Article

Challenges and Opportunities in Machine Learning for Geometry

Rafael Magdalena-Benedicto [1,*,†], Sonia Pérez-Díaz [2,†] and Adrià Costa-Roig [3,†]

1. Department of Electronic Engineering, University of Valencia, 46010 Valencia, Spain
2. University of Alcalá, Department of Physics and Mathematics, 28871 Alcalá de Henares, Spain; sonia.perez@uah.es
3. Department of Pediatric Surgery, La Fe University and Polytechnic Hospital, 46026 Valencia, Spain; adriacostaroig@gmail.com
* Correspondence: rafael.magdalena@uv.es
† These authors contributed equally to this work.

Abstract: Over the past few decades, the mathematical community has accumulated a significant amount of pure mathematical data, which has been analyzed through supervised, semi-supervised, and unsupervised machine learning techniques with remarkable results, e.g., artificial neural networks, support vector machines, and principal component analysis. Therefore, we consider as disruptive the use of machine learning algorithms to study mathematical structures, enabling the formulation of conjectures via numerical algorithms. In this paper, we review the latest applications of machine learning in the field of geometry. Artificial intelligence can help in mathematical problem solving, and we predict a blossoming of machine learning applications during the next years in the field of geometry. As a contribution, we propose a new method for extracting geometric information from the point cloud and reconstruct a 2D or a 3D model, based on the novel concept of *generalized asymptotes*.

Keywords: algebraic geometry; machine learning; generalized asymptotes

MSC: 14Q20; 14Q05; 68T01

1. Introduction

The use of machine learning (ML) is gaining popularity in the scientific community, especially in domains such as data analysis, optimization, and statistics. ML algorithms are used to detect patterns in data and can be leveraged to solve a broad range of mathematical problems. Mathematics has benefited significantly from ML, particularly in the realm of data analysis. Through ML algorithms, mathematicians can analyze vast datasets and uncover underlying relationships and patterns that may be elusive using conventional statistical techniques. The discovery of these patterns and relationships have provided novel insights in various fields such as engineering, biology, and finance [1].

Machine learning also plays an essential role in optimization, which refers to finding the most optimal solution for a given problem. Machine learning algorithms are particularly useful in searching for the best solution in high-dimensional spaces. Furthermore, through the application of machine learning techniques to analyze vast datasets and uncover patterns, mathematicians can develop new mathematical models that are better equipped to address the complexity of real-world systems. Machine learning can help in the creation of novel mathematical models and algorithms.

The structure of the paper is as follows (see Figure 1): In Section 2, we discuss the outbreak of machine learning in certain fields closely related to mathematics and geometry, and the unstoppable growth of machine learning in the coming years in this field of knowledge, reasoning the causes for this trend.
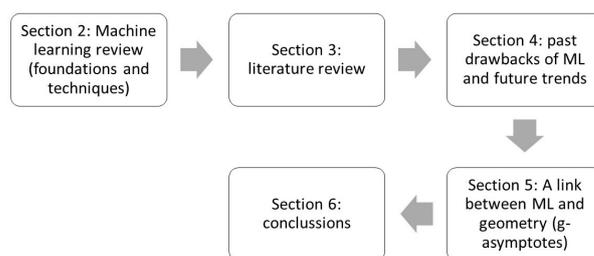
Figure 1. Structure of the paper, with the target for every section.

In the following section (Section 3), we review the latest publications and tendencies in which machine learning is being applied to various fields of mathematics and geometry. We briefly summarize the algorithms used and the strategies with which the problem is approached, from the data science point of view, for its resolution, as well as the knowledge provided by this perspective.

In Section 4, we highlight some of the problems, both formal and technical, that our discipline may face in this advent of machine learning.

In Section 5, we present a method using asymptotes for point clouds reconstruction that involves fitting a set of asymptotes to the point clouds. The asymptotes can be defined from the infinity branches that can be constructed from the point clouds. Once the asymptotes have been fit to the point clouds, they can be used to reconstruct a 2D or 3D model by interpolating between the points and generating a curve or a surface that follows the asymptotes. The novelty is that one may use the asymptotes that are not necessarily lines but *g-asymptotes*. At this point, we underline the the great advantage provided by the use of g-asymptotes and the problem it avoids in the reconstruction of 2D or 3D models. When the given point clouds include points at the "infinity" (i.e., points having large coordinates), the construction of effective method needs some other approach, since otherwise the prediction of the geometry of the 2D or 3D model could not be the expected one. For instance, in Section 5 introduction, point clouds at $[-5, 5] \times [-5, 5]$ and at $[-15, 15] \times [-15, 15]$ are provided. The behavior of the curve one is looking for modeling the point clouds is totally different if one looks at the squares with smaller coordinate points, than if one "goes to the infinity" where the distortion of the model seems to indicate that we have different curves (compare figures in Section 5 introduction, where point clouds at $[-100, 100] \times [-100, 100]$ and at $[-1000, 1000] \times [-1000, 1000]$ are provided). More precisely, in the square $[-N, N] \times [-N, N]$ with N small enough, some well-known machine learning techniques allow us to accurately determine the model. However, to correctly predict the geometric object, we need to model the infinity accurately, and the essential tools from which we can extract geometric information from the point cloud and reconstruct a 2D or a 3D model are the *g-asymptotes*, which generalize the classical notion of asymptote. In figures of Example 2, one may find that although the approximation in the squares of smaller lengths is also good, the error in this area is much greater than in the infinity where the asymptotes describe the point clouds perfectly. Therefore, the use of g-asymptotes seems to be a successful and novel technique to deal with this problem.

We finish the paper with a section of conclusions and future work (see Section 6).

2. Data Analysis from a Mathematical Point of View

Geometry is a branch of mathematics that focuses on the study of shapes, sizes, positions, and dimensions of objects in space. The concepts of geometry have been studied for thousands of years, and its principles and formulas are still used today in various applications. Through the study of geometry, we can gain a deeper understanding of the world around us and develop problem-solving skills that can be applied in numerous contexts. Algebraic geometry is a branch of mathematics that combines algebra and geometry to study the solutions of polynomial equations. It deals with geometric objects that are defined by algebraic equations and seeks to understand their properties and

structures. In algebraic geometry, the focus is on studying the geometric shapes that are solutions to polynomial equations rather than the specific numerical values of those solutions. The field has numerous applications, including in physics, computer science, and cryptography. The study of algebraic geometry requires a solid understanding of abstract algebra, topology, and complex analysis. With its focus on the relationship between algebra and geometry, algebraic geometry has been instrumental in advancing the fields of modern mathematics and theoretical physics [2,3].

Without getting into debates about the philosophy of mathematics, it is generally agreed that mathematics is not an exception to the scientific method: the appearance of certain data, whether from physical reality, mathematical reality, or the plane of ideas, leads to a number of conjectures, which are then proved or disproved through further analysis. Up until the introduction of the first computers, the entire procedure was traditionally completed mentally.

An interesting approach can be found in [4], from the mathematics mechanization point of view. This paper reviews the state-of-the-art on developing symbolic algorithms from manipulating mathematical objects, aided by computers or artificial intelligence. These methods enable automated proving or discovering geometry theorems. Moreover, the methods always work in the symbolic realm, leaving apart the analysis of numerical datasets coming from geometry-related problems. Automated Deduction in Geometry conferences [5] are a valuable resource of work for this approach. Nevertheless, as the latest advances in machine learning and artificial intelligence (AI) bring us closer to achieving universal AI, this will mark a significant leap in knowledge discovery in mathematics.

Given that mathematics is the language through which nature is expressed [6], it is not unexpected that there is a significant crossover between mathematics and physics. The works of Kepler, Newton, Fermat, Gauss, and even more recently, Einstein, come to mind. In each of them, we may see a collection of tables or numerical records from which we must deduce the mathematical expressions that support, clarify, or model the facts we are dealing with. All of this was performed manually before the invention of the computer.

However, computers are also able to provide huge datasets representing a theoretical or practical problem. This is the crucial aspect of the situation: [7,8] are just two examples of enormous data sets that frequently emerge in mathematics as a result of numerical simulations. Any numerical mathematical simulation generates a lot of data that, unlike real process data, are accurate: they include no noise at all. An example of this may be the creation of numerically rational curves modeling a specific problem [9]. These days, it is usual to use computers for these tasks due to the cheapness of storage and the current computing capability of CPUs and GPUs.

Significant contributions to these developments have also came from the study of theoretical physics and cosmology. On the one hand, some of their problems are conceptually and formally treated from a purely mathematical perspective; on the other hand, numerical simulations have become crucial to scientific advancement almost since the arrival of the computer.

Symmetry is a critical concept in all physical theories, and it is described using mathematical groups. Understanding a theory's symmetry greatly simplifies calculations and aids in developing intuition [10–12]. The theory of relativity was a groundbreaking paradigm shift in the way space and time were perceived. Relativity frames were based on speeds, and a maximum speed of light was established as an unbreakable limit. As the theory evolved into general relativity, incorporating accelerating objects and massive bodies, it became clear that gravity was based on the geometry of spacetime [13–15].

Other hot topic in theoretical physics is the superfluid vacuum theory (SVT). The vacuum, which appears to be empty, actually includes a superfluid that permeates the entire cosmos. It is hypothesized that this superfluid, which has peculiar characteristics such as zero viscosity and limitless compressibility, is the cause of a variety of phenomena, including the presence of mass and the functioning of gravity. The scientific community

still views SVT as a novel and contentious concept, and much research and discussions are being conducted to examine its potential applications and veracity.

Fundamental particle objects are changed to fields in specific representations of the Lorentz group, resulting in quantum field theory, which incorporates the key concepts of special relativity into quantum theory [16–20].

Quantum computing is an advancement in physics and mathematics. In order to tackle complicated problems at a previously unheard-of speed, quantum computing is a breakthrough area that merges the ideas of quantum physics and computer technology. It has the potential to transform many industries, including artificial intelligence, medicine development, and cryptography. The development and use of quantum computing heavily rely on mathematics. Mathematics is at the core of quantum computing, from creating quantum algorithms to determining the mathematical foundation to characterize quantum systems. By providing the required mathematical tools and insights, mathematicians have considerably aided the development of quantum computing. We can examine the connection between mathematics and quantum computing in [21–24] and highlight some important mathematical concepts used in quantum computing.

We concluded that the vast majority of scientific disciplines heavily rely on data and mathematics. It is inevitable that one of the most well-known fields of our era—machine learning—will emerge with massive mathematical data sets. We can take into account a variety of choices when we have enormous data sets, the output of simulations, or the outcome of assigning numerical values to a problem. Every research has a target, whether it be to investigate particular high-level concepts or to respond to particular questions. Exploratory questions may aim to find unusual data trends or locate unusual records. Confirmatory questions, on the other hand, are more focused and involve tasks such as identifying group differences or monitoring attribute changes over time.

Intelligent data analysis finds its roots in various disciplines, but statistics and machine learning are arguably the most significant [1,25]. Although statistics is the older of the two, the emergence of machine learning has added a distinct culture, interests, emphases, aims, and objectives that diverge from those of statistics. This divergence has created a creative tension between the two disciplines at the core of intelligent data analysis, leading to the development of innovative data analytic tools. Despite their differences, both statistics and machine learning contribute to the advancement of intelligent data analysis.

There are different types of machine learning algorithms, such as supervised learning, unsupervised learning, and reinforcement learning, and each is used in different contexts and for different purposes [1]. Supervised learning is a type of machine learning in which a labeled data set is used to train a machine learning algorithm. The labeled data are those that have already been "tagged" with the correct answers, and are used to train the algorithm to make predictions or decisions. Supervised learning is useful when we have labeled data available and want to predict a specific outcome or make a decision based on that data. In a way, we are incorporating prior human knowledge and experience of the problem into the solution of the problem. The algorithm has all that knowledge as a starting point, and it evolves and learns from the knowledge that we have provided it. Examples of supervised algorithms could be linear regression, logistic regression, neural networks, or support vector machines (SVM).

Unsupervised learning is a type of machine learning in which you do not provide labeled data to the machine learning algorithm. Instead, the algorithm is trusted to discover patterns and relationships in the data on its own. Unsupervised learning is useful when we do not have a labeled data set available and want to discover patterns and trends in the data. However, unsupervised learning does not allow us to make specific predictions or decisions based on the data, as it does not provide the correct answers to train the algorithm. Unsupervised algorithms are clustering algorithms and dimensionality reduction algorithms.

In semi-supervised learning, the data set contains both labeled and unlabeled data. Typically, the amount of unlabeled data is much larger than the number of labeled examples.

The goal of a semi-supervised learning algorithm is the same as that of a supervised algorithm. The idea is that using the unlabeled data in addition to the labeled data allows the algorithm to find a better model.

We may come across reinforced learning algorithms on occasion. In these circumstances, the algorithm is immersed in the problem and may observe the problem's or the environment's state, which is converted into a feature vector. The algorithm can then operate on each state and change it. The algorithm is rewarded (or punished) based on the outcome of the action, whether it changes to a better or desired state or not. The algorithm's goal is to generate an action strategy that solves the problem.

3. Machine Learning Techniques for Geometry

Machine learning and geometry are two fields that have become increasingly interconnected in recent years. Geometry provides a powerful toolset for understanding and analyzing the structure of data, while machine learning algorithms offer a framework for processing and making predictions based on that data. As a result, machine learning has found many applications in geometry, and geometric methods have become increasingly important in machine learning [8,26,27].

Geometry has been employed in machine learning in a variety of ways, including the development of novel algorithms [7,28]. Geometric algorithms, for example, have been utilized to build clustering approaches that group comparable data points together based on their geometric qualities. Geometric algorithms have also been used to create classification algorithms, which use geometric features to assign new data points to one of several pre-defined categories.

Another important application of geometry in machine learning is in the analysis of high-dimensional data [29–32]. High-dimensional data are common in many machine learning applications, but it can be difficult to understand and analyze using traditional statistical methods. Geometric methods provide a way to represent and analyze high-dimensional data in a way that is more intuitive and interpretable. Geometric deep learning is another area of research where machine learning and geometry intersect [33–35]. In geometric deep learning, the goal is to develop deep learning algorithms that can operate directly on geometric structures such as graphs, point clouds, and meshes. This method has demonstrated potential in a number of applications, including 3D object identification and drug development.

Finally, machine learning has also been used to advance the field of geometry itself. Machine learning algorithms have been used to automate the generation of geometric models, to predict geometric properties of materials, and to develop new geometric optimization algorithms [7].

One example of a supervised learning algorithm used in geometry is the support vector machine (SVM) (Figure 2). SVMs are a popular tool for classification and regression tasks. They work by finding the hyperplane that maximally separates two classes in a high-dimensional space (see Figure 3) [1]. An example of the application of SVMs can be found in [36]. In this paper, the authors use the database of weighted-P4s which admit Calabi–Yau 3-fold. This is a classic problem in string theory, closely related to algebraic geometry, that has been faced with machine learning tools, mainly due to the existence of numerical datasets [37,38]. Unsupervised techniques identified an unanticipated almost linear dependence of the topological data on the weights. This then allowed them to identify a previously unnoticed clustering in the Calabi–Yau data. Supervised techniques were successful in predicting the topological parameters of the hypersurface from its weights with an accuracy of R2 > 95%. Supervised learning also allowed them to identify weighted-P4s, which admit Calabi–Yau hypersurfaces to 100% accuracy by making use of partitioning supported by the clustering behavior.
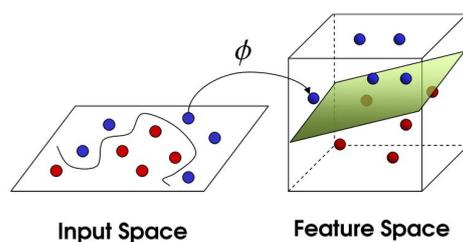
Figure 2. Kernel-SVM for separating classes using higher dimensional spaces. Source: Shehzadex (https://commons.wikimedia.org/wiki/File:Kernel_yontemi_ile_veriyi_daha_fazla_dimensiyonlu_uzaya_tasima_islemi.png, accessed on 1 June 2023), https://creativecommons.org/licenses/by-sa/4.0/legalcode, accessed on 1 June 2023.
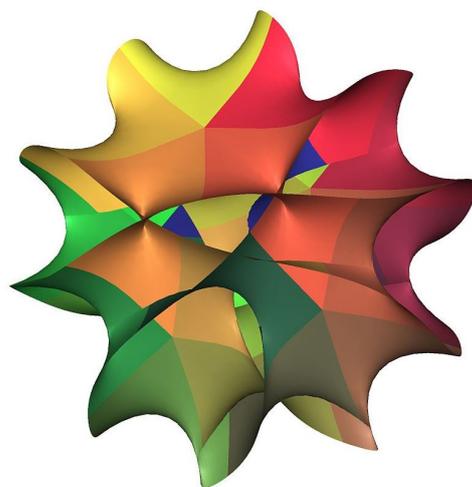
Figure 3. The Calabi–Yau quintic (local 2D cross-section of the real 6D manifold). ML techniques have been used for discover hidden clusters in data in [36]. Source: Andrew J. Hanson, Indiana University. (https://commons.wikimedia.org/wiki/File:CalabiYau5.jpg, accessed on 1 June 2023), "CalabiYau5", https://creativecommons.org/licenses/by-sa/3.0/legalcode, accessed on 1 June 2023.

Another example of a supervised learning algorithm used in geometry is the convolutional neural network (CNN). CNNs are a type of deep learning algorithm that use layers of convolutional filters to extract features from images or other spatial data [39]. In geometry, CNNs can be used to segment images of geometric shapes or recognize patterns in point clouds [40]. For example, a CNN could be trained to identify the boundaries between different regions of a 3D surface [41–44]. It has been also applied in the Amoebae problem in [45]. Amoebae, introduced in [46], are regions in $\mathbb{R}_n$ with several holes and straight narrowing tentacles reaching to the infinity, constructed from polynomials in n complex variables. Amoebae from tropical geometry and the Mahler measure from number theory play important roles in quiver gauge theories and dimer models.

Regression tasks in geometry are also an example of supervised learning algorithms. In fact, the aforementioned Amoebae problem has also been tackled with regression in [47]. Their dependencies on the coefficients of the Newton polynomial closely resemble each other, and they are connected via the Ronkin function. Genetic symbolic regression methods are employed to extract the numerical relationships between the 2D and 3D amoebae components and the Mahler measure.

Another common unsupervised learning problem in geometry is clustering, where the target is to group data points together based on some measure of similarity. This can be useful for tasks such as image segmentation or identifying patterns in complex data sets. Another popular algorithm is principal component analysis (PCA), which finds the directions of greatest variance in the data and projects the data onto those directions, effectively reducing the dimensionality of the data. Berman, in [36], provides some analysis of the

fundamentals of the dataset using PCA, topological data analysis (TDA), and other unsupervised machine learning methods, just as a previous stage before applying supervised machine learning methods.

Unsupervised learning has demonstrated success in generative models, which aim to generate new data that are similar to the training data. According to [48], one prominent strategy is generative adversarial networks (GANs), in which two neural networks are trained concurrently: one generates new data, and the other attempts to distinguish between the produced data and the real data. This results in a feedback loop in which the generator learns to generate increasingly realistic data and the discriminator improves its ability to discern between actual and fake data.

In geometry, GANs have been used to generate 3D shapes and textures, as well as to interpolate between different shapes. For example, GANs have been used to generate realistic 3D models of chairs, cars, and other objects, which can be useful in fields such as architecture and product design. GANs have also been used to interpolate between different shapes, allowing for the creation of new shapes that are similar to existing ones but with variations that may not have been manually designed. More complex approaches, related to geometry, can be found in [49,50].

Reinforcement learning has also been applied in geometry, i.e., in the optimization of geometric shapes. Given a set of points, an algorithm can use reinforcement learning to find the shape that maximizes a certain criterion, such as the area or perimeter. The algorithm would start with an initial guess of the shape, and then iteratively modify it based on the feedback received from the environment. The feedback could be the value of the criterion, or a measure of the distance between the shape and a target shape. Another application of reinforcement learning in geometry is the discovery of new mathematical structures. An algorithm could learn to generate graphs that satisfy certain properties, such as being planar or having a certain degree distribution. The algorithm would start with a random graph and then iteratively modify it based on the feedback received from the environment. The feedback could be the value of a metric that measures how well the graph satisfies the desired properties. Some applied examples can be found in [51,52].

Neural networks and deep learning have changed the field of machine learning, and their impact on geometry has been significant. A neural network (ANN) is a computational model that is designed to simulate the way the human brain works. It is composed of interconnected nodes or neurons, each of which is assigned a weight and a bias value [1]. These neurons receive inputs from other neurons and perform a computation before passing their output to other neurons. By adjusting the weights and biases of these neurons, the neural network can be trained to recognize patterns in data. One application can be seen in [53], where they apply machine learning algorithms to the study of lattice polytopes. With ANN, they are able to predict standard properties, such as volume, dual volume, reflexivity, etc, with accuracies up to 100%. The paper applies to 2D polygons and 3D polytopes with Plücker coordinates as input, which out-perform the usual vertex representation. Same author also applies ANN to amoebae in [45], applying multilayer perceptron as well as CNN.

Deep learning takes this concept a step further by using neural networks with many layers [25]. Each layer in a deep neural network performs a different computation on the input data, with the output of one layer serving as the input to the next. This allows the network to learn more complex features and patterns in the data, and can result in more accurate predictions. One of the key benefits of deep learning in geometry is its ability to learn from large amounts of data. This is especially useful in situations where traditional geometric algorithms may be too computationally expensive or too complex to implement. Deep learning algorithms can be trained on large datasets of images or geometric models, allowing them to learn from a vast amount of information and make accurate predictions [8,33,35]. Another advantage is its ability to generalize to new, unseen data. Once a neural network has been trained on a particular dataset, it can be applied to

new data with similar properties. This has many applications in fields such as computer graphics, computer vision, and robotics [8,54,55] (see Figure 4).
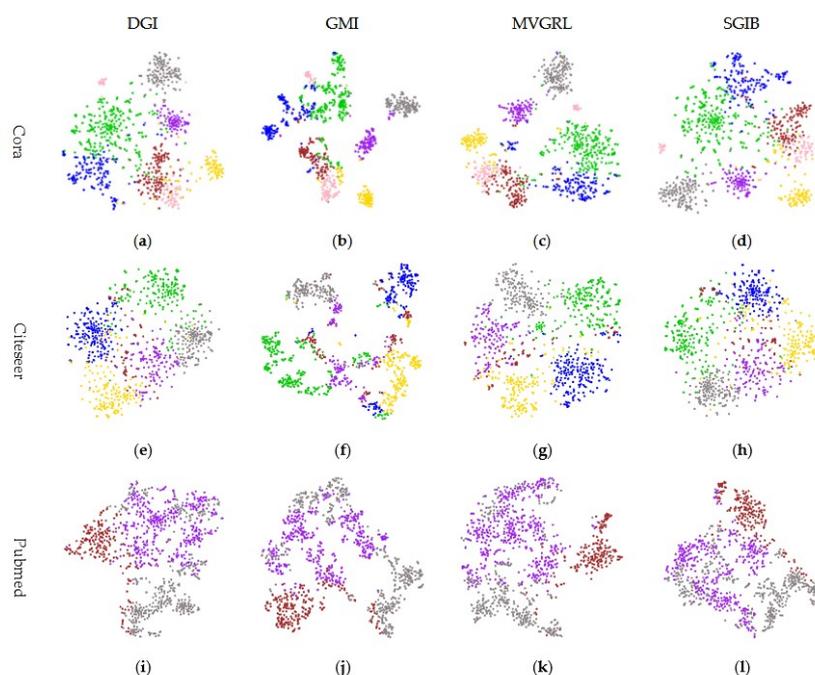
Figure 4. Embeddings over datasets, after applying clustering in [55], using graph neural networks. (**a**) DGI for Cora; (**b**) GMI for Cora; (**c**) MVGRL for Cora; (**d**) SGIB for Cora; (**e**) DGI for Citeseer; (**f**) GMI for Citeseer; (**g**) MVGRL for Citeseer; (**h**) SGIB for Citeseer; (**i**) DGI for Pubmed; (**j**) GMI for Pubmed; (**k**) MVGRL for Pubmed; (**l**) SGIB for Pubmed. Source: Reprinted from Ref. [55].

Table 1 summarizes most popular methods in machine learning, ordered by the function performed (clustering, regression, classification or dimensionality reduction). Figure 5 offers a short guide to choose the most suited algorithm.

Table 1. Popular machine learning algorithms organized by task.

| Supervised (Classification)	Supervised (Regression)	Unsupervised (Clustering)	Unsupervised (Dimensionality Reduction)
Kernel SVM & Linear SVM	Random forest	dbscan	PCA
ANN	ANN	k-means	Singular Value Decomposition
CNN	Linear regression	K-modes	Latent Dirichlet Analysis
Logistic Regression	Gradient boosting tree	hierarchical	
Decision tree & gradient boosting tree	Decision tree	Gaussian Mixture mode	
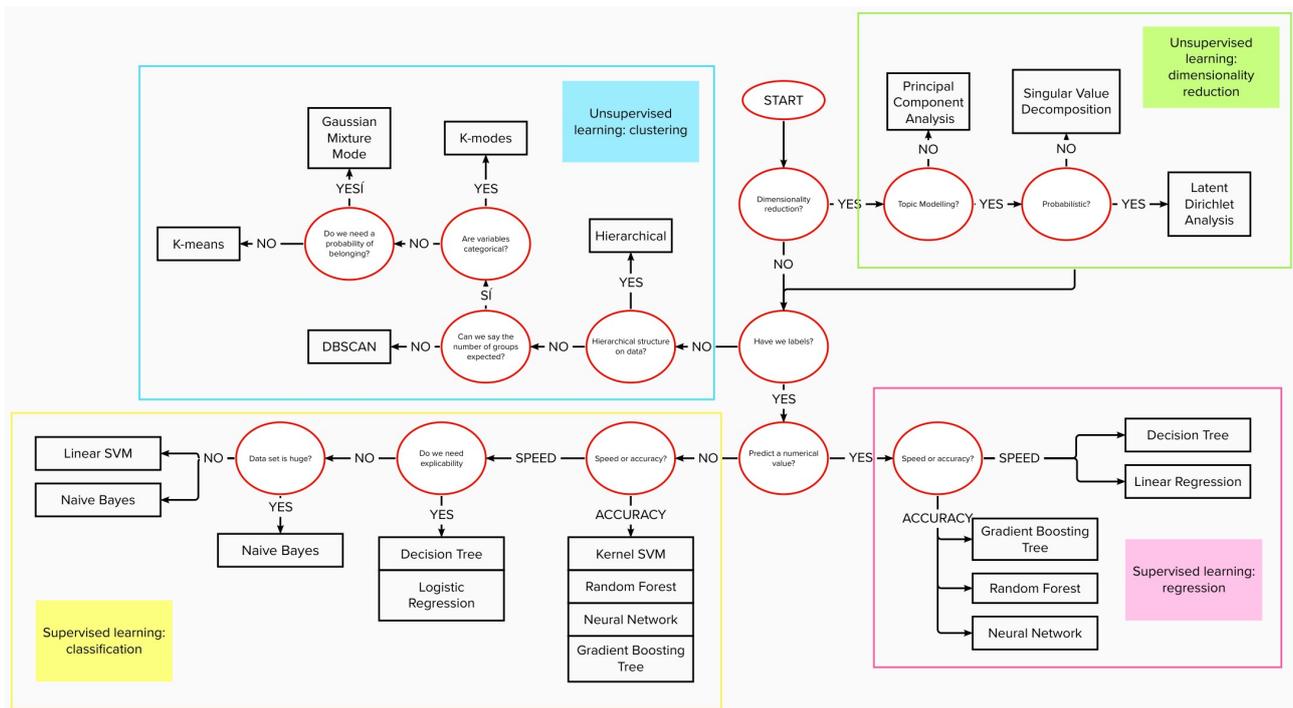
Figure 5. Cheat-code guide for choosing ML algorithms.

The fields of mathematics and geometry are poised for a significant revolution in the form of machine learning, which will impact virtually every area of study. It is essential for mathematicians to remain at the forefront of these inevitable advancements to guide their development.

4. Challenges in Machine Learning for Geometry

Math and geometry are no exception to how machine learning approaches have altered how we process and interpret data. We consider that machine learning in geometry has a number of opportunities and challenges. Our survey shows that this method has potential to be both helpful and disruptive in the coming years. However, there are some historical ML flaws that should be carefully considered when used in mathematics and geometry. Despite these challenges, here are also several opportunities in applying machine learning to geometry. For example, machine learning can be used to extract meaningful features from high-dimensional geometry data, such as point clouds, meshes, and curves. These features can then be used for tasks such as classification, segmentation, and reconstruction.

One of the most significant challenges in machine learning for geometry is overfitting and underfitting. Overfitting occurs when a model is too complex and tries to fit the noise in the data, resulting in poor generalization performance. On the other hand, underfitting happens when a model is too simple and fails to capture the underlying patterns in the data, leading to poor performance on both the training and test data. Defeating overfitting and underfitting in geometry requires careful attention to the choice of model and the amount of data used for training. To prevent overfitting or underfitting, it is crucial to find a balance between model complexity and the volume of training data. This is particularly difficult in geometry because the data's dimensionality might be very great. Additionally, the theoretical basis and prior understanding of the issue should be considered. This additional information can be used to improve the models or possibly create some new ones based on a thorough theoretical understanding of the issue.

Another drawback in machine learning for geometry is the lack of labeled data. Unlike in other fields, such as computer vision or natural language processing, where large labeled datasets are available, geometry often requires manual labeling, which is time-consuming and expensive. Nevertheless, symbolic and numerical computer software is able to generate

datasets that represent a mathematical problem. Machine learning applied over this dataset can offer mathematics solutions regarding classification, prediction of modeling, three of the key features of ML methods.

Alternatively, unsupervised learning techniques can be used to learn from unlabeled data. Unsupervised learning can be used to discover meaningful structure and patterns in the data without the need for explicit labels. For example, unsupervised learning can be used to learn a low-dimensional representation of high-dimensional geometry data, such as point clouds or meshes. This low-dimensional representation can then be used for downstream tasks such as classification, segmentation, and reconstruction.

One more way to address the limited availability of data is to develop techniques that can learn from few or zero-shot examples. Few-shot learning aims to learn from a small number of examples, while zero-shot learning aims to learn from a set of examples without any direct training. These techniques are particularly useful in geometry, where it is often challenging to obtain large labeled datasets. Few-shot and zero-shot learning can enable machines to recognize new shapes and structures with minimal training data, making them valuable tools in geometry processing and modeling.

Machine learning can also be used to enhance the accuracy and efficiency of traditional geometry processing techniques, such as surface fitting, shape optimization, and geometric modeling. For example, machine learning can be used to predict the behavior of complex geometric structures, such as composite materials, and to optimize their design.

Because machine learning for geometry frequently uses sophisticated mathematical operations and transformations that are difficult to understand or explain, interpretability is a challenge when using machine learning methods. In order to overcome this difficulty, researchers are creating methods for interpreting and explaining the decisions made by machine learning algorithms. To increase understanding of the models' decision-making process, one option is to use visualization techniques. Another strategy is to create models with built-in interpretability. For instance, because they are simple to comprehend and explain, decision trees and rule-based models are frequently utilized in situations where interpretability is important. Explainability can support or extend previous knowledge, and also gain insight on mathematical problems. Previous knowledge of the problem or similar problems, and the theoretical existing corpus, can help on decide how to choose between the different solutions machine learning gives for a problem. In mathematics, theoretical knowledge is always a key factor.

One of the opportunities in applying machine learning to geometry is the potential for collaboration and interdisciplinary research. Although several fields have a strong relationship with geometry, namely theoretical physics, other fields lying in the results of mathematical processes, more than theoretical developments, can be largely benefited. Machine learning techniques can be used in conjunction with traditional geometry processing techniques, such as surface reconstruction, shape optimization, and geometric modeling. This collaboration can enable researchers to develop more efficient and accurate techniques for solving challenging geometric problems.

Another chance for collaboration is the development of shared benchmarks and datasets. Researchers may compare and assess the performance of various algorithms and models by working together on benchmarks and datasets, allowing the discipline to advance more swiftly and effectively. Mathematician can generate huge synthetic datasets, and machine learning can look for patterns, anomalies, or make predictions of the spatial or time evolution. This collaboration can bring insight for mathematical open problems, besides developing new tailored machine learning algorithms starting from the deep mathematical expertise in the problem.

5. A New Practical Application in Algebraic Geometry

As we stated above, machine learning can be used for point cloud reconstruction, which is the process of creating a 2D or a 3D model from a set of 2D or 3D points captured by

a scanner or generated by some other means. Point cloud reconstruction is an important step in a wide range of applications such as printing, virtual reality, and autonomous driving.

There are several machine learning techniques that can be used for point cloud reconstruction. One popular approach is to use deep learning models such as convolutional neural networks (CNNs) or graph neural networks (GNNs) to learn a mapping from the input point cloud to the output model. These models can be trained on a large dataset of point clouds and corresponding models, and can learn to generalize to new, unseen data.

Another approach is to use traditional machine learning algorithms such as *k*-nearest neighbors or random forests to predict the geometry of the model from the input point cloud. These methods can be effective in certain situations, but may not be as powerful as deep learning models when it comes to handling complex, high-dimensional data.

Machine learning has the potential to revolutionize the field of point cloud reconstruction by enabling faster, more accurate, and more automated methods for creating models from point clouds.

However, when point clouds include points at the "infinity" (i.e., points having large coordinates), the construction of effective method needs some other approach since otherwise, the prediction of the geometry of the 2D or 3D model could not be the expected one. More precisely, let us consider the point clouds given in Figures 6 and 7.
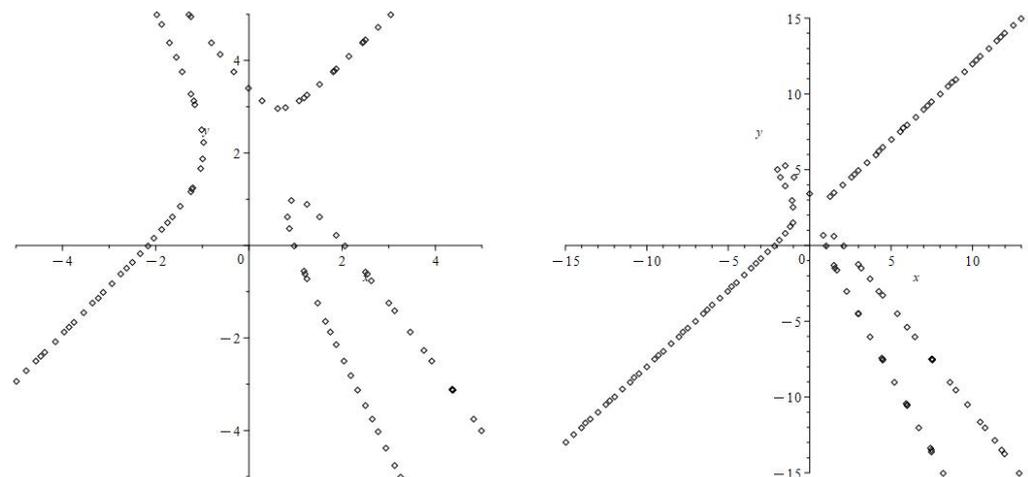
Figure 6. Point clouds at $[-5, 5] \times [-5, 5]$ (**left**) and point clouds at $[-15, 15] \times [-15, 15]$ (**right**).

The behavior of the curve we are looking for, for modeling the point clouds, is totally different if we look at the squares with smaller coordinate points, than if we "go to the infinity". Here, the distortion of the model seems to indicate that we have different curves. In the square $[-N, N] \times [-N, N]$ with N small enough, some well-known machine learning techniques allow to accurately determine the model. However, to correctly predict the geometric object, we need to model the infinity accurately and the essential tool for this problem is the asymptotes.

Thus, in this section, we use a new important tool from which we can extract geometric information from the point cloud and reconstruct a 2D or a 3D model, the *generalized asymptotes* or *g-asymptotes*. A curve may have more general curves than lines describing the status at the points with "large coordinates". More precisely, a curve $\widetilde{C}$ is a *generalized asymptote* (or *g-asymptote*) of another curve C if the distance between $\widetilde{C}$ and C tends to zero as they tend to infinity, and C can not be approached by a new curve of lower degree. This notion, introduced and studied by S. Pérez-Díaz in some previous papers ([56–60]), generalizes the classical concept of an asymptote of a curve C defined as a line such that the distance between C and the line approaches zero as they tend to infinity (see [61–63]).
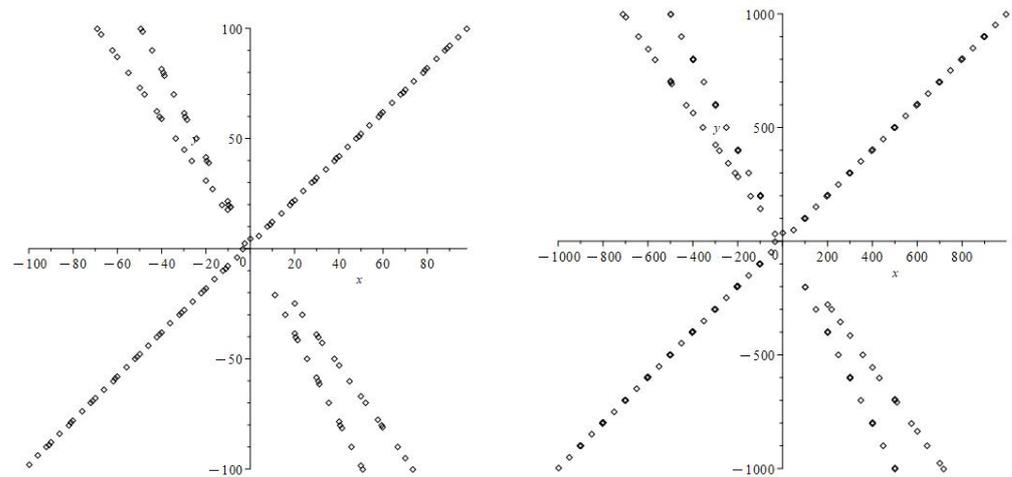
Figure 7. Point clouds at $[-100, 100] \times [-100, 100]$ (**left**) and point clouds at $[-1000, 1000] \times [-1000, 1000]$ (**right**).

The approach using asymptotes for point clouds reconstruction involves fitting a set of asymptotes to the point clouds. The asymptotes can be defined from the infinity branches that can be constructed from the point clouds. Once the asymptotes have been fit to the point clouds, they can be used to reconstruct a 2D or 3D model by interpolating between the points and generating a curve or a surface that follows the asymptotes. This approach can be particularly useful for reconstructing smooth, curve and surfaces, where other methods such as voxel-based reconstruction (see [64]) may not be as effective.

The novelty of this paper is to use the asymptotes that are not necessarily lines but *g-asymptotes*. For this purpose, first we start with some previous notions and we introduce the concept of *infinity branch* and *g-asymptote* from which one may obtain an algebraic plane curve that follows the point clouds. We present the method for the case of plane curves, but this approach can be easily generalized to the *n*-dimensional space (see [58] where the g-asymptotes for algebraic curves in *n*-dimensional space are introduced). The case of surfaces, can be dealt in a similar way, but for this purpose, we need the theory concerning g-asymptotes, which is currently being studied by the authors of this paper (see [65,66]).

The use of infinity branches and g-asymptotes opens up a promising field at intersection with machine learning. In general, any method that seeks boundaries of separation between classes can rely on these concepts to try to look for curves, planes, or hypersurfaces that, in some way, are defined by the asymptotic behavior of the point cloud determined by a given class. Without forgetting the predictive capacity that asymptotes have in themselves, since their own concept is the projection of a tendency towards extreme values of the coordinates.

In the following, let $\mathcal{C}$ be a plane curve over the complex field $\mathbb{C}$ defined by the (irreducible) polynomial $f(x, y) \in \mathbb{R}[x, y]$. Its corresponding projective curve denoted as $\mathcal{C}^*$, is defined by the (homogeneous) polynomial $F(x, y, z) = f_d(x, y) + z f_{d-1}(x, y) + \cdots + z^d f_0 \in \mathbb{R}[x, y, z]$, where $d := \deg(\mathcal{C})$ and $f_j(x, y)$ are the homogeneous forms of degree j, for $j = 0, \ldots, d$. Throughout this section, we assume w.l.o.g. that $(0:1:0)$ is not an infinity point of $\mathcal{C}^*$ (otherwise, we apply a linear change of coordinates).

To obtain the infinity branches of $\mathbb{C}$, we consider the curve defined by the polynomial $g(y, z) = F(1 : y : z)$ and we compute the series expansion for the solutions of $g(y, z) = 0$ around $z = 0$. We obtain $\deg_y(g)$ solutions defined by the (different) Puiseux series that can be grouped into conjugacy classes. That is, if $\varphi(z) = m + a_1 z^{N_1/N} + a_2 z^{N_2/N} + a_3 z^{N_3/N} + \cdots \in \mathbb{C}\langle\langle z \rangle\rangle$, $a_i \neq 0$, $\forall i \in \mathbb{N}$, where $N \in \mathbb{N}$, $N_i \in \mathbb{N}$, $i \in \mathbb{N}$, and $0 < N_1 < N_2 < \cdots$, is a Puiseux series (i.e., $g(\varphi(z), z) = 0$), and $\nu(\varphi) = N$ (N is the called *ramification index* of φ), the series $\varphi_j(z) = m + a_1 c_j^{N_1} z^{N_1/N} + a_2 c_j^{N_2} z^{N_2/N} + a_3 c_j^{N_3} z^{N_3/N} + \cdots$, where $c_j^N = 1$, $j \in \{1, \ldots, N\}$, are the *conjugates* of φ. The set of all the conjugates of φ is called the *conjugacy class* of φ and it contains $\nu(\varphi)$ different series.

Since $g(\varphi(z), z) = 0$ in some neighborhood of $z = 0$ where $\varphi(z)$ converges, there exists $M \in \mathbb{R}^+$ with $F(1 : \varphi(t) : t) = g(\varphi(t), t) = 0$ for $t \in \mathbb{C}$ and $|t| < M$, which implies that $F(t^{-1} : t^{-1}\varphi(t) : 1) = f(t^{-1}, t^{-1}\varphi(t)) = 0$, for $t \in \mathbb{C}$ and $0 < |t| < M$. Set $t^{-1} = z$. We find that $f(z, r(z)) = 0$ for $z \in \mathbb{C}$ and $|z| > M^{-1}$ where

$$r(z) = z\varphi(z^{-1}) = mz + a_1 z^{1 - N_1/N} + a_2 z^{1 - N_2/N} + a_3 z^{1 - N_3/N} + \cdots, \quad a_i \neq 0, \forall i \in \mathbb{N}$$

$N, N_i \in \mathbb{N}$, $i \in \mathbb{N}$, and $0 < N_1 < N_2 < \cdots$.

One may reason likewise with the N different series in the conjugacy class. However, in [57], we prove that all the results hold independently on the chosen series in the conjugacy class. Thus, in the following, we consider any representant in the conjugacy class and we introduce the notion of *infinity branch of a plane curve $\mathcal{C}$*.

Definition 1. *An infinity branch of a plane curve $\mathcal{C}$ associated to the infinity point $P = (1{:}m{:}0)$, $m \in \mathbb{C}$, is a set $B = \{(z, r(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > M\}$, $M \in \mathbb{R}^+$,*

$$r(z) = z\varphi(z^{-1}) = mz + a_1 z^{1 - N_1/N} + a_2 z^{1 - N_2/N} + a_3 z^{1 - N_3/N} + \cdots, \tag{1}$$

where $N, N_i \in \mathbb{N}$, $i \in \mathbb{N}$, and $0 < N_1 < N_2 < \cdots$.

Now, we provide the concepts of *convergent branches* and *approaching curves*. These notions will allow us to study if two curves approach each other (Theorem 2). In addition, Theorem 1 characterizes the convergence of two infinity branches (these notions and the proofs of the theorems can be found in [56,57]).

Definition 2. *Two infinity branches, $B = \{(z, r(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > M\} \subset B$ and $\overline{B} = \{(z, \overline{r}(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > \overline{M}\} \subset \overline{B}$, are convergent if $\lim_{z \to \infty}(\overline{r}(z) - r(z)) = 0$.*

Theorem 1. *Two branches $B = \{(z, r(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > M\}$ and $\overline{B} = \{(z, \overline{r}(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > \overline{M}\}$ are convergent iff the terms with non-negative exponent in $r(z)$ and $\overline{r}(z)$ are the same. Therefore, two convergent infinity branches are associated with the same infinity point.*

The classical concept of asymptote has to be with a line that *approaches* a given curve at the infinity. In the following, we generalize this idea and we say that two curves *approach each other* if they have two infinity branches that converge (see Definition 3 and Theorem 2).

Definition 3. *Let $\mathcal{C}$ be a plane curve with an infinity branch B. A curve $\overline{\mathcal{C}}$ approaches $\mathcal{C}$ at B if $\lim_{z \to \infty} d((z, r(z)), \overline{\mathcal{C}}) = 0$.*

Theorem 2. *Let $\mathcal{C}$ be a plane curve with an infinity branch B. A plane curve $\overline{\mathcal{C}}$ approaches $\mathcal{C}$ at B iff $\overline{\mathcal{C}}$ has an infinity branch, $\overline{B}$, such that B and $\overline{B}$ are convergent.*

Now, we consider $\mathcal{C}$ a plane curve and B an infinity branch of $\mathcal{C}$. We have just described how $\mathcal{C}$ can be approached at B by a new curve $\overline{\mathcal{C}}$, and now we consider that $\deg(\overline{\mathcal{C}}) < \deg(\mathcal{C})$. Then, one may say that $\mathcal{C}$ *degenerates* since it behaves at the infinity as a curve of smaller degree. For example, one may think on a hyperbola that is a curve of degree two having two real asymptotes. This could make us deduce that the hyperbola degenerates at the infinity in two lines. Similarly, an ellipse has two asymptotes that, in this case, are complex lines. The asymptotic behavior of a parabola is different since it cannot be approached at the infinity by any line. This leads us to the notion of *perfect curve* and *g-asymptotes*.

Definition 4. *A curve of degree d is a perfect curve if it cannot be approached by any curve of degree less than d.*

Definition 5. *Let $\mathcal{C}$ be a curve with an infinity branch B. A g-asymptote (or generalized asymptote) of $\mathcal{C}$ at B is a perfect curve that approaches $\mathcal{C}$ at B.*

The notion of *g-asymptote* is a generalization of the classical concept of asymptote since as one may deduce, a g-asymptote is not necessarily a line, but a perfect curve (Definition 4). Throughout this section, we refer to *g-asymptote* simply as *asymptote*.

Every infinity branch of a given plane curve implicitly defined has, at least, one asymptote and now, we show how to compute it. For this purpose, we rewrite Equation (1) defining a branch B (Definition 1) as

$$r(z) = mz + a_1 z^{1-n_1/n} + \cdots + a_k z^{1-n_k/n} + a_{k+1} z^{1-N_{k+1}/N} + \cdots \tag{2}$$

where $0 < N_1 < \cdots < N_k \leq N < N_{k+1} < \cdots$ and $\gcd(N, N_1, \ldots, N_k) = b$, $N = n \cdot b$, $N_j = n_j \cdot b$, $j \in \{1, \ldots, k\}$. That is, we simplify the non-negative exponents such that $\gcd(n, n_1, \ldots, n_k) = 1$. Remark that $0 < n_1 < n_2 < \cdots$, and $n_k \leq n$, and $N < N_{k+1}$, i.e., the terms $a_j z^{1-N_j/N}$ with $j \geq k+1$ are those which have negative exponent. We denote these terms as $A(z) := \sum_{\ell=k+1}^{\infty} a_\ell z^{-q_\ell}$, where $q_\ell = 1 - N_\ell/N \in \mathbb{Q}^+$, *ell* $\geq k+1$. We say that n is *the degree of B*, and we denote it by $\deg(B)$.

Taking into account Theorems 1 and 2, we find that any curve $\overline{\mathcal{C}}$ approaching $\mathcal{C}$ at B should have an infinity branch $\overline{B} = \{(z, \overline{r}(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > \overline{M}\}$ such that the terms with non-negative exponent in $r(z)$ and $\overline{r}(z)$ are the same. In the most simple case, if $A = 0$ (there are no terms with negative exponent, see Equation (2)), we obtain

$$\tilde{r}(z) = mz + a_1 z^{1-n_1/n} + a_2 z^{1-n_2/n} + \cdots + a_k z^{1-n_k/n}, \tag{3}$$

where $a_1, a_2, \ldots \in \mathbb{C} \setminus \{0\}$, $m \in \mathbb{C}$, $n, n_1, n_2 \ldots \in \mathbb{N}$, $\gcd(n, n_1, \ldots, n_k) = 1$, and $0 < n_1 < n_2 < \cdots$. We observe that $\tilde{r}$ has the same terms with non-negative exponent as r, and $\tilde{r}$ does not have terms with negative exponent.

Let $\widetilde{\mathcal{C}}$ be the plane curve containing the branch $\widetilde{B} = \{(z, \tilde{r}(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > \widetilde{M}\}$. We have that

$$\widetilde{\mathcal{Q}}(t) = (t^n, mt^n + a_1 t^{n-n_1} + \cdots + a_k t^{n-n_k}) \in \mathbb{C}[t]^2,$$

where $n, n_1, \ldots, n_k \in \mathbb{N}$, $\gcd(n, n_1, \ldots, n_k) = 1$, and $0 < n_1 < \cdots < n_k$, is a polynomial parametrization of $\widetilde{\mathcal{C}}$ and it is proper (see Lemma 3 in [56]). In Theorem 2 in [56], we prove that $\widetilde{\mathcal{C}}$ is a g-asymptote of $\mathcal{C}$ at B.

In the following, we illustrate this process by means of an example.

Example 1. *Let $\mathcal{C}$ be a curve of degree $d = 4$ defined by (see Figure 8)*

$$f(x, y) = -81x^4 - 144x^3y + 102x^2y^2 + 104xy^3 + 19y^4 + 146x^3 + 729x^2y + 28xy^2 - 7y^3 + 444x^2 - 698xy + 230y^2 - 863x + 618y + 577 \in \mathbb{R}[x, y].$$
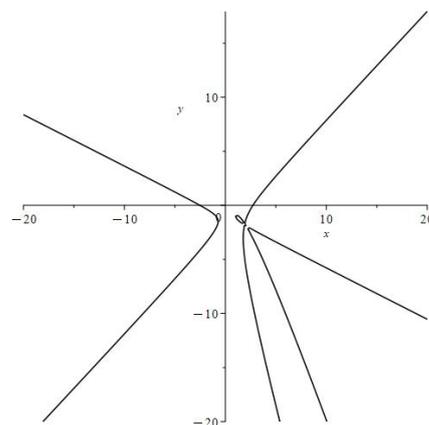

Figure 8. Input curve.

The infinity points are $P_1 = (1 : -3 : 0)$, $P_2 = (1 : 1 : 0)$, and $P_3 = (1 : -19/9 : 0)$. We first consider P_1 and we compute its associated branches and asymptotes.

There exists only one branch associated to P_1, $B_1 = \{(z, r_1(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > M_1\}$, where

$$r_1(z) = -3z + 37703z^{-3}/59049 - 7577437\sqrt{3}z^{-5/2}/20155392 + 3602z^{-2}/6561 - 44111\sqrt{3}z^{-3/2}/93312 + 233z^{-1}/486 - 863\sqrt{3}z^{-1/2}/648 + 31/18 + 5\sqrt{3}\sqrt{z}/3 + \dots$$

(we compute r_1 by using the command puiseux, which is included in the algcurves package of the computer algebra system Maple).

We obtain $\tilde{r}_1(z) = -3z + 31/18 + 5\sqrt{3}\sqrt{z}/3$ and hence, the parametrization of the asymptote $\widetilde{C}_1$ is

$$\widetilde{Q}_1(t) = (t^2, -3t^2 + 31/18 + 5\sqrt{3}t/3).$$

Now, we analyze the point P_2. We have one infinity branch associated to P_2, $B_2 = \{(z, r_2(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > M_2\}$, where

$$r_2(z) = z - 85/64z^{-3} - 9/8z^{-2} - 5/4z^{-1} - 2 + \cdots.$$

We obtain that $\tilde{r}_2(z) = -2 + z$ and thus, the parametrization of the asymptote $\widetilde{C}_2$ is given by $\widetilde{Q}_2(t) = (t, -2 + t) \in \mathbb{R}[t]^2$.

Now, we analyze the point P_3. We have one infinity branch associated to P_3, $B_3 = \{(z, r_3(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > M_2\}$, where

$$r_3(z) = -9z/19 + 193181/3779136z^{-3} + 1417/52488z^{-2} - 14495/18468z^{-1} + \cdots.$$

We obtain that $\tilde{r}_3(z) = -9z/19$. The parametrization of the asymptote $\widetilde{C}_3$ is given by $\widetilde{Q}_3(t) = (t, -9t/19) \in \mathbb{R}[t]^2$.

In Figure 9, we plot the curve C, and the asymptotes $\widetilde{C}_1$, $\widetilde{C}_2$, and $\widetilde{C}_3$.
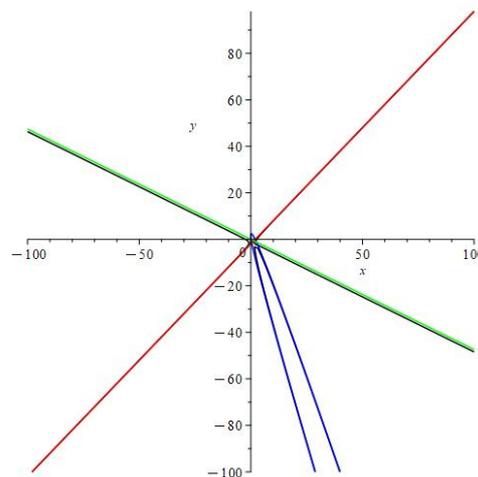
Figure 9. Input curve with the asymptotes in $[-100, 100] \times [-100, 100]$.

Observe that Figure 9 is plotted in the square $[-100, 100] \times [-100, 100]$. Note that in this square, where the points have "sufficient large coordinates", asymptotes approach perfectly to the input curve. However, if we plot C with the asymptotes $\widetilde{C}_1$, $\widetilde{C}_2$, and $\widetilde{C}_3$ in a smaller square, $[-20, 20] \times [-20, 20]$ (see Figure 10), one may check that the asymptotes approach C worse. That is, as one knows, the approach of asymptotes to the curve is good at the infinity and in fact, the asymptotes are the only tool we have to approach at the infinity.
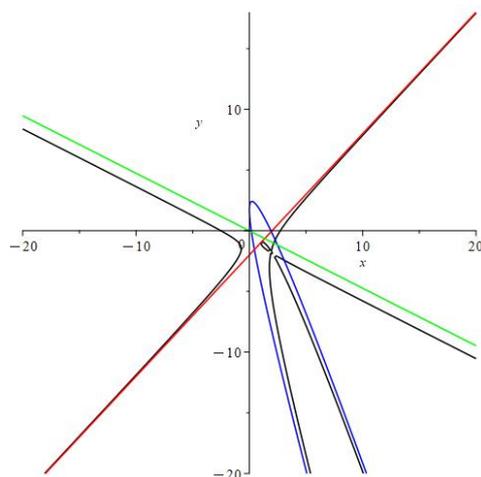
Figure 10. Input curve with the asymptotes in $[-20, 20] \times [-20, 20]$.

Now, let us assume that we are given a point clouds as in Figure 7, and we are interested in making specific predictions or decisions based on the data. For this purpose, one has to develop methods that should generate geometric models and analyze its geometric properties. In fact as we stated above, although some machine learning algorithms are developed in this sense, some new tools are necessary to understand the behavior at the infinity, which is summarized in constructing the asymptotes.

For this purpose, the idea we provide in this paper is the following: from the point clouds at the infinity and in each of the directions, the infinity branches, $B = \{(z, r(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > M\}$, passing through those points are constructed. One can consider branches of degree $n \geq 1$ according to the given point clouds. The ramification index to be considered (i.e., the value of N) and the number of terms in $r(z)$ can be as large as one wishes, depending on the number of points one wants to use. As more points one considers, a better approximation is obtained.

As one can deduce, this method only involves linear systems and their solution provides the infinity branches and hence the asymptotes. That is, we compute $(t^n, r(t^n))$, where $r(z)$ has as many terms as one wishes. Depending on the approximation purposes, either the whole infinity branch can be used, or only the asymptote determined from it. Note that an infinity branch (with a finite number of terms) is, in the background, a parametrization however, it is not polynomial (as the asymptote is) and its degree is N, which could be much larger than the degree of the asymptote (n).

Additionally and in order to measure the error, for each given point one may compute the distance to the nearest point on the asymptote. An error will be acceptable depending on the objectives of the problem being addressed.

This new technique, as we mentioned above, can serve as a basis for new clustering algorithms based on the distance of points to a hypersurface. The approach could also be combined with dimensionality reduction methods or methods based on manifolds to search for spaces in which points can be separated by lower-order surfaces, as SVM kernels do. All the mathematical theory developed so far can be used as a starting point for such algorithms, by means of the numerical approximations presented above, setting as parameters the maximum orders or the errors committed, for example.

Therefore, this method can be used as a basis for a new classification algorithm, and also, once the families of asymptotes that solve the problem have been determined, they can be used as a basis for predicting the classes in the extreme locations of the points, whether they are time variables or other types of variables. These asymptotes, together with plausibility conditions imposed by the problem, can lead to knowledge discovery in problems analyzed in this way. It also opens up a new area of work in the study of asymptotes that can represent acceptable solutions, as opposed to those that provide false solutions.

This proposal for the use of asymptotes is being developed by the authors and it will be the subject of publications and developments in forthcoming articles.

In the following, we illustrate these ideas by means of several examples. In the first one, the asymptotes computed have a degree of one. However, the second provides asymptotes of degree two.

Example 2. *We are given a set of point clouds that we plot in Figures 6 and 7.*

One should note that the input points are given, in general, in floating point arithmetic. From the figures, we observe that we have three different directions and then one should obtain three infinity branches. We consider $N = 4$ and $n = 1$ (if the approximation is not good enough, one may increase the value of n).

The first branch obtained is $B_1 = \{(z, r_1(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > M_1\}$, where

$$r_1(z) = -2z - 0.02961946858z^{-4} + 0.1088248743z^{-3} - 0.05349794239z^{-2}$$
$$+0.5185185185z^{-1} + 1.333333333.$$

We compute $\tilde{r}_1(z)$, and we have that $\tilde{r}_1(z) = -2z + 1.333$. Hence, the parametrization of the asymptote $\widetilde{\mathcal{C}}_1$ is $\widetilde{\mathcal{Q}}_1(t) = (t, -2t + 1.33)$.

Now, we determine $B_2 = \{(z, r_2(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > M_2\}$, where

$$r_2(z) = z + 0.09375000000z^{-4} + 0.01562500000z^{-3} + 0.2500000000z^{-2} - 0.2500000000z^{-1}$$
$$+ 2.$$

We obtain that $\tilde{r}_2(z) = 2 + z$ and thus, the parametrization of the asymptote $\widetilde{\mathcal{C}}_2$ is given by $\widetilde{\mathcal{Q}}_2(t) = (t, 2 + t) \in \mathbb{R}[t]^2$.

Finally, we obtain the infinity branch $B_3 = \{(z, r_3(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > M_2\}$, where

$$r_3(z) = -1.4z - 0.06413053142z^{-4} - 0.1244498743z^{-3} - 0.1965020576z^{-2}$$
$$-0.2685185185z^{-1} + 3.066666667.$$

We obtain that $\tilde{r}_3(z) = -1.4z + 3.0666$, and the parametrization of the asymptote $\widetilde{\mathcal{C}}_3$ is given by $\widetilde{\mathcal{Q}}_3(t) = (t, -1.4t + 3.0666) \in \mathbb{R}[t]^2$.

In Figure 11, we plot the curve $\mathcal{C}$, and the asymptotes $\widetilde{\mathcal{C}}_1$, $\widetilde{\mathcal{C}}_2$ and $\widetilde{\mathcal{C}}_3$ in $[-1000, 1000] \times [-1000, 1000]$.
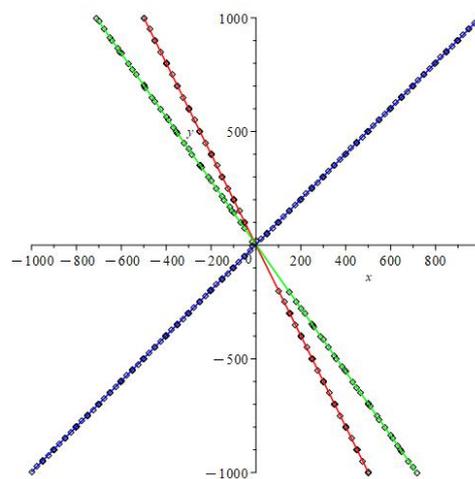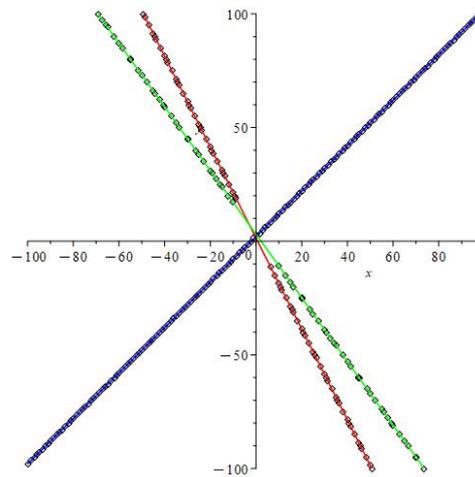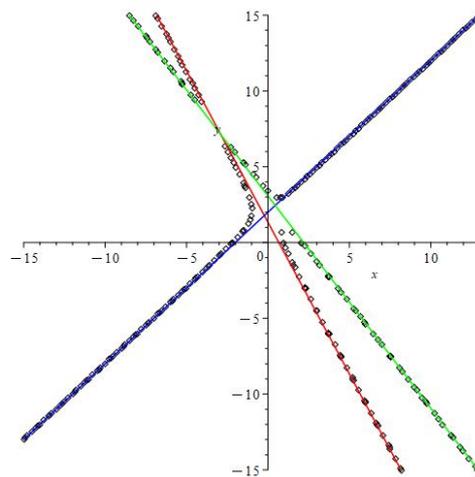

Figure 11. Input curve with asymptotes in $[-1000, 1000] \times [-1000, 1000]$.

In Figure 12, we plot the curve $\mathcal{C}$, and the asymptotes in $[-100, 100] \times [-100, 100]$.
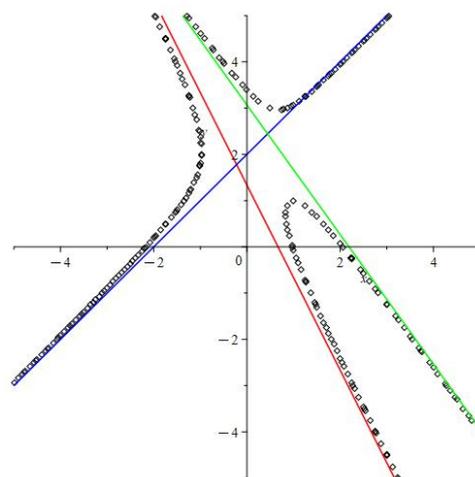
Figure 12. Input curve with asymptotes in $[-100, 100] \times [-100, 100]$.

In Figure 13, we plot the curve $\mathcal{C}$, and the asymptotes in $[-15, 15] \times [-15, 15]$.

Figure 13. Input curve with asymptotes in $[-15, 15] \times [-15, 15]$.

*In Figure 14, we plot the curve $\mathcal{C}$, and the asymptotes in $[-5, 5] \times [-5, 5]$.*

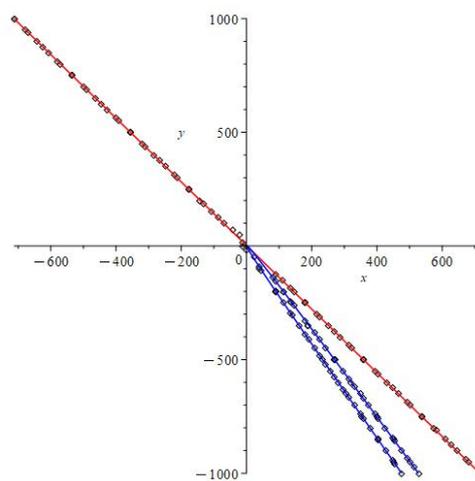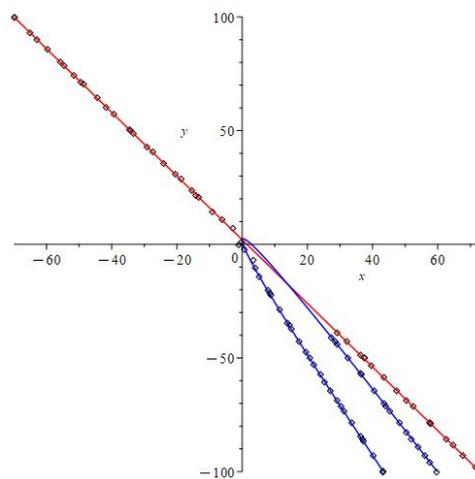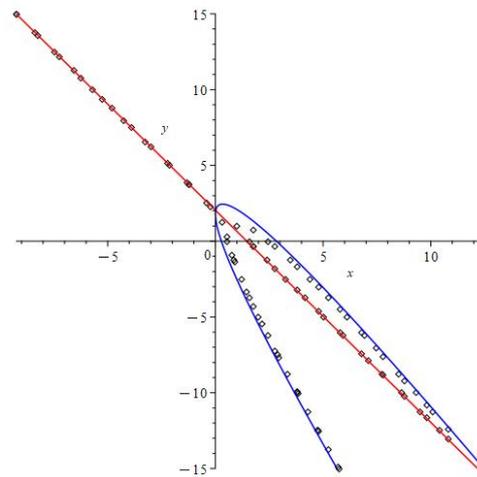**Figure 14.** Input curve with asymptotes in $[-5, 5] \times [-5, 5]$.

One may observe that, although the approximation in the squares of smaller length is also good, the error in this area is much greater than in the infinity. More precisely, if for each given point we calculate the distance to the nearest point on the nearest line, we obtain that for the first case ($[-1000, 1000]^2$) the error is less or equal than 10^{-6}, for $[-100, 100]^2$ the error is less or equal than 10^{-3}, for $[-15, 15]^2$ the error is less or equal than 10^{-2}, and for $[-5, 5]^2$ the error is less or equal than 10^{-1}. However, is important to note that topologically, the asymptotes describe the curve perfectly at the infinity but not in the area near the origin.

In the following example, we have that asymptotes of degree two have to be used to obtain a better approximation at one of the directions of the infinity.

Example 3. *We are given a set of point clouds that we plot in Figures 15 and 16.*
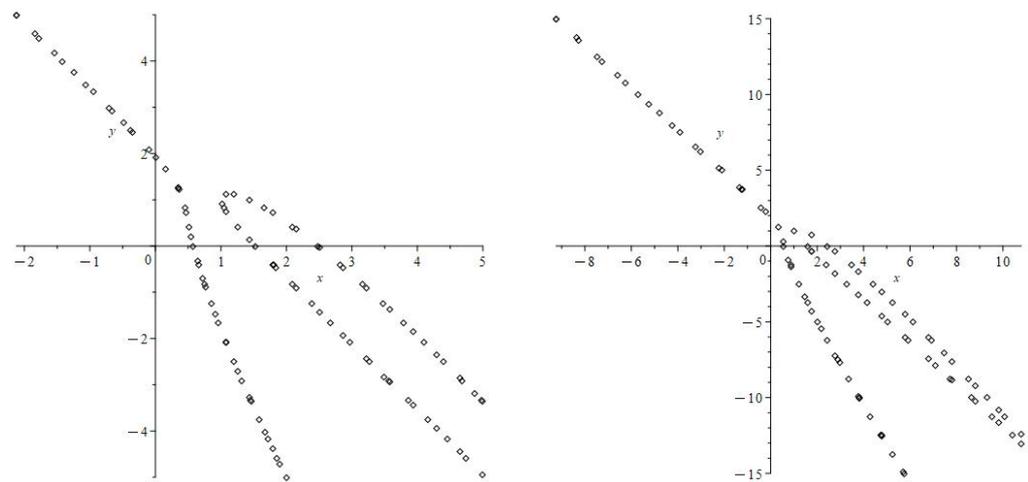
Figure 15. Point clouds at $[-5, 5] \times [-5, 5]$ (**left**) and point clouds at $[-15, 15] \times [-15, 15]$ (**right**).
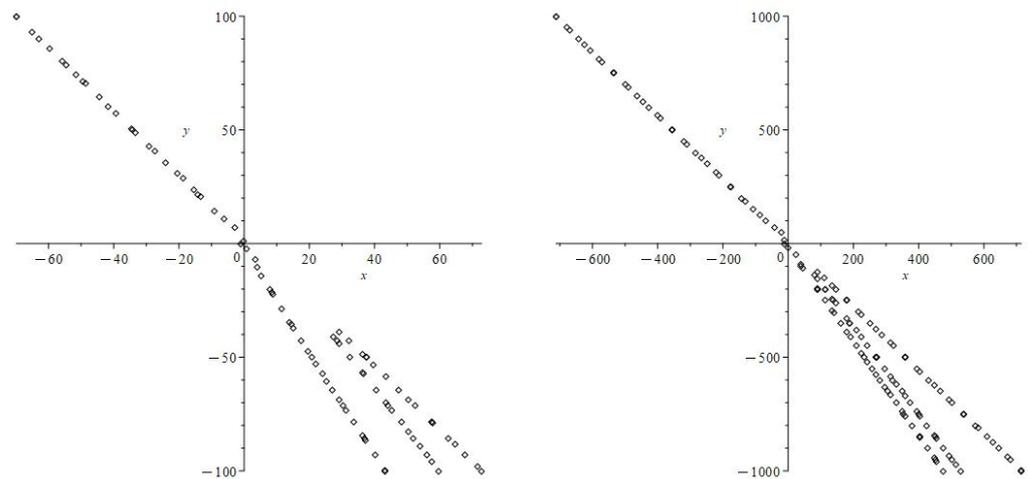
Figure 16. Point clouds at $[-100, 100] \times [-100, 100]$ (**left**) and point clouds at $[-1000, 1000] \times [-1000, 1000]$ (**right**).

From these figures, we observe that we have two different directions and then one should compute two infinity branches. We consider $N = 4$ and $n = 1$ (if the approximation is not good one can increase the value of n).

The first branch obtained is $B_1 = \{(z, r_1(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > M_1\}$, where

$$r_1(z) = -1.4z + 0.02264890280z^{-4} + 0.03151619841z^{-3} + 0.04587715287z^{-2} + 0.06995884774z^{-1} + 2.044444444.$$

We compute $\tilde{r}_1(z)$, and we have that $\tilde{r}_1(z) = -1.4z + 2.0444$. Hence, the parametrization of the asymptote $\widetilde{\mathcal{C}}_1$ is $\widetilde{\mathcal{Q}}_1(t) = (t, -1.4t + 2.0444)$.

Now, we determine $B_2 = \{(z, r_2(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, |z| > M_2\}$. If one considers $n = 1$ one does not obtain a nice approximation. Thus, let $n = 2$. We find

$$r_2(z) = -2z^2 + 2.30940107700000z + 1.77777777800000 - 0.652191970900000z^{-1}$$
$$- 0.0349794238699994z^{-2} - 0.102090614700000z^{-3} - 0.0229385764400001z^{-4}.$$

We obtain that $\tilde{r}_2(z) = -2z^2 + 2.30940107700000z + 1.77777777800000$ and thus, the parametrization of the asymptote $\widetilde{\mathcal{C}}_2$ is given by $\widetilde{\mathcal{Q}}_2(t) = (t^2, -2t^2 + 2.30940107700000t + 1.77777777800000) \in \mathbb{R}[t]^2$.

In Figure 17, we plot the curve $\mathcal{C}$, and the asymptotes $\widetilde{\mathcal{C}}_1$, and $\widetilde{\mathcal{C}}_2$ in $[-1000, 1000] \times [-1000, 1000]$.

Figure 17. Input curve with asymptotes in $[-1000, 1000] \times [-1000, 1000]$.

In Figure 18, we plot the curve $\mathcal{C}$, and the asymptotes in $[-100, 100] \times [-100, 100]$.

Figure 18. Input curve with asymptotes in $[-100, 100] \times [-100, 100]$.

In Figure 19, we represent the curve $\mathcal{C}$, and the asymptotes in $[-15, 15] \times [-15, 15]$.

Figure 19. Input curve with asymptotes in $[-15, 15] \times [-15, 15]$.

In Figure 20, we plot the curve $\mathcal{C}$, and the asymptotes in $[-5, 5] \times [-5, 5]$.
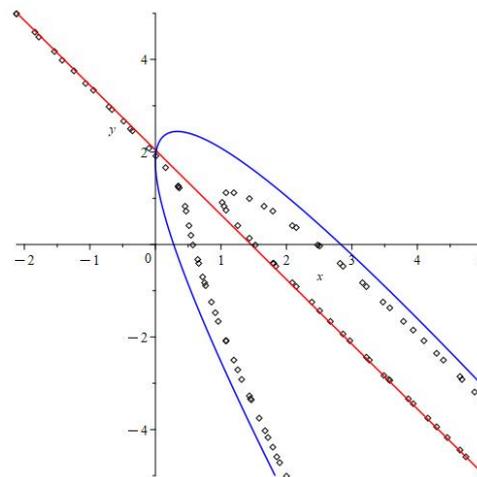

Figure 20. Input curve with asymptotes in $[-5, 5] \times [-5, 5]$.

Now, for each given point we calculate the distance to the nearest point on the asymptotes. We obtain that for the first case ($[-1000, 1000]^2$) the error is less or equal than 10^{-5}, for $[-100, 100]^2$ the error is less or equal than 10^{-4}, for $[-15, 15]^2$ the error is less or equal than 10^{-1}, and for $[-5, 5]^2$ the error is less or equal than 0.5.

6. Conclusions

Machine learning is a highly promising technique that has demonstrated significant value in pattern recognition, knowledge discovery, prediction, and classification. Artificial intelligence is one of the most disruptive technologies today, with machine learning being a significant contributor. The discipline of machine learning is currently experiencing a surge in growth, boosted by advancements in computer processing power and the abundance of data available for analysis.

In addition to traditional mathematical development, computers play a crucial role in modern mathematics. With the rise of numerical and symbolic calculus software, computer-assisted mathematics and geometry have become increasingly prevalent. This opens the door for machine learning to be an powerful tool into the realm of geometry by generating datasets that represent open problems that cannot be solved using conventional methods, either due to their complexity, novelty, or unfamiliarity. Machine learning has the potential to model multi-dimensional point clouds, approximate curves numerically,

and identify singularities, anomalies, and patterns that can later be mathematically proven. The possibilities for machine learning in this field are virtually limitless.

In particular, in this paper we present a method using asymptotes for point clouds reconstruction that involves fitting a set of asymptotes to the point clouds. The asymptotes can be defined from the infinity branches that can be constructed from the point clouds. Once the asymptotes have been fitted to the point clouds, they can be used to reconstruct a 2D or 3D model by interpolating the points and generating a curve or a surface that follows the asymptotes. The novelty is that one may use the asymptotes that are not necessarily lines but *g-asymptotes*.

Our presented method, using asymptotes and point reconstruction, is an example of one of the lines of joint work between geometry and machine learning. On the one hand, we have that the theoretical constructs are able to generate series of computed points, with user-controlled accuracy, on which the usual machine learning algorithms can be applied to perform predictions, modeling, classification, or pattern search.

Each of these approaches can provide different and novel solutions to a mathematical problem, from interpolation or extrapolation issues, to the issue of calculating functions or asymptotes, or even the discovery of singularities or anomalies not predicted. The machine learning approach to numerical sets may provide geometry and mathematics with discoveries of the same kind that machine learning is now providing to different areas of knowledge.

In addition, the methods used in geometry can be applied to extend current machine learning algorithms, or even generate new ones. The very concept of asymptote is a modeling of a behavior at infinity. While up to now, asymptotes have been calculated on curves or surfaces, in the case of generalized asymptotes, the calculation of numerical asymptotes on point clouds can open up new ways of modeling real processes, or of calculating evolutions and predictions about such processes. This is, for example, a current line of work being conducted by two of the authors.

References

1. Alpaydin, E. *Introduction to Machine Learning*; The MIT Press: Cambridge, MA, USA; London, UK, 2020.
2. Gelf, I.M.; Alekseyevskaya, T. *Geometry*; Birkhäuser: New York, NY, USA, 2020.

3. Griffiths, P.; Harris, J. *Principles of Algebraic Geometry*; John Wiley and Sons, Inc.: Hoboken, NJ, USA, 2011.
4. Wu, W.; Gao, X. Mathematics mechanization and applications after thirty years. *Front. Comput. Sc. China* **2007**, *1*, 1–8. [CrossRef]
5. Janićixcx, P.; Kovács, Z. In Proceedings of the 13th International Conference on Automated Deduction in Geometry, Hagenberg, Austria, 15–17 September 2021.
6. Dirac, P. XI.—The relation between mathematics and physics. *Proc. R. Soc. Edinb.* **1940**, *59*, 122–129. [CrossRef]
7. Davies, A.; Veličković, P.; Buesing, L.; Blackwell, S.; Zheng, D.; Tomašev, N.; Tanburn, R.; Battaglia, P.; Blundell, C.; Juhász, A.; et al. Advancing mathematics by guiding human intuition with AI. *Nature* **2021**, *600*, 70–74. [CrossRef] [PubMed]
8. He, Y.; Kim, M. Learning algebraic structures: Preliminary investigations. *Int. J. Data Sci. Math. Sci.* **2023**, *1*, 3–22. [CrossRef]
9. He, Y.; Lee, K.; Oliver, T. Machine learning invariants of arithmetic curves. *J. Symb. Comput.* **2023**, *115*, 478–491. [CrossRef]
10. Bobev, N.; Fischbacher, T.; Pilch, K. Properties of the new N = 1 AdS4 vacuum of maximal supergravity. *J. High Energy Phys.* **2020**, *2020*, 99. [CrossRef]
11. Comsa, I.; Firsching, M.; Fischbacher, T. SO(8) supergravity and the magic of machine learning. *J. High Energy Phys.* **2019**, *2019*, 57. [CrossRef]
12. Krishnan, C.; Mohan, V.; Ray, S. Machine Learning Gauged Supergravity. *Fortschritte Der Phys.* **2020**, *68*, 2000027. [CrossRef]
13. Angulo, R.; Hahn, O. Large-scale dark matter simulations. *Living Rev. Comput. Astrophys.* **2022**, *8*, 1. [CrossRef]
14. Barsotti, D.; Cerino, F.; Tiglio, M.; Villanueva, A. Gravitational wave surrogates through automated machine learning. *Class. Quantum Gravity* **2022**, *39*, 085011. [CrossRef]
15. Dieselhorst, T.; Cook, W.; Bernuzzi, S.; Radice, D. Machine learning for conservative-to-primitive in relativistic hydrodynamics. *Symmetry* **2021**, *13*, 2157. [CrossRef]
16. Bachtis, D.; Aarts, G.; Lucini, B. Quantum field-theoretic machine learning. *Phys. Rev. D* **2021**, *103*, 074510. [CrossRef]
17. Bachtis, D.; Aarts, G.; Lucini, B. Quantum field theories, Markov random fields and machine learning. *J. Phys. Conf. Ser.* **2022**, *2207*, 012056. [CrossRef]
18. Carrasquilla, J. Machine learning for quantum matter. *Adv. Phys. X* **2020**, *5*, 1797528. [CrossRef]
19. Kudyshev, Z.; Shalaev, V.; Boltasseva, A. Machine Learning for Integrated Quantum Photonics. *ACS Photonics* **2021**, *8*, 34–46. [CrossRef]
20. Zhang, Y.; Ginsparg, P.; Kim, E. Interpreting machine learning of topological quantum phase transitions. *Phys. Rev. Res.* **2020**, *2*, 023283. [CrossRef]
21. Ding, Y.; Martín-Guerrero, J.; Sanz, M.; Magdalena-Benedicto, R.; Chen, X.; Solano, E. Retrieving Quantum Information with Active Learning. *Phys. Rev. Lett.* **2020**, *124*, 140504. [CrossRef]
22. Houssein, E.; Abohashima, Z.; Elhoseny, M.; Mohamed, W. Machine learning in the quantum realm: The state-of-the-art, challenges, and future vision. *Expert Syst. Appl.* **2022**, *194*, 116512. [CrossRef]
23. Khan, T.; Robles-Kelly, A. Machine Learning: Quantum vs. Classical. *IEEE Access* **2020**, *8*, 219275–219294. [CrossRef]
24. Schuld, M.; Sinayskiy, I.; Petruccione, F. An introduction to quantum machine learning. *Contemp. Phys.* **2015**, *56*, 172–185. [CrossRef]
25. Janiesch, C.; Zschech, P.; Heinrich, K. Machine learning and deep learning. *Electron. Mark.* **2021**, *31*, 685–695. [CrossRef]
26. Bernal, E.; Hauenstein, J.; Mehta, D.; Regan, M.; Tang, T. Machine learning the real discriminant locus. *J. Symb. Comput.* **2023**, *115*, 409–426. [CrossRef]
27. Pérez-Díaz, S.; Sendra, J.; Sendra, J. Parametrization of approximate algebraic curves by lines. *Theor. Comput. Sci.* **2004**, *315*, 627–650. [CrossRef]
28. Hutter, M. Universal Algorithmic Intelligence: A Mathematical Top→Down Approach. *Cogn. Technol.* **2007**, *8*, 227–290.
29. Ayesha, S.; Hanif, M.; Talib, R. Overview and comparative study of dimensionality reduction techniques for high dimensional data. *Inf. Fusion* **2020**, *59*, 44–58. [CrossRef]
30. Bouveyron, C.; Girard, S.; Schmid, C. High-dimensional data clustering. *Comput. Stat. Data Anal.* **2007**, *52*, 502–519. [CrossRef]
31. Doraiswamy, H.; Tierny, J.; Silva, P.; Nonato, L.; Silva, C. TopoMap: A 0-dimensional homology preserving projection of high-dimensional data. *IEEE Trans. Vis. Comput. Graphys.* **2021**, *27*, 561–571. [CrossRef] [PubMed]
32. Wang, Z.; Scott, D. Nonparametric density estimation for high-dimensional data—Algorithms and applications. *Wiley Interdiscip. Rev. Comput. Stat.* **2019**, *11*, e1461. [CrossRef]
33. Atz, K.; Grisoni, F.; Schneider, G. Geometric deep learning on molecular representations. *Nat. Mach. Intell.* **2021**, *3*, 1023–1032. [CrossRef]
34. Cao, W.; Yan, Z.; He, Z.; He, Z. A Comprehensive Survey on Geometric Deep Learning. *IEEE Access* **2020**, *8*, 35929–35949. [CrossRef]
35. Sommer, S.; Bronstein, A. Horizontal Flows and Manifold Stochastics in Geometric Deep Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 811–822. [CrossRef] [PubMed]
36. Berman, D.; He, Y.; Hirst, E. Machine learning Calabi-Yau hypersurfaces. *Phys. Rev. D* **2022**, *105*, 066002. [CrossRef]
37. Carifio, J.; Halverson, J.; Krioukov, D.; Nelson, B. Machine learning in the string landscape. *J. High Energy Phys.* **2017**, *2017*, 157. [CrossRef]
38. Krefl, D.; Seong, R. Machine learning of Calabi-Yau volumes. *Phys. Rev. D* **2017**, *96*, 066014. [CrossRef]
39. Feng, H.; Li, J.; Zhou, D. Approximation Analysis of CNNs from Feature Extraction View. 2022. Available online: https://ssrn.com/abstract=4294503 (accessed on 1 June 2023).

40. Bellaard, G.; Bon, D.; Pai, G.; Smets, B.; Duits, R. Analysis of (sub-)Riemannian PDE-G-CNNs. *arXiv* **2022**. arXiv:2210.00935.
41. Chen, Z.; Wu, B.; Liu, W. Mars3dnet: Cnn-based high-resolution 3d reconstruction of the martian surface from single images. *Remote Sens.* **2021**, *13*, 839. [CrossRef]
42. Li, J.; Zhang, H.; Wan, W.; Sun, J. Two-class 3D-CNN classifiers combination for video copy detection. *Multimed. Tools Appl.* **2020**, *79*, 4749–4761. [CrossRef]
43. Li, H.; Todd, Z.; Bielski, N.; Carroll, F. 3D lidar point-cloud projection operator and transfer machine learning for effective road surface features detection and segmentation. *Vis. Comput.* **2022**, *38*, 1759–1774. [CrossRef]
44. Rehman, M.; Ahmed, F.; Khan, M.; Tariq, U.; Alfouzan, F.; Alzahrani, N.; Ahmad, J. Dynamic hand gesture recognition using 3D-CNN and LSTM networks. *Comput. Mater. Contin.* **2022**, *70*, 4675–4690. [CrossRef]
45. Bao, J.; He, Y.; Hirst, E. Neurons on amoebae. *J. Symb. Comput.* **2023**, *116*, 1–38. [CrossRef]
46. Gelf, I.; Kapranov, M.; Zelevinsky, A. *Discriminants, Resultants, and Multidimensional Determinants*; Birkhäuser: New York, NY, USA, 1994.
47. Chen, S.; He, Y.; Hirst, E.; Nestor, A.; Zahabi, A. Mahler Measuring the Genetic Code of Amoebae. *arXiv* **2022**, arXiv:2212.06553.
48. Luo, J.; Huang, J. Generative adversarial network: An overview. *Chin. J. Sci. Instrum.* **2019**, *40*, 74–84.
49. Assouli, M.; Missaoui, B. Deep Learning for Mean Field Games with non-separable Hamiltonians. *arXiv* **2023**, arXiv:2301.02877.
50. Stinis, P.; Daskalakis, C.; Atzberger, P. SDYN-GANs: Adversarial Learning Methods for Multistep Generative Models for General Order Stochastic Dynamics. *arXiv* **2023**, arXiv:2302.03663.
51. Lonjou, A. Sur lhyperbolicite de graphes associes au groupe de Cremona. *E Pijournal GéOméTrie AlgéBrique* **2019**, *3*, 4895. [CrossRef]
52. Peifer, D.; Stillman, M.; Halpern-Leistner, D. Learning selection strategies in Buchberger's algorithm. In Proceedings of the 37th International Conference on Machine Learning, Online, 13–18 July 2020; Volume 119.
53. Bao, J.; He, Y.; Hirst, E.; Hofscheier, J.; Kasprzyk, A.; Majumder, S. Polytopes and Machine Learning. *arXiv* **2021**, arXiv:2109.09602.
54. Heal, K.; Kulkarni, A.; Sertöz, E. Deep Learning GaussManin Connections. *Adv. Appl. Clifford Algebr.* **2022**, *32*, 24. [CrossRef]
55. Gu, J.; Zheng, Z.; Zhou, W.; Zhang, Y.; Lu, Z.; Yang, L. Self-Supervised Graph Representation Learning via Information Bottleneck. *Symmetry* **2022**, *14*, 657. [CrossRef]
56. Blasco, A.; Pérez-Díaz, S. Asymptotes and Perfect Curves. *Comput. Aided Geom. Des.* **2014**, *31*, 81–96. [CrossRef]
57. Blasco, A.; Pérez-Díaz, S. Asymptotic Behavior of an Implicit Algebraic Plane Curve. *Comput. Aided Geom. Des.* **2014**, *31*, 345–357. [CrossRef]
58. Blasco, A.; Pérez-Díaz, S. Asymptotes of Space Curves. *J. Comput. Appl. Math.* **2015**, *278*, 231–247. [CrossRef]
59. Blasco, A.; Pérez-Díaz, S. A New Approach for Computing the Asymptotes of a Parametric Curve. *J. Comput. Appl. Math.* **2020**, *364*, 112350. [CrossRef]
60. Campo-Montalvo, E.; de Sevilla Fernández, M.; Pérez-Díaz, S. A simple formula for the computation of branches and asymptotes of curves and some applications. *Comput. Aided Geom. Des.* **2022**, *94*, 102084. [CrossRef]
61. Kečkić, J.D. A Method for Obtaining Asymptotes of Some Curves. *Teach. Math. III* **2000**, *1*, 53–59.
62. Maxwell, E.A. *An Analytical Calculus*; Cambridge University Press: Cambridge, UK, 1962.
63. Zeng, G. Computing the Asymptotes for a Real Plane Algebraic Curve. *J. Algebra* **2007**, *316*, 680–705. [CrossRef]
64. Tahir, R.; Bux, S.A.; Habib, Z. Voxel-Based 3D Object Reconstruction from Single 2D Image Using Variational Autoencoders. *Mathematics* **2021**, *9*, 2288. [CrossRef]
65. Campo-Montalvo, E.; de Sevilla Fernández, M.; Pérez-Díaz, S. Asymptotic behavior of a surface implicitly defined. *Mathematics* **2022**, *10*, 1445. [CrossRef]
66. de Sevilla Fernández, M.; Magdalena-Benedicto, R.; Pérez-Díaz, S. Asymptotic Behavior of Parametric Algebraic Surfaces. *Contemp. Math.* **2023**, *accepted*.