Unsupervised Learning: Basic Concepts and Application to Particle Dynamics

Noel Jakse

Université Grenoble Alpes, CNRS, Grenoble INP, SIMaP, 38000 Grenoble, France

1 Introduction

Machine learning (ML) approaches attracted significant interest in many scientific fields due to their potential ability to uncover patterns, make predictions, and extract valuable insights from large and complex body of data [EH21]. Web-based data collection platforms, scientific instruments and computer simulations are creating exponentially increasing data stores. they triggered new scientific methods to analyze and organize huge amount of data, giving the possibility to find subtle effects missed previously [HT20, HTTG09].

ML tools can be broadly categorized into three types: supervised, unsupervised, and semi-supervised learning. Each type offers distinct approaches to learning from data, with specific strengths and limitations [EH21, HTFF09]. Supervised learning is one of the most used approach and involves training a model using labeled data, where both input features and corresponding output labels are provided. The model learns to map inputs to their respective outputs, generalizing to make predictions for new, unseen data. Applications of supervised learning include regression and classification tasks that will be be the subject of Chapters 2 and 4 of the present Book.

However, obtaining labeled data can be time-consuming and expensive, especially for large datasets. Unsupervised learning, on the other hand, works with unlabeled data, where the model discovers patterns and structures within the data without prior knowledge of the desired output. It is particularly useful in exploratory data analysis, clustering, dimensionality reduction, and anomaly detection. Finally, semi-supervised learning combines aspects of both supervised and unsupervised learning, leveraging a small amount of labeled data with a large amount of unlabeled data. The method is based on unsupervised techniques to extract features from the unlabeled data, and supervised learning is then applied to improve the trained model using the labeled data. This approach often results in improved model performance compared to purely

supervised or unsupervised learning, especially when labeled data is limited.

Unsupervised learning, at the heart of the present Chapter, is useful in situations where labeled data are scarce, enabling insights from larger datasets. It can be useful to deal with large datasets of unlabelled instance, and is particularly relevant when hidden patterns and structures in data have to be extracted without prior knowledge on any label or output. It plays a vital role in scientific fields such as materials science [Cer19, SMBM19] and geophysics [WT10, KA21], where labeled data are often unavailable, scarce or expensive to calculate or to measure. It enables to analyze large datasets and identify essential relationships, leading to the discovery of new materials or the understanding of complex geological processes.

The Chapter is mainly divided in two parts. The first part is devoted to the comprehensive description of the basic concepts and most popular techniques of unsupervised learning. For the latter, the choice was made to describe two main branches, which are the Clustering and Dimensionality Reduction techniques [HTFF09]. Unsupervised learning has numerous real-world applications in many domains. As an example of applications, the second part of this Chapter illustrates an application of unsupervised learning to the discovery of patterns in particles dynamics. A particular focus is made on large scale molecular dynamics simulations performed with up to 10 million atoms for the purpose of describing the early stages of solidification of a materials so-called Homogeneous nucleation [SCC$^+$16, BDMJ22b]. While specific to material science, this example can surely transposed at higher scales with Discrete Particle Dynamics (DPD) modelling in geophysics, which is at the heart of this Book.

2 Basic concepts

2.1 Representation of the data: Feature extraction and selection

Feature extraction and representation are fundamental aspects of unsupervised learning as in most of the ML techniques. The basic idea sketched in Fig. 1 is to transform raw data into a format, often a vector, that can be easily processed and analyzed by ML algorithms, such a representation is so-called hereafter a descriptor. Having say that, the major objective of this transformation is then to identify and extract the most relevant and informative features from the data, intending to capture the underlying patterns and structures while reducing noise and redundancy.

Feature extraction techniques can be divided into two categories: feature construction and feature selection [KKN14]. As mentioned above, feature construction involves creating new features from the original data, often through mathematical transformations or domain-specific knowledge. For example, in image processing, features can be extracted using edge detection or texture analysis. In text analysis, natural language processing techniques like tokenization, stemming, and term frequency-inverse document frequency (TF-IDF) can be employed to extract meaningful features [NQY18] in the so-called Bag of Words representation. In the application treated in the present Chapter regarding atomic scale simulations of homogeneous crystal nucleation, a topological

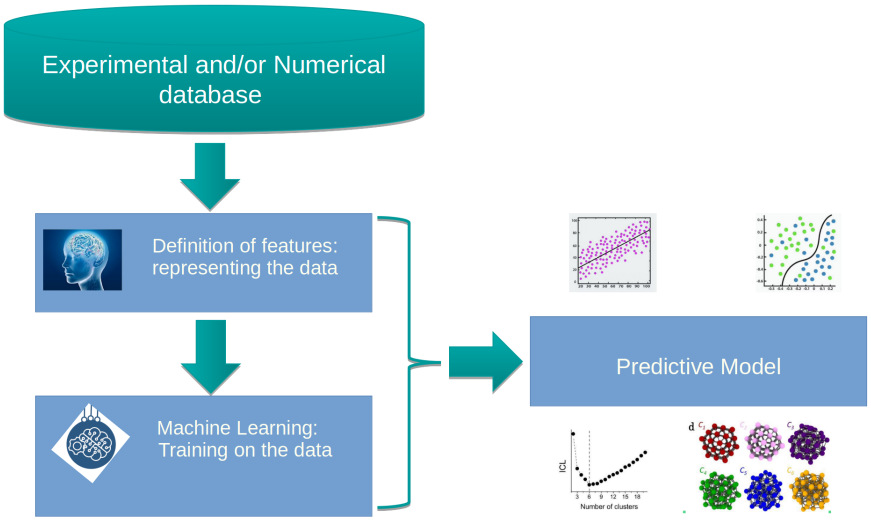

Figure 1: Typical machine learning flowchart that learns a predictive model from the data.

descriptor is built [BDMJ22a] based on persistence homology (PH) in the framework of the topological data analysis (TDA) [PSG$^+$18, Car20].

This poses naturally the question of the dimensionality of the descriptor after extraction. High-dimensional feature spaces can pose challenges for unsupervised learning algorithms, as the increase in dimensions can lead to increased computational complexity, reduced model interpretability, and over-fitting. This phenomenon is known as the "curse of dimensionality" [HTFF09, EH21]. Feature selection addresses this problem by selecting a subset of the original features that are most relevant to the task at hand, in order to reduce the dimensionality of the data. There are three primary methods for feature selection among others. Filter methods evaluate the relevance of individual features based on their statistical properties, independent of any learning algorithm. Examples of filter methods include correlation coefficients, mutual information, and chi-squared tests. Wrapper methods use a specific learning algorithm's performance to guide the feature selection process, by iteratively adding or remove features based on their impact on the model's performance (*i.e* forward selection, backward elimination, and recursive feature elimination).

Finally, embedded methods integrate feature selection into the model training process, combining the advantages of both filter and wrapper methods. One advantage is the assessment the importance of features during the training phase, automatically selecting the most relevant ones. Emblematic examples of embedded methods include LASSO,

ridge regression, and decision trees.

2.2 Distance and similarity metrics

In order to quantify the relationship between data points in the feature space of dimension D, defining metrics is at the heart of unsupervised learning. These metrics provide a basis for comparing and grouping similar instances in clustering.

The most widely used distance metric, measuring the straight-line distance between two points in Euclidean D-dimensional feature space. Considering two points p and q, it is given by

$$d(p,q) = \sqrt{\sum_{i=1}^{D}(p_i - q_i)^2}. \tag{1}$$

This applies for continuous features and is sensitive to the scale of the data. A simpler version is the Manhattan distance, also known as the L_1-norm, which measures the sum of the absolute differences namely

$$d_{L_1}(p,q) = \sum_{i=1}^{N}|p_i - q_i|, \tag{2}$$

and is less sensitive to outliers compared to Euclidean distance. When the dimensionality of the feature space is high or when the magnitude of the vectors is less relevant, such as in text analysis, the cosine similarity metrics is relevant and measures the angle between two feature vectors $\mathbf{p}$ and $\mathbf{q}$ as

$$\text{cosine similarity}(\mathbf{p},\mathbf{q}) = \frac{\mathbf{p}\cdot\mathbf{q}}{\|\mathbf{p}\|\|\mathbf{q}\|} = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2}\sqrt{\sum_{i=1}^{n} q_i^2}}. \tag{3}$$

If sets of data have to be compared, the Jaccard similarity metrics can be used for comparing binary or set-based data, measuring the ratio of the intersection to the union of two sets A and B is given by:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}. \tag{4}$$

Alternatively, Pearson correlation coefficient measures the linear relationship between two variables of size n, ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation) with

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}. \tag{5}$$

These are the most common ones, however the metrics to be used should be decided given the specificity of the data at hand. The choice can affect the results of the learning.

3 Unsupervised Learning Techniques

Unsupervised learning comprises a diverse range of methods and algorithms dealing with unlabeled data [EH21]. This section provides an overview of some popular unsupervised learning techniques and their applications, focusing only on *clustering* and *dimensionality reduction* methods. A corresponding, comprehensive tutorial through a JUPYTER notebook attached to this Chapter will explore the most common ones.

What will not be treated in the present Chapter is the combination of powerful deep learning models with unsupervised learning techniques [HWWT13]. These so-called *deep unsupervised techniques* are Autoencoders and Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), Self-supervised Learning and Deep Clustering.

3.1 Clustering

Clustering is the grouping process of data points based on their features. Many clustering methods were already proposed [HTFF09]. The key notion is the degree of similarity (or dissimilarity) between the individual objects being associated. Indeed, the method attempts to group observations with respect to some similarity criterion. Combinatorial methods were proposed to avoid a modeling through a probability distribution function. However, an exact solution is computable only when the dataset is very small. As a matter of fact, testing every combination of 20 observations into 4 clusters needs 10^{10} comparisons, which is basically unreachable).

Approximations of these methods, were proposed through iterative process. The famous K-means method is a standard combinatorial algorithm, based on the Euclidean distance given by Eq. (1). A standard version of the algorithm given in Al. 1. Given a number K of points in the dataset of size n $X = \{x_1, x_2, \ldots, x_n\}$ are randomly denoted as the initial K centroids, and each observation is assigned to the cluster with the closest centroid. Then, the centroids are updated, and the process is iterated. As an illustration, Figure 2 schematize the application of the K-means to as dataset randomly created as a collection of three distinct blobs [1] as a guide for the eyes, and with the same color to indicate the fact that the data are unlabelled.

Adding a probabilistic view on this modeling with a soft assignment of each observation to each cluster, leads to Gaussian Mixture Models, with spherical covariance matrices, proportional to the identity matrix. The idea behind is that the data distribution can be well approximated by a mixture of K multivariate Gaussian distributions:

$$\sum_{k=1}^{K} \pi_k \Phi(\mu_k, \Sigma_k), \tag{6}$$

[1] see Scikit-learn blobs

Algorithm 1 K-means

Require: Data points $X = \{x_1, x_2, \ldots, x_n\}$, number of clusters K
Ensure: Cluster assignments $C = \{c_1, c_2, \ldots, c_K\}$, centroids $\mu = \{\mu_1, \mu_2, \ldots, \mu_K\}$
Initialize centroids $\mu_1, \mu_2, \ldots, \mu_K$ randomly from X
repeat
Assign each point x_i to the nearest centroid: $c_i = \arg\min_k \|x_i - \mu_k\|^2$
Update centroids by computing the mean of assigned points: $\mu_k = \frac{\sum_{i=1}^{n} I(c_i=k) x_i}{\sum_{i=1}^{n} I(c_i=k)}$
until Convergence (centroids do not change significantly or a maximum number of iterations is reached)

with Φ the multivariate Gaussian distribution, where the cluster $k \in \{1, \ldots, K\}$ is described by its proportion π_k among the full dataset, its mean μ_k and its covariance matrix Σ_k. Estimation of parameters is classically done using the Expectation Maximization (EM) algorithm [DLR77].

The main issue of most clustering methods is the number of clusters, that has to be set beforehand by a human. To circonvent the arbitrariness of choosing a number of cluster, and when there is a probability density function, the likelihood should be considered. It consists in measuring how close the samples are to the specified clustering distribution. However, this criterion will always be better for a higher number clusters, which lead to the classical problem of over-fitting. If the well-known elbow criterion was largely used to penalize such criterion, it is not related to any theoretical background, and is always seem very objective. Model selection criterion can be used, adapting the likelihood by penalizing it with respect to the dimension (related to the number of clusters) of each model. Usually, Akaïke Information Criterion (AIC) [Aka73] or Bayesian Information Criterion (BIC) [Sch78] are proposed, but the Integrated Completed Likelihood ICL [BCG00] should be preferred when focusing on clustering, because there is also a penalty term about the purity of each cluster based on a Shanon entropy term.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [EKSX96] is another famous clustering algorithm, known to be more robust. Observations lying in low-density regions are labeled as outliers. No assumptions on the form of the clusters is done, allowing for convex or non convex clusters. Contrarily to K-means and GMM, DBSCAN does not directly rely on a given number of clusters, but on hyperparameters that are closely related to the number of clusters: thresholds on the similarity and outliers.

3.2 Dimensionality Reduction

Dimensionality reduction techniques aim to project high-dimensional data into a lower-dimensional space while preserving the underlying structure and relationships in the data at best. These techniques are widely used for visualization, data compression, and noise reduction, for instance. This Section focuses on the *Principal Component*

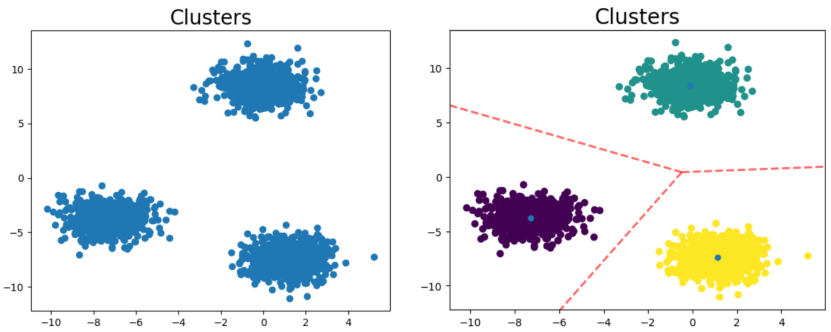

Figure 2: Schematic representation of a K-means clustering. Initial unlabeled (same color) data points are shown in the left panel, on which the K-means is applied, choosing obviously $K = 3$ cluster. After convergence, the K-means model is represented on the right panel together with the blue points being the positions of the centroids. The red dashed lines highlights the separation in the three domains given the Euclidean distance metrics: the cluster space. Thus, with this model, each new point can now unambiguously be associated to one of the three clusters given its position in the feature space (here, the x- and y-axis in arbitrary units).

Analysis (PCA) and a non-linear variant, namely the *t-Distributed Stochastic Neighbor Embedding* (t-SNE).

PCA is a widely-used linear technique that projects the data onto a lower-dimensional subspace, maximizing the variance along the new axes. The principal components, which form the basis of the new subspace, are orthogonal and capture the directions of maximum variance in the data. The measure of the variance in the data is based on the calculation of the covariance matrix of the n data points in D-dimensional feature space. For two points x and y, it is given by:

$$\Sigma = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{pmatrix}, \tag{7}$$

where σ_x^2 and σ_y^2 are the variances of x and y, respectively, and σ_{xy} and σ_{yx} are the covariances between x and y. The variances read

$$\sigma_x^2 = \frac{1}{D-1} \sum_{i=1}^{D} (x_i - \bar{x})^2, \tag{8}$$

where $\bar{x}$ is the mean of x, and the same for Y. The covariance between x and Y reads

$$\sigma_{xy} = \sigma_{yx} = \frac{1}{D-1} \sum_{i=1}^{D} (x_i - \bar{x})(y_i - \bar{y}). \tag{9}$$

Algorithm 2 Principal Component Analysis

Require: Data points of dimension D, $X = \{x_1, x_2, \ldots, x_n\}$target dimensionality k
Ensure: Transformed data points $Y = \{y_1, y_2, \ldots, y_n\}$
Compute the mean vector $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$
Center the data points by subtracting the mean: $X_{centered} = X - \mu$
Compute the covariance matrix Σ
Calculate eigenvectors and eigenvalues of Σ: $(\lambda_1, v_1), \ldots, (\lambda_n, v_n)$
Sort eigenvectors by decreasing eigenvalues: $v_1, \ldots, v_n$
Select the top k eigenvectors: $V_k = [v_1, \ldots, v_k]$
Project the centered data onto the principal components: $Y = X_{centered} V_k$

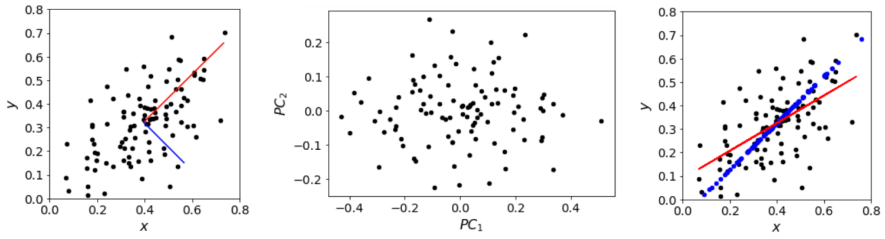

Figure 3: Principal Component analysis of simple bivariate random Gaussian dataset with two different variances, and having a linear correlation between the two random variables. The left panel shows the point-cloud of the 200 point dataset in the x-y plane together with the eigenvector with the largest eigenvalue in red, and the second one in blue, from the PCA analysis. Center panel displays the dataset in the basis of these two eigenvectors red and blue, respectively denoted PC1_1 and PC_2, and called the first and second principal components. The right panel shows in blue the PCA representation of the point cloud along PC_1 only. The red line is a linear regression model that was learned on the same dataset (see Chapter 2).

The lower-dimensional representation $Y = \{y_1, y_2, \ldots, y_n\}$ with target dimensionality k of a dataset $X = \{x_1, x_2, \ldots, x_n\}$ of dimensionality D is obtained through Algorithm 2. The n data points are projected on the space of the k eigenvectors whose eigenvalues are the largest and sorted by decreasing values.

Figure 2 illustrates the application of the PCA on a simple arbitrary bivariate random Gaussian dataset with 200 point-cloud using the `Scikit-learn` PYTHON package [2]. Interestingly, projection of the data on the first principal component, PC_1, is compared to a linear regression model learned on the same dataset. The good comparison indicates indeed that PC_1 carry the most important part of the information about the dataset.

Contrarily to PCA, t-SNE is a nonlinear dimensionality reduction technique. It is often used for visualizing high-dimensional data, such as images or text embeddings, and

[2] PCA decomposition module of the Scikit-learn package

is effective at revealing clusters and structures in the data, most often in 2D and 3D for visualization purposes [VdMH08]. t-SNE works by minimizing the divergence between two probability distributions, one representing pairwise similarities in the high-dimensional space

$$p_{j|i} = \frac{\exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-||x_i - x_k||^2/2\sigma_i^2)} \tag{10}$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{11}$$

using a Gaussian kernel and the other representing pairwise similarities in the low-dimensional space

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k\neq l}(1 + ||y_k - y_l||^2)^{-1}} \tag{12}$$

using the Student t-statistics. The algorithm consists in minimizing the Kullback-Leibler divergence between the two similarity matrices using gradient descent [HR02, VdMH08]. The perplexity parameter p is a user-defined parameter that balances the focus on local and global structure in the data.

4 Application to particle dynamics

Unsupervised learning has a broad range of applications across various domains, as it can uncover hidden structures and relationships in data without the need for labeled examples. We restrict here to a specific cased in materials science, namely the atomic scale description of homogeneous nucleation, for which the unsupervised learning together with a topological descriptor was applied successfully for the first time very recently [BDMJ22b, BDMJ22a]. Interestingly enough, this concerns particle dynamics and therefore could undoubtedly be applied at larger scales in geophysics.

Crystal nucleation, the early stages where the liquid-to-solid transition occurs upon undercooling, initiates at the atomic level on nanometer length and sub-picoseconds time scales and involves complex multidimensional mechanisms with local symmetry breaking that can hardly be observed experimentally in the very details. In such cases, atomic-level simulations and more particularly molecular dynamics (MD) with a suitable interaction model [AT17] is the dedicated tool. However, reaching statistically meaningful nucleation events, large scale simulations up to million-to billion-atom scale is required [SCC$^+$16].

In this Section, an analysis is proposed for the MD simulations performed previously on the solidification of pure zirconium [BDMJ20]. The liquid state above the melting point (approximately $T_M = 2128$ K) was quenched down to the deep undercooled liquid at $T = 1250$ K, using a simulation box of 1 million atoms. The technical details of the MD simulation are beyond the scope of the present Chapter, and the reader is referred to Ref. [BDMJ20]. During the simulation, the liquid undergoes homogeneous nucleation on this isotherm as shown in Figure 4. In order to unveil their structural

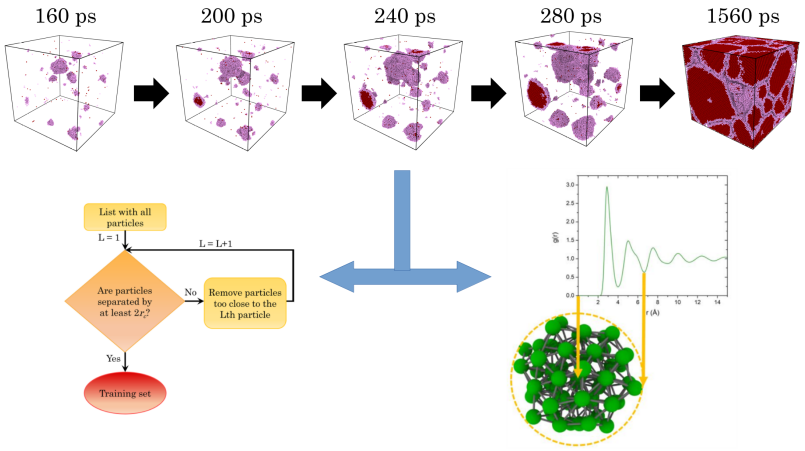

Figure 4: Snapshot of a one-million atom MD simulation of zirconium during nucleation along the $T = 1250$ K isotherm in the undercooled states at different times after cooling (upper panels). Only atoms having a bcc or distorted bcc crystalline ordering are drawn as detected by the unsupervised topological learning approach (see text). From the snapshots, independent local atomic environment up to the second nearest-neighbors defined by the second minimum of the radial-distribution $g(r)$ (lower right panel), so-called local structures, are sampled by an the algorithm given in the lower left panel (After Ref. [BDMJ22b]).

features during nucleation in such huge simulations, without *a priori*, an unsupervised learning approach founded on topological descriptors loaned from persistent homology concepts was built, and will be described in the following.

4.1 Topological description of local structures

Topological data analysis (TDA) [CM21, Mot18] is a growing mathematical field with applications in a wide range of other field such as biology, computer science, physics, and materials science. Persistent homology is an effective and flexible tools to study the underlying topological shapes of a point cloud. In atomic-scale simulations considered here, the point cloud usually corresponds to an atom assembly in a simulation box. Let's consider a dataset as the point-cloud denoted $\mathcal{X}_0$ in a vector space and a parameter $r \geq 0$. The topological space $\mathcal{X}_r$ is defined by the union of all the balls (Euclidean or with another metric) of radius r, each of them being centered on a each point of the point-cloud. It consists in following the evolution of the topology of $\mathcal{X}_r$, or more precisely the persistence of its topological features as r grows from 0 about the initial set of points $\mathcal{X}_0$, to r big enough so that $\mathcal{X}_r$ has a the topology of a big ball containing all the points.

To encode or quantify the algebraic topology, a dedicated tool is homology [Hat02],

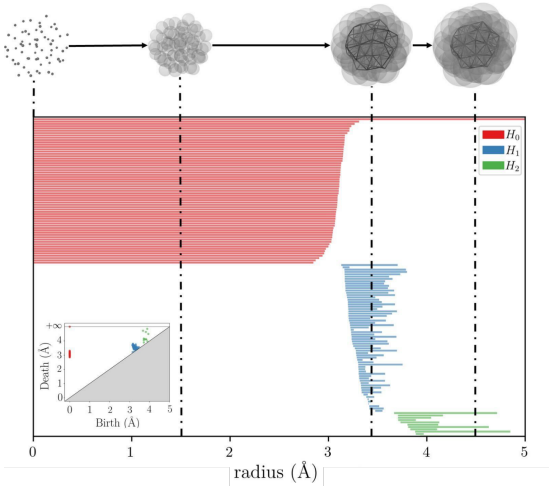

Figure 5: Persistence of a local structure consisting of a central atom and its two neighbourhood shells by means of a barecode description. The local structures with the ball growing as the radius increase are shown on the top. The barecode gives the lifespan of all topological features with colors corresponding to the homological dimension. In the bottom left corner there is the corresponding persistent diagram (with the same corresponding colors for the dimensions).

which derives vector spaces $H_n(\mathcal{X})$, $n \geq 0$, from a space $\mathcal{X}$, generated by n-dimensional topological features. 0-dimensional features correspond to connected components, 1-dimensional ones correspond to "holes" in the space, 2-dimensional to cavities and so on. Then persistent homology gives the persistence of these topological features, *i.e.* their lifespan when r is growing. The tracking of this persistence is commonly represented either through a barecode, with each bare corresponding to the lifespan of a specific topological feature, or a persistence diagram (PD) where each topological feature is associated to the point $(\text{birth}, \text{death})$ in the plane as can be seen in Figure 5).

While persistent homology in molecular dynamics simulations usually consists in determining the persistence diagram from all atoms of the box as the point cloud. The originality here to define from the PH a topological descriptor of the local environment of each atom as shown in Figure 4. In this case, the point cloud is the local environment and contains a bit less than one hundred atoms for the second neighbor shell. Finally a persistence diagram is obtained for each local structure as a description of its topology, and is associated to its central atom.

Among the various representations of these PDs as a vector, a classical method that has been successfully used to study 3D-shapes [COO15] is chosen here. Each coordinate of the topological vector is associated to a pair of points (x, y) in a persistence diagram

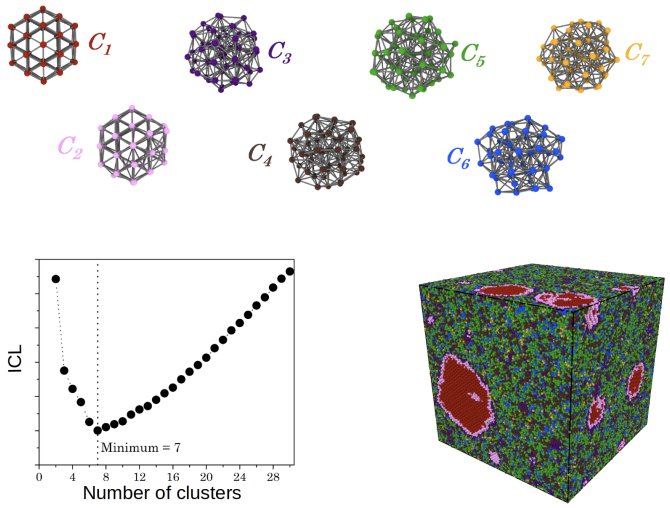

Figure 6: TDA-GMM clustering model of undercooled liquid Zirconium during nucleation with 7 clusters C_1 to C_7, which are closest to the centroids. Evolution of the Integrated Completed Likelihood (ICL) criterion (lower left panel) as a function of number of clusters predicting the optimal number of 7 clusters, trained on the configuration in lower right panel. Adapted from Ref. [BDMJ22a]

D for a fixed level of homology, except the infinite point, and is calculated by

$$m_D(x, y) = \min\{\|x - y\|_\infty, d_\Delta(x), d_\Delta(y)\}, \tag{13}$$

where $d_\Delta(\cdot)$ denotes the ℓ^∞ distance to the diagonal, and those coordinates are sorted by decreasing order. The resulting topological space is high-dimensional with often between 100 and 300 components.

4.2 Clustering local environments during nucleation

For the clustering, a model-based method is used, namely Gaussian Mixture Models (GMM) [HTFF09] and its estimation by an Expectation Maximization (EM) algorithm [DLR77] as mentioned in Section 3.1. The number of clusters is selected by Integrated Criterion Likelihood (ICL, [BCG00]), a refinement for clustering of Bayesian Integrated Likelihood (BIC, [Sch78]).

In a first step, a training set is built from a configuration during the nucleation, where the supercooled liquid coexists with crystalline nuclei, to capture all structural atomic events of interest. This is done by sampling a number of approximately 5000 independent atom centred local structures as described in Figure 4. Using PYTHON packages `gudhi` [MBGY14] and `ripser` [TSBO18], the PD of each individual local atomic

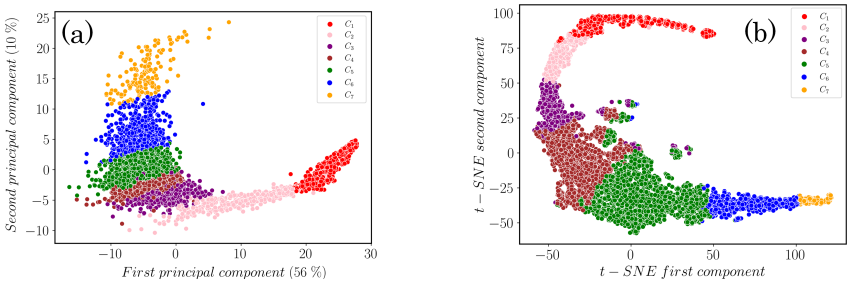

Figure 7: analysis of a configuration of undercooled Zr during nucleation with the PCA (left panel) and t-SNE (right panel) after each atom being labelled according to the TDA-GMM model with 7 clusters.

structures of the training set computed up to homological dimensions H_0 , H_1 , and H_2 and the topological feature is calculated using Eq. 13.

In a second step, the unsupervised learning with the Gaussian Mixture Model is performed in the topological space iteratively with cluster numbers 2 to 30. The ICL criterion is computed for each of them giving rise to the curve shown in Figure 6. The optimal number of clusters is chosen from the minimum of ICL, leading to a model with seven clusters numbered C_1 to C_7 whose representative local structure (closest to the centroids) are show in Figure 6. The inferred model from this method is called hereafter TDA-GMM.

Finally, with the learned TDA-GMM as such, each atom of the simulation box is assigned to C_1 to C_7 for any given configuration that enable one to identify and describe the structural properties of the system and morphological properties of the nuclei as can be seen on the specific configuration shown in Figure 6. Moreover, the crystal nucleation and its evolution as a function of time as can be seen strikingly in Figure 4 where only the crystalline local structures (C_1 and C_2) are drawn. All the results and findings can be found in Refs. [BDMJ22b, BDMJ22a], and probably the main physical outcome in these works is that the liquid in the undercooled state appears very heterogeneous and the nucleation is triggered from fluctuations with lowest icosahedral ordering, always present in the liquid at various degrees, the latter being known as incompatible with the long-range ordering of the crystalline state [Fra52, Tur52].

To close this Section, a question arise from the relevance of the TDA-GMM unsupervised modeling and especially the chosen number of clusters (seven for zirconium) in such a high-dimensional topological descriptor space with roughly $D \approx 200$. This can be visualized by performing a dimensionality reduction with the PCA or the t-SNE. Figure 7 displays a two-dimensional analysis for the two principal components of the topological vectors for all atoms in the simulation box shown in Figure 6. They are colored with the cluster they belong to. Interestingly, both methods lead to a similar

representation, in which the clusterized local structures vary as a quasi continuum but do not overlap mainly (except very partially for C_4 and C_5) in the t-SNE represenation). The PCA and tSNE distinguishe the liquid on one component and the crystal on the second one. Clusters C_3, C_4, and C_5 that are known to retain partially liquid and crystalline orderings are located as expected at the crossover of the two components.

5 Conclusion

This Chapter was devoted to unsupervised learning approaches that emerged as a powerful approach for uncovering hidden patterns, relationships, and structures in data without the need for labeled examples. The key concepts and techniques in unsupervised learning were discussed, focusing only on the basic approach clustering and dimensionality reduction, letting aside association rule mining and unsupervised deep learning methods, which could be treated as topics in themselves. The power of clustering was highlighted through a application in materials science that helped us to monitor efficiently the structural evolution during crystal nucleation of a liquid metal, and uncover hidden correlations without *a priori* in this process from the huge amount of data of these large-scale molecular dynamics simulations with millions of atoms. Such an unsupervised approach is deemed to be sufficiently general to be transposed to discrete particle dynamics at a larger scale in geophysics.

Acknowledgement

We acknowledge the CINES and IDRIS under Project No. INP2227/72914, as well as CIMENT/GRICAD for computational resources. This work was performed within the framework of the Centre of Excellence of Multifunctional Architectured Materials CEMAM-ANR-10-LABX-44-01 funded by the "Investments for the Future" Program. This work has been partially supported by MIAI@Grenoble Alpes (ANR-19-P3IA-0003). Discussions within the French collaborative network in high-temperature thermodynamics GDR CNRS3584 (TherMatHT) and in artificial intelligence in materials science GDR CNRS 2123 (IAMAT) are also acknowledged.

References

[Aka73] Hirotogu Akaike. *Information Theory and an Extension of the Maximum Likelihood Principle*, pages 199–213. Springer New York, New York, NY, 1973.

[AT17] Michael P Allen and Dominic J Tildesley. *Computer simulation of liquids*. Oxford university press, 2017.

[BCG00] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725, 2000.

[BDMJ20] S. Becker, E. Devijver, R. Molinier, and N. Jakse. Glass-forming ability of elemental zirconium. *Physical Review B*, 102, 2020.

[BDMJ22a] Sébastien Becker, Emilie Devijver, Rémi Molinier, and Noël Jakse. Physical review e 105 , 045304 (2022) unsupervised topological learning for identification of atomic structures. 045304:1–10, 2022.

[BDMJ22b] Sébastien Becker, Emilie Devijver, Rémi Molinier, and Noël Jakse. Unsupervised topological learning approach of crystal nucleation. *Scientific Reports*, 12:1–9, 2022.

[Car20] Gunnar Carlsson. Topological methods for data modelling. *Nature Reviews Physics*, 2:697–708, 2020.

[Cer19] Michele Ceriotti. Unsupervised machine learning in atomistic simulations, between predictions and understanding. *The Journal of chemical physics*, 150(15):150901, 2019.

[CM21] Frédèric Chazal and Bertrand Michel. An introduction to topological data analysis: Fundamental and practical aspects for data scientists. *Frontiers in Artificial Intelligence*, 4, 2021.

[COO15] Mathieu Carrière, Steve Y. Oudot, and Maks Ovsjanikov. Stable topological signatures for points on 3d shapes. *Eurographics Symposium on Geometry Processing*, 34:1–12, 2015.

[DLR77] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[EH21] Bradley Efron and Trevor Hastie. *Computer age statistical inference, student edition: algorithms, evidence, and data science*, volume 6. Cambridge University Press, 2021.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of 2nd International Conference on Knowledge Discovery and*, pages 226–231, 1996.

[Fra52] Frederick Charles Frank. Supercooling of liquids. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 215(1120):43–46, 1952.

[Hat02] A. Hatcher. *Algebraic Topology*. Algebraic Topology. Cambridge University Press, 2002.

[HR02] Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15, 2002.

[HT20] Tony Hey and Anne Trefethen. The fourth paradigm 10 years on. *Informatik Spektrum*, 42:441–447, 2020.

[HTFF09] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[HTTG09] Tony Hey, Stewart Tansley, Kristin Tolle, and Jim Gray. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, October 2009.

[HWWT13] Yongzhen Huang, Zifeng Wu, Liang Wang, and Tieniu Tan. Feature coding in image classification: A comprehensive study. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):493–506, 2013.

[KA21] K. Karapiperis and J.E. Andrade. Nonlocality in granular complex networks: Linking topology, kinematics and forces. *Extreme Mechanics Letters*, 42:101041, 2021.

[KKN14] Samina Khalid, Tehmina Khalil, and Shamila Nasreen. A survey of feature selection and feature extraction techniques in machine learning. In *2014 science and information conference*, pages 372–378. IEEE, 2014.

[MBGY14] Clément Maria, Jean-Daniel Boissonnat, Marc Glisse, and Mariette Yvinec. The gudhi library: Simplicial complexes and persistent homology. In *The Gudhi Library: Simplicial Complexes and Persistent Homology*, 06 2014.

[Mot18] Francis C Motta. Topological data analysis: Developments and applications. *Advances in Nonlinear Geosciences*, pages 369–391, 2018.

[NQY18] Joel Nothman, Hanmin Qin, and Roman Yurchak. Stop word lists in free open-source software packages. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 7–12, 2018.

[PSG+18] Mariam Pirashvili, Lee Steinberg, Francisco Belchi Guillamon, Mahesan Niranjan, Jeremy G. Frey, and Jacek Brodzki. Improved understanding of aqueous solubility modeling through topological data analysis. *Journal of Cheminformatics*, 10:1–14, 2018.

[SCC+16] Gabriele C. Sosso, Ji Chen, Stephen J. Cox, Martin Fitzner, Philipp Pedevilla, Andrea Zen, and Angelos Michaelides. Crystal nucleation in liquids: Open questions and future challenges in molecular dynamics simulations. *Chemical Reviews*, 116:7078–7116, 2016.

[Sch78] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, March 1978.

[SMBM19] Jonathan Schmidt, Mário RG Marques, Silvana Botti, and Miguel AL Marques. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, 5(1):83, 2019.

[TSBO18] Christopher Tralie, Nathaniel Saul, and Rann Bar-On. Ripser.py: A lean persistent homology library for python. *Journal of Open Source Software*, 3(29):925, 2018.

[Tur52] David Turnbull. Kinetics of solidification of supercooled liquid mercury droplets. *The Journal of chemical physics*, 20(3):411–424, 1952.

[VdMH08] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[WT10] David M. Walker and Antoinette Tordesillas. Topological evolution in dense granular materials: A complex networks perspective. *International Journal of Solids and Structures*, 47(5):624–639, 2010.