal{L}_{\text{R}}(\mathbf{W_e}, \mathbf{W_d}) + \gamma \mathcal{L}_{K\text{-means}}(\mathbf{W_e}, \{\mathbf{m}_k\}_1^K, \{b_i\}_1^N) \text{ where } b_i := \underset{k}{\text{argmin}} \, ||\mathbf{z}_i - \mathbf{m}_k||^2$$

$$= \frac{1}{N} \sum_i ||\tilde{\mathbf{x}}_i - \mathbf{x}_i||_2^2 + \gamma \frac{1}{N} \sum_i ||\mathbf{z}_i - \mathbf{m}_{b_i}||_2^2$$



Alternating procedure:

1. Update cluster assignments
2. Jointly update network parameters and centroids

Back to menu

Comparison with other deep SOM approaches

Table 5: Comparison of the properties of deep SOM models.

Model	Latent space	AE	Rec. loss	SOM loss	Neighborhood	Joint	Pretraining
Deep neural maps [Pesteie et al., 2018]	continuous	AE	MSE	KL+SOM	Gaussian	✓	✓
DASOM [Ferles et al., 2018]	continuous	DAE	MSE	SOM	-	✓	✓
ConvSOM [Elend and Kramer, 2019]	continuous	AE	MSE	SOM	Gaussian	✗	✓
DESOM [Forest et al., 2019b]	continuous	AE	MSE	SOM	Gaussian	✓	✗
SOM-VAE [Fortuin et al., 2019]	discrete	VQ-VAE	ELBO	VQ+SOM	fixed	✓	✓
DPSOM [Manduchi et al., 2020]	continuous	VAE	ELBO	KL+SOM	fixed	✓	✓

Back to menu

Visualizing DESOM during training

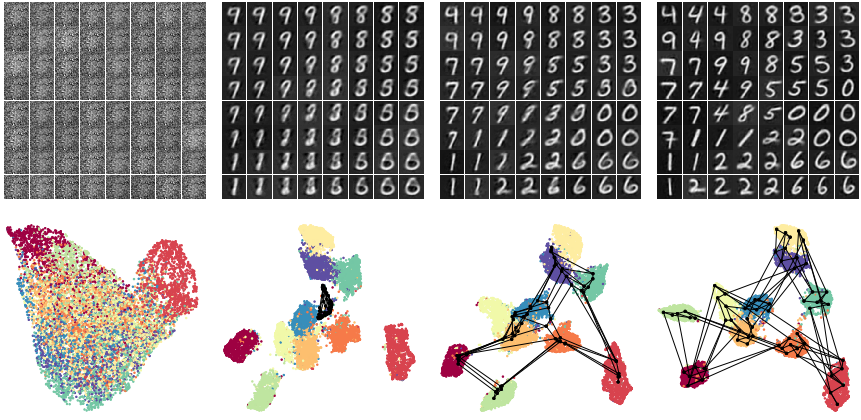


Figure 6: (Top) DESOM decoded prototypes and (bottom) UMAP visualization of latent space after 0, 10, 20 and 40 training epochs.

Back to menu

Comparison of DESOM with SOM

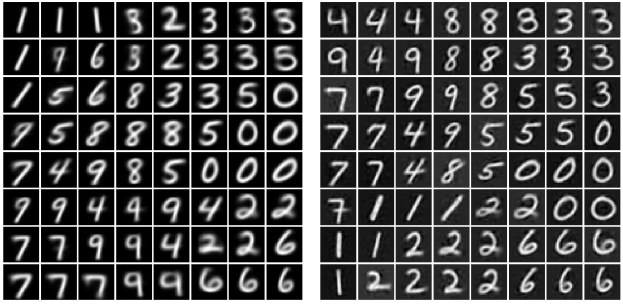


Figure 7: SOM (left) and DESOM (right) maps of the MNIST data set.

Back to menu

Comparison of DESOM with SOM

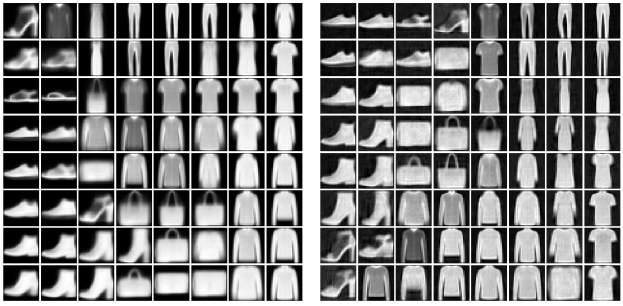


Figure 8: SOM (left) and DESOM (right) maps of the Fashion-MNIST data set.

Back to menu

Comparison of DESOM with VAE latent space

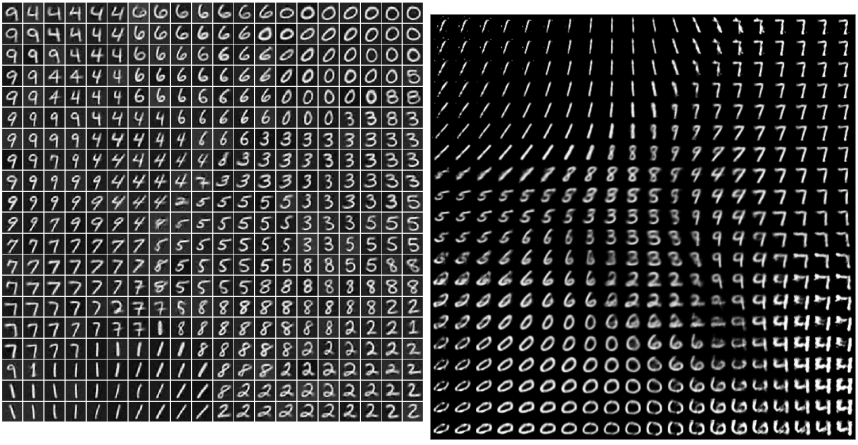


Figure 9: DESOM map (left) and VAE latent space visualization (right) for MNIST.

Back to menu

Properties of SOM-regularized latent space

Comparison of **quantization/topographic/combined errors** and **topographic product**.

Table 6: In original space:

Method	QE	TE	CE	TP
		MNIST		
SOM	**5.345**	**0.518**	**15.78**	**-0.073**
DESOM	5.848	0.597	21.74	-0.104
		Fashion-MNIST		
SOM	**4.537**	**0.477**	**12.40**	**-0.026**
DESOM	4.755	0.536	15.22	-0.046
		USPS		
SOM	**3.693**	**0.474**	**10.35**	**-0.055**
DESOM	4.025	0.556	14.62	-0.082
		Reuters-10k		
SOM	42.70	**0.595**	**102.4**	-0.206
DESOM	**41.81**	**0.754**	**113.8**	**-0.147**

Table 7: In latent space:

Method	$\hat{QE}$	$\hat{TE}$	$\hat{CE}$	$\hat{TP}$
		MNIST		
AE+SOM	1.231	**0.510**	4.429	-0.066
DESOM-AE+SOM	**0.205**	**0.514**	**0.713**	**-0.055**
DESOM	**0.205**	**0.534**	**0.727**	**-0.057**
		Fashion-MNIST		
AE+SOM	0.960	**0.532**	3.973	-0.059
DESOM-AE+SOM	**0.166**	0.572	**0.664**	**-0.044**
DESOM	**0.167**	**0.556**	**0.661**	**-0.045**
		USPS		
AE+SOM	3.926	0.689	19.88	-0.098
DESOM-AE+SOM	**0.278**	**0.554**	**1.174**	**-0.065**
DESOM	**0.280**	**0.563**	**1.184**	**-0.069**
		Reuters-10k		
AE+SOM	30.00	0.934	270.7	-0.146
DESOM-AE+SOM	**0.527**	**0.710**	**3.391**	**-0.071**
DESOM	**0.524**	**0.696**	**3.102**	**-0.069**

Back to menu

Hyperparameter influence studies

Table 8: Impact of DESOM hyperparameters.

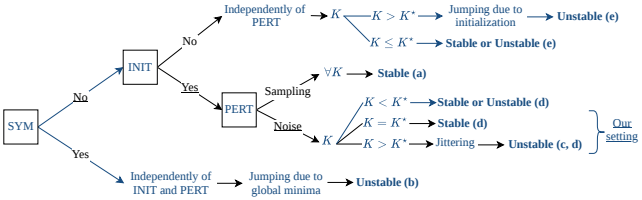
Parameter	Notation	Selected value	Studied range	Clustering metric Pur	Clustering metric NMI	Map metric QE	Map metric TE
Gamma	γ	10^{-3}	$10^{-4} - 10^{0}$	⇓	⇓	⇓	⇑
Latent code dimension	L	10	$2 - 100$	∩	∩	∅	∅
Map size	-	8×8	$5 \times 5 - 20 \times 20$	⇑	⇓	⇓	⇑
Initial temperature	$T_{\max}$	8.0	0.1, 8.0	∅	∅	∅	⇓
Batch size	n_b	256	$16 - 256$	⇑	⇑	∅	∅

"When <parameter> increases, <metric> ⇑ increases, ⇓ decreases, ∩ has an optimal value." ∅ is not significantly impacted." quality becomes better." worse."

AE pre-training and SOM initialization do not improve performance.

Back to menu

Sources of instability



Justified by or related to:
(a) **Theorem 10** (Stability theorem) [1]
Lemma 1 (Stability and global optima of the objective function) [3]
(b) **Theorem 15** (Instability from symmetry) [1]
Lemma 1 (Stability and global optima of the objective function) [3]
(c) **Theorem 4** (High instability implies cut in high density region) [2]
Conclusion 3 (Instable clusterings) [3]
(d) **Conjecture 4** (Stable clusterings) [3]
Conclusion 5 (Stability of idealized K-means detects whether K is too large) [3]
(e) **Conjecture 8** (Stability of the actual K-means algorithm) [3]

Notations:
SYM: Symmetries in the data distribution
INIT: Effective initialization scheme
PERT: Perturbation process

References:
[1] Ben-David and von Luxburg (2006)
[2] Ben-David and von Luxburg (2008)
[3] von Luxburg (2010)

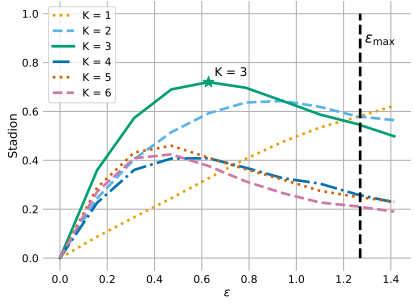
Figure 10: Diagram explaining sources of instability in different settings for K-means.

Back to menu

Whenever $K^\star$ is not the best partition

(a) $K = 2$ (ARI = 0.74) (b) $K = 3$ (ARI = 0.92) (c) $K^\star = 4$ (ARI = 0.58) (d) $K = 5$ (ARI = 0.65)

K	ARI	StabB	StabW	Stadion	
1	0.00	++	- -	0	✗
2	0.74	++	-	+	✗
3	0.92	++	+	+++	✓
4	0.58	- -	+	-	✗
5	0.65	- -	++	0	✗



Benchmark results

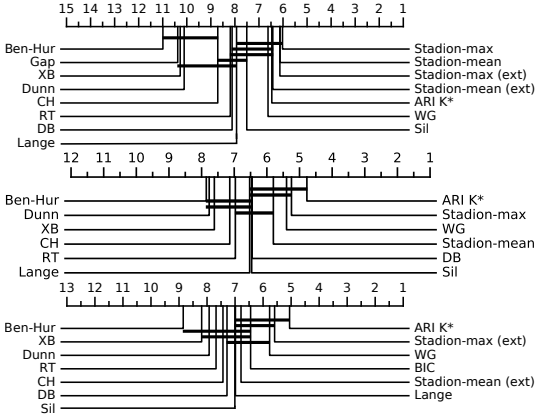


Figure 11: CD diagram after Wilcoxon-Holms test on ARI performance across 73 artificial data sets for *K*-means, Ward, and GMM.

Hyperparameter study

Stadion hyperparameters and conclusions:

- $D \in \{1, \ldots, \mathbf{10}\}$
  $\rightarrow$ No significant influence
- noise: **uniform** or Gaussian
  $\rightarrow$ No significant influence
- $\mathbf{\Omega} \in \{2, 3, 5, 10, \{2, \ldots, 5\}, \{\mathbf{2}, \ldots, \mathbf{10}\}, \{10, \ldots, 20\}, \{2, \ldots 20\}\}$
  $\rightarrow$ Best with $\{2, \ldots, 5\}$ or $\{2, \ldots, 10\}$
- $s \in \{\mathbf{ARI}_1, \mathrm{ARI}_2, \mathrm{RI}, \mathrm{FM}, \mathrm{JACC}, \mathrm{MI}, \mathrm{AMI}, \mathrm{VI}, \mathrm{NVI}, \mathrm{ID}, \mathrm{NID}, \mathrm{NMI}_1, \mathrm{NMI}_2, \mathrm{NMI}_3, \mathrm{NMI}_4, \mathrm{NMI}_5\}$
  $\rightarrow$ No significant influence for most adjusted/normalized measures

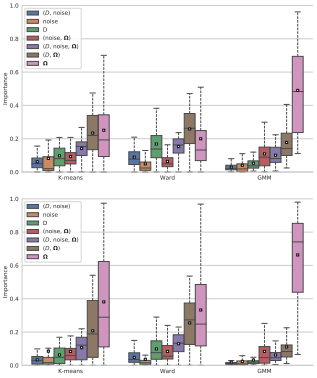
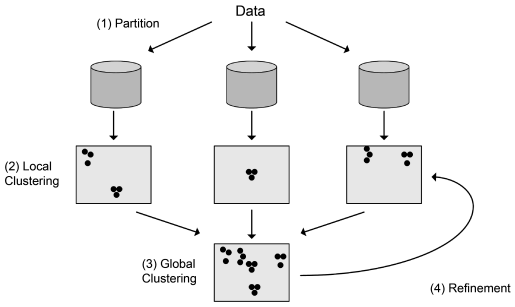


Figure 12: fANOVA importance of hyperparameters and their interactions for Stadion-max (top) and mean (bottom).

Distributed clustering algorithms [Aggarwal and Reddy, 2013]



Back to menu

Vibration profiles SOM visualizations

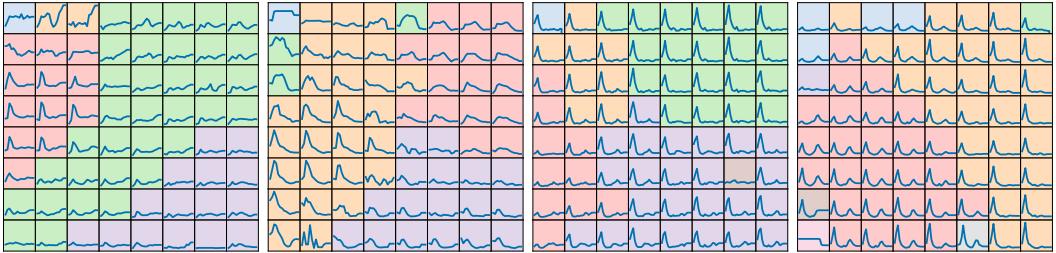


Figure 13: SOM maps of signature 1, 2, 3 and 4 (LP-ACC1 vs N1, LP-ACC2 vs N1, HP-ACC1 vs N2, HP-ACC2 vs N2).

Back to menu

References i

Aggarwal, C. C. and Reddy, C. K. (2013).
Data Clustering: Algorithms and Applications.

Anouar, F., Badran, F., and Thiria, S. (1998).
Probabilistic self-organizing map and radial basis function networks.
Neurocomputing, 20(1-3):83–96.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., and Perona, I. (2013).
An extensive comparative study of cluster validity indices.
Pattern Recognition, 46(1):243–256.

Attaoui, M. O., Azzag, H., Lebbah, M., and Keskes, N. (2020).
Subspace data stream clustering with global and local weighting models.
Neural Computing and Applications, 0123456789.
https://doi.org/10.1007/s00521-020-05184-z.

Bastard, G., Lacaille, J., Coupard, J., and Stouky, Y. (2016).
Engine Health Management in Safran Aircraft Engines.
In *Annual Conference of the PHM Society*.

Ben-David, S. (2018).
Clustering - What both theoreticians and practitioners are doing wrong.
AAAI Conference on Artificial Intelligence, pages 7962–7964.

References ii

Ben-David, S. and Von Luxburg, U. (2008).
Relating clustering stability to properties of cluster boundaries.
Conference on Learning Theory (COLT), pages 379–390.

Ben-David, S., Von Luxburg, U., and Pál, D. (2006).
A sober look at clustering stability.
Lecture Notes in Computer Science, 4005(2002):5–19.

Ben-Hur, A., Elisseeff, A., and Guyon, I. (2002).
A stability based method for discovering structure in clustered data.
Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing, 17:6–17.

Benabdeslem, K. and Lebbah, M. (2007).
Feature selection for self-organizing map.
In *International Conference on Information Technology Interfaces (ITI)*, pages 45–58.

Bengio, Y. (2012).
Deep Learning of Representations for Unsupervised and Transfer Learning.
JMLR: Workshop and Conference Proceedings, 27:17–37.

Biernacki, C., Celeux, G., and Govaert, G. (2000).
Assessing a Mixture Model for Clustering with Integrated Completed likelihood.
IEEE Transactions on Pattern Analysis and Machine Learning, 22(7):1899–1906.

References iii

Bishop, C. M., Svensen, M., and Williams, C. K. I. (1998).
GTM: The Generative Topographic Mapping.
Neural Computation, 10(1):215–234.
http://www.mitpressjournals.org/doi/10.1162/089976698300017953.

Bouveyron, C., Girard, S., and Schmid, C. (2007).
High-dimensional data clustering.
Computational Statistics and Data Analysis, 52(1):502–519.

Chavent, M., Lacaille, J., Mourer, A., and Olteanu, M. (2020).
Sparse k -means for mixed data via group-sparse clustering.
In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*.

Côme, E., Cottrell, M., Verleysen, M., and Lacaille, J. (2010a).
Aircraft engine health monitoring using Self-Organizing Maps.
In *Industrial Conference on Data Mining*.

Côme, E., Cottrell, M., Verleysen, M., and Lacaille, J. (2010b).
Self organizing star (SOS) for health monitoring.
In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, number April, pages 99–104.

References iv

Côme, E., Cottrell, M., Verleysen, M., and Lacaille, J. (2011).
Aircraft engine fleet monitoring using Self-Organizing Maps and Edit Distance.
In *International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM)*, pages 298–307.

Cottrell, M., Gaubert, P., Eloy, C., François, D., Hallaux, G., Lacaille, J., and Verleysen, M. (2009).
Fault prediction in aircraft engines using Self-Organizing Maps.
In *International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM)*.

De La Torre, F. and Kanade, T. (2006).
Discriminative cluster analysis.
In *International Conference on Machine Learning (ICML)*.

De Soete, G. and Carroll, J. D. (1994).
K-means clustering in a low-dimensional Euclidean space.
In Diday, E., Lechevallier, Y., Schader, M., Bertrand, P., and Burtschy, B., editors, *New Approaches in Classification and Data Analysis*, pages 212–219, Berlin, Heidelberg. Springer Berlin Heidelberg.

Desgraupes, B. (2013).
ClusterCrit: Clustering Indices.
`cran.r-project.org/web/packages/clusterCrit`.

Dilokthanakul, N., Mediano, P. A. M., Garnelo, M., Lee, M. C. H., Salimbeni, H., Arulkumaran, K., and Shanahan, M. (2017).
Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders.

References v

Ding, C. and Li, T. (2007).
Adaptive dimension reduction using discriminant analysis and K-means clustering.
In *International Conference on Machine Learning (ICML)*.

Dizaji, K. G., Herandi, A., Deng, C., Cai, W., and Huang, H. (2017).
Deep Clustering via Joint Convolutional Autoencoder Embedding and Relative Entropy Minimization.
In *ICCV*, pages 5747–5756.

Elend, L. and Kramer, O. (2019).
Self-Organizing Maps with Convolutional Layers.
In *International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM)*.

Elhamifar, E. and Vidal, R. (2013).
Sparse subspace clustering: Algorithm, theory, and applications.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(11):2765–2781.

Fard, M. M., Thonet, T., and Gaussier, E. (2018).
Deep k-Means: Jointly Clustering with k-Means and Learning Representations.
`http://arxiv.org/abs/1806.10069`.

Faure, C., Olteanu, M., Bardet, J.-M., and Lacaille, J. (2017).
Using self-organizing maps for clustering and labelling aircraft engine data phases.
In *International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM)*.

References vi


Ferles, C., Papanikolaou, Y., and Naidoo, K. J. (2018).
Denoising Autoencoder Self-Organizing Map (DASOM).
Neural Networks, 105:112–131.
https://doi.org/10.1016/j.neunet.2018.04.016.


Forest, F., Cochard, Q., Noyer, C., Cabut, A., Joncour, M., Lacaille, J., Lebbah, M., and Azzag, H. (2020a).
Large-scale Vibration Monitoring of Aircraft Engines from Operational Data using Self-organized Models.
In *Annual Conference of the PHM Society*.


Forest, F., Lacaille, J., Lebbah, M., and Azzag, H. (2018).
A Generic and Scalable Pipeline for Large-Scale Analytics of Continuous Aircraft Engine Data.
In *IEEE International Conference on Big Data*.


Forest, F., Lebbah, M., Azzag, H., and Lacaille, J. (2019a).
Deep Architectures for Joint Clustering and Visualization with Self-Organizing Maps.
In *PAKDD Workshop on Learning Data Representations for Clustering (LDRC)*.


Forest, F., Lebbah, M., Azzag, H., and Lacaille, J. (2019b).
Deep Embedded SOM: Joint Representation Learning and Self-Organization.
In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*.

References vii

Forest, F., Lebbah, M., Azzag, H., and Lacaille, J. (2020b).
Carte SOM profonde : Apprentissage joint de représentations et auto-organisation.
In *CAp: Conférence d'Apprentissage*.
https://hal.archives-ouvertes.fr/hal-02859997.

Fortuin, V., Hüser, M., Locatello, F., Strathmann, H., and Rätsch, G. (2019).
SOM-VAE: Interpretable Discrete Representation Learning on Time Series.
In *International Conference on Learning Representations (ICLR)*.

Fritzke, B. (1995).
A Growing Neural Gas Learns Topologies.
In *NIPS*, volume 7, pages 625–632.

Guo, X., Gao, L., Liu, X., and Yin, J. (2017).
Improved deep embedded clustering with local structure preservation.
In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1753–1759.

Harchaoui, W., Mattei, P.-A., Alamansa, A., and Bouveyron, C. (2019).
Wasserstein Adversarial Mixture for Deep Generative Modeling and Clustering.
In *AISTATS*.

Hu, W., Miyato, T., Tokui, S., Matsumoto, E., and Sugiyama, M. (2017).
Learning discrete representations via information maximizing self-augmented training.
In *International Conference on Machine Learning (ICML)*, volume 4, pages 2467–2481.

References viii


Jain, A. K. (2010).
Data clustering: 50 years beyond K-means.
Pattern Recognition Letters, 31(8):651–666.
http://dx.doi.org/10.1016/j.patrec.2009.09.011.


Jiang, Z., Zheng, Y., Tan, H., Tang, B., and Zhou, H. (2017).
Variational Deep Embedding : An Unsupervised and Generative Approach to Clustering.
In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1965–1972.


Kaly, F., Niang, N., Ouattara, M., Niang, A., Thiria, S., Marticorena, B., and Janicot, S. (2004).
Two step soft subspace SOM : une méthode de classification multi-bloc avec sélection de variables.


Kilinc, O. and Uysal, I. (2018).
Learning latent representations in neural networks for clustering through pseudo supervision and graph-based activity regularization.
In *International Conference on Learning Representations (ICLR)*.


Kleinberg, J. (2003).
An impossibility theorem for clustering.
Advances in Neural Information Processing Systems.


Kohonen, T. (1982).
Self-organized formation of topologically correct feature maps.
Biological Cybernetics, 43(1):59–69.

References ix

Lange, T., Roth, V., Braun, M. L., and Buhmann, J. M. (2004).
Stability-based validation of clustering solutions.
Neural Computation, 16(6):1299–1323.

Lee, J. A. and Verleysen, M. (2007).
Nonlinear Dimensionality Reduction.
Springer.

Manduchi, L., Hüser, M., Rätsch, G., and Fortuin, V. (2020).
DPSOM: Deep Probabilistic Clustering with Self-Organizing Maps.
http://arxiv.org/abs/1910.01590.

Martinetz, T. and Schulten, K. (1991).
A "Neural-Gas" Network Learns Topologies.
http://web.cs.swarthmore.edu/$\sim$meeden/DevelopmentalRobotics/fritzke95.pdf.

Martinetz, T. and Schulten, K. (1994).
Topology representing networks.
Neural Networks, 7(3):507–522.

Mourer, A., Forest, F., Lebbah, M., Azzag, H., and Lacaille, J. (2020).
Selecting the Number of Clusters K with a Stability Trade-off: an Internal Validation Criterion.
https://arxiv.org/abs/2006.08530.

References x

Mukherjee, S., Asnani, H., Lin, E., and Kannan, S. (2019).
ClusterGAN: Latent Space Clustering in Generative Adversarial Networks.
AAAI Conference on Artificial Intelligence, 33:4610–4617.
https://aaai.org/ojs/index.php/AAAI/article/view/4385.

Pesteie, M., Abolmaesumi, P., and Rohling, R. (2018).
Deep Neural Maps.
In *ICML workshop*.
http://arxiv.org/abs/1810.07291.

Randall, R. B. (2011).
***Vibration-based condition monitoring*.**
Wiley.

Roth, V., Lange, T., Braun, M., and Buhmann, J. (2002).
A Resampling Approach to Cluster Validation.
Compstat, pages 123–128.

Sarazin, T., Azzag, H., and Lebbah, M. (2014).
SOM clustering using spark-MapReduce.
In *International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*.

Shalev-Shwartz, S. and Ben-David, S. (2013).
***Understanding machine learning: From theory to algorithms*, volume 9781107057.**

References xi

Song, C., Huang, Y., Liu, F., Wang, Z., and Wang, L. (2014).
Deep auto-encoder based clustering.
Intelligent Data Analysis, 18(6).

Sun, W., Wang, J., and Fang, Y. (2012).
Regularized k-means clustering of high-dimensional data and its asymptotic consistency.
Electronic Journal of Statistics, 6(April 2011):148–167.

Von Luxburg, U. (2009).
Clustering stability: An overview.
Foundations and Trends in Machine Learning, 2(3):129–168.

von Luxburg, U., Williamson, R. C., and Guyon, I. (2012).
Clustering: Science or Art?
JMLR: Workshop and Conference Proceedings, 27:6579.

Wang, F., Wang, Q., Nie, F., Li, Z., Yu, W., and Wang, R. (2019).
Unsupervised Linear Discriminant Analysis for Jointly Clustering and Subspace Learning.
IEEE Transactions on Knowledge and Data Engineering, 4347(c).

Witten, D. M. and Tibshirani, R. (2010).
A framework for feature selection in clustering.
Journal of the American Statistical Association, 105(490):713–726.

References xii


Xie, J., Girshick, R., and Farhadi, A. (2016).
Unsupervised Deep Embedding for Clustering Analysis.
In *International Conference on Machine Learning (ICML)*, volume 48.
http://arxiv.org/abs/1511.06335.


Yang, B., Fu, X., Sidiropoulos, N. D., and Hong, M. (2017).
Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering.
In *International Conference on Machine Learning (ICML)*.
http://arxiv.org/abs/1610.04794.


Yang, J., Parikh, D., and Batra, D. (2016).
Joint Unsupervised Learning of Deep Representations and Image Clusters.
http://arxiv.org/abs/1604.03628.


Yang, X., Deng, C., Zheng, F., Yan, J., and Liu, W. (2019).
Deep spectral clustering using dual autoencoder network.
In *CVPR*, pages 4061–4070.


Ye, J. (2007).
Discriminative K-means for Clustering.
In *NIPS*.