UNIVERSITÀ DEGLI STUDI DI MILANO

Scuola di Dottorato in Fisica, Astrofisica e Fisica Applicata
Dipartimento di Fisica

Corso di Dottorato in Fisica, Astrofisica e Fisica Applicata
Ciclo XXXIV

Aspects of Data Structure in Machine Learning

Settore Scientifico Disciplinare FIS/02

Supervisore: Professor Sergio Caracciolo

Coordinatore: Professor Matteo Paris

Tesi di Dottorato di:
Vittorio Erba

Anno Accademico 2020/21

Final examination:

September 21, 2021

Dipartimento di Fisica, Università degli Studi di Milano, Milano, Italy

Commission of the final examination:

External Member:
Professor Chiara Cammarota, Dipartimento di Fisica, Sapienza Università di Roma

External Member:
Professor Pierluigi Contucci, Dipartimento di Matematica, Alma Mater Studiorum Università di Bologna

Internal Member:
Professor Sergio Caracciolo, Dipartimento di Fisica, Università degli Studi di Milano

External referees:

Enrico Maria Malatesta, Artificial Intelligence Lab, Università Bocconi, Milano

Gabriele Sicuro, Disordered Systems Group, Mathematics Department, King's College, London

Cover illustration:

The cover illustration was generated using VQGAN+CLIP [ERO20; RSK+21], a generative machine learning model that synthesizes images from text descriptions. This image was generated from the following text, combining the description with modifying keywords to obtain a peculiar graphical effect: *"A painting of an apple in a fruit bowl | psychedelic | surreal:0.5 | weird:0.25"*. The code is available at `https://github.com/nerdyrodent/VQGAN-CLIP`.

MIUR subjects:

FIS/02

Abstract

It is widely believed that understanding data structure is a crucial ingredient to push forward our comprehension on how (and why) modern machine learning works. Still, most of the theoretical results we have are obtained under very simplifying assumptions on the structure of the training data.

In this Thesis, I review some novel results on the problem of characterizing the geometric structure of datasets and the consequences that this structure has on learning algorithms. I also provide pedagogical introductions to manifold learning, random geometric graphs theory and supervised binary classification.

I focus on three different aspects of the problem. First, I spend some time reviewing techniques to characterize the intrinsic dimensionality of datasets: this is the first "experimental" step towards proper theoretical modelling of data. Then, I focus on the problem of finding null models of data in high-dimension: does Euclidean structure survive when the dimensionality of data becomes larger and larger? Finally, I study how geometric data structure alters the expressive potential of simple classifiers.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Deep neural networks have profoundly changed the way we look at machine learning in the last 10 years. State-of-the-art architectures perform comparably, or even better, than humans; yet, humans outperform algorithms whenever distortion or noise alters the data [GTR+18]. Moreover, deep neural networks can be described effectively as "black boxes": we can test their performances experimentally, but in the vast majority of cases we are not able to explain their inner workings.

These observations foreshadow a near future where learning algorithms will aid, if not substitute completely, humans in decision making tasks, possibly with repercussions on how we understand, for example, healthcare and policy making, without many guarantees of robustness or explainability. It is thus of primary importance to understand at the theoretical level the ins and outs of modern machine learning algorithms. Currently, we lack a satisfying analytical framework.

One of the many difficulties of modelling deep neural networks, but also simpler architectures such as linear models and support-vector machines, is that many ingredients conspire together to make them work. Obviously one has to take into account the specific architecture that learns to solve a task. Fully-connected networks, Gaussian processes, ResNets, they all differ crucially in how they process data. But this is not nearly enough. The nature of the task to be learned imposes constraints on how to engineer the learning architecture: image classification calls for convolutional neural networks due to translational invariance, speech recognition calls for recurrent architectures that store some memory of past signals due to the contextuality of languages, and the list goes on [BBC+21]. Again, this is not enough! Training is often modeled as an high-dimensional risk minimization problem, which is often non-convex, presenting many global and local minima. The outcome of learning algorithms will thus be non-unique, depending on the training data and on the specific initialization of the architecture. And finally, the nature of the training data will itself play a role in what the algorithms learns, how fast and reliably it learns it and how well the architecture will generalize to unseen examples. All these ingredient are interconnected even in the simplest learning tasks, so that analytical models often fail to describe realistic settings.

In this Thesis, I would like to take a step back from all the complexity to look at the least studied aspect: data structure. Both Statistical Learning Theory (SLT), which is the branch of statistics that aims to characterize learning algorithms, and Statistical Physics, which has often been used as a tool to understand average properties of optimization algorithms, have traditionally studied overly-simplified models of data in order to model all the remaining components faithfully. Classic SLT provides bounds on the generalization performances of learners in the worst-case scenario, or uniformly over large classes of model functions. This approach is bound to be of low usefulness, as worst-case/uniform analyses have to account for rare and pathological situations. On the other side, Statistical Physics looks at the average case, which while hopefully being more directly useful, needs simplifying assumptions on data structure in order to obtain usable analytical results. To be fair, on both sides there have been recent progresses towards understanding more complex data structures, see for example [CLS16; GMK$^+$20; RLG20], but there is still a long way to go.

In this Thesis, I approach data structure from three different points of view. In Chapter 2, I focus on the problem of estimating structure from real data, which is a crucial step towards understanding which models of data structure are realistic. In Chapter 3, I focus on understanding how high-dimensional datasets behave with respect to their Euclidean properties, such as their mutual distances. Indeed, most real world datasets live in extremely high-dimensional spaces, where concentration properties constrain the allowed geometries. Finally, in Chapter 4, I present some results that generalize classical expressivity studies of linear classifiers in order to take into account a particular form of geometric structure, inspired by concepts from neurobiology.

1.2 Organization and main results

This Thesis is organized in three self-contained chapters that can be read independently. Each chapter presents some novel results obtained by my collaborators and myself in a pedagogical way, introducing broadly the matter before delving into the details. Many technical details that are not fundamental to follow the discussion are enclosed in grey boxes: feel free to skip them if you are not interested in technicalities!

The main theme is the Euclidean structure of datasets, declined in different ways from chapter to chapter.

Chapter 2 focuses on the problem of intrinsic dimension estimation, i.e. the problem of estimating how many degrees of freedom are really necessary to encode an high-dimensional dataset. Intrinsic dimension estimation can be performed using a long list of algorithm, and features many possible applications in Physics and Computer Science, but its limitations hinder the usability of many of the results. With Marco Gherardi and Pietro Rotondo, we proposed a novel estimator [EGR19] that tries (and in some cases succeeds) to overcome these limitations. Chapter 2 motivates the estimation problems in the more general setting of manifold learning, and presents some simple, yet paradigmatic estimators to equip the reader with some basics on the subject. It then presents the novel estimator in detail.

Chapter 3 focuses on high-dimensional random geometric graphs, i.e. graphs whose nodes are connected only if they are near enough in some high-dimensional ambient space. The community is still puzzled by a seemingly simple question: in high-dimension, where concentration properties greatly affect the geometry of sets of points - forcing them to be all roughly orthogonal and equidistant - are geometric graphs distinguishable from common random graphs, with no underlying Euclidean structure? With Sebastiano Ariosto, Marco Gherardi and Pietro Rotondo, we explored this question through the lenses of a set of local observables, k-cliques densities, and found that at least in some cases the answer to the previous question is no, with the Euclidean structure of the graph still detectable through local observables [EAG+20] even in infinite dimension. Chapter 3 frames the problem in the language of random graphs, before delving into central limit theorems and simulations to motivate our findings.

Chapter 4 focuses on machine learning, and more precisely on how the geometric structure of datasets alters the properties of simple classification tasks. This work fits nicely into a broader and very active line of research investigating the implicit biases that structured data impose onto learning architectures such as neural networks and support vector machines. With Marco Gherardi, Mauro Pastore and Pietro Rotondo, we studied a simple model of structured learning inspired by biological considerations, and found that data structures greatly affects the expressivity of linear classifiers [PRE+20]. Data structure introduces also a new phase-transition which appears to be of universal character. Chapter 4 starts with a broad introduction to supervised learning problems, detailing commonly used models of data structure and measures of expressivity for binary classifiers. It then presents in detail the novel results.

1.3 What I left out

I would like to spend a couple of phrases on other projects and results that contributed to my Ph.D. experience, but didn't make it to this Thesis. The least common denominator of these projects is again given by Euclidean correlations in disordered systems, but the flavor was far away enough from the applications to data science and machine learning to justify their absence from this Thesis.

A big part of my Ph.D. has been dedicated to the study of Euclidean random combinatorial optimization problems, i.e. optimization problems that depend on the random position of points in some Euclidean space. In particular, I studied the concave-cost 1d random Euclidean bipartite matching problem, i.e. the problem of finding a minimum assignment between N white and N black points on the line, with cost function depending on the distance between points as $c(x) = x^p$, $0 < p < 1$. The analytical structure of this problem is very rich, and bears connections with lattice paths (Dyck paths) and unusual probability distributions (the area-Airy distribution). In a series of works with Sergio Caracciolo, Matteo D'Achille and Andrea Sportiello [CDE+20; CES20; CES21] we investigated the structural properties of the optimal solution to this problem with techniques from analytic combinatorics.

More recently, with Mauro Pastore and Pietro Rotondo we investigated analytically and numerically a deterministic spin model that features a glassy-like phase without explicit quenched disorder [EPR21]. The glassy phase is induced by the peculiar Euclidean form of the interaction. This model has direct experimental applications, as it models the interaction of Bose-Einstein condensates in confocal electromagnetic cavities.

1.4 Chapter bibliography

[BBC+21] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: grids, groups, graphs, geodesics, and gauges, 2021. arXiv: 2104.13478 [cs.LG].

[CDE$^+$20] Sergio Caracciolo, Matteo P. D'Achille, Vittorio Erba, and Andrea Sportiello. The dyck bound in the concave 1-dimensional random assignment model. *Journal of Physics A: Mathematical and Theoretical*, 53(6):064001, January 2020. DOI: 10.1088/1751-8121/ab4a34.

[CES20] Sergio Caracciolo, Vittorio Erba, and Andrea Sportiello. The p-airy distribution, 2020. arXiv: 2010.14468 [math.CO].

[CES21] Sergio Caracciolo, Vittorio Erba, and Andrea Sportiello. The number of optimal matchings for euclidean assignment on the line. *Journal of Statistical Physics*, 183(1):3, March 2021. ISSN: 1572-9613. DOI: 10.1007/s10955-021-02741-1.

[CLS16] SueYeon Chung, Daniel D. Lee, and Haim Sompolinsky. Linear readout of object manifolds. *Phys. Rev. E*, 93:060301, 6, June 2016. DOI: 10.1103/PhysRevE.93.060301.

[EAG$^+$20] Vittorio Erba, Sebastiano Ariosto, Marco Gherardi, and Pietro Rotondo. Random geometric graphs in high dimension. *Phys. Rev. E*, 102:012306, 1, July 2020. DOI: 10.1103/PhysRevE.102.012306.

[EGR19] Vittorio Erba, Marco Gherardi, and Pietro Rotondo. Intrinsic dimension estimation for locally undersampled data. *Scientific Reports*, 9(1):17133, November 2019. ISSN: 2045-2322. DOI: 10.1038/s41598-019-53549-9.

[EPR21] Vittorio Erba, Mauro Pastore, and Pietro Rotondo. Self-induced glassy phase in multimodal cavity quantum electrodynamics. *Phys. Rev. Lett.*, 126:183601, 18, May 2021. DOI: 10.1103/PhysRevLett.126.183601.

[GMK$^+$20] Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová. Modeling the influence of data structure on learning in neural networks: the hidden manifold model. *Phys. Rev. X*, 10:041044, 4, December 2020. DOI: 10.1103/PhysRevX.10.041044.

[GTR$^+$18] Robert Geirhos, Carlos R. M. Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. Generalisation in humans and deep neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/0937fb5864ed06ffb59ae5f9b5ed67a9-Paper.pdf.

[PRE$^+$20] Mauro Pastore, Pietro Rotondo, Vittorio Erba, and Marco Gherardi. Statistical learning theory of structured data. *Phys. Rev. E*, 102:032119, 3, September 2020. DOI: 10.1103/PhysRevE.102.032119.

[RLG20] Pietro Rotondo, Marco Cosentino Lagomarsino, and Marco Gherardi. Counting the learnable functions of geometrically structured data. *Phys. Rev. Research*, 2:023169, 2, May 2020. DOI: 10.1103/PhysRevResearch.2.023169.

CHAPTER 2

Intrinsic dimension estimation

Intrinsic dimension estimation is the art of guessing how much geometrical information is contained in a cloud of points. A quick example: a set of aligned points in 300 dimension could be easily described by a single parameter, as it lays on a one-dimensional manifold, while points sampled from the 300-dimensional hypercube need 300 parameters to be described. This information is crucial in order to properly compress data, to study high-dimensional datasets (coming from large scale experiments, for example) and to apply dimensionality reduction techniques for visualization purposes.

In this chapter, I will guide you into the basic methods and assumptions of intrinsic dimension estimation. I will then present a novel estimation framework [EGR19].

Contents

2.1 Manifold learning

2.1.1 The need for manifold learning

You are given a dataset of 100 points in 3 dimensions, that is, a set of 100 vectors in $\mathbb{R}^3$. You have no prior knowledge about this dataset, and I ask you to obtain some insight on it (nothing specific, just tell me something about this bunch of vectors). What would you do? Well, if I were you, I would start by plotting the dataset to take a good look at it[1]. At this point, we may realize that the points lie (for example) on the surface of a sphere, and suddenly we know that we may gather very useful information on the dataset by estimating the radius of the sphere, or something along this line. Visualization is crucial in order to study datasets. It allows researchers to use one of the best pattern recognition tools, their own vision, to come up with good observables to study and to spot general trends in the data. Conclusions, or new theories, then follow easily (well, sort of).

But if you are reading this boring Ph.D. thesis, I guess you wouldn't be very impressed by my small, three-dimensional dataset. To level up the game, you may want to try the same task on a more complex dataset, maybe composed by 60000 points in 784 dimensions (these numbers are not random, they come from the MNIST [LCB10] dataset, a famous benchmark dataset of images of handwritten digits). Well, this is much harder, as we have no idea how to plot sets of 784-dimensional points! Thus, in order to study high-dimensional dataset we have to come up with a set of analytical and computational tools that replace our own vision in performing qualitative analysis on the data.

This is, at its core, the motivation for *manifold learning*, a set of techniques that allow the user to gain some intuition on the datasets they have to study. To give you a flavor of what manifold learning can do, here you have some examples.

Intrinsic dimension estimation More often than not, datasets are represented in a redundant fashion. In low dimension, you may have three-dimensional data that lay along a curved line. In that case, two of the three dimensions are effectively spurious, unneeded to fully describe the dataset. To give you a less trivial example, think of a dataset of images of flying eagles. Most of the pixels representing the sky will be colored with a uniform, bluish color, and will be completely uninformative. In both the examples, we would like, as a first step, to estimate the minumum number of degrees of freedom (dimensions, pixels) that we need to fully encode the variability inside the dataset. This is typically called *intrinsic dimension estimation.* I will give you a more detailed look at it in what follows, as this is the main topic of the chapter.

Dimensionality reduction After gaining some intuition on the number of degrees of freedom actually needed to describe the dataset, you may want to estimate which of the dimensions of your dataset are the important ones, or which of the pixels of your images are the most informative. To get back to the trivial three dimensional example described in the previous point, you may want to highlight a point along your line to serve as an origin and give a notion of distance along the line so that each of the points in your dataset

[1] This was actually one of the very first pieces of mathematical advice that I have ever got. When you are given some mathematical object to study, you should always ask: "How do I draw it?". Thanks Prof. Palazzi!

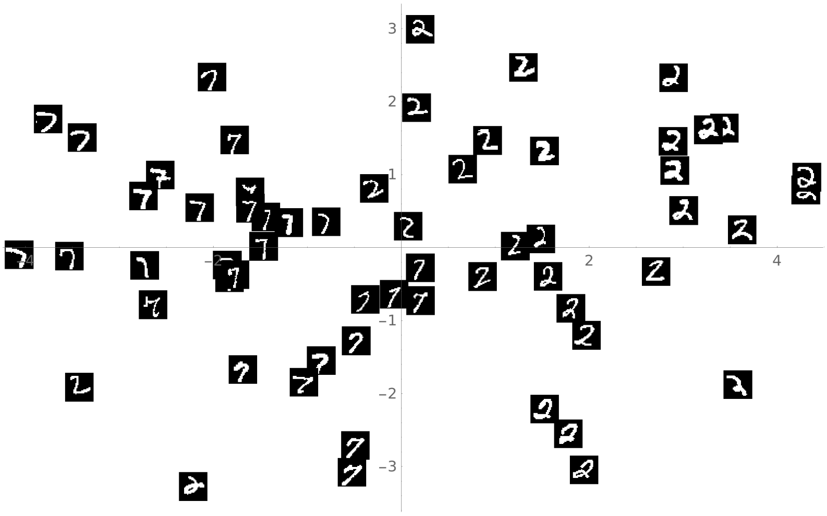


Figure 2.1: Example of dimensionality reduction. The plot is a two-dimensional representation of a sample of 60 images taken from MNIST, 30 representing the digit 2 and 30 representing the digit 7. The reduction has been performed using the t-SNE algorithm [vdMH08]. In the low dimensional representation, we see that images representing the two digits are basically linearly separable, with the exception of a couple of images of 2's in the bottom left portion of the plot. Notice how similar images are projected next to one another.

can actually be represented by giving just one number, its distance from the origin along the line. This task, that is finding low-dimensional coordinates for your dataset, is called *dimensionality reduction*.

In the following, I will give you some details about the most famous dimensionality reduction technique, which is Principal Component Analysis (in short, PCA), as it combines dimensionality estimation (see above) and dimensionality reduction. As you may imagine, there exists a whole world of techniques that go well beyond PCA. If you are interested, here are some good starting points to get into the literature [Row00; Ten00; BML06; LV07; vdMH08].

Figure 2.1 shows an example of dimensionality reduction applied to the MNIST dataset.

Clustering On a different line, you may want to know if your data is *clustered*, i.e. it is composed by multiple different clouds of points. For example, think about MNIST, the dataset of handwritten digits. You may have the intuition that the images representing ones are well separated by the imaged representing zeros in the space of pixels, as they mostly use different pixels (the ones use pixels around the vertical mid line of the image, the zeros use pixels distributed over a circle). The task of recognising that your data is clustered and of giving the rules that identify that a datapoint is in one cluster or in another is called, guess what, *clustering*. Again, there are many techniques to cluster data; see [SPG$^+$17] for a thorough review.

Figure 2.2 shows an example of clustering on a dataset of points extracted by a mixture of Gaussian distributions.

A word of caution is needed here. You may now think that manifold learning solves

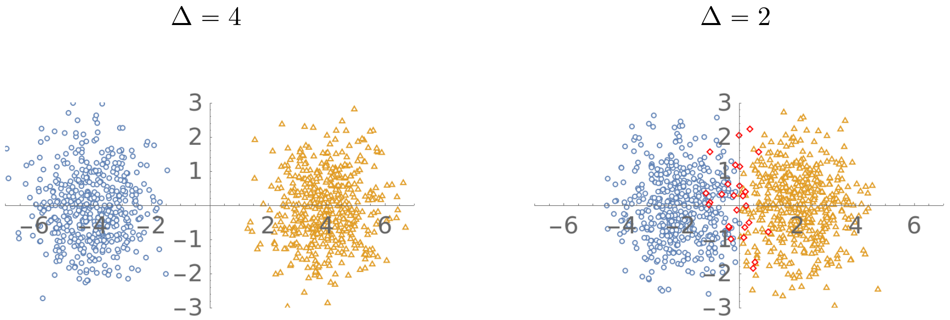


Figure 2.2: Example of clustering. The plot represents the result of the k-means clustering algorithm [Ste56; Llo82] applied to a dataset of points sampled from two different $2d$ Gaussians (with mean $(\pm\Delta, 0)$ and identity covariance). Blue and orange points are points that the clustering algorithm correctly classifies as belonging to the left or right Gaussian cluster. Red points are misclassified points. We see that as Δ gets smaller we get more and more misclassified points.

our visualization problems in high dimensions, allowing us to get a full overview of the geometrical properties of a dataset just as our vision does in low dimensions. This is sadly not completely true. Often, manifold learning techniques are proven to be effective under a restrictive set of assumptions on the dataset, assumptions that in practice one cannot explicitly verify or that are explicitly violated. Even worse, most of the tools may only be benchmarked on synthetic datasets (i.e. artificially generated datasets whose properties are known by construction). Let me give you an example. Suppose that you have to benchmark an intrinsic dimension estimation algorithm. Of course, you can verify its performance on a dataset of points that lay on the surface of a d-dimensional hypersphere: the algorithm works only if it returns the dimension of the sphere, d. But how can you measure its performance on real datasets, for which the intrinsic dimension is not known *a priori*?

Thus, my personal advice if you would like to do some manifold learning is to use and compare many different tools (for example, different intrinsic dimension estimators) on the same task, and only moderately trust their results.

2.1.2 Some applications of intrinsic dimension estimation

Before delving into the mathematical details of dataset modelling and intrinsic dimension estimation, I would like to give a brief overview of the possible applications of this kind of estimation.

The main application of intrinsic dimension estimation algorithms in the early days (late 1970's) was the characterization of the behaviour of dynamical systems. At the time, chaotic behaviour[2] had just been discovered, along with the concept of strange attractors, i.e. fractal subsets of a dynamical system's phase space around which the orbits described by the evolution of the system may converge. Measuring the fractal dimension of these attractors was a very relevant problem that motivated the introduction of the first intrinsic dimension estimators [Tru68; FO71; GP83a; GP83b].

[2] See [Str94] for a wonderful beginner introduction to the subject.

Around the same time, another problem gained wide interest, i.e. that of determining whether a time series is stochastic or is deterministic in a chaotic regime. Takens's embedding theorem and the related line of works [Tak81; Tak85; SM90; Cao97] provide a way of answering this question if the intrinsic dimension of a peculiar high-dimensional representation of the time series can be measured accurately. Among the direct applications, one could then distinguish whether meteorological or financial time series depend on a stochastic component or not.

More recently, intrinsic dimension estimators moved further and further away from particular applications and became a general purpose tools for data analysis. To mention some results, intrinsic dimension estimation has been used to track the compression effect of hierarchic structure in deep neural networks [ALM+19], to data-mine physical systems in order to automatically detect phase transitions [MTD+21] and characterize quantum partition functions [MAR+21], to characterize the variability of observed protein sequences [FPR+19] and to devise novel clustering algorithms [AFD+20].

See [CRH10] for more examples.

2.1.3 A model of dataset

In order to start developing manifold learning algorithms we need a mathematical model of what a dataset is. In this section I want to introduce a common model, the so called *manifold model* (hence the name manifold learning), that we will use throughout the chapter. It is a constructive model, i.e. a model that characterizes datasets by telling us how they were generated in the first place.[3]

The first assumption is that there exists a smooth d-dimensional manifold, the *intrinsic manifold*, equipped with a probability measure. The intrinsic manifold describes the intrinsic geometry of the dataset, i.e. the minimal information that we would need to reproduce the dataset. Of course, there is no guarantee that when we are given a dataset it is represented in the most efficient way. We model our own representation of the intrinsic data, as well as possible data corruption, by assuming that there exists a sufficiently regular map (say, smooth and injective) from the intrinsic manifold into D-dimensional Euclidean space that we will call *embedding map*; here $D \geqslant d$. Our dataset will then be generated by sampling i.i.d. points on the intrinsic manifold according to the given probability distribution, and mapping each point into Euclidean space using the embedding map. To fix some nomenclature, d is the intrinsic dimension of the dataset, D is the embedding dimension.

Let me try to justify why the manifold model may be a good model for datasets. To fix our ideas, let me use the MNIST dataset as an example, and let me focus on the subset of handwritten "ones". The idea that there exists an intrinsic manifold underlying this dataset comes from the intuition that there is a prototype image of a "one", and that all other images can be generated from the first one by smooth transformations in intrinsic space that translate to rotations, translations, rescalings and complicated local distortions in the pixel space. The intrinsic manifold is generated by this set of meaning-preserving transformations. In this picture, it is clear that the intrinsic manifold will

[3] Disclaimer: this section mentions a fair deal of concepts borrowed from differential geometry. I will not enter into the details, and I will be sloppy in many places. For example, I will use the word "embedding" in a non-technical sense, even though it has a very precise meaning in differential geometry. See [PM13] for a more precise treatment.

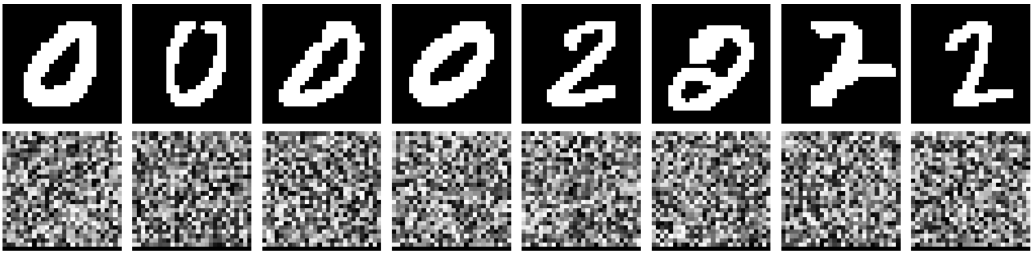


Figure 2.3: Comparison between MNIST images and random images. Images carrying semantical informations, like the images of handwritten digits of the MNIST dataset shown in the top row, lie on a low-dimensional intrinsic manifold generated by smooth transformations that preserve meaning. For comparison, the bottom row shows random samples from the high-dimensional space of 28 × 28 bitmap images.

have a dimension smaller then the number of pixel used to represent the images, so that there must be an embedding map linking the intrinsic representation and the embedded one. The embedding map may depend on how we decide to represent that data (fixing the resolution of the images is a trivial example), and may contain statistical noise corrupting our data. See Figure 2.3.

I want to highlight an important fact. In this model of dataset, we have three sources of complexity: the intrinsic manifold, which can be a non-trivial manifold, the probability distribution associated to the intrinsic manifold, which may be highly non-uniform, and finally the embedding map, which again may be complex. I will often restrict my attention to simple probability distributions, as highly non-uniform distribution denature the intrinsic geometry of the dataset in practice. On the other hand, we will see examples of non-trivial intrinsic manifolds and embeddings.

Notice that the manifold model can be trivially extended to account for *multidimensional datasets*, i.e. union of datasets with different intrinsic dimensions. Each component of the multidimensional dataset is described by the manifold model separately, but with a common embedding space.

Examples of synthetic datasets

See Figure 2.4 for the graphical representation of some of the following datasets.

Linear datasets The intrinsic manifold is a full-dimensional subset of $\mathbb{R}^d$ equipped with the uniform probability distribution. The most common example is given by the d-dimensional hypercube $\mathcal{H}_d$, i.e. $[-1,1]^d$. The embedding map is a linear map from $\mathbb{R}^d$ to $\mathbb{R}^D$, such as the natural inclusion map, or the inclusion map followed by a rotation. A noisy version of the dataset can be considered, where after embedding the dataset we corrupt it with an Gaussian noise (factorized over each coordinate) with variance σ^2.

Digital dataset The intrinsic manifold is the set of vertices of the d-dimensional hypercube equipped with the uniform probability distribution. The embedding map is a linear map from $\mathbb{R}^d$ to $\mathbb{R}^D$.

Spherical datasets The intrinsic manifold is a d-dimensional hypersphere, i.e. $\mathcal{S}_d = \{x \in \mathbb{R}^{d+1} \mid ||x|| = 1\}$, equipped with the uniform probability distribution. The embedding

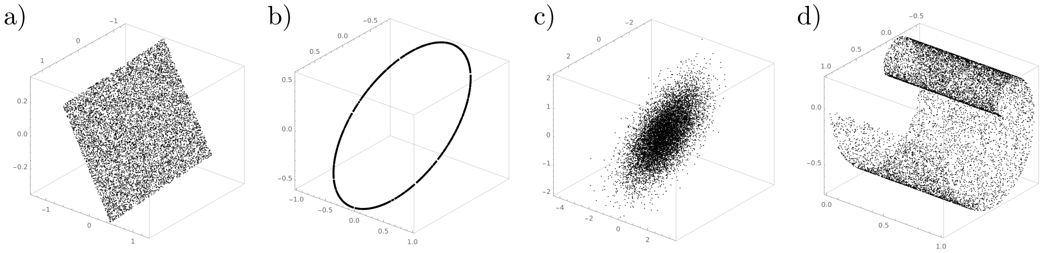


Figure 2.4: Examples of synthetic datasets. (a) A $d = 2$ linear dataset embedded in $D = 3$ dimensions. (b) A $d = 1$ spherical dataset embedded in $D = 3$ dimensions. (c) A $d = 2$ Gaussian dataset embedded in $D = 3$ dimensions. (d) A $d = 2$ swiss roll dataset embedded in $D = 3$ dimensions. In all cases, $P = 10000$ points are shown.

map is the composition of the natural inclusion of $\mathcal{S}_d$ into $\mathbb{R}^{d+1}$ and of a linear map from $\mathbb{R}^{d+1}$ to $\mathbb{R}^D$.

Gaussian datasets The intrinsic manifold is $\mathbb{R}^d$ equipped with the standard multivariate Gaussian measure, i.e.

$$p_{\text{Gauss}}(x) = \frac{e^{-\frac{1}{2}||x||^2}}{(2\pi)^{\frac{d}{2}}} \,. \tag{2.1}$$

The embedding map is a linear map from $\mathbb{R}^d$ to $\mathbb{R}^D$.

The swiss roll dataset The intrinsic manifold is the two-dimensional square $[0, 1]^2$ sampled with uniform probability. The embedding map is

$$\phi(x, y) = (x \cos(2\pi y), y, x \sin(2\pi y)) \,. \tag{2.2}$$

The name is a reference to a cake[4] typical of Central Europe.

The Hein dataset This is a peculiar dataset first introduced in the literature by [HA05], and is characterized by a non-trivial extrinsic curvature. The intrinsic manifold is the d-dimensional hypercube $[0, 2\pi]^d$ sampled with uniform probability. The embedding map is

$$\phi(x_1, \ldots, x_d) = (x_2 \cos(x_1), x_2 \sin(x_1), \ldots, x_1 \cos(x_d), x_1 \sin(x_d)) \,, \tag{2.3}$$

and the minimal embedding dimension is $D = 2d$. Higher embedding dimensions can be considered by using an additional linear embedding map from $\mathbb{R}^{2d}$ to $\mathbb{R}^D$.

[4] see https://en.wikipedia.org/wiki/Swiss_roll.

2.2 Intrinsic dimension estimation

I hopefully have convinced you that manifold learning is a useful tool, and that the manifold model provides us with a variegated playground of synthetic datasets. We are now ready to get into the details of intrinsic dimension estimation. Let me state the problem for synthetic datasets, where the framework is clear: I give you a set of P points in $\mathbb{R}^D$ generated with the manifold model, with unknown intrinsic manifold and embedding map. The intrinsic dimension estimation task is to recover the value of the intrinsic dimension d. The idea is that when we have an estimator that works reliably on synthetic datasets, we can apply it to real datasets with some confidence that it will provide a reasonable intrinsic dimension estimation.

2.2.1 Two paradigmatic estimators: CorrDim and PCA

To give you an idea on how intrinsic dimension estimation works in practice, and for which reasons it typically fails, I will give you a detailed overview of two simple estimators: Correlation Dimension (CorrDim) and Principal Component Analysis (PCA). I chose to get into the details of CorrDim and PCA not only because they are simple to understand and to numerically implement, but also because they represent well the two main classes of estimators: CorrDim is a representative of the so called *geometric* estimators, while PCA represents the *projective* estimators. I will come back to this classification later. Finally, these estimators suffer from complementary drawbacks.

CorrDim and the curse of dimensionality

The CorrDim estimator was first introduced in the literature by Grassberger and Procaccia in the eighties [GP83b] in order to measure the fractal dimension of strange attractor of dynamical systems exhibiting chaotic behaviour. Let me explain how it works.

I will denote our dataset as $X = \{x^\mu\}_{\mu=1}^P$, where each $x^\mu \in \mathbb{R}^D$. The idea is to consider the normalized density of neighbours of each point at scale r, namely

$$\rho_\mu(r) = \frac{1}{P-1} \sum_{\substack{\nu=1 \\ \nu\neq\mu}}^{P} \theta(r - ||x^\mu - x^\nu||) \,, \tag{2.4}$$

where $\theta(x)$ is the Heaviside step function, that equals 1 if its argument is non-negative and 0 otherwise. This observable measures the fraction of points other than x^μ that are within a distance r from x^μ. Why should we do this? Because we expect that the scaling of $\rho_\mu(r)$ at small scales r probes the local geometric structure of the dataset, i.e. the embedded intrinsic manifold.

If the nearest neighbours of the μ-th point are almost uniformly distributed on the surface of a d-dimensional submanifold, we can approximate the density $\rho_i(r)$ by considering x^μ to be the center of a d-dimensional disk (the sphere plus the volume it bounds), and the nearest neighbours as uniformly drawn points in the disk. Both these approximations are valid for small values of r if the intrinsic manifold, the probability distribution over the intrinsic manifold and the embedding map are smooth. In fact: (i) smooth manifold are locally linear, i.e. they are well approximated by their tangent spaces; (ii) smooth probability distributions can be Taylor-expanded, and at zeroth order they are constant; (iii) smooth embedding maps can be again Taylor-expanded, and at first order they map linear spaces into linear spaces (non-trivially if they are injective).

Thus, for $r \ll 1$, we have that

$$\rho_\mu(r) = \int dy \, p_{\text{Disk}}(y) \, \theta(r - ||y||) \tag{2.5}$$

where $p_{\text{Disk}}(y)$ is the uniform measure on a d-dimensional disk with unit radius. Here we are implicitly assuming that $\rho_\mu(r)$ is self-averaging, which is true for large number of samples P. By moving to spherical coordinates, we obtain

$$\rho_\mu(r) \propto r^d \,. \tag{2.6}$$

We see that the local number of neighbours depends on r through a power-law behaviour with exponent given by the intrinsic dimension d.

We could average the normalized density of neighbours over all points in the dataset in order to wash out non-uniformities, obtaining the so called *correlation integral*

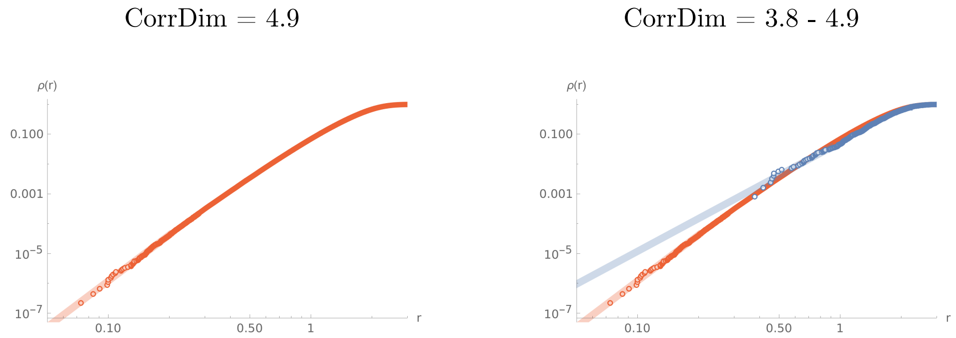
$$\rho(r) = \frac{2}{P(P-1)} \sum_{1 \leqslant \mu < \nu \leqslant P} \theta(r - ||x^\mu - x^\nu||) \propto r^d \,, \tag{2.7}$$

where to write the final scaling we supposed that around each point x^μ the intrinsic manifold has the same intrinsic dimension (this may be false if one considers a dataset built as a union of two different embedded intrinsic manifolds, i.e. multidimensional datasets). Again, this derivation holds locally, i.e. for small values of r, and for a sufficiently large number of local neighbours.

Now, it's easy to come up with an intrinsic dimension estimator: compute the correlation integral $\rho(r)$ on your dataset and fit it at small distances with a power-law behaviour to extract the intrinsic dimension d. The fit is easily done by considering the log-log version of the correlation integral, namely $\rho_{\log}(r') = \log \rho(e^{r'})$, as in log-log scale the fit becomes linear and the intrinsic dimension can be recovered as the slope of the interpolating line. Refer to Figure 2.5, left panel, for an example.

It seems that CorrDim is a very simple, yet very powerful estimator. Indeed, it ignores all the complexities of the intrinsic manifold and of the embedding map by exploiting the local limit. You may be wandering if it has any pitfalls. CorrDim has a major drawback: it heavily relies on the availability of data points at small distances. In other words, it needs densely populated local neighbourhoods in order to work. While this may seem an innocuous request in low dimensions, it quickly becomes a problem when the intrinsic dimension of the dataset grows. Imagine that you are sitting on the barycenter of a d-dimensional volume. To sample all directions, you would need to throw some points in front of you and behind you, at your left and your right, above and below you, and so on for all the d dimensions. Thus, to minimally sample a d-dimensional volume, you would need roughly a number of data points that scales exponentially with the intrinsic dimension d. This phenomenon is known as *curse of dimensionality* (see Figure 2.6).

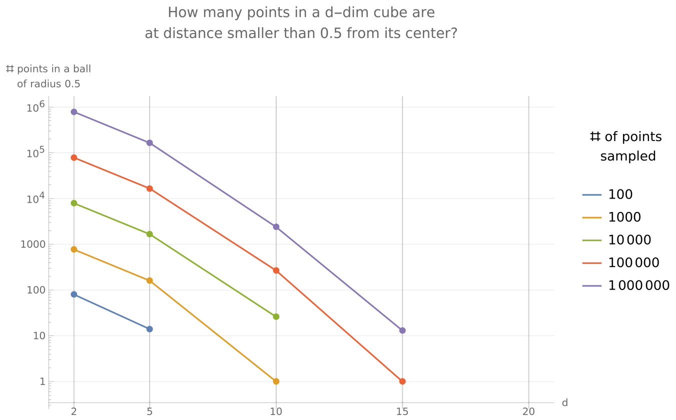
In practice, the curse of dimensionality results in a systematic underestimation issue for the CorrDim estimator that takes place starting at intrinsic dimension $d \sim 6$ for $P \sim 10^3$ samples [ER92]. See Figure 2.5, right panel, for a minimal example of this underestimation effect.



CorrDim = 4.9

CorrDim = 3.8 - 4.9

Figure 2.5: Example of CorrDim estimation. (Left) CorrDim intrinsic dimension estimation of a linear dataset with intrinsic dimension $d = 5$, embedding dimension $D = 20$ and $P = 3000$ samples. Open markers denote the empirical correlation integral of the dataset (in log-log scale), and the shaded line is a linear fit of the first 200 points. CorrDim estimates the intrinsic dimension as the slope of this linear fit, which in this case equals 4.93. (Right) Same estimation task as before. Red markers refer to a dataset with $P = 3000$ samples while blue markers refer to a dataset with $P = 50$. CorrDim correctly estimates the intrinsic dimension in the $P = 3000$ case (CorrDim = 4.9), but it underestimates it in the $P = 50$ case (CorrDim = 3.8) due to undersampling of the intrinsic manifold.



How many points in a d–dim cube are at distance smaller than 0.5 from its center?

Figure 2.6: Example of curse of dimensionality. High dimensional datasets require an exponential number of samples in order to be accurately represented. The plot (linear-log scale) shows the number of points falling withing distance 0.5 from the center of an hypercube as a function of the dimension of the hypercube for different values of total number of i.i.d. points sampled on the hypercube. The number of points in the neighbourhood of the origin decays exponentially fast as the intrinsic dimension grows.

To get a rough idea of the effect of the curse of dimensionality on the CorrDim estimation, we follow the concise argument of [ER92]. Two conditions must be met in order for CorrDim to be effective:

- the distances r to be used in the estimation procedure must be much smaller than the diameter δ of the intrinsic manifold in the embedding space, i.e. $\frac{r}{\delta} \ll 1$;
- the distances r must be sufficiently large in order for the number of pairs of samples at distance smaller than r, let's call it $N(r)$, to be large. This is necessary in order to avoid statistical fluctuations. The condition thus is given by $N(r) \gg 1$.

But

$$N(r) \sim \frac{P^2}{2}\left(\frac{r}{\delta}\right)^d \tag{2.8}$$

where we used the power-law behaviour of the correlation integral, and the fact that at scales $r \sim \delta$ all pairs of points are at distance lesser than r, i.e. $N(\delta) \sim \frac{P^2}{2}$. The second condition implies

$$d \lesssim \frac{2\log P}{\log\left(\frac{\delta}{r}\right)} \,. \tag{2.9}$$

and the first requires that $\frac{\delta}{r} \gg 1$. Using $\frac{\delta}{r} = 10$ and $P = 10^3$ gives $d \lesssim 6$ as announced previously. Moreover, this rough bound makes it clear that to estimate large intrinsic dimension an exponential number of points P is required.

PCA and curvature

PCA is not directly, or at least not only, an intrinsic dimension estimator. PCA answers a more general question: what is the most convenient choice of orthonormal basis to study my dataset? The underlying idea is that by "aligning" some of the axis of an orthonormal basis to the directions in which dataset varies the most, we could hopefully express our dataset using a smaller set of coordinates (dimensionality reduction) or we could try to interpret each axis as a hidden feature of our dataset (responsible for a large variance).

I will state the result first: the optimal basis to encode the variance of the dataset is given by the basis of eigenvectors of the empirical covariance matrix $C = \frac{1}{P}X^T X$, where X is the $P \times D$ matrix obtained by stacking vertically the P row vectors representing the data points, and under the assumption that the dataset is centered, i.e. $\sum_{\mu=1}^{P} x^\mu$ is the null vector. Along each of the axes of the basis, the dataset has variance equal to the corresponding eigenvalue of the covariance matrix. Let me explain why this is the case.

As a first step, we look for the direction along which the projection of the dataset has maximum variance, i.e.

$$w^{(1)} = \underset{\substack{w\in\mathbb{R}^D \\ \text{s.t. } ||w||=1}}{\arg\max} \left[\frac{1}{P}\sum_{\mu=1}^{P}(w \cdot x^\mu)^2\right] = \underset{\substack{w\in\mathbb{R}^D \\ \text{s.t. } ||w||=1}}{\arg\max} \left[w^T C w\right] . \tag{2.10}$$

Notice that C is both symmetric

$$C_{i,j} = \frac{1}{P}\sum_{\mu=1}^{P} x_i^\mu x_j^\mu \tag{2.11}$$

and positive semi-definite

$$w^T C w = \frac{1}{P}\sum_{\mu=1}^{P} (x^\mu \cdot w)^2 \geqslant 0 \quad \text{for all} \quad w \neq 0\,. \tag{2.12}$$

Getting back to Equation (2.10), we can diagonalize C as $C = U^T \Lambda U$, with U orthogonal and Λ diagonal (with non-negative eigenvalues due to positive semi-definiteness), obtaining

$$\begin{aligned} w^{(1)} &= \underset{\substack{w\in\mathbb{R}^D \\ \text{s.t. } ||w||=1}}{\arg\max} \left[(Uw)^T \Lambda (Uw)\right] = U^T \underset{\substack{y\in\mathbb{R}^D \\ \text{s.t. } ||y||=1}}{\arg\max} \left[y^T \Lambda y\right] \\ &= U^T \underset{\substack{y\in\mathbb{R}^D \\ \text{s.t. } ||y||=1}}{\arg\max} \left[\sum_{i=1}^{D} \lambda_i y_i^2\right] \end{aligned} \tag{2.13}$$

where λ_i are the eigenvalues of C sorted in decreasing order, and where we used the orthogonality of U to bijectively change variables from w to y. It is now easy to see (for example by Lagrange constrained optimization, or by bounding $y^T \Lambda y \leqslant \lambda_1 ||y||^2 = \lambda_1$ and showing that $y_1 = 1$ saturates the bound) that the maximum is obtained when $y_1 = 1$ and all other coordinates are zero, i.e. when $w^{(1)}$ is precisely the eigenvector of C corresponding to its largest eigenvalue.

We can now look for a new direction $w^{(2)}$, orthogonal to $w^{(1)}$, that encodes the maximum residual variance. This is easily achieved considering a modified covariance matrix obtained by subtracting from C the projector onto its first eigenspace, i.e. $\lambda_1 w^{(1)} (w^{(1)})^T$. The new covariance matrix has the same spectral structure of the original matrix, with the replacement $\lambda_1 \to 0$, allowing us to repeat the above procedure in order to find $w^{(2)}$.

By iteration, this gives that the optimal orthonormal basis is given by the basis of eigenvectors of the covariance matrix C, and the variances of the dataset along the axis of the basis are given by the eigenvalues of C.

Notice that we implicitly assumed that C is non degenerate, which we expect to be the case at least in the finite P case; the degenerate case can be treated analogously nonetheless.

Now we know how to find a convenient basis for the study of our dataset, and we know the magnitude of the variances of the dataset along these axes. How can we use this information to estimate the dimensionality of the dataset? The idea is to sort the directions of variance by decreasing eigenvalue, i.e. decreasing contribution to the variance of the dataset, and to consider the directions *relevant* up until the residual variance is less than 5% (for example) of the total. The intrinsic dimension is then estimated as the number of relevant directions (see Figure 2.7, blue plot). Typically, the spectrum of the empirical correlation matrix features clear jumps in the magnitude of the eigenvalues.

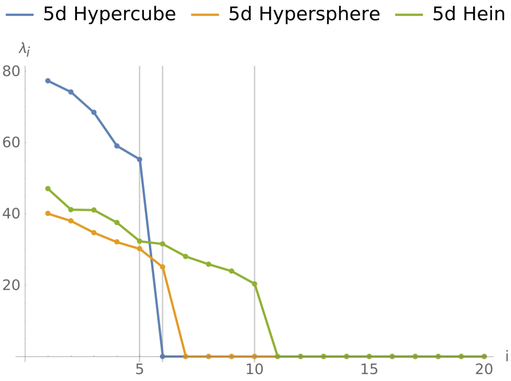


Figure 2.7: Example of PCA estimation. The plot represents the eigenvalues of the empirical correlation matrix of three datasets: (blue) a linear dataset, (orange) a spherical dataset, (green) a Hein dataset. In all cases, the intrinsic dimension is $d = 5$, the embedding dimension is $D = 20$ and the number of sampled points is $P = 200$. The PCA-estimated intrinsic dimensions are, respectively, $d = 5, 6, 10$, and are obtained by looking for sharp jumps in the magnitude of the eigenvalues (in this case, this is equivalent to the residual variance criterion explained in the main text). On linear datasets, PCA easily recovers the correct intrinsic dimension. On curved datasets, PCA overestimates the intrinsic dimension.

This aids the intrinsic estimation procedure, and grants that the estimation is stable against modifications of the residual variance threshold, which is somewhat arbitrary otherwise.

Again, you could ask: is there any drawback that I should be aware of? PCA has a severe limitation: it is a linear method, in the sense that it intrinsically reasons in terms of orthonormal bases and linear subspaces. If two coordinates are correlated, think for example to a circle in two dimensions, PCA will consider both dimensions as relevant, as they both contribute significantly to the variance of the dataset. Yet, as the two coordinates are correlated, they could in principle be described by a single parameter, resulting in an overestimation of the intrinsic dimension computed using PCA. In general, non-linear intrinsic manifolds and non-linear embedding maps will contribute to this overestimation effect (see Figure 2.7, orange and green plots). On the other side, PCA is not hindered by large intrinsic dimension as CorrDim is: it only needs roughly $\sim d \log d$ points [LMR17], independently on the dimension D, in order to reliably estimate the correlation matrix.

A further drawback that has to be highlighted is that, contrary to the examples shown in Figure 2.7, the empirical covariance matrix may not have any large jump in the magnitude of its eigenvalues. If this is the case, the 5% criterion on the residual variance becomes somewhat arbitrary, and may be subject to interpretation.

2.2.2 A brief overview of intrinsic dimension estimators

CorrDim and PCA are simple intrinsic dimension estimators, yet they are paradigmatic of more advanced algorithms. CorrDim is the prototype of the so-called *geometric* estimators, that combine the local limit (looking at small neighbourhoods) with the measurement of smart observables, which are explicitly computable in the local limit, and explicitly dependent on the intrinsic dimension d. PCA, on the other hand, is the prototype for the *projective* estimators, that look for useful decompositions or representation of the embedding space in order to highlight relevant and irrelevant directions.

In general, all geometric estimators suffer from the curse of dimensionality (severe undersampling in high dimension) and need exponentially many points (in the intrinsic dimension) to work properly, while projective estimators typically rely on matrix decompositions/factorizations, and need the number of points to scale roughly linearly with the intrinsic dimension. On the other hand, projective estimators systematically overestimate the intrinsic dimension of curved or non-linearly embedded intrinsic manifolds, while geometric estimators avoid all this complexity by exploiting the local limit.

In this section, I will give you a brief description of some more recent geometric estimators without entering into much details; the list will not be exhaustive. I will not cover projective estimators, as they are less widely used (with the notable exception of PCA), and I am way less familiar with them; some useful references are [Cam03; LLJ+09; Cer14; LMR17].

Geometric estimators

Most geometric estimators are variations over the CorrDim estimator. The Takens estimator [Tak85], for example, is given by

$$d_{\text{Takens}} = \left\langle \log\left(\frac{h}{||x^\mu - x^\nu||}\right)\right\rangle^{-1}, \tag{2.14}$$

where angular brackets denote the average over all pairs of points x^μ and x^ν at distance lesser than h. h is a cutoff distance that must be fixed externally. Again, the idea is that the number of pairs of points at distance lesser than h behaves as a power-law with exponent d for small r.

A maximum likelihood estimator based on the same principles was proposed by Levina & Bickel [LB04]

$$d_{\text{MLE}}(x) = \left[\frac{1}{k-1}\sum_{j=1}^{k-1}\log\frac{||x-x_k||}{||x-x_j||}\right]^{-1}, \tag{2.15}$$

where x_j it the j-th nearest neighbour of point x. They also provide theoretical predictions for the bias and variance of the estimator as the number of points in the dataset goes to infinity. They observe a systematic negative bias caused by the undersampling issue, and possibly by data point lying at edges of the intrinsic manifold.

Hein & Audibert [HA05] further elaborated on CorrDim and Takens's idea by introducing not only an adjustable reference scale h, but also the possibility of preprocessing the distances between points using a positive kernel function k. The central object of their analysis is given by

$$U_{P,h,d'}(k) = \frac{2}{P(P-1)}\sum_{1\leqslant\mu<\nu\leqslant P}\frac{1}{h^{d'}}k\left(\frac{||x^\mu - x^\nu||}{h^2}\right) \tag{2.16}$$

where h is the reference scale, d' is a candidate intrinsic dimension and k is a non-negative, non-increasing function. They then provide, for fixed kernel function and candidate dimension, a prescription to optimize the reference scale h. The intrinsic dimension is then obtained by finding the candidate dimension that better fits the theoretical behaviour of $U_{P,h,d'}(k)$. On the level of performance, while they claim to be performing on average better than CorrDim and of the Takens estimator, it seems that on complex tasks, like the dataset they present in section 4.5, they perform comparably with respect to other estimators.

On a slightly different line from the already mentioned estimators, the minimal neighbourhood estimator proposed by Facco and collaborators [FdR+17] uses the fact that, in the local approximation, the distribution of the ratio μ between the distance between a point and its second nearest neighbour and the distance between the point and its first nearest neighbour is known, and equals

$$f(\mu) = d\mu^{-d-1}\theta(\mu - 1)\,, \tag{2.17}$$

with cumulative distribution function

$$F(\mu) = (1 - \mu^{-d})\theta(\mu - 1)\,. \tag{2.18}$$

Thus, the intrinsic dimension can be estimated by fitting the empirical cumulative distribution F to this analytic form. While this estimator is not based on CorrDim, it is reasonable to expect that it performs similarly, as it is based on a local geometric observable.

In the spirit of revisiting non-trivially CorrDim, Kégl proposed an estimator based on packing numbers [Kég02], that is

$$d_{\text{Packing}} = -\lim_{r\to 0} \frac{\log N(r)}{\log r} \tag{2.19}$$

where $N(r)$ is the number of D-dimensional disks of radius r needed to fully cover the dataset. Again, the estimator is based on a power-law behaviour in the intrinsic dimension at small scales. By performing an analysis at different scales, the author shows that on noisy or non-uniform data the packing estimator may perform better than CorrDim.

Another explored possibility is that of heuristically modifying CorrDim in order to improve its performances on high-dimensional data. This is the path followed by Camastra & Vinciarelli [CV01; CV02]. They propose to compute calibration curves for CorrDim for various sizes of datasets and intrinsic dimensions using synthetic datasets (for example hypercubes), and to use them to mitigate the curse of dimensionality.

A final line of research worth mentioning is that focusing on enhancing CorrDim-like methods with informations on the angular distribution of nearest-neighbours [LRC+11; CBR+14; DQV19].

2.3 The Full Correlation Integral estimator

In the previous sections we saw that intrinsic dimension estimation has two major enemies. One one side, generic dataset may be well represented by highly curved intrinsic manifolds, or by non-linearly embedded simple manifolds; this introduces a great deal of variability in the possible geometries that an estimator needs to be able to deal with. On the other hand, high-dimensional datasets suffer a severe undersampling issue, the curse of dimensionality, that hinders the possibility of using local observables to extract informations on intrinsic geometry. Sadly, we cannot fully escape the consequences of these obstacles with any intrinsic dimension estimator. How can we improve over existing estimators?

The proposal of my collaborators and myself is to try to combine some features of geometric estimators with those of projective estimators. In the next few sections I will give you the details of our new procedure, the Full Correlation Integral (FCI) estimator, but let me first provide you with the general picture.

We start by observing that the empirical correlation integral is analytically tractable in a specific simple case, i.e. for uniformly sampled points on the surface of a d-dimensional hypersphere, embedded in $d+1$ dimensions using the natural inclusion map. By analytically tractable I mean that: (i) the empirical correlation integral is self-averaging, i.e. its probability distribution concentrates around the average value; (ii) the average correlation integral has a closed analytical form depending parametrically on the intrinsic dimension d. This suggests a procedure to estimate the intrinsic dimension of spherically-sampled and isometrically-embedded dataset:

1. center the dataset, i.e. subtract from each sample the position of the barycenter $\frac{1}{P}\sum_{\mu=1}^{P} x^\mu$;
2. project each sample on the unit hypersphere by normalizing it;
3. compute the empirical correlation integral on the normalized data;
4. recover the intrinsic dimension by fitting the analytic correlation integral (computed for the hypersphere) to the empirical correlation integral, and by adding one to it to account for the normalization procedure (that *a priori* lowers the total number of degrees of freedom by one unit).

In a sense, this estimator is more similar to PCA than it is to CorrDim: it assumes some *global* structure on the dataset, and uses geometric information at all length scales to estimate the dimension. I will show you that, actually, FCI has some advantages over PCA: for example, it is more reliable in the undersampled regime $P < d$. Moreover, the FCI estimator is robust: when the assumptions on the dataset are mildly violated, it can nonetheless estimate correctly the intrinsic dimension.

I would like to acknowledge that a very similar approach was proposed by Granata and Carnevale [GC16]. Contrary to our work, they focus on the probability density associated with the correlation integral (seen as a cumulative distribution function). This causes multiple numerical issues with respect to our proposal, resulting in poorer performances on high-dimensional datasets.

All these ingredients are crucial to the last, and most important step. The FCI estimator is easily multi-scalable, i.e. it can be applied to local subsets of the dataset to obtain multiple local intrinsic dimension estimation, in order to estimate the dimension of more complex datasets. The underlying idea is again that of geometric estimators, i.e. that locally the datasets are trivial (linear, uniformly sampled and linearly embedded), and thus the FCI estimator should recover the correct intrinsic dimension locally (up to the limits induced by the curse of dimensionality). The robustness to undersampling of the FCI estimator is the key to optimally mitigate the effects of the curse of dimensionality that inevitably arise when the estimator is multi-scaled.

2.3.1 The correlation integral of spherical datasets

Let me start by studying the analytical properties of the correlation integral

$$\rho(r) = \frac{2}{P(P-1)} \sum_{1\leqslant\mu<\nu\leqslant P} \theta(r - ||x^\mu - x^\nu||) \tag{2.20}$$

on spherical datasets, i.e. with $\{x^\mu\}_{\mu=1}^P$ i.i.d. samples on the $(d-1)$-dimensional hypersphere. I want to prove two results:

- the average correlation integral equals

$$\langle\rho(r)\rangle = \frac{1}{2}\left(1+\frac{\Omega_{d-1}}{\Omega_d}(r^2-2)\,{}_2F_1\left(1-\frac{d}{2},\frac{1}{2};\frac{3}{2};\frac{(r^2-2)^2}{4}\right)\right), \tag{2.21}$$

  where $\Omega_d = 2\pi^{\frac{d+1}{2}}/\Gamma\left(\frac{d+1}{2}\right)$ is the d-dimensional solid angle, and ${}_2F_1$ is the hypergeometric function. The average is performed over the distribution of the P samples $\{x^\mu\}$.

- the correlation integral is self-averaging, i.e. its distribution concentrates around the average value as the number of samples P diverges. In practice, we will show that

$$\mathrm{Var}(\rho(r)) = \mathcal{O}(P^{-1}) \tag{2.22}$$

  as $P \to \infty$.

Average value First of all, as all samples in the dataset are i.i.d., we have that

$$\langle\rho(r)\rangle = \left\langle \frac{2}{P(P-1)}\sum_{1\leqslant\mu<\nu\leqslant P}\theta(r-||x^\mu-x^\nu||)\right\rangle = \langle\theta(r-||x-y||)\rangle \tag{2.23}$$

where angular brackets denote the average with respect to the positions of the points in the dataset, and in the last expression x and y are two i.i.d. points sampled from the $(d-1)$-dimensional hypersphere. Thus

$$\langle\rho(r)\rangle = \int_{\mathbb{R}^d}dx\int_{\mathbb{R}^d}dy\,\frac{\delta(||x||-1)}{\Omega_d}\,\frac{\delta(||y||-1)}{\Omega_d}\theta\left(r^2-||x-y||^2\right) \tag{2.24}$$

where we separately squared the addends inside the step function as they are all positive numbers, Ω_d is the d-dimensional solid angle[a] and the Dirac's delta functions enforce the spherical constraints.

To go on, we move to spherical coordinates. By choosing the azimutal axis of y to be aligned with x, we can write $||x-y||^2$ as

$$||x-y|| = 2(1-\cos(\beta_d))\,, \tag{2.26}$$

where β_d is the azimutal angle of y, or analogously the angle between x and y. Thus

$$\begin{aligned}\langle\rho(r)\rangle = \frac{1}{\Omega_d^2}&\int_0^{2\pi}d\alpha_1\int_0^{\pi}d\alpha_2\sin(\alpha_2)\ldots\int_0^{\pi}d\alpha_d\sin^{d-1}(\alpha_d)\times\\ &\int_0^{2\pi}d\beta_1\int_0^{\pi}d\beta_2\sin(\beta_2)\ldots\int_0^{\pi}d\beta_d\sin^{d-1}(\beta_d)\theta(r^2-2(1-\cos(\beta_d))) \quad (2.27)\\ = \frac{\Omega_{d-1}}{\Omega_d}&\int_0^{\arccos\left(1-\frac{r^2}{2}\right)}d\beta_d\sin^{d-1}(\beta_d)\end{aligned}$$

where in the last passage we performed all the trivial integrals in $\alpha_1,\ldots,\alpha_d,\beta_1,\ldots,\beta_{d-1}$ using the definition of Ω_d and Ω_{d-1}. The integration can be performed exactly by first changing variable to $t = \cos(\beta_d)$, and then by expanding the integrand in Taylor series around the origin

$$\begin{aligned}
\langle\rho(r)\rangle &= \frac{\Omega_{d-1}}{\Omega_d}\int_0^{\arccos\left(1-\frac{r^2}{2}\right)} d\beta_d \sin^{d-1}(\beta_d) \\
&= \frac{\Omega_{d-1}}{\Omega_d}\int_{1-\frac{r^2}{2}}^{1} dt\,\left(1-t^2\right)^{\frac{d}{2}-1} \\
&= \frac{\Omega_{d-1}}{\Omega_d}\sum_{n\geqslant 0}\frac{\Gamma\left(n+1-\frac{d}{2}\right)}{n!\,\Gamma\left(1-\frac{d}{2}\right)}\int_{1-\frac{r^2}{2}}^{1} dt\, t^{2n} \\
&= \frac{\Omega_{d-1}}{\Omega_d}\sum_{n\geqslant 0}\frac{\Gamma\left(n+1-\frac{d}{2}\right)}{n!\,\Gamma\left(1-\frac{d}{2}\right)}\frac{1}{2n+1}\left(1-\left(1-\frac{r^2}{2}\right)^{2n+1}\right).
\end{aligned} \tag{2.28}$$

Finally, using the fact that

$$\sum_{n\geqslant 0}\frac{\Gamma(n+a)}{n!\,\Gamma(a)}\frac{1}{2n+1}b^{2n+1} = b\,{}_2F_1\left(a,\frac{1}{2};\frac{3}{2};b\right) \tag{2.29}$$

where ${}_2F_1$ is the hypergeometric function. we obtain

$$\langle\rho(r)\rangle = \frac{1}{2}\left(1+\frac{\Omega_{d-1}}{\Omega_d}(r^2-2)\,{}_2F_1\left(1-\frac{d}{2},\frac{1}{2};\frac{3}{2};\frac{(r^2-2)^2}{4}\right)\right). \tag{2.30}$$

Variance The variance of $\rho(r)$ equals

$$\begin{aligned}
\left\langle\rho(r)^2\right\rangle &= \left(\frac{2}{P(P-1)}\right)^2 \\
&\times\sum_{1\leqslant\mu<\nu\leqslant P}\;\sum_{1\leqslant\sigma<\tau\leqslant P}\left\langle\theta(r-||x^\mu-x^\nu||)\,\theta(r-||x^\sigma-x^\tau||)\right\rangle.
\end{aligned} \tag{2.31}$$

It's easy to see that, for large P, the double sum is dominated by the terms with all different indices, and that at leading order in P the number of such terms is $P^2/4$, giving that

$$\left\langle\rho(r)^2\right\rangle = \langle\rho(r)\rangle^2\left(1+\mathcal{O}(P^{-1})\right) \tag{2.32}$$

or, equivalently,

$$\mathrm{Var}(\rho(r)) = \mathcal{O}(P^{-1}). \tag{2.33}$$

[a] The d-dimensional solid angle measures the surface of the d dimensional unit hypersphere, and is given by

$$\Omega_d = \int_0^{2\pi} d\alpha_1\int_0^{\pi} d\alpha_2\sin(\alpha_2)\ldots\int_0^{\pi} d\alpha_d\sin^{d-1}(\alpha_d) = \frac{2\pi^{\frac{d+1}{2}}}{\Gamma\left(\frac{d+1}{2}\right)}, \tag{2.25}$$

2.3.2 Robustness on simple datasets

As I already mentioned, the analytical results presented in the previous section suggest the following intrinsic dimension estimator:

1. center the dataset, i.e. subtract from each sample the position of the barycenter $\frac{1}{P}\sum_{\mu=1}^{P} x^{\mu}$;
2. project each sample on the unit hypersphere by normalizing it;
3. compute the empirical correlation integral on the normalized data;
4. recover the intrinsic dimension by fitting the analytic correlation integral (computed for the hypersphere) to the empirical correlation integral, and by adding one to it to account for the normalization procedure (that *a priori* lowers the total number of degrees of freedom by one unit).

This procedure is exact if the dataset has a linear intrinsic manifold sampled with a rotation invariant probability distribution and is isometrically embedded[5] in higher dimensions. In fact, in this case the centered and normalized dataset is a uniform sample from a $d-1$ dimensional hypersphere, and all distances can be equivalently measured in $\mathbb{R}^d$ or $\mathbb{R}^D$ by isometry.

It is extremely important to characterize the performance of the FCI estimator on datasets that fail to satisfy the linearity, isotropy and isometry conditions. Figure 2.8 presents experimental results that suggest that the FCI estimator is reliable even if:

- the dataset is not perfectly linear, isotropically sampled or isometrically embedded. The left panels of Figure 2.8 show that the correlation integral of (centered and normalized) linear, Gaussian and digital datasets are well fitted by Equation (2.21), even though they show manifold-dependent features. Notice that the full analytical form of Equation (2.21) is needed in order to correctly fit the correlation integral in the case of digital datasets (top left panel of Figure 2.8); a local fit (for example at small or intermediate r) would be tricked by manifold-dependent features into an incorrect estimation;

- the dataset is extremely undersampled. The central and top right panels of Figure 2.8 show that, in the case of linear datasets, the FCI estimator correctly estimates the intrinsic dimension (up to a relative error of the order of 1%) even in the extremely undersampled case in which the number of samples P is lesser than the intrinsic dimension d. Notice that PCA needs at least $\sim d \log d$ samples to correctly identify the intrinsic dimension;

- the dataset is corrupted by noise. The bottom right panel of Figure 2.8 shows the intrinsic dimension estimation for a linear dataset corrupted by i.i.d. Gaussian noise. The FCI estimator has a sharp transition between a correct estimation when the noise is not corrupting the data, and an incorrect one when the noise completely covers the original geometric information of the dataset.

[5] An isometrical embedding is a map that preserves distances. For example, the natural inclusion map of $\mathbb{R}^d \to \mathbb{R}^D$, followed by a rotation is an isometrical map.

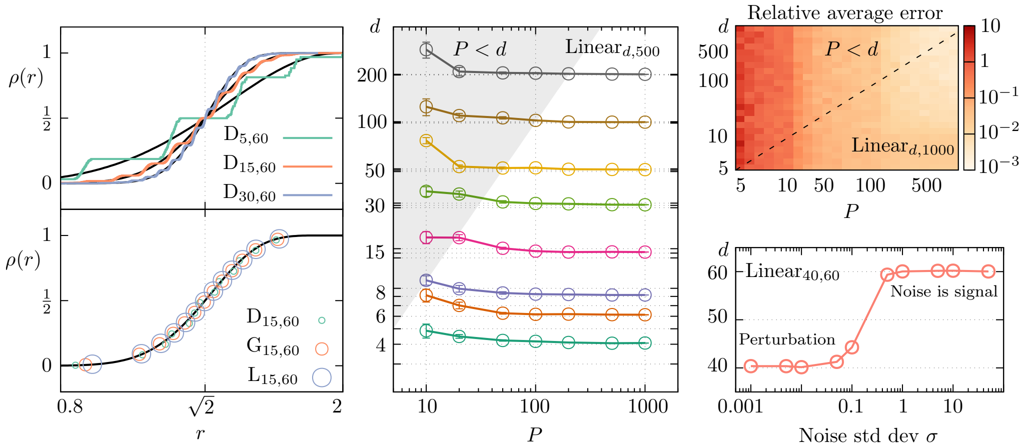


Figure 2.8: Robustness of the FCI estimator. (Top left) Correlation integral of three digital datasets (centered and normalized) of increasing intrinsic dimension $d = 5, 15, 30$, common embedding dimension $D = 60$ and sample size $P = 500$, overlaid to the analytical form Equation (2.21). Even though the curves present manifold-dependent features, the global fitting procedure of the FCI estimator averages out these imperfections and correctly estimates the intrinsic dimension. (Bottom left) Correlation integral of a digital, a Gaussian and a linear dataset (centered and normalized) with $d = 15$, $D = 60$ and $P = 500$ (open markers) overlaid to the analytic form of Equation (2.21). (Center) Estimated dimension (using the FCI estimator) vs number of samples P for several linear datasets with varying intrinsic dimension d and common embedding dimension $D = 500$. Error bars denote the variance over 10 different datasets at fixed P and d. The shaded region is the extremely undersampled regime where $P < d$. (Top right) Average relative error $|(d_{\text{est}} - d)/d|$ (over 20 instances) for different values of P and d. For $P \sim 100$, the average relative error is roughly 1% independently on d, allowing for a reliable estimation even in the extremely undersampled regime. (Bottom right) Estimated dimension vs noise level for a linear dataset ($d = 40$, $D = 60$) corrupted by Gaussian noise with variance σ. As a function of the noise level, the estimated dimension has a somewhat sharp transition between the correct value $d = 40$ and the incorrect value $d = 60$ corresponding to the ID of the noise. (Reprinted from [EGR19]).

Summarizing, the FCI estimator is a robust and reliable estimators even on datasets that do not completely comply with its assumptions, on extremely undersampled datasets and on noisy datasets.

Of course, the FCI estimators has some obvious limitations. On one hand, when the dataset is highly curved, the FCI estimator will overestimate the intrinsic dimension in a similar way as in the case of PCA. This can be observed in Figure 2.9, left panel, in the case of a Hein dataset. On the other hand, when the geometry of the dataset is even more complex, the correlation integral of the dataset will not be well-fitted by the functional form in Equation 2.21, and the dimension estimation given by the FCI estimator will be meaningless. For an example on the MNIST dataset, see Figure 2.9, right panel.

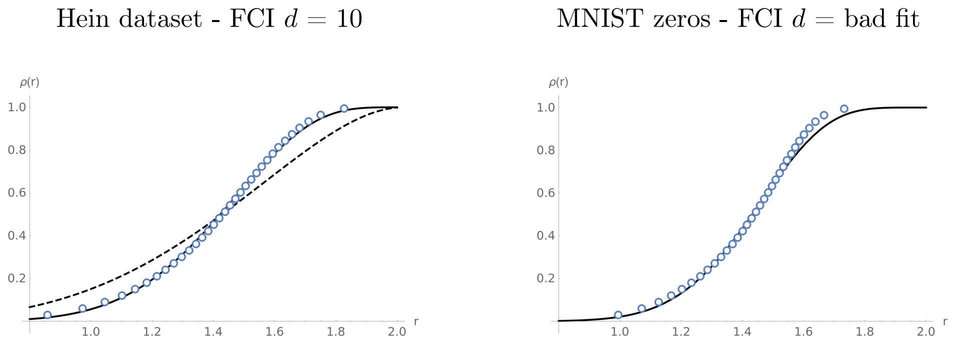


Figure 2.9: Limitation of FCI estimator. (Left) Correlation integral of a (centered and normalized) Hein dataset, $d = 5$, $D = 10$ and $P = 200$. The dashed line is Equation (2.21) with $d = 4$ (not $d = 5$ due to the normalization procedure), the solid line with $d = 9$. The FCI estimator overestimates the dimension of curved datasets similarly to PCA. (Right) Correlation integral of a (centered and normalized) sample of $P = 200$ MNIST's zeros. The solid line is Equation (2.21) with $d = 14$. The empirical correlation integral is not well-fitted by (2.21), as it is particularly evident for $r > 1.6$.

2.3.3 The multi-scale approach

In order to tackle more complex datasets, such as spherical datasets[6], the Hein dataset or MNIST, we have to come up with some trick to ignore all such complexity. The inspiration comes from geometric estimators. Locally, synthetic datasets are well-described (keeping in mind the limits of the curse of dimensionality) by uniformly sampled linear datasets. It is then natural to try to use global estimators such as PCA and the FCI estimator locally by applying them only to small patches of the dataset U, i.e. to subsets

$$U(c, r) = \{x^\nu \in U \mid ||c - x^\nu|| \leqslant r\}\,, \tag{2.34}$$

where r is a scale at which our dataset is well-approximated by the linear approximation, and c is a random point in U. This procedure, repeated for various *centers* c and various (small) scales r, provides us with a set of local intrinsic dimension estimations $d(c, r)$, that we can study as functions of c and r. Two questions arise immediately. How can we choose the scale r so that the linear approximation is valid? How can we aggregate all the data of different local patches to get a single intrinsic dimension estimation? Unfortunately, it is very difficult to answer theoretically these questions. In this section, I will try to convince you that we might use some heuristic principles in order to guide our estimation procedure.

It's good to start from an example. Figure 2.10, top left panel, shows a swiss roll dataset. Let us apply the FCI estimator to two local neighbourhoods of this datasets (the respective centers are shaded in Figure 2.10), and plot the resulting FCI estimations as

[6] You may wonder why spherical datasets are considered complex, given that the correlation integral can be analytically computed on spheres. The reason is that you don't have the *a priori* knowledge that a dataset is spherical. In practice, when normalizing a spherical dataset you lose no information, contrary on what happens on any other dataset. Thus, the last step of the FCI estimator, i.e. adding 1 to the fitted dimension, overestimates the dimension of a spherical dataset by 1.

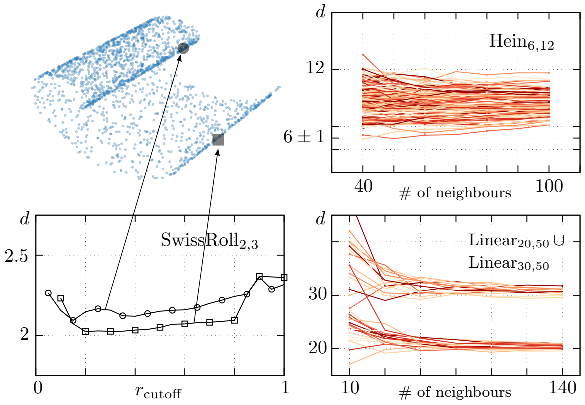


Figure 2.10: Performance of the multi-scale FCI estimator on synthetic datasets. (Left) Multi-scale FCI estimation on a swiss roll dataset ($d = 2$, $D = 3$, $P = 2000$). Higher local intrinsic dimension estimations correspond to zones of higher curvature in the dataset. The correct intrinsic dimension can be recovered by measuring the height of the lowest *plateau* in the plot. (Top right) Multi-scale FCI estimation on a Hein dataset ($d = 6$, $D = 12$, $P = 10^4$). In this dataset curvature effects are extremely strong, hence the vertical spread of the local intrinsic dimensionality curves in the plot. Again, the lowest *plateau* identifies the correct intrinsic dimension. (Bottom right) Multi-scale FCI estimation on a multidimensional dataset, composed by the union of two linear datasets ($d_1 = 20$, $d_2 = 30$, $D = 50$, $P = 1000$ for each of the two datasets). The local quality of the multi-scale FCI estimator allows to correctly separate and identify multiple intrinsic dimensions in the dataset. (Reprinted from [EGR19]).

functions of the scale r (see Figure 2.10, bottom left panel). We can observe some crucial phenomena:

- the estimations related to the innermost patch of the dataset are systematically higher than those related to the outermost patch. This is a clear consequence of curvature: where the datasets is most curved, the FCI estimator will overestimate the intrinsic dimension;

- the estimations are *persistent*. They are mostly constant as functions of the scale r, meaning that the geometrical properties of the dataset are mostly constant in this range of scales;

- at small and large scales, we observe higher intrinsic dimension estimations. At small scales, this is due to the lack of datapoints (even though the FCI is robust to undersampling, it will fail when the number of samples is too small, $P \sim 10$). At large scales, the FCI starts probing the global structure of the dataset, and overestimates the dimension due to curvature;

- the intrinsic dimension estimation of the lowest *plateau* is near 2, which is the correct intrinsic dimension.

These observations suggest the following heuristic procedure, that we call *multi-scale FCI* estimator:

1. perform many FCI estimations $d(c, r)$ on small patches of the dataset $U(c, r)$, varying both the center c and the scale r. A notable variant is to replace the metric scale r with the number k of nearest neighbours, i.e. to define

$$U(c,k) = \{x^{\nu} \in U \mid x^{\nu} \text{ is one of the first } k \text{ nearest neighbours of } c \text{ in } U\}\,; \quad (2.35)$$

2. plot $d(c, r)$ as a function of r, for all centers c. This allows to have a concise representation of many local intrinsic dimension estimations;

3. consider reliable the curves that show a *plateau* as functions of r. Longer *plateaux* should correspond to neighbourhoods on the dataset where the geometric properties vary less;

4. select the height of the lowest *plateau* as the aggregated intrinsic dimension estimation for the dataset. Selecting the lowest *plateau* should favor neighbourhoods of the dataset with lowest curvature.

Of course, all this considerations are qualitative, and may break in many situations, for example if the dataset is severely undersampled. In order to confirm that this heuristic is reasonable, let us take a look at Figure 2.10, right panels, where the multi-scale FCI has been applied to a pair of challenging synthetic datasets: the highly-curved Hein dataset ($d = 6$, $D = 12$, top right panel) and a multidimensional dataset composed by a union of linear datasets with common embedding space ($d_1 = 20$, $d_2 = 30$, $D = 50$, bottom right panel). In both cases, we observe that the lowest *plateaux* identifies the correct intrinsic dimension (or dimensions, in the second case). In the Hein dataset case, where the curvature is higher, we see that the majority of the curves $d(c, r)$ identify higher local intrinsic dimensions as expected, and that selecting the lowest *plateau* is crucial to mitigate curvature effects. In the multidimensional case, the local estimates are more concentrated around the values of $d = 20$ and $d = 30$ as the curvature effects are not present. Notice that in both cases we decided to use the k-nearest neighbour version of the multi-scale FCI estimator due to computational convenience.

Thus, the multi-scale FCI estimator provides a useful heuristic to treat complex datasets. Notice that, in principle, every intrinsic dimension estimator can be multi-scaled as we did in this section. For example, multi-scale PCA and its generalizations [LLJ+09; LMR17] have been analyzed in depth in the past. I stress that the multi-scalability of the FCI estimator in particular is enormously helped by its robustness to imperfections in the datasets and to undersampling, allowing to mitigate both local curvature effects and the curse of dimensionality. Other estimators do not share this feature.

2.3.4 Multi-scale FCI on datasets of synthetic bitmap images

To better evaluate the performance of the multi-scale FCI estimator, and to asses whether it is reasonable to expect that it will work on real datasets such as MNIST, it is useful to introduce a new kind of synthetic dataset. We consider a dataset of synthetic bitmap images representing light spots, see Figure 2.11, right panel. We generate each bitmap

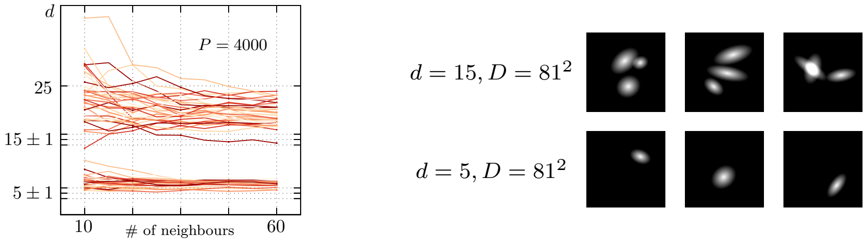


Figure 2.11: Performance of the multi-scale FCI estimator on bitmap images. (Left) Multi-scale FCI estimation for two datasets ($D = 81^2$, $P = 4000$) of synthetic bitmap images representing respectively one ($d = 5$) or three ($d = 15$) light spots. The heuristics of considering the lowest *plateau* is again effective, leading to the correct intrinsic dimension estimation. (Right) Some samples taken from the two datasets of synthetic bitmap images. (Reprinted from [EGR19]).

image by assigning to each pixel (i, j) of a 81×81 image[7] the value $v_{i,j}$ given by

$$\begin{aligned} a_{i,j} &= \cos\theta(j - \Delta x) + \sin\theta(i + \Delta y) \\ b_{i,j} &= -\sin\theta(j - \Delta x) + \cos\theta(i + \Delta y) \\ v_{i,j} &= 1 - \sqrt{\frac{a_{i,j}^2 + e^2 b_{i,j}^2}{(1 + e^2)s^2}} \end{aligned} \tag{2.36}$$

with parameters

Δx	horizontal translation	uniform in $(-20, 20)$
Δy	vertical translation	uniform in $(-20, 20)$
s	size	uniform in $(1, 3)$
e	eccentricity	uniform in $(5, 10)$
θ	angle of the major axis	uniform in $(-\pi/2, \pi/2)$

An image of a light spot has thus five degrees of freedom, i.e. intrinsic dimension $d = 5$ (horizontal and vertical position of its center, size, eccentricity and rotation), and embedding dimension $D = 81^2$. Multiple images can be easily stacked by selecting for each pixel the maximum value between all images, allowing for images with up to three easily distinguishable light spots, i.e. up to $d = 15$ (more than three light spots gives unclear images).

Datasets of bitmap images are particularly challenging for intrinsic dimension estimators. Simple transformations such as translations in the image translate to very complex transformations in intrinsic space.

The multi-scale analysis for two of these datasets (the one and three light blobs per image cases) is shown in Figure 2.11, left panel. Notice that the $d = 15$ case is an example of a very high-dimensional complex dataset. Even in this very complex case, the multi-scale FCI estimator provides a very good estimate of the intrinsic dimension.

[7] The pixel $(0, 0)$ is the central one.

Estimator	$\text{SwissRoll}_{2,3}$	$\text{Linear}_{20,50}$ $\cup$ $\text{Linear}_{30,50}$	$\text{Hein}_{6,12}$	$\text{Images}_{5,81^2}$	$\text{Images}_{15,81^2}$
CorrDim [GP83b]	1.98	12.53	5.93	5	13.5
Takens [Tak85]	1.97	12.01	5.77	N.A.	N.A.
Hein et al. [HA05]	2	13	6	N.A.	N.A.
PCA	3	20 & 30	12	40	40
Multi-scale FCI	2	20 & 30	6	5	15

Table 2.1: Comparison between ID estimators on curved and multi-dimensional datasets. Green digits represent correct estimates, red digits wrong estimates.

2.3.5 Comparison with other estimators

Benchmarking intrinsic dimension estimators is a challenge by itself [LRC+11]. To provide a preliminary assessment on the performance of multi-scale FCI, we compared a selection of estimators on the challenging datasets presented above: the swiss roll, a multidimensional dataset, the Hein dataset and the synthetic bitmap images datasets. Table 2.1 reports the results[8].

As expected, geometric and projective estimators have complementary performances: low-dimensional curved datasets are correctly treated by geometric estimators, while PCA can deal with the multidimensional dataset (in the eigenvalue plot, two magnitude jumps are visible). On high-dimensional curved datasets, such as the $d = 15$ synthetic bitmap images dataset, both classes of estimators fail. In all cases, the multi-scale FCI estimator can estimate the correct intrinsic dimension.

2.4 Perspectives

In this chapter, I hopefully gave you an introduction to manifold learning, and in particular to intrinsic dimension estimation. I presented the FCI estimator and its multi-scaled version, and I argued that the latter has notable advantages over existing intrinsic dimension estimators. I think that now it may be a good time for the bad news: throughout the chapter I have ignored some issues that I should now highlight, and that directly motivate the possible future research directions related to this topic.

The first issue concerns the significance of the manifold model itself. Given a discrete set of points, it is always possible to describe it as a random sample over a one-dimensional, possibly very contorted, manifold. This makes it clear that the existence (and uniqueness) of the intrinsic manifold is at least a delicate point. It also raises the question of whether the real intrinsic dimension of a dataset (in the manifold model) is the actually the number that we would like to compute. Consider, for example, a one-dimensional curve densely-

[8] The CorrDim, Takens and Hein estimations were obtained using the code available at `https://www.ml.uni-saarland.de/code/IntDim/IntDim.htm`. N.A. means that the code could not obtain an intrinsic dimension estimation. We however expect very similar results to those of CorrDim. The FCI estimations were obtained using the code available at `https://github.com/vittorioerba/pyFCI`.

packed in a two-dimensional square. At which level of packing should we say that a discrete sample of the curve is one or two dimensional? And how these questions interact with the curse of dimensionality, that practically hinders our possibility of probing the local geometry of datasets? These questions are of fundamental and crucial importance, and constitute a first possible line of further research.

A second issue is given by the level of rigour and usability of the multi-scale FCI estimator. As we have seen, the multi-scale FCI estimator relies heavily on the user to perform the actual estimation. In fact, at the current time, there is no automated procedure to extract the final intrinsic dimension estimation from the local dimensionality curves $d(c, r)$. A possible step towards the formalization of the multi-scale FCI is to develop an error measure to attach to each local estimate. My collaborators and myself briefly explored the possibility of using the mean-square error of the fit in the FCI estimation procedure as an error measure, but we found no promising result.

Finally, and based on a solution to the previous issue, it will be very important to benchmark the multi-scale FCI estimator using state-of-the-art benchmarking protocols, such as the one described [LRC+11].

2.5 Chapter bibliography

[AFD+20] Michele Allegra, Elena Facco, Francesco Denti, Alessandro Laio, and Antonietta Mira. Data segmentation based on the local intrinsic dimension. *Scientific Reports*, 10(1):16449, October 2020. ISSN: 2045-2322. DOI: 10.1038/s41598-020-72222-0.

[ALM+19] Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/cfcce0621b49c983991ead4c3d4d3b6b-Paper.pdf.

[BML06] Yoshua Bengio, Martin Monperrus, and Hugo Larochelle. Nonlocal estimation of manifold structure. *Neural Computation*, 18(10):2509–2528, 2006. DOI: 10.1162/neco.2006.18.10.2509.

[Cam03] Francesco Camastra. Data dimensionality estimation methods: a survey. en. *Pattern Recognition*, 36(12):2945–2954, December 2003. ISSN: 00313203. DOI: 10.1016/S0031-3203(03)00176-6.

[Cao97] Liangyue Cao. Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D: Nonlinear Phenomena*, 110(1-2):43–50, December 1997. DOI: 10.1016/s0167-2789(97)00118-8.

[CBR+14] Claudio Ceruti, Simone Bassis, Alessandro Rozza, Gabriele Lombardi, Elena Casiraghi, and Paola Campadelli. Danco: an intrinsic dimensionality estimator exploiting angle and norm concentration. *Pattern recognition*, 47(8):2569–2581, August 2014. DOI: 10.1016/j.patcog.2014.02.013.

[Cer14] Claudio Ceruti. *Novel techniques for intrinsic dimension estimation*. PhD thesis, Scuola di Dottorato in Matematica e Statistica per le Scienze Computazionali - XXVII ciclo - Dipartimento di Matematica "Federigo Enriques", 2014. DOI: 10.13130/ceruti-claudio_phd2014-12-16.

[CRH10] K.M. Carter, R. Raich, and A.O. Hero. On local intrinsic dimension estimation and its applications. *IEEE Transactions on Signal Processing*, 58(2):650–663, February 2010. DOI: 10.1109/tsp.2009.2031722.

[CV01] Francesco Camastra and Alessandro Vinciarelli. Intrinsic dimension estimation of data: an approach based on grassberger–procaccia's algorithm. *Neural Processing Letters*, 14(1):27–34, August 2001. ISSN: 1573-773X. DOI: 10.1023/A:1011326007550.

[CV02] F. Camastra and A. Vinciarelli. Estimating the intrinsic dimension of data with a fractal-based method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(10):1404–1407, October 2002. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2002.1039212.

[DQV19] Mateo Díaz, Adolfo J. Quiroz, and Mauricio Velasco. Local angles and dimension estimation from data on manifolds. *Journal of Multivariate Analysis*, 173:229–247, September 2019. DOI: 10.1016/j.jmva.2019.02.014.

[EGR19] Vittorio Erba, Marco Gherardi, and Pietro Rotondo. Intrinsic dimension estimation for locally undersampled data. *Scientific Reports*, 9(1):17133, November 2019. ISSN: 2045-2322. DOI: 10.1038/s41598-019-53549-9.

[ER92] J.-P. Eckmann and D. Ruelle. Fundamental limitations for estimating dimensions and Lyapunov exponents in dynamical systems. en. *Physica D: Nonlinear Phenomena*, 56(2-3):185–187, May 1992. ISSN: 01672789. DOI: 10.1016/0167-2789(92)90023-G.

[FdR+17] Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7(1), September 2017. DOI: 10.1038/s41598-017-11873-y.

[FO71] K. Fukunaga and D.R. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, C-20(2):176–183, February 1971. DOI: 10.1109/t-c.1971.223208.

[FPR+19] Elena Facco, Andrea Pagnani, Elena Tea Russo, and Alessandro Laio. The intrinsic dimension of protein sequence evolution. *PLOS Computational Biology*, 15(4):1–16, April 2019. DOI: 10.1371/journal.pcbi.1006767.

[GC16] Daniele Granata and Vincenzo Carnevale. Accurate Estimation of the Intrinsic Dimension Using Graph Distances: Unraveling the Geometric Complexity of Datasets. en. *Scientific Reports*, 6(1):31377, November 2016. ISSN: 2045-2322. DOI: 10.1038/srep31377.

[GP83a] Peter Grassberger and Itamar Procaccia. Characterization of strange attractors. *Physical Review Letters*, 50(5):346–349, January 1983. DOI: 10.1103/physrevlett.50.346.

[GP83b] Peter Grassberger and Itamar Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1-2):189–208, October 1983. ISSN: 0167-2789. DOI: 10.1016/0167-2789(83)90298-1.

[HA05] Matthias Hein and Jean-Yves Audibert. Intrinsic dimensionality estimation of submanifolds in rd. In *Proceedings of the 22nd international conference on Machine learning -ICML 05*. ACM Press, 2005. DOI: 10.1145/1102351.1102388.

[Kég02] Balázs Kégl. Intrinsic dimension estimation using packing numbers. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, pages 697–704, Cambridge, MA, USA. MIT Press, 2002. URL: http://dl.acm.org/citation.cfm?id=2968618.2968705.

[LB04] Elizaveta Levina and Peter J. Bickel. Maximum likelihood estimation of intrinsic dimension. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, pages 777–784, Vancouver, British Columbia, Canada. MIT Press, 2004. URL: http://dl.acm.org/citation.cfm?id=2976040.2976138.

[LCB10] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[LLJ+09] Anna V. Little, Jason Lee, Yoon-Mo Jung, and Mauro Maggioni. Estimation of intrinsic dimensionality of samples from noisy low-dimensional manifolds in high dimensions with multiscale SVD. In *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*. IEEE, August 2009. DOI: 10.1109/ssp.2009.5278634.

[Llo82] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. DOI: 10.1109/TIT.1982.1056489.

[LMR17] Anna V. Little, Mauro Maggioni, and Lorenzo Rosasco. Multiscale geometric methods for data sets I: Multiscale SVD, noise and curvature. en. *Applied and Computational Harmonic Analysis*, 43(3):504–567, November 2017. ISSN: 10635203. DOI: 10.1016/j.acha.2015.09.009.

[LRC+11] Gabriele Lombardi, Alessandro Rozza, Claudio Ceruti, Elena Casiraghi, and Paola Campadelli. Minimum neighbor distance estimators of intrinsic dimension. In *Machine Learning and Knowledge Discovery in Databases*, pages 374–389. Springer Berlin Heidelberg, 2011. DOI: 10.1007/978-3-642-23783-6_24.

[LV07] John A. Lee and Michel Verleysen, editors. *Nonlinear Dimensionality Reduction*. Springer New York, 2007. DOI: 10.1007/978-0-387-39351-3.

[MAR+21] Tiago Mendes-Santos, Adriano Angelone, Alex Rodriguez, Rosario Fazio, and Marcello Dalmonte. Intrinsic dimension of path integrals: data mining quantum criticality and emergent simplicity, 2021. arXiv: 2103.02640 [cond-mat.stat-mech].

[MTD+21] T. Mendes-Santos, X. Turkeshi, M. Dalmonte, and Alex Rodriguez. Unsupervised learning universal critical behavior via the intrinsic dimension. *Phys. Rev. X*, 11:011040, 1, February 2021. DOI: 10.1103/PhysRevX.11.011040.

[PM13] Dominique Perraul-Joncas and Marina Meila. Non-linear dimensionality reduction: riemannian metric estimation and the problem of geometric discovery, 2013. arXiv: 1305.7255 [stat.ML].

[Row00] S. T. Roweis. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000. DOI: 10.1126/science.290.5500.2323.

[SM90] George Sugihara and Robert M. May. Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature*, 344(6268):734–741, April 1990. DOI: 10.1038/344734a0.

[SPG$^+$17] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin. A review of clustering techniques and developments. *Neurocomputing*, 267:664–681, 2017. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.06.053.

[Ste56] Hugo Steinhaus. Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci*, 1(804):801, 1956.

[Str94] Steven H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. en. Studies in Nonlinearity. Addison-Wesley Pub, Reading, Mass, 1994. ISBN: 978-0-201-54344-5.

[Tak81] Floris Takens. Detecting strange attractors in turbulence. In *Lecture Notes in Mathematics*, pages 366–381. Springer Berlin Heidelberg, 1981. DOI: 10.1007/bfb0091924.

[Tak85] F. Takens. On the numerical determination of the dimension of an attractor. In *Lecture Notes in Mathematics*, pages 99–106. Springer Berlin Heidelberg, 1985. DOI: 10.1007/bfb0075637.

[Ten00] J. B. Tenenbaum. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000. DOI: 10.1126/science.290.5500.2319.

[Tru68] G.V. Trunk. Statistical estimation of the intrinsic dimensionality of data collections. *Information and Control*, 12(5):508–525, May 1968. DOI: 10.1016/s0019-9958(68)90591-3.

[vdMH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

CHAPTER 3

Random geometric graphs in high dimension

Spatial networks are a key ingredient in Statistical Physics and Computation, modelling interaction networks between particles and nearest-neighbour graphs for machine learning applications. It is still not clear whether high-dimensional spatial networks behave as geometrically unstructured graphs, or if it is possible to detect their Euclidean geometry. In this chapter, I will explore this question following [EAG+20]. I will focus on specific observables, M-clique densities, and show that in some cases, even in the limit of infinite dimension, spatial network may be distinguished from unstructured graphs.

Contents

3.1 Network theory

3.1.1 Graphs are a key tool in modern science

In the last twenty years, the sudden availability of big telecommunication, social and epidemic datasets fostered a novel interest for graph theory [AB02]. In fact, graphs are the best mathematical object to describe relational structure in datasets. Consider, for example, a social network [Net13]. While it is certainly interesting to have data about all its users, the crucial information here is given by the relations between the users: who follows whom?

Statistical Physics contributed greatly to the study of complex systems through the lenses of graph theory [CSS+19]. It was discovered that many models of graphs undergo phase transitions in the limit of an infinite number of nodes. For example, as the number of edges increases, many models of large random graphs quickly transition from a phase in which the nodes are mostly disconnected between each other to a phase in which all the nodes belong to the same connected component. This has practical implications: as the users of a growing social network connect more and more with each other, there is a sharp transition from mostly-disconnected, small communities to a phase where every user belongs to the same community. Moreover, physicists noticed that many graphs arising from real complex systems show some form of criticality, meaning that their properties show fractal behaviour. For example, the average degree of the nodes, i.e. the average number of nodes linked to a given node, often follows a power-law distribution. This implies that there is no intrinsic scale describing the number of neighbours in the graph.

Graphs arise often also in Computer Science [BCD+18]. Almost every optimization problem will be defined on a graph, the more complex the more non-trivial the problem is. For example, consider what Google Maps does when you try to find the optimal route between Milan and Naples. It looks for the path of least travel time on the graph of cities, routes and intersections of Italy. Figure 3.1 shows an example of shortest path on a simpler graph.

A key model of graph is given by spatial graphs [Pen03; Bar11]. Spatial graphs are special in the fact that their nodes are points in some geometrical space, for example d-dimensional Euclidean space. In non-spatial graphs, there is no limit to the connectivity between nodes in general. In spatial graphs, geometry affects the connectivity between nodes. For example, consider the network of telecommunications between mobile phones and cell towers. In this graph, if a cell phone is far from a given cell tower, the probability that they will be connected will be very small [GGD15]. The same is true for epidemic spreading in a small community: nearer nodes (people) will have a much stronger epidemic influence one on the other than farther nodes.

Spatial graphs find their use also in optimization problems and machine learning. Datasets are often represented as points in some high-dimensional space, and they are studied using local algorithms that use not only the position of the nodes, but also the position of the neighbours of each node, to infer properties. For example, many dimensionality reduction algorithms try to project high-dimensional points into a low-dimensional space while keeping the local geometry of a node and its nearest neighbours intact (see for example Isomap [Ten00], LLE [Row00] or t-SNE [vdMH08]).

While for the study of complex systems low-dimensional spatial networks (dimension lesser than three) are often enough, it is crucial to understand the properties of spatial networks in high-dimension for machine learning applications (the dimension of the MNIST dataset is 784, and this is considered a "simple" dataset). Nonetheless, analytical

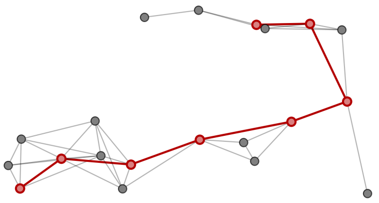


Figure 3.1: Example of shortest path on a graph. The graph drawn in gray is generated by connecting nearby nodes. The highlighted subgraph is the shortest path between its two end-nodes.

results for high-dimensional spatial networks are scarce, and mainly motivated by pure mathematical interest [DC02; DGL+11; BDE+16; AB20]. The main question is whether spatial graphs in very high dimension are still significantly constrained by the underlying geometry, that is, whether they converge to simpler, non-geometrical graphs or not. A natural setting to investigate this question is given by random graphs, i.e. graphs where the attributes of nodes (their position, for example) and their connectivities are determined by a random process [FK15]. By looking at average properties, we may hope to obtain some generic insight on certain families of graphs, without limiting our analysis to specific realizations.

In this chapter I will compare the behaviour of spatial networks to that of Erdös-Rényi graphs [ER60], i.e. unstructured graphs in which each pair of nodes is connected by an edge with independent probability p, by studying the average number of highly-connected clusters that can be found in the graph. I will show that, in some regimes, geometric structure can be detected even in very high dimension by looking at the statistics of highly-connected clusters [EAG+20].

3.1.2 A primer in graph theory

Before starting, let me recollect some basic notions and notations about graphs. A graph is defined by a set of nodes (or vertices) V and by a set of edges $E \subseteq V \times V$ (edges are denoted by their end vertices). Whenever the order of end vertices does not matter we say that the graph is *undirected*, that is edges have no orientation. In the following we will always consider undirected graphs without self-loops, i.e. with no edges connecting a node to itself. In mathematical notation $(i,i) \notin E$ for all $i \in V$. The *degree* of a node is the number of other nodes directly connected to it by an edge.

The connectivity structure of a graph $G = (V, E)$ with P nodes can be encoded into the *adjacency matrix*, i.e. the $P \times P$ matrix defined by

$$A(G)_{ij} = \begin{cases} 1 & (i,j) \in E \\ 0 & (i,j) \notin E \end{cases}, \tag{3.1}$$

where the vertex set V was given an arbitrary ordering.

A *path* in a graph is a set of contiguous edges, i.e. edges sharing a common node. A graph is *connected* if there exists a path between each pair of nodes; in other words, a graph is connected if a walker can travel between each pair of nodes by walking only on edges of the graph. A graph is *fully-connected* if all of its nodes are connected to each other by an edge.

A subgraph $H = (G', E')$ of a graph $G = (V, E)$ is a graph such that $G' \subseteq G$, i.e. whose nodes are a subset of the nodes of the original graph, and such that $E' \subseteq (G' \times G') \cap E$, i.e. whose edges are a subset of the original edge set, and are restricted to the new node set G'.

A M-clique H in a graph G is a fully-connected subgraph of G with M vertices. M-cliques in a graph denote clusters of highly-connected nodes. The maximum number of M-cliques in a graph with P nodes is given by $\binom{P}{M}$, and is achieved only by fully-connected graphs. The density of M-cliques of a graph G is defined as

$$\rho_M(G) = \frac{\#\text{ of } M\text{-cliques in } G}{\binom{P}{M}}\,, \tag{3.2}$$

that is the fraction of M-cliques with respect to the fully-connected graph.

In this chapter we will not need more graph theory than this. Some good general references on graph theory and more in general complex network theory are [BM08; Bar11; CSS$^+$19].

3.2 Random graphs

When dealing with the study of general properties of graphs, it is often useful to introduce models of random graphs. In this way, one can study average properties in the ensemble of choice.

From a probabilistic point of view, random graphs (with bounded number of nodes) are no different from any other discrete random variable. Indeed, a graph with P nodes can be described by $\binom{P}{2}$ binary variables A_{ij} that equal one if the i-th and j-th nodes (in some order) are connected by an edge, and zero otherwise. Notice that there may exist permutations of the nodes that leave the connectivity variables A_{ij} unchanged: these should be factored out if the nodes are indistinguishable.

There are many models of random graphs that are constructed in order to display certain properties: random graphs with constant degree, random graphs with a chosen degree distribution, random graphs with clustered nodes (i.e. with subsets of nodes having a higher degree of connections between them). I will not provide a review of all these models: see [FK15] for a complete survey, and [HL81] for a nice maximum-entropy approach to distributions of random graphs.

3.2.1 Unstructured graphs: the Erdös-Rényi model

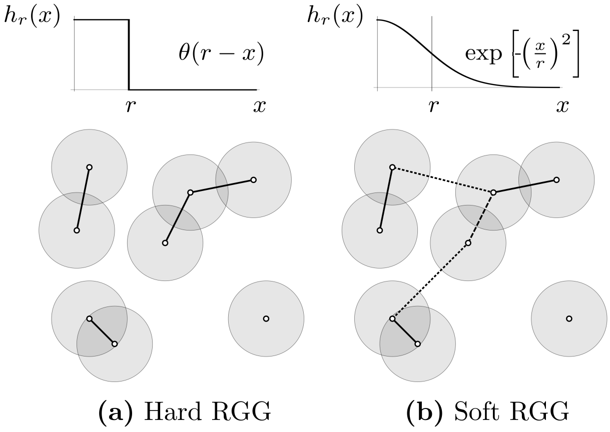
The simplest model of random graph is given by the Erdös-Rényi model [ER60]. Given a set of P nodes, an Erdös-Rényi graph is generated by connecting each pair of nodes with independent probability p.

The average density of M-cliques in an Erdös-Rényi graph (as a function of the connection probability p) is given by

$$\rho_M(p) = p^{\binom{M}{2}}\,. \tag{3.3}$$

3.2.2 Spatial networks: hard and soft random geometric graphs

Random spatial networks are usually called *random geometric graphs* (RGG). The monograph by Penrose [Pen03] is the bible of the topic. RGGs are widely used to model



(a) Hard RGG **(b)** Soft RGG

Figure 3.2: Example of hard and soft random geometric graph. Small circles denote nodes embedded in $\mathbb{R}^2$ drawn randomly with the uniform measure on $[0,1]^2$ and shaded circles highlight a region of radius $r/2$ around nodes. Solid lines highlight the actual edges of the represented graphs. On the top of the graph representations, the activation function used to build them are displayed. **(a)** In a hard random geometric graph at cutoff r, the only selected edges are those with nodes closer than r (in the picture, the nodes whose shaded regions intersect). **(b)** In a soft random geometric graph, edges are selected based on a continuous activation function $h_r(x)$. If two nodes are at distance d between each other, then the edge that connects them will be chosen with probability $h_r(d)$. In the picture, dotted edges are those edges that have been chosen by the soft random geometric graphs even though the distance between nodes was larger than r. Vice versa, dashed edges are those at distance smaller than r, but not selected in that specific instance of the soft random geometric graph. (Reprinted from [EAG+20]).

systems in which geometry plays a crucial role: transport networks [BLG17], wireless and 5G networks [GGD15] and social networks [Net13].

RGGs are defined by randomly extracting d-dimensional coordinates for each node, and by connecting with edges each pair of nodes whose mutual distance is lesser than a fixed threshold r. Thus, the randomness of the model is limited to the position of the nodes, and the connectivity structure of the graph is then determined by the mutual distance between points.

To be more precise, a random geometric graph is defined by:

- a random process that generates P points in $\mathbb{R}^d$. The positions of the points may in principle be correlated, and their probability density may be supported on complex domains. In most settings, and in the following, it is hard enough to consider the simplest case, in which all the positions of the points are i.i.d. random variables. Moreover, I will consider the *factorized* case, i.e. the probability distribution of the position of a point $\nu(x)$ is factorized in the coordinates, i.e.

$$\nu(x) = \prod_{i=1}^{d} \tau(x_i) \tag{3.4}$$

for some probability distribution τ on the real line with finite first and second moment. In this case, all the coordinates of all the nodes are i.i.d. random variables, with law τ. An example is given by RGGs on the hypercube, in which

$$\tau(x) = \theta(x)\theta(1-x) \tag{3.5}$$

where θ is the Heaviside step function. A notable exception to the factorized case is given by RGGs on the hypersphere, extensively treated in the literature of high-dimensional RGGs [DGL$^+$11; BDE$^+$16; AB20].

In low dimension, many models of RGGs on different geometries have been considered in the literature, see [ES15; All18] for example.

- a distance function $d(x, y)$ on $\mathbb{R}^d$. As always, Euclidean distance is the natural choice. In the following, we will consider the more general notion of p-norm distances $d(x, y) = ||x-y||_p^{\min(1,p)}$, where

$$||x||_p = \left[\sum_{i=1}^{d} |x_i|^p\right]^{\frac{1}{p}} . \tag{3.6}$$

  This is motivated by the observation that in some machine learning applications, using the $0 < p < 1$ pseudo-norms improves performances, especially in the high dimensional setting [FWV07];

- a linking probability $h(x)$, also called *activation function*, that determines the probability that a pair of nodes at distance x is connected be an edge. While other options are possible, we will consider activation function that are monotone decreasing, to model the fact that nearer nodes should be linked with higher probability, and such that $h(0) = 1$ and $h(\infty) = 0$. Usually, the activation function is labeled by a parameter $r \in \mathbb{R}^+$ that describes the typical distance at which a pair of nodes will be considered close enough to be linked with a nontrivial probability, for example $h_r(r) = \frac{1}{2}$. In this case, the statistical properties of RGGs can be investigated as functions of r.

  The case discussed in the brief introduction to this section is that of classic RGGs, or *hard* RGGs, in which the linking process is deterministic due to the activation function

$$h_r^{\text{hard}}(x) = \theta(r-x) . \tag{3.7}$$

  See Figure 3.2, left panel, for an example of hard RGG.

  Another possibility is given by *soft* RGGs, in which the activation function is continuous. A common example is given by the so-called Reyleigh fading activation function (often used in telecommunication modelling) [GGD15; KBD19]

$$h_r^{\text{Reyleigh}}(x) = \exp\left[-\xi\left(\frac{x}{r}\right)^{\eta}\right] \tag{3.8}$$

  where the choice $\xi = \log(2)$ enforces that $h_r(r) = \frac{1}{2}$. See Figure 3.2, right panel, for an example of soft RGG.

Of course, RGGs at finite dimension are quite different from Erdös-Rényi graphs (for the sake of comparison, consider Erdös-Rényi graphs embedded in Euclidean space by

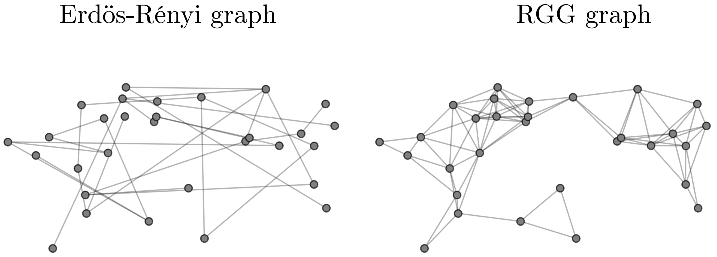


Figure 3.3: Comparison between an Erdös-Rényi graph embedded in $2d$ and an hard RGG. On the same set of 30 nodes embedded in the $2d$ unit square, we construct an Erdös-Rényi graph (left) with connection probability $p = 0.1$ and an hard RGG (right) with connection radius $r = 0.3$. The Euclidean structure of the RGG is evident: nearer nodes are linked, and further nodes are not. On the other hand, the edges of the Erdös-Rényi graph have no correlation with the Euclidean position of the nodes, even at small values of the connection probability.

assigning random positions to their nodes): the former typically have a higher number of connections between nearby nodes, while the latter has no such constraint. See Figure 3.3 for an example.

To get an intuition on what may happen in high-dimension, and to why Erdös-Rényi graphs are a natural choice for comparison, notice that in high dimension Euclidean norms, distances and angles tend to concentrate (this will be made precise in what follows), so that the geometrical structure of finite sets of points trivializes into sets of orthogonal points lying on the surface of a sphere. With this in mind, one may guess that RGGs could become less and less structured as the dimension grows, converging to Erdös-Rényi graphs.

This is what is found in a series of technical papers [DGL+11; BDE+16; AB20] that consider the case of hard RGGs on the sphere. The authors there prove that RGGs on the sphere converge to Erdös-Rényi graphs in the *total variation distance*, that is

$$d_{\mathrm{TV}}(G, G') = \sum_{g} |\, \mathrm{Prob}(G = g) - \mathrm{Prob}(G' = g)| \,, \tag{3.9}$$

where G, G' are two random geometric graphs defined on a shared set of nodes, and the sum runs over all graphs g on the same set of nodes. They also note that while this convergence criterion is extremely strong, the rate of convergence provided by their proofs is quite poor, requiring that the dimension d of the RGG scales as $d \sim 2^{P^2}$ in the number of nodes P. Nonetheless, simple graph observables may converge much faster, as they explicitly show in the case of the clique number, i.e. the size of the largest clique in a graph.

On the other hand, for the case of hard RGGs on the hypercube, it seems that RGGs retain some geometrical information in the high-dimensional limit. In [DC02], the authors study the cluster coefficient of RGGs, i.e. the probability that two nodes sharing a common neighbour are connected, and observe that in the high-dimensional limit it is different from that of Erdös-Rényi graphs with comparable connectivity.

Thus, it seems that the high-dimensional limit of hard RGGs is delicate.

3.3 A central limit theorem for distances in high dimension

The first step towards understanding the behaviour of RGGs in high dimension is to understand how the set of mutual distances between M[1] points behaves as the dimension grows to infinity. In other words, we would like to compute

$$\Pi(d_{(1,2)}, d_{(1,3)}, \dots d_{(M-1,M)}) = \int \prod_{\rho=1}^{M} \nu(x^\rho) dx^\rho \prod_{1 \leqslant \rho < \sigma \leqslant M} \delta\left(d_p(x^\rho, x^\sigma) - d_{(\rho,\sigma)}\right) , \quad (3.10)$$

that is the probability that the $\binom{M}{2}$ distances between the M nodes have values $d_{(\rho,\sigma)}$ for $1 \leqslant \rho < \sigma \leqslant M$. Recall that the random positions of the nodes are controlled by the probability density $\nu(x)$ which is factorized over coordinates as in Equation (3.4).

3.3.1 The central limit theorem

Due to the factorization of $\nu(x)$, each distance $d_p(x^\rho, x^\nu)$ is a function of the sum of d i.i.d. random variables, that we expect to concentrate to the mean due to the law of large numbers. Thus, it's natural to define the centered and rescaled variables

$$q_{(\rho,\sigma)} = \frac{[d_p(x^\rho, x^\sigma)]^{\max(1,p)} - d\mu}{\sqrt{d}} = \frac{1}{\sqrt{d}} \sum_{k=1}^{d} \left(|x_k^\rho - x_k^\sigma|^p - \mu\right) = \frac{1}{\sqrt{d}} \sum_{k=1}^{d} q_{(\rho,\sigma)}^k \quad (3.11)$$

where

$$\mu = \int dx\, dy\, \tau(x)\tau(y)|x-y|^p . \quad (3.12)$$

Again, notice that $q_{(\rho,\sigma)}^k$ are, at fixed $1 \leqslant \rho < \sigma \leqslant M$, i.i.d. random variables with null mean.

By the multivariate central limit theorem (CLT), in the limit $d \to \infty$, the distribution of the vector $\boldsymbol{q} = (q_{(1,2)}, q_{(1,3)}, \dots, q_{(M-1,M)})$[2] converges to a multivariate Gaussian distribution with null mean and covariance $\boldsymbol{\Sigma}_{(\rho,\sigma),(\eta,\zeta)} = \left\langle q_{(\rho,\sigma)}^1 q_{(\eta,\zeta)}^1 \right\rangle$, where the average is taken over the values of $x_1^\rho, x_1^\sigma, x_1^\eta, x_1^\zeta$ all distributed with law $\tau(x)$.

The general proof of the multivariate CLT can be found in [Van00, Proposition 2.17]. Here, I sketch a more down-to-the-earth argument.

The probability that the rescaled distances $q_{(\rho,\sigma)}$ assume values $h_{(\rho,\sigma)}$ is defined as:

[1] I don't use P for the number of nodes in this section in order to stress that here the number of nodes is strictly finite.

[2] I will use boldface to denote vectors or matrices whose indices run over pairs of points, in contrast with standard Euclidean vectors such as the positions of the points.

$$\begin{aligned}&\mathrm{Prob}\left(\left\{\frac{1}{\sqrt{d}}\sum_{k=1}^{d} q^k_{(\rho,\sigma)} = h_{(\rho,\sigma)}\right\}_{\forall\, 1\leqslant\rho<\sigma\leqslant M}\right)\\ &\quad= \int \prod_{k=1}^{d}\prod_{\mu=1}^{M}\Big(dx^\mu_k\,\tau(x^\mu_k)\Big)\prod_{\rho<\sigma}\delta\Big(h_{(\rho,\sigma)} - \frac{1}{\sqrt{d}}\sum_{k=1}^{d} q^k_{(\rho,\sigma)}\Big)\,,\end{aligned} \tag{3.13}$$

where we recall that $q^k_{(\rho,\sigma)}$ is a function of x^ρ_k and x^σ_k. Using the Fourier representation of Dirac's delta function, we obtain:

$$\begin{aligned}&\mathrm{Prob}\left(\left\{\frac{1}{\sqrt{d}}\sum_{k=1}^{d} q^k_{(\rho,\sigma)} = h_{(\rho,\sigma)}\right\}_{\forall\, 1\leqslant\rho<\sigma\leqslant M}\right)\\ &= \int \prod_{k=1}^{d}\prod_{\mu=1}^{M}\Big(dx^\mu_k\,\tau(x^\mu_k)\Big)\int\prod_{\rho<\sigma}\frac{d\lambda_{(\rho,\sigma)}}{2\pi}\exp\left(i\lambda_{(\rho,\sigma)}\Big(h_{(\rho,\sigma)} - \frac{1}{\sqrt{d}}\sum_{k=1}^{d} q^k_{(\rho,\sigma)}\Big)\right)\\ &= \int \mathcal{D}\lambda \prod_{k=1}^{d}\left[\int\prod_{.\mu=1}^{M}\Big(dx^\mu_k\,\tau(x^\mu_k)\Big)\exp\Big(-\frac{i}{\sqrt{d}}\sum_{\rho<\sigma}\lambda_{(\rho,\sigma)}q^k_{(\rho,\sigma)}\Big)\right],\end{aligned} \tag{3.14}$$

where we have defined $\mathcal{D}\lambda = \prod_{\rho<\sigma}\frac{d\lambda_{(\rho,\sigma)}}{2\pi}e^{i\lambda_{(\rho,\sigma)}h_{(\rho,\sigma)}}$. Now we use the fact that k is mute index, so that

$$\begin{aligned}&\mathrm{Prob}\left(\left\{\frac{1}{\sqrt{d}}\sum_{k=1}^{d} q^k_{(\rho,\sigma)} = h_{(\rho,\sigma)}\right\}_{\forall\, 1\leqslant\rho<\sigma\leqslant M}\right)\\ &= \int \mathcal{D}\lambda\left[\int\prod_{\mu=1}^{M}\Big(dx^\mu\,\tau(x^\mu)\Big)\exp\Big(-\frac{i}{\sqrt{d}}\sum_{\rho<\sigma}\lambda_{(\rho,\sigma)}\Big(|x^\rho - x^\sigma|^p - \mu\Big)\Big)\right]^d.\end{aligned} \tag{3.15}$$

If M is not scaling to infinity with d, so that the sum in the exponential is finite, and d is large, we can expand to second order the exponential, obtaining

$$\begin{aligned}&\mathrm{Prob}\left(\left\{\frac{1}{\sqrt{d}}\sum_{k=1}^{d} q^k_{(\rho,\sigma)} = h_{(\rho,\sigma)}\right\}_{\forall\, 1\leqslant\rho<\sigma\leqslant M}\right)\\ &\simeq \int \mathcal{D}\lambda\Bigg[\int\prod_{\mu=1}^{M}\Big(dx^\mu\,\tau(x^\mu)\Big)\Big(1 - \frac{i}{\sqrt{d}}\sum_{\rho<\sigma}\lambda_{(\rho,\sigma)}\Big(|x^\rho - x^\sigma|^p - \mu\Big)\\ &\qquad - \frac{1}{2d}\sum_{\rho<\sigma}\sum_{\eta<\zeta}\lambda_{(\rho,\sigma)}\lambda_{(\eta,\zeta)}\Big(|x^\rho - x^\sigma|^p - \mu\Big)\Big(|x^\eta - x^\zeta|^p - \mu\Big)\Big)\Bigg]^d\end{aligned} \tag{3.16}$$

The integrals in $dx^\mu\tau(x^\mu)$ can be explicitly solved. The linear term is null due to the definition of μ, and in the quadratic term the integrals give exactly $\mathbf{\Sigma}_{(\rho,\sigma),(\eta,\zeta)}$.

Moreover, we can now re-sum the terms in to an exponential, obtaining

$$\begin{aligned}\text{Prob}&\left(\left\{\frac{1}{\sqrt{d}}\sum_{k=1}^{d} q^k_{(\rho,\sigma)} = h_{(\rho,\sigma)}\right\}_{\forall\, 1\leqslant\rho<\sigma\leqslant M}\right)\\ &\simeq \int \frac{d\mathbf{\Lambda}}{2\pi} e^{i\mathbf{\Lambda}^T\boldsymbol{H}} e^{-\mathbf{\Lambda}^T\mathbf{\Sigma}\mathbf{\Lambda}} = \frac{e^{-\frac{1}{2}\boldsymbol{H}^T\mathbf{\Sigma}^{-1}\boldsymbol{H}}}{\sqrt{(2\pi)^{\binom{M}{2}}\det(\mathbf{\Sigma})}}\,,\end{aligned} \tag{3.17}$$

where we introduced bold symbols to denote vectors indexed by pairs of points.

This computation proves that, in the limit of dimension which tends to infinity, the probability distribution of the normalized distances among M points tends to a $\binom{M}{2}$-dimensional multivariate normal distribution with mean 0 and covariance matrix given by $\mathbf{\Sigma}$. This matrix is the most important object in our result, so in the following sections we study in depth its properties.

It is worth noticing that the corrections to the $d \to \infty$ limit are of order $\frac{1}{\sqrt{d}}$, and may depend on M. Thus, this $d \to \infty$ limit is performed at fixed M, and the result can be used either to treat generic observables for graphs where the total number of nodes is fixed, or to treat observables that depend only on a finite number of nodes at a time in graphs where the total number of nodes may scale with d. Moreover, the CLT presented above holds for the variable $\boldsymbol{q}$, and not for the actual distances. However this is not an issue, as the joint distribution for distances can be derived by a simple coordinate change, factorized over each direction. Moreover, as we will see in the following, it is often easy to obtain the observables of interest in terms of the $\boldsymbol{q}$ variable.

3.3.2 The structure of the covariance matrix

The structure of the covariance matrix $\mathbf{\Sigma}$ is quite peculiar. Recall that

$$\mathbf{\Sigma}_{(\rho,\sigma),(\eta,\zeta)} = \big\langle(|y^\rho - y^\sigma|^p - \mu)(|y^\eta - y^\zeta|^p - \mu)\big\rangle \tag{3.18}$$

where all the y's are distributed with law τ.

By permutational symmetry, there are only three different matrix elements:

- **Diagonal correlations ($\rho = \eta$ and $\sigma = \zeta$)**

$$\alpha := \mathbf{\Sigma}_{(\rho,\sigma),(\rho,\sigma)} = \int dx\, dy\, \tau(x)\tau(y)|x-y|^{2p} - \mu^2\,; \tag{3.19}$$

- **Triangular correlations ($\rho = \eta$ and $\sigma \neq \zeta$ or $\rho \neq \eta$ and $\sigma = \zeta$)**

$$\begin{aligned}\beta &:= \mathbf{\Sigma}_{(\rho,\sigma),(\rho,\zeta)} = \mathbf{\Sigma}_{(\rho,\sigma),(\eta,\sigma)}\\ &= \int dx\, dy\, dz\, \tau(x)\tau(y)\tau(z)|x-y|^p|x-z|^p - \mu^2\,;\end{aligned} \tag{3.20}$$

- **Pair-pair correlations ($\rho, \sigma, \eta, \zeta$ are all distinct)**

$$\gamma := \mathbf{\Sigma}_{(\rho,\sigma),(\eta,\zeta)} = \left(\int dx\, dy\, \tau(x)\tau(y)|x-y|^p\right)^2 - \mu^2 = 0\,. \tag{3.21}$$

Notice that $\gamma = 0$ due to the definition of μ.

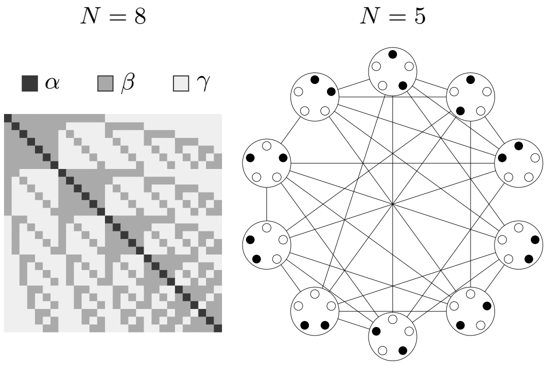


Figure 3.4: (Left) Example of a matrix $\boldsymbol{\Delta}(N, \alpha, \beta, \gamma)$ for N=8. The entries with value equal to β have the same structure of the adjacency matrix of the Johnson graph. (Right) Example of Johnson graph with N=5. The Johnson graph $J(N, 2)$ is the line graph of the complete graph over N nodes. It has all the distinct pairs of the original nodes as its vertices, and its vertices are linked if their pairs share an original node. (Reprinted from [EAG+20]).

In the case of simple distributions $\tau(x)$, we can explicitly evaluate μ, α and β.

Uniform distribution on hypercube In this case, $\tau(x) = \theta(x)\theta(1-x)$, and

$$\begin{aligned}
\mu^{\text{cube}} &= \frac{2}{(p+1)(p+2)}\,, \\
\alpha^{\text{cube}} &= \frac{p^2(p+5)}{(p+1)^2(p+2)^2(2p+1)}\,, \\
\beta^{\text{cube}} &= \frac{2}{(p+1)^2}\left(\frac{p^2-2}{(p+2)^2(2p+3)} + \frac{\Gamma^2(p+2)}{\Gamma(2p+4)}\right),
\end{aligned} \tag{3.22}$$

where $\Gamma(x)$ is the Euler gamma function.

The general form of a matrix with the symmetries of $\boldsymbol{\Sigma}$ is given by (see Figure 3.4).

$$\begin{aligned}
\boldsymbol{\Delta}_{(\rho,\sigma)(\eta,\zeta)}(M, \alpha, \beta, \gamma) = {} & (\alpha - 2\beta + \gamma)\delta_{\rho,\eta}\delta_{\sigma,\zeta} \\
& + (\beta - \gamma)(\delta_{\rho,\eta} + \delta_{\rho,\zeta} + \delta_{\sigma,\eta} + \delta_{\sigma,\zeta}) + \gamma\,,
\end{aligned} \tag{3.23}$$

where $\delta_{i,j}$ is the Kronecker delta, and $\binom{M}{2} \times \binom{M}{2}$ is the size of the matrix. We collect here some properties of these kind of matrices, mainly regarding its spectral decomposition and its inverse which are useful to numerically sample the multivariate Gaussian distribution of Section 3.3.1.

An easier decomposition The matrix $\boldsymbol{\Delta}$ can be more easily manipulated when rewritten as

$$\boldsymbol{\Delta} = (\alpha - \gamma)\boldsymbol{I} + (\beta - \gamma)\boldsymbol{J} + \gamma\boldsymbol{U} \tag{3.24}$$

where $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{U}$ is the matrix with unit entries and $\boldsymbol{J}$ is the adjacency matrix (with null diagonal) of the Johnson graph $J(M, 2)$. The Johnson graph $J(M, 2)$

is the so-called *line graph* of the complete graph over M vertices, (see Figure 3.4, right panel). It has all the distinct pairs of the original nodes as its vertices, and its vertices are linked if the corresponding pairs share an original node.

Notice that the three matrices commute. The only non-trivial pair is $(\boldsymbol{J}, \boldsymbol{U})$; the two matrices commute as $\boldsymbol{J}$ satisfies

$$\sum_k \boldsymbol{J}_{ik} = \sum_k \boldsymbol{J}_{kj} = 2(M-2) \tag{3.25}$$

for all rows i and columns j, as all vertices of the Johnson graph $J(M,2)$ have degree $2(M-2)$.

Spectrum As $\boldsymbol{I}$, $\boldsymbol{J}$ and $\boldsymbol{U}$ commute, they can be simultaneously diagonalized. The contribution of $\boldsymbol{I}$ to the spectrum is trivial, shifting all the eigenvalues by a constant. $\boldsymbol{J}$ and $\boldsymbol{U}$ share a common non-degenerate eigenvector, the one with all unit coordinates. In the corresponding 1d subspace, $\boldsymbol{J}$ acts by multiplying by $2(M-2)$, and $\boldsymbol{U}$ by multiplying by $\binom{M}{2}$. In the orthogonal subspace, i.e. the space of vectors with null sum of coordinates, $\boldsymbol{U}$ is represented by the null operator. Thus, the spectrum is fully determined by that of $\boldsymbol{J}$ [Bur17], which is given by an $(M-1)$-degenerate eigenvalue $\lambda_2^{\boldsymbol{J}} = M-4$ and by a $M(M-3)/2$-degenerate eigenvalue $\lambda_3^{\boldsymbol{J}} = -2$.

To summarize, the spectrum of the matrix $\Delta(M, \alpha, \beta, \gamma)$ is given by

- $\lambda_1 = \alpha + 2(M-2)\beta + \frac{(M-2)(M-3)}{2}\gamma$ with multiplicity 1;
- $\lambda_2 = \alpha + (M-4)\beta - (M-3)\gamma$ with multiplicity $M-1$;
- $\lambda_3 = \alpha - 2\beta + \gamma$ with multiplicity $\frac{M(M-3)}{2}$;

Trace and determinant The spectrum gives immediate access to the trace and the determinant of $\boldsymbol{\Delta}$:

$$\begin{aligned} \mathrm{Tr}(\boldsymbol{\Delta}) &= \binom{M}{2}\alpha\,, \\ \det(\boldsymbol{\Delta}) &= \lambda_1 \lambda_2^{M-1} \lambda_3^{\frac{M(M-3)}{2}}\,. \end{aligned} \tag{3.26}$$

Inverse matrix The inverse matrix $\boldsymbol{\Delta}^{-1}$ can be explicitly computed by noticing that $\boldsymbol{\Delta}$ and $\boldsymbol{\Delta}^{-1}$ share the same eigenvectors, and thus

$$\boldsymbol{\Delta}^{-1}(M, \alpha, \beta, \gamma) = \boldsymbol{\Delta}(M, \alpha', \beta', \gamma')\,, \tag{3.27}$$

where the new parameters α', β' and γ' are functions of the old parameters α, β and γ, and can be determined by solving the linear system

$$\lambda_i(\alpha', \beta', \gamma') = \frac{1}{\lambda_i(\alpha, \beta, \gamma)}\,, \tag{3.28}$$

for $i = 1, 2, 3$.

3.4 M-clique densities in random geometric graphs

We are now ready to compute observables on random geometric graphs in the limit of infinite dimensions; in particular, we aim to characterize the average number of subgraph with a given structure.

3.4.1 Average number of generic subgraphs

In general, the average number of a certain subgraph G with M nodes of a random geometric graph with P nodes can be factored in two terms. The first one is a combinatorial factor $\binom{P}{M}$, that accounts for the number of ways in which one can extract M nodes from a set of P of them.

The second one is the so-called density $\rho_G(r)$ of the subgraph G at scale r, that is the probability that M random points are close enough with respect to the cutoff radius r to form a subgraph with the same adjacency matrix of G. Recalling the definition of the joint probability of the distances between M points given in Equation (3.10), we have that

$$\rho_G(r) = \int d\boldsymbol{y}\, \Pi(\boldsymbol{y}) \prod_{1 \leqslant \rho < \sigma \leqslant M} \left[h_r\left(y_{(\rho,\sigma)}\right)\right]^{A_{\rho\sigma}(G)} , \tag{3.29}$$

where $y_{(\rho,\sigma)}$ is the distance between nodes ρ and σ, and A is the adjacency matrix of the subgraph. We can rescale the variables $y_{(\rho,\sigma)}$ as in Equation (3.11), and exploit the fact that for large dimension $d\boldsymbol{y}\, \Pi(\boldsymbol{y}) \sim d\boldsymbol{q}\, \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})(\boldsymbol{q})$ (here $\mathcal{N}$ is a multivariate Gaussian, see Section 3.3.1) to obtain an expression for $\rho_G(r)$ that is valid in the limit of large dimension:

$$\begin{aligned} \rho_g(r) &= \int d\boldsymbol{y}\, \Pi(\boldsymbol{y}) \prod_{1 \leqslant \rho < \sigma \leqslant M} \left[h_r\left(y_{(\rho,\sigma)}\right)\right]^{A_{\rho\sigma}(G)} \\ &\sim \int d\boldsymbol{q}\, \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})(\boldsymbol{q}) \prod_{1 \leqslant \rho < \sigma \leqslant M} \left[h_r\left(\left[d\,\mu + \sqrt{d}\, q_{(\rho,\sigma)}\right]^{\min\left(1,\frac{1}{p}\right)}\right)\right]^{A_{\rho\sigma}(G)} , \end{aligned} \tag{3.30}$$

where $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ is the multivariate Gaussian with null mean and covariance $\boldsymbol{\Sigma}$ (given in Equation (3.18)), i.e.

$$\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})(\boldsymbol{q}) = \frac{e^{-\frac{1}{2}\boldsymbol{q}^T \boldsymbol{\Sigma} \boldsymbol{q}}}{\sqrt{(2\pi)^{\binom{M}{2}} \det \boldsymbol{\Sigma}}} . \tag{3.31}$$

In the rest of the section, all results hold in the limit of large dimension.

I want to stress again that this high-dimensional limit is valid only when M is not scaling to infinity as fast as d. This is always the case if the class of subgraphs G considered has a fixed finite number of vertices M.

3.4.2 The case of M-cliques

As a paradigmatic example, we consider the average density of M-cliques $\rho_M(r)$, i.e. fully connected subgraphs with M vertices, on random geometric graphs with generic activation function $h_r(x)$; in this specific case, $A_{\rho\sigma}$ has only unit entries, so that

$$\rho_M(r) = \int d\boldsymbol{q}\, \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})(\boldsymbol{q}) \prod_{1 \leqslant \rho < \sigma \leqslant M} h_r\left(\left[d\,\mu + \sqrt{d}\, q_{(\rho,\sigma)}\right]^{\min\left(1,\frac{1}{p}\right)}\right) . \tag{3.32}$$

We now distinguish the two main classes of RGGs: hard RGGs, where the activation function is discontinuous, and soft RGGs with continuous activation function.

Hard RGGs In the case of hard activation function h^{hard}, we observe that

$$\begin{aligned} h_r^{\text{hard}}(x) &= h_{r^p}^{\text{hard}}(x^p) \\ h_r^{\text{hard}}(x+c) &= h_{r-c}^{\text{hard}}(x)\,, \quad \forall c \in \mathbb{R} \\ h_r^{\text{hard}}(x) &= h_{c\,r}^{\text{hard}}(c\,x)\,, \quad \forall c \in \mathbb{R}^+ \end{aligned} \tag{3.33}$$

so that the p-th root can be discarded along with a factor of $\sqrt{d}$, and the integral reduces to

$$\rho_M^{\text{hard}}(r) = \overline{\rho}_M^{\text{hard}}\left(\frac{r^{\max(1,p)} - d\,\mu}{\sqrt{d}}\right)\,, \tag{3.34}$$

with

$$\begin{aligned} \overline{\rho}_M^{\text{hard}}(x) &= \int d\boldsymbol{q}\,\mathcal{N}(\boldsymbol{0},\boldsymbol{\Sigma})(\boldsymbol{q}) \prod_{1\leqslant\rho<\sigma\leqslant M} h_x^{\text{hard}}\left(q_{(\rho,\sigma)}\right) \\ &= \int d\boldsymbol{q}\,\mathcal{N}(\boldsymbol{0},\boldsymbol{\Sigma})(\boldsymbol{q}) \prod_{1\leqslant\rho<\sigma\leqslant M} \theta\left(x - q_{(\rho,\sigma)}\right) \end{aligned} \tag{3.35}$$

which is a multivariate Gaussian cumulative distribution function. Equation (3.34) highlights the simple dependence of $\overline{\rho}_M^{\text{hard}}$ on the parameters p, d and μ.

In the case $M = 2$, the integral in Equation (3.35) can be explicitly solved as it reduces to the computation of the standard error function, giving

$$\overline{\rho}_2^{\text{hard}}(x) = \frac{1}{2}\left[1 + \text{Erf}\left(\frac{x}{\sqrt{2\alpha}}\right)\right]\,. \tag{3.36}$$

In the case $M > 2$, the integral in Equation (3.36) admits an explicit solution only for special and simple forms of the covariance matrix Σ, which do not match our case.

Soft RGGs In the case of soft random geometric graphs with continuous activation functions, one can expand $h_r(x)$ to the 0-th order in powers of $1/\sqrt{d}$ for any value of r, obtaining that in the limit of high dimension

$$\begin{aligned} \rho_M^{\text{soft}}(r) &= \left[\rho_2^{\text{soft}}(r)\right]^{\binom{M}{2}}\,, \\ \rho_2^{\text{soft}}(r) &= h_r\left((d\mu)^{\min\left(1,\frac{1}{p}\right)}\right)\,. \end{aligned} \tag{3.37}$$

In the special case of Rayleigh fading activation function h^{rayleigh}, one has

$$\rho_2^{\text{rayleigh}}(r) = \exp\left[-\xi\left(\frac{d\mu}{r}\right)^{\eta\min\left(1,\frac{1}{p}\right)}\right]\,. \tag{3.38}$$

Intuitively, the difference between hard and soft RGGs depends on the freedom in performing the rescaling of the cutoff radius in the former case [see Equation (3.34)], which is lost in the latter, and on the regularity of the activation function, which allows to take the high-dimensional limit in the straight-forward way.

3.4.3 Comparison between hard RGGs, soft RGGs and Erdös-Rényi graphs

It's useful to notice that $\rho_2(r)$ can be interpreted as the probability that two random nodes in the RGG will be linked at scale r. $\rho_2(r)$ is monotone increasing in r, so that its inverse is well defined.

The function $r(x) = (\rho_2)^{-1}(x)$ allows to re-parametrize RGGs by expressing the connectivity scale r as a function of the probability x that two nodes are linked, which is analogous to the connection probability p in Erdös-Rényi graphs (here we use x to denote it to avoid confusion with p-norms). This highlights a common ground to compare RGGs and Erdös-Rényi graphs. In particular, we introduce the modified M-clique densities

$$\omega_M(x) = (\rho_M \circ (\rho_2)^{-1})(x)\,, \tag{3.39}$$

which measure the probability of observing an M-clique as a function of the probability that two nodes are linked. Notice that the graph of $\omega_M(x)$ lies in the square $[0,1]\times[0,1]$, and that, in practice, it can be easily plotted by producing a scatter plot of $\rho_M(r)$ versus $\rho_2(r)$ as r grows. Notice that the recent literature on RGGs on hyperspheres [DGL+11; BDE+16; AB20] adopts this parametrization too.

In the case of hard RGGs, the simple dependence of $\rho_M(r)$ on p, d and μ allows to write

$$\omega_M^{\text{hard}}(x) = (\rho_M^{\text{hard}} \circ (\rho_2^{\text{hard}})^{-1})(x) = (\overline{\rho}_M^{\text{hard}} \circ (\overline{\rho}_2^{\text{hard}})^{-1})(x)\,, \tag{3.40}$$

where we notice that the dependence of ω_M on p, d and μ cancels out.

In the case of soft RGGs, we have that $\omega_M(x)$ reduces to the simple form of Erdös-Rényi graphs (see Equation (3.41)), i.e.

$$\omega_M^{\text{soft}}(x) = \omega_M^{\text{ER}}(x) = x^{\binom{M}{2}}\,. \tag{3.41}$$

The question now is simple: does $\omega_M^{\text{hard}}(x) = \omega_M^{\text{soft}}(x)$ due to some magical simplifications? We have to resort to numerical simulations to answer this question, but it's easy to expect that the answer will be a strong no, as multivariate Gaussian cumulative distributions are complex objects.

Numerical simulations To compare RGGs and Erdös-Rényi graphs by computing $\omega_M(x)$, we have to numerically compute the multivariate Gaussian cumulative function of Equation (3.35). While this is not difficult, it requires more labour than a naïve Monte Carlo integration. In fact, the dimension of the integration domain grows as $\sim M^2$ if M is the clique order, and the performance of naïve Monte Carlo integration degrades quickly as the dimension grows. To numerically approximate Equation (3.35), we resorted to the procedure described in [Gen92], which I summarize here for completeness.

We wish to compute numerically integrals of the kind

$$F(a,b) = \int_{a_1}^{b_1}\int_{a_2}^{b_2}\dots\int_{a_m}^{b_m} d\theta\, \frac{e^{-\frac{1}{2}\theta^T\Sigma^{-1}\theta}}{\sqrt{(2\pi)^m \det(\Sigma)}}\,, \tag{3.42}$$

where a and b are m-dimensional vectors defining the integration domain, possibly with some components equal to $\pm\infty$, and θ is a m-dimensional integration variable. We wish to compute the integral by Monte Carlo sampling. In this form, this is not very efficient for large m, as the integration domain may be infinite and as the region in which the integrand is non-null may be difficult to sample. The idea is to perform a series of manipulations on the integral in order to trivialize the integrand while avoiding to complicate too much the integration domain.

Without entering too much into the details, the manipulations are the following:

1. compute the Cholesky decomposition C [Cho05] of Σ, and perform the change of variable $\theta = Cy$. This allows to factorize the integrand into the variables $y_1, \ldots, y_m$, but couples the integration variables with the integration boundaries
$$F(a,b) = \frac{1}{\sqrt{(2\pi)^m}} \int_{a'_1}^{b'_1} e^{-\frac{y_1^2}{2}} \int_{a'_2(y_1)}^{b'_2(y_1)} e^{-\frac{y_2^2}{2}} \ldots \int_{a'_m(y_1,\ldots,y_{m-1})}^{b'_m(y_1,\ldots,y_{m-1})} e^{-\frac{y_m^2}{2}} dy \tag{3.43}$$
where
$$\begin{aligned} a'_i(y_1,\ldots,y_{i-1}) &= \frac{a_i - \sum_{j=1}^{i-1} C_{ij} y_j}{C_{ii}} \, , \\ b'_i(y_1,\ldots,y_{i-1}) &= \frac{b_i - \sum_{j=1}^{i-1} C_{ij} y_j}{C_{ii}} \, . \end{aligned} \tag{3.44}$$

2. perform the change of variable $y_i = \Phi^{-1}(z_i)$, where
$$\Phi(y) = \int_{-\infty}^{y} \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}} dx \, . \tag{3.45}$$
This trivializes the integrand, while hiding all the complexity into the integration boundaries
$$F(a,b) = \int_{d_1}^{e_1} \int_{d_2(z_1)}^{e_2(z_1)} \ldots \int_{d_m(z_1,\ldots,z_{m-1})}^{e_m(z_1,\ldots,z_{m-1})} dz \, , \tag{3.46}$$
where
$$\begin{aligned} d_i(z_1,\ldots,z_{i-1}) &= \Phi\left(\frac{a_i - \sum_{j=1}^{i-1} C_{ij}\Phi^{-1}(z_j)}{C_{ii}}\right) , \\ e_i(z_1,\ldots,z_{i-1}) &= \Phi\left(\frac{b_i - \sum_{j=1}^{i-1} C_{ij}\Phi^{-1}(z_j)}{C_{ii}}\right) . \end{aligned} \tag{3.47}$$

3. finally, we put the integral in a constant-limit form by performing the change of variable $z_i = d_i + w_i(e_i - d_i)$. This introduces a natural priority ordering on the integration variables w_i that allows Monte Carlo sampling to be more effective, and lowers the number of integration variables by one unit
$$F(a,b) = (e_1 - d_1) \int_0^1 (e_2 - d_2) \ldots \int_0^1 (e_m - d_m) dw \, . \tag{3.48}$$

Notice that if $a_i = -\infty$, $d_i = 0$, and if $b_i = +\infty$, $e_i = 1$.

In this form, the integral is more suitable for Monte Carlo sampling, giving an easy algorithmic procedure to evaluate the integral.

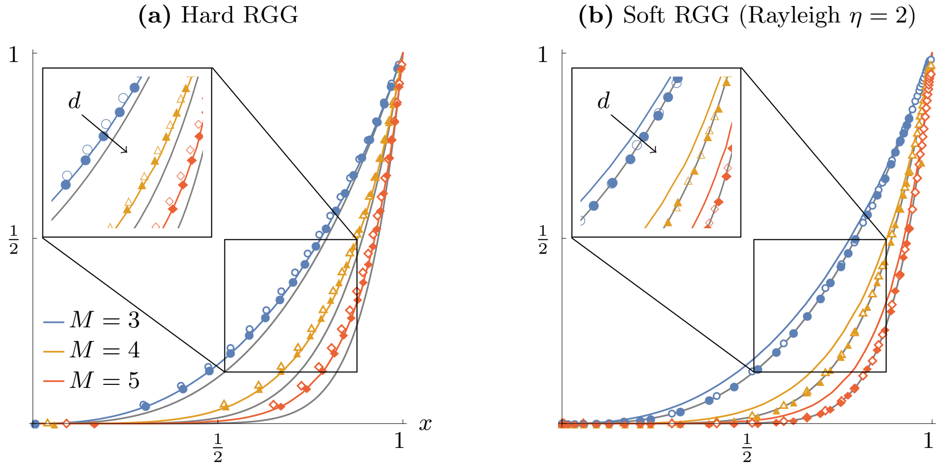


Figure 3.5: Comparison between finite d simulations and infinite d analytical predictions. Colored solid lines represent the analytical predictions for $\omega_M^{\text{hard}}(x)$ obtained from Equation (3.35), for $M = 3, 4, 5$ (blue (first line from above), orange (third line from above) and red (fifth line from above) respectively). Gray solid lines represent $\omega_M^{\text{ER}}(x)$ for the Erdös-Rényi graph (Equation (3.41)) for comparison for the same values of $M = 3, 4, 5$ (second, fourth and sixth line from above respectively). Open and filled markers show numerical simulations at $d = 20, 200$ respectively, $p = 2$ and $\nu = \nu^{\text{cube}}$, for hard RGG **(a)** and soft RGG with Rayleigh activation function $\eta = 2$ **(b)**, for the same values of $M = 3, 4, 5$ (circles, triangles and diamonds respectively). In practice, $\omega_M(x)$ can be represented by producing a scatter plot of $\rho_M(r)$ versus $\rho_2(r)$. (Reprinted from [EAG+20]).

The results are presented in Figure 3.5, where Equation (3.35), Equation (3.41) and numerical experiments on hard and soft RGGs are compared. The two main messages are:

1. hard RGGs behave differently from soft RGGs and Erdös-Rényi graphs in high dimension. In particular, for any linking probability x, the probability of finding an M-clique is systematically higher in the hard case;

2. numerical experiments on hard and soft RGGs confirm the goodness of the theoretical analysis presented in the previous sections. Notice that the agreement between theory and experiments starts at very low dimension $d \sim 20$, suggesting that the high-dimensional limit (and its finite dimensional corrections) may be used to study perturbatively RGGs in low dimension. More quantitatively, we observe relative deviations from analytical predictions are of the order of $\sim 10\%$ at $d = 20$, and $\sim 2\%$ at $d = 200$ for both hard and soft RGGs and $M = 3$. Larger $M = 4, 5$ shows higher relative errors, possibly due to the harder random sampling procedure required by larger cliques.

Thus, M-clique densities can be used as an observable to detect high-dimensional geometry in hard RGGs. High-dimensional soft RGGs instead become more and more indistinguishable from Erdös-Rényi graphs as the dimension grows, at least in the sense of convergence of the average M-clique densities.

In the following, I give some details on the numerical experiments.

Hard RGG To compute the density of M-cliques in simulated hard RGGs, we implemented a simple random sampling procedure, as exhaustive enumeration of M-cliques scales poorly, i.e. as $\mathcal{O}\left(P^M\right)$, with the total number of nodes P. For each realization of the nodes (with ν^{cube} and $P = 10^4$), we extracted $\sim 5 \cdot 10^5$ M-uples of nodes, computing the minimum cutoff distance at which they formed a clique. The cumulative distribution of the minimal distances obtained, averaged over different realization of the nodes, reconstructs $\rho_M^{\text{hard}}(r)$. We noticed that as P grows, the last average is well approximated by a single realization of the nodes, suggesting a self-averaging property for the density of M-cliques; in practice, not averaging does not affect the results of the simulations.

Soft RGG To compute the density of cliques in simulated soft RGGs with generic activation function, we implemented again a random sampling procedure. This time, for each realization of the nodes (as above) and for a fixed radius r, we counted how many of $\sim 10^4$ M-uples of nodes $\{y_i\}_{i=1}^M$ where M-cliques, considering each of them to be a M-clique with probability

$$\prod_{1 \leqslant \rho < \sigma \leqslant M} h_r(d(\vec{y}_\rho, \vec{y}_\sigma)) \,. \tag{3.49}$$

Normalizing the count over the total number of candidate cliques and averaging over different realizations of the nodes (order 10^2) gives an empirical estimation for $\rho_M(r)$ in the soft case.

3.5 Perspectives

In this chapter I presented a multivariate version of the central limit theorem to compute average observables of random geometric graphs in the limit of infinite dimension. In particular, the average number of M-cliques in hard and soft RGGs for different distance functions induced by p-norms is different in the limit of infinite dimensionality.

This approach highlights that convergence to the Erdös-Rényi graphs prediction for local observables depends on the choice of the ensemble: soft RGGs in particular seem to approach this naive limit for $d \to \infty$, whereas hard RGGs whose probability distribution of the nodes fulfils the CLT hypothesis deviate systematically from it. This result suggests that the latter provide a non-trivial null model to benchmark empirical data.

A potentially useful application of these results lies in their guidance with regards to the choice of *null models*, which are essential if one is to extract meaningful information from the data. For example, consider data points from an empirical data set (such as MNIST, for instance), and a graph constructed on these points, where a link exist whenever two data points are closer than a given cutoff radius (determining this graph is the starting point for many learning algorithms, from hierarchical clustering to manifold learning). Now, say the number of cliques in this graph deviates from the Erdös-Rényi prediction. If we erroneously believe that RGGs in high dimension are Erdös-Rényi graphs, then we should conclude that the behavior is due to specificities of the data (e.g., deviations from the assumption of independence). This conclusion would be misleading, since, for the hard activation function, there are systematic deviations from the Erdös-Rényi prediction even if the data points are uncorrelated and identically distributed. The analysis presented in this chapter makes clear that ruling out the null hypothesis of RGG in high dimension is fundamentally different from ruling out the hypothesis of being a

Erdös-Rényi graphs.

This is a first line of possible development for these results: checking in practice whether graphs constructed on real data deviate from the Erdös-Rényi or the hard-RGG prediction for the average density of M-cliques in order to validate both models as possible null-models for complex data.

Since the CLT can be formulated in a much more general setting than the one reported in this manuscript, we expect that our findings hold (possibly with slight modifications) for several probability distributions of the nodes not included here, e.g. not factorized over coordinates, but with mild inter-coordinate correlations; factorized over coordinates, but not identically distributed; factorized over coordinates, but with infinite second moment. The wide basin of attraction of the Gaussian limit hints to the possibility that the properties of high-dimensional structured datasets may be faithfully described by our approach. In this manuscript we worked with the simplest version of the CLT, as random geometric graphs are commonly studied with nodes that are independently drawn in the hypercube. The very relevant case of *structured data* [BLR+19; RLG20; PRE+20; RPG20] calls for more sophisticated CLTs, which may be addressed with the same tools developed here.

Another potentially interesting case is that of RGGs whose vertex measure is supported on low-dimensional manifolds but is embedded in a much higher-dimensional ambient space with noise. Which observables will be hidden by the added noise? And which will be robust, allowing to recover non-trivial properties of the underlying geometry?

Finally, our numerical simulations show that the infinite dimensional limit is a good approximation even in finite dimensions of order $d \sim 10$. This hints at the possibility to improve our results by computing higher order corrections to the CLT, and using d as a perturbative parameter, to access the low dimensional regime of RGGs.

3.6 Chapter bibliography

[AB02] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, 1, January 2002. DOI: 10.1103/RevModPhys.74.47.

[AB20] Konstantin E. Avrachenkov and Andrei V. Bobu. Cliques in high-dimensional random geometric graphs. *Applied Network Science*, 5(1):92, November 2020. ISSN: 2364-8228. DOI: 10.1007/s41109-020-00335-6.

[All18] Alfonso Allen-Perkins. Random spherical graphs. *Physical Review E*, 98(3), September 2018. DOI: 10.1103/PhysRevE.98.032310.

[Bar11] Marc Barthélemy. Spatial networks. *Physics Reports*, 499(1):1–101, 2011. ISSN: 0370-1573. DOI: https://doi.org/10.1016/j.physrep.2010.11.002.

[BCD+18] Alessandro Benfenati, Emilie Chouzenoux, Laurent Duval, Jean-Christophe Pesquet, and Aurélie Pirayre. A review on graph optimization and algorithmic frameworks. Research Report, LIGM - Laboratoire d'Informatique Gaspard-Monge, October 2018. URL: https://hal.archives-ouvertes.fr/hal-01901499.

[BDE$^+$16] Sébastien Bubeck, Jian Ding, Ronen Eldan, and Miklós Z. Rácz. Testing for high-dimensional geometry in random graphs. en. *Random Structures & Algorithms*, 49(3):503–532, October 2016. ISSN: 1098-2418. DOI: 10.1002/rsa.20633.

[BLG17] Arianna Bottinelli, Rémi Louf, and Marco Gherardi. Balancing building and maintenance costs in growing transport networks. *Phys. Rev. E*, 96:032316, 3, September 2017. DOI: 10.1103/PhysRevE.96.032316.

[BLR$^+$19] Francesco Borra, Marco Cosentino Lagomarsino, Pietro Rotondo, and Marco Gherardi. Generalization from correlated sets of patterns in the perceptron. *Journal of Physics A: Mathematical and Theoretical*, 52(38):384004, August 2019. DOI: 10.1088/1751-8121/ab3709.

[BM08] J Adrian Bondy and Uppaluri SR Murty. Graph theory. In Springer, 2008.

[Bur17] Amanda Burcroff. Johnson schemes and certain matrices with integral eigenvalues. *University of Michigan, Tech. Rep*, 2017. URL: http://math.uchicago.edu/~may/REU2017/REUPapers/Burcroff.pdf.

[Cho05] André-Louis Cholesky. Sur la résolution numérique des systèmes d'équations linéaires. *Bulletin de la Sabix. Société des amis de la Bibliothèque et de l'Histoire de l'École polytechnique*, (39):81–95, 2005.

[CSS$^+$19] Giulio Cimini, Tiziano Squartini, Fabio Saracco, Diego Garlaschelli, Andrea Gabrielli, and Guido Caldarelli. The statistical physics of real-world networks. *Nature Reviews Physics*, 1(1):58–71, January 2019. ISSN: 2522-5820. DOI: 10.1038/s42254-018-0002-6.

[DC02] Jesper Dall and Michael Christensen. Random geometric graphs. en. *Physical Review E*, 66(1):016121, July 2002. ISSN: 1063-651X, 1095-3787. DOI: 10.1103/PhysRevE.66.016121.

[DGL$^+$11] Luc Devroye, András György, Gábor Lugosi, and Frederic Udina. High-Dimensional Random Geometric Graphs and their Clique Number. en. *Electronic Journal of Probability*, 16(0):2481–2508, 2011. ISSN: 1083-6489. DOI: 10.1214/EJP.v16-967.

[EAG$^+$20] Vittorio Erba, Sebastiano Ariosto, Marco Gherardi, and Pietro Rotondo. Random geometric graphs in high dimension. *Phys. Rev. E*, 102:012306, 1, July 2020. DOI: 10.1103/PhysRevE.102.012306.

[ER60] Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

[ES15] Ernesto Estrada and Matthew Sheerin. Random rectangular graphs. *Physical Review E*, 91(4):042805, April 2015. DOI: 10.1103/PhysRevE.91.042805.

[FK15] Alan Frieze and Michał Karoński. *Introduction to Random Graphs*. Cambridge University Press, 2015. DOI: 10.1017/CBO9781316339831.

[FWV07] Damien Francois, Vincent Wertz, and Michel Verleysen. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19(7):873–886, 2007. DOI: 10.1109/TKDE.2007.1037.

[Gen92] Alan Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1(2):141–149, 1992. DOI: 10.1080/10618600.1992.10477010.

[GGD15] Alexander P. Giles, Orestis Georgiou, and Carl P. Dettmann. Betweenness centrality in dense random geometric networks. en. In *2015 IEEE International Conference on Communications (ICC)*, pages 6450–6455, London. IEEE, June 2015. ISBN: 978-1-4673-6432-4. DOI: 10.1109/ICC.2015.7249352.

[HL81] Paul W. Holland and Samuel Leinhardt. An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33–50, 1981. DOI: 10.1080/01621459.1981.10477598.

[KBD19] Alexander P. Kartun-Giles, Marc Barthelemy, and Carl P. Dettmann. The shape of shortest paths in random spatial networks. en. *Physical Review E*, 100(3):032315, September 2019. ISSN: 2470-0045, 2470-0053. DOI: 10.1103/PhysRevE.100.032315.

[Net13] David F. Nettleton. Data mining of social networks represented as graphs. *Computer Science Review*, 7:1–34, 2013. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2012.12.001.

[Pen03] Mathew Penrose. *Random Geometric Graphs*. Oxford University Press, May 2003.

[PRE$^+$20] Mauro Pastore, Pietro Rotondo, Vittorio Erba, and Marco Gherardi. Statistical learning theory of structured data. *Phys. Rev. E*, 102:032119, 3, September 2020. DOI: 10.1103/PhysRevE.102.032119.

[RLG20] Pietro Rotondo, Marco Cosentino Lagomarsino, and Marco Gherardi. Counting the learnable functions of geometrically structured data. *Phys. Rev. Research*, 2:023169, 2, May 2020. DOI: 10.1103/PhysRevResearch.2.023169.

[Row00] S. T. Roweis. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000. DOI: 10.1126/science.290.5500.2323.

[RPG20] Pietro Rotondo, Mauro Pastore, and Marco Gherardi. Beyond the storage capacity: data-driven satisfiability transition. *Phys. Rev. Lett.*, 125:120601, 12, September 2020. DOI: 10.1103/PhysRevLett.125.120601.

[Ten00] J. B. Tenenbaum. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000. DOI: 10.1126/science.290.5500.2319.

[Van00] Aad W Van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge university press, 2000.

[vdMH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

CHAPTER 4

Expressivity of linear classifiers on geometrically structured data

Linear models are the simplest and better understood architectures to perform regression or classification on generic datasets. They often appear as building blocks of more complex learning models, such as kernel methods, support-vector machines and, more recently, deep neural networks. In this chapter, I investigate how a simple model of data structure can alter the expressivity properties of linear classifiers following [RLG20]. I finally present recent results on a novel phase transition for the expressivity of linear classifiers that is specifically induced by data structure [PRE+20].

Contents

4.1 Supervised learning

4.1.1 Learning paradigms in artificial intelligence

What do we mean when we say that "a deep neural network can learn to distinguish images of cats from images of dogs"?

In machine learning, we typically mean that we have a dataset of P samples $\{x^\mu\}_{\mu=1}^P$ (think of images embedded in Euclidean space by assigning to each pixel a different dimension) labelled with labels $\{y^\mu\}_{\mu=1}^P$ (these may be real numbers, binary numbers, abstract categories), and an adjustable learning architecture, i.e. an adjustable function that can associate to each sample a label (possibly incorrectly). The learning architecture is then randomly initialized, meaning that to each image the learner associates a random label, and it is trained by adjusting its parameters in such a way that, after each adjustment, its predictions of the labels for the samples in the training set get better and better. This paradigm is usually called *supervised learning.*

We already covered another paradigm of machine learning in Chapter 2, namely manifold learning, where from an unlabelled dataset, under the hypothesis that there is an underlying geometrical structure, we try to infer properties of the intrinsic geometry.

There are other paradigms that are worth mentioning:

- *unsupervised learning* is similar in spirit to manifold learning. In this case, from an unlabelled dataset one tries to learn a compression/decompression algorithm that is able to reduce the embedding dimensionality of the dataset while preserving all the meaningful information. More qualitatively, in unsupervised learning one hopes that the learning algorithm can infer some underlying structure from the training data in a similar way as humans infer general properties from particular observations. See [HS+99] for more details;

- *reinforcement learning* takes a different approach. Instead on relying on sample/label pairs, it uses as learning material a set of possible moves and a function to evaluate the performance of a sequence of moves. For example, the task may be "to learn playing Super Mario", the set of possible moves may be the combination of inputs that a player can provide to the main character, and the performance function may be the distance from the end of the level after the characters' death. The learner then optimizes its strategy, i.e. its sequence of moves, trial after trial in order to optimize the performance. See [SB18] for more details.

In this chapter, we will focus on the setting of supervised learning, which is the most relevant to study modern classification architectures.

4.1.2 The ingredients of a supervised learning task

Let me start by fixing some notations and discussing the components of a supervised learning task more in detail. A supervised learning task is defined by a **training set**, a family of **predictor functions** (a.k.a. model functions), a **cost function** (a.k.a. error function, loss function) and a **generalization cost function**.

Briefly, the idea is that the functions belonging to the model set are parametrized by a set of adjustable "knobs" (parameters), that can be tuned to represent a variety of different relationships between inputs and outputs. The training set is a set of input/output pairs that can be used to tune the parameters of the predictor functions, and the tuning itself is done by minimizing a cost function (that depends on both the training set and the current realization of the predictor function). Up to this point, we are just describing a fitting procedure to learn the training set by heart. The last ingredient, that transforms fitting into supervised learning, is that cost minimization is just a proxy for the real deal, generalization cost minimization. In fact, one is not interested into learning by heart training samples, but into obtaining a predictor that perform well on unseen examples.

Training set The training set is a collection of P pairs of samples x^μ with associated labels y^μ. The samples are typically elements of some metric space, usually $\mathbb{R}^N$. The labels can be either real numbers or vectors (of dimension N_{out}), or categorical variables; in the following, we will call $\mathcal{Y}$ the set in which the labels take value. In the former case one talks about *regression*, in the latter about *classification*.

Binary categorical variables are usually encoded into binary variables $y^\mu \in \{-1, 1\}$ or $y^\mu \in \{0, 1\}$. Categorical variables of higher cardinality are encoded in the so-called *one-hot vector* representation: if the number of categories is C and the μ-th sample is in category i, then y^μ is a C dimensional vector with components $y^\mu_j = \delta_{i,j}$ for $j = 1, 2, \ldots, C$. In other words, only the i-th component of the label vector is turned on.

In general, the training set should be considered as given once and for all, with no prior information on the process that generated it. In the analytical practice however it is useful to consider specific generative models for the dataset. In Section 4.1.3 we discuss some of the options in this direction.

Predictor function The predictor function is a parametrized set of functions $f_w : \mathbb{R}^N \to \mathcal{Y}$, where $w \in \mathbb{R}^Q$ is the vector of parameters. The idea is that, as the parameters vary, this set of functions can represent a variety of possible relationships between the samples and their labels.

Learning architectures typically differ one from another due to the choice of predictor functions. For example, linear models are defined by (here $Q = N$)

$$f_w(x) = w \cdot x \,, \tag{4.1}$$

while deep, fully-connected neural networks are defined by

$$f_{w^1,w^2,\ldots,w^L}(x) = w^L \cdot \sigma\Big(\ldots \sigma\big(w^2 \cdot \sigma(w^1 \cdot x)\big)\Big) \tag{4.2}$$

where $w^i \in \mathbb{R}^{N_i \times N_{i-1}}$, $N_0 = N$, $N_L = N_{\text{out}}$ and $\sigma : \mathbb{R} \to \mathbb{R}$ is an activation function.

In the case of classification, both models above can be adapted with minor changes. For example, linear binary classifiers are just given by the composition of a linear model with a sign function.

Learning architectures may also differ just due to different parametrizations of the same set of functions. For example, consider a deep fully-connected neural network with the trivial choice $\sigma(x) = x$ and $N_i \geqslant \min(N, N_{\text{out}})$ for $i = 1, 2, \ldots L-1$. f_w is a composition of linear maps, and thus it is equivalent to a strangely-parametrized linear model[1]. Re-parametrization may be useful in order to explore different learning dynamics for the same set of predictor functions [SMG19].

[1] The condition on the widths of the layers grants that in the deep case we are not artificially restricting the rank of the linear map.

Cost function The cost function is a function that measures how well a particular predictor function is representing the relationship between training samples and labels. It is usually factorized over the training patterns, i.e.

$$\mathcal{L}_{\text{data}}[f] = \sum_{\mu=1}^{P} \ell(f(x^\mu), y^\mu)\,, \tag{4.3}$$

and may have additional regularization terms $\mathcal{L}_{\text{reg}}[f_w] = g(w)$ penalizing wildly-varying values of the parameters w[2]. The function $\ell(f(x), y)$ computes how badly a function f recovers the correct label y of a sample x.

Most supervised learning problems describe learning as the minimization of the cost function over the set of predictor functions, i.e. the learned predictor $\hat{f} = f_{\hat{w}}$ is determined by the optimal parameters

$$\hat{w} = \arg\min_{w} \left[\mathcal{L}_{\text{data}}[f_w] + \mathcal{L}_{\text{reg}}[f_w]\right]\,. \tag{4.4}$$

In regression problems, a common choice for the function ℓ is the squared loss, i.e.

$$\ell(y, y') = (y - y')^2\,, \tag{4.5}$$

but other choices are possible.

In classification problems, the natural choice for ℓ is the error counting loss, i.e.

$$\ell(y, y') = \delta_{y,y'}\,, \tag{4.6}$$

where the Kronecker's delta equals one if y and y' describe the same category, and equals zero otherwise. In practice though, it is often useful to consider differentiable cost functions. In fact, they can be minimized using local methods (for example, gradient-based methods). A common choice for the loss function is given by the cross-entropy loss.

Notice that, in general, the cost minimization problem is an high-dimensional non-convex optimization task, which may have many different global and local minimizers and are in general difficult to solve (deciding whether cost minimization with a deep neural network can achieve zero error, for example, is an NP-complete problem [BR92]). Thus, the actual minimization procedure (i.e. the training) is usually performed using gradient-based local algorithms, and the optimal set of parameters $\hat{w}$ is a function of the details of the algorithm (initialization, properties of the training set, etc...). This is not a problem, as the aim of supervised learning is not cost minimization, but generalization cost minimization, as we will see better in the next paragraph. Thus, it may as well be that local minima of the cost function have better generalization properties that global minima, and so on . I will not delve deeper into this issue in this Thesis; I would just like to remark that the relationship between algorithms, cost minimization and generalization is one of the most challenging open problems of the field.

[2] Common choices are the L2, or ridge, regularization $g(w) = \sum_i w_i^2$ [HK70] and the L1, or LASSO, regularization $g(w) = \sum_i |w_i|$ [Bre95]. Notice that the regularization depends directly on the weights and on the specific parametrization of the function f_w.

Generalization cost function Up to this point, the aim of supervised learning seems to be to fit as precisely as possible the training set. At least, this is what "learning as cost minimization" seems to suggest. Actually, there is one last ingredient in supervised learning: a measure of the generalization performance of the optimal predictor $\hat{f}$. By generalization performance we mean the ability of the optimal predictor to correctly label samples that were never used in the training process.

First of all, where are these unseen examples coming from? When only a fixed training set is available, it is common practice to split the P samples of the training set into an actual training set of P_{train} samples, and a test set of P_{test} samples. Training is then performed using the P_{train} training samples, and generalization is assessed by evaluating the cost function over the P_{test} test samples. Notice that the test cost function may in principle be different from the cost function used in training. This is especially true in the case of classification, where the cost function may be a differentiable function for minimization purposes, while the test cost function may be the simple error counting function.

If a generative model for the training set is available, the test set can be easily generated by sampling from the generative model P_{test} new patterns, with P_{test} possibly very large.

It is worth noticing that cost minimization does not imply generalization cost minimization in general. Indeed, even if the cost and the generalization cost functions are equal and the training and test set are generated by the same generative process, there may be finite-size effect due to the limited dataset size and learning-dependent effects due to the choice of minimization algorithm. Moreover, it is common knowledge that the generalization cost of a trained classifier may be a non-monotonic function of the training set size (all other ingredients fixed). This phenomena go under the name of "bias-variance trade-offs" [KW+96] and "double descent behaviours" [NKB+20], and are caused by, for example, noise corrupting the training samples and labels, or noise due to random components of the predictor function (as happening in the Random Feature model [DRB+20; MM20], for example).

Most of the buzz around Machine Learning is caused by the empirical observation that cost minimization performed using noisy gradient-based algorithms seems to lead to unexpected generalization cost minimization [Sej20].

Now that we have an overview on the components of a supervised learning task, we shall focus on two crucial aspects: the structure of the training set (meaning the structure of correlations between samples and labels) and the expressivity of a set of predictor functions (meaning, roughly, the number of sample/label relationship that the model class can represent). We will see that data structure, at least in a particular declination, has remarkable effects on the ability of simple predictors to correctly classify data. We will explore this relationship in a particular case, that of binary linear classification tasks ($\mathcal{Y} = \{-1, +1\}$) with the error counting cost function, without worrying about trainability or generalization.

4.1.3 The training set: models of data structure

To fix a model of data, we resort to the usual setting of Statistical Learning Theory (SLT) [MRT18; Wol18]. We suppose that there exists a true sample/label probability distribution $P_{X,Y}$ on $\mathbb{R}^N \times \mathcal{Y}$, and we extract the P training samples $\{x^\mu, y^\mu\}_{\mu=1}^P$ i.i.d.

from $P_{X,Y}$[3]. In this setting, we see that:

- the generalization error has a natural expression as an average over the true sample/label distribution, i.e.

$$\epsilon_{\text{gen}}[f] = \mathbb{E}\left[\ell(f(x), y)\right] , \tag{4.7}$$

  where ℓ is some loss function, and $\mathbb{E}$ denotes the average over $(x, y) \sim P_{X,Y}$. In practice, one can sample a large number P_{test} of sample/label pairs, and approximate the generalization error as

$$\epsilon_{\text{gen}}[f] = \frac{1}{P_{\text{test}}} \sum_{\mu=1}^{P_{\text{test}}} \ell(f(x^\mu), y^\mu) ; \tag{4.8}$$

- if we define the cost function as

$$\mathcal{L}_{\text{data}}[f] = \frac{1}{P} \sum_{\mu=1}^{P} \ell(f(x^\mu), y^\mu) , \tag{4.9}$$

  we see that the training error will converge to the generalization error as P grows, justifying the idea of cost minimization to achieve generalization cost minimization.

In this setting, unstructured data is defined by a factorized $P_{X,Y} = P_X P_Y$, so that each sample has the same probability of being labelled in a certain way, independently on its position in $\mathbb{R}^N$. Given that $\mathcal{Y}$ is binary in our case, P_Y will be given in general by a Bernoulli distribution

$$P_{\text{Bernoulli}}(y; q) = q\delta_{y,1} + (1 - q)\delta_{y,-1} . \tag{4.10}$$

The two most commonly studied paradigms to model non-trivial data structure are given by the **teacher-student setting** and the **perceptual manifolds model**.

Teacher-student setting The teacher-student model has been studied for a long time (see [EV01] for more informations). It defines data structure in a very practical way. In this model, samples are i.i.d. samples from a given probability distribution P_X as in the unstructured case. Structure is imposed by generating the label deterministically using a *teacher* architecture, i.e. a particular predictor function f^T which may or may not belong to the set of predictor functions of the learner, also called *student*. More formally

$$P_Y(y|x) = \delta(y - f^T(x)) . \tag{4.11}$$

See Figure 4.1 for an example. Thus, in this model, data structure is given solely by correlations between a sample and its label.

The teacher-student model is extremely versatile. For example, one could account for data corruption by adding some form of statistical noise to the deterministic label-generating function. Moreover, data structure is completely encoded into the teacher

[3] Notice that this is not the only possibility. Indeed, we will focus later on a model of dataset in which subsets of k samples are jointly distributed with non-trivial correlations.

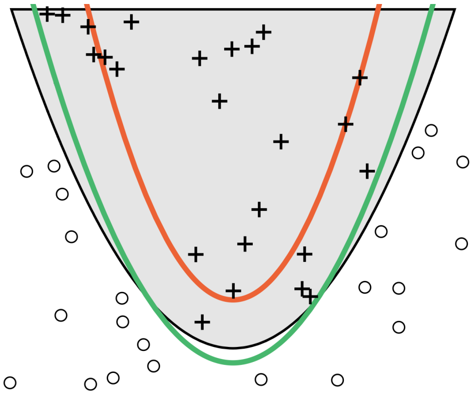


Figure 4.1: Example of teacher-student setting. In this example, we generate random points in the plane, and we label them using a teacher from the predictor function set $f_{a,b}(x) = \text{sign}(ax^2 + b)$, with parameters $(a, b)_{\text{teacher}} = (1, -4)$ (black parabola). More explicitly, the model calls labels points above the parabola with label $+1$, and points below the parabola with label -1. The red curve is an example of a "bad" student $(a, b)_{\text{bad}} = (2, -3)$, i.e. a predictor function that makes unnecessary labelling mistakes. Indeed, all the points falling between the red curve and the black parabola are mistakenly classified by the bad student. The green curve is a "good" student $(a, b)_{\text{good}} = (1.3, -4.3)$, i.e. a student perfectly reproducing the classification of the teacher, even though it has different values of the parameters.

function, which can be compared to the student architecture using techniques from functional analysis (see for example the Reproducing Kernel Hilbert Spaces techniques used in teacher-student models for kernel regression and classification [DOS99; CBP21]).

More recently, the teacher-student model has been generalized to account for more complex forms of data structure. The hidden-manifold model for example is an evolution of the classic teacher-student setup recently introduced in [GMK+20]. In the teacher-student setup, labels depend directly on the corresponding samples. In the hidden-manifold model, both samples and labels are generated based on an intrinsic, latent representation.

More in detail, for each sample/label pair, a latent representation $c^\mu \in \mathbb{R}^{N_{\text{latent}}}$ is generated according to some given probability distribution P_{latent}. Then, the sample x^μ is given by

$$x^\mu = \sigma(F \cdot c^\mu) \tag{4.12}$$

or some fixed matrix $F \in \mathbb{R}^{N \times N_{\text{latent}}}$, called feature matrix, and component-wise non-linearity σ. The labels are generated as in the classical teacher-student setup, with the crucial difference that they directly depend on the latent representations c^μ, and not on the actual sample x^μ.

In this model, structure is given by a non-trivial correlation between samples and labels, mediated by the latent representations. Again, each sample/label pair is independently generated from all others.

Perceptual manifolds The perceptual manifolds model is motivated by the biological intuition that similar inputs (that a priori excites neurons in different way) should be classified similarly. This intuition resonates with recent concepts put forward in neuroscience,

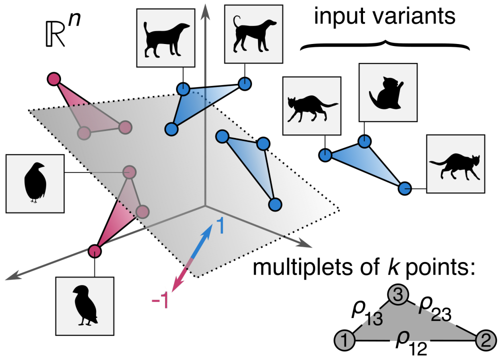


Figure 4.2: Sketch of the linear classification of a perceptual manifolds dataset. In this qualitative example, the labels ±1 correspond respectively to mammals and birds, and each sample in a k-plet (here $k = 3$) is an image of the same animal in different positions. The linear classifier, i.e. hyperplane, shaded in grey classifies the k-plets with an admissible labelling, as all samples of each k-plet is labelled coherently with its companions. The geometry of the k-plet is fixed by the overlaps $\rho^{a,b}$. Notice that for the sake of representation here the samples are not distributed on the unit hypersphere as specified in the main text. (Reprinted from [RLG20]).

and more recently in machine learning, of invariant recognition (different representation of the same objects give rise to similar neural responses) [SL00; ALR+16] and object manifolds classification (sets of inputs giving rise to the same neural response should be coherently classified) [CLS16; CLS18; CCL+20].

In this model, the dataset is defined as follows. Samples are points on the unit $(N-1)$-dimensional hypersphere[4], and are grouped in P k-plets $\{x^{\mu,a}\}_{a=1,\ldots,k}^{\mu=1,\ldots,P}$. Each k-plet is generated uniformly on the sphere with the only constraint that

$$x^{\mu,a} \cdot x^{\mu,b} = \rho^{a,b} \qquad \forall \mu = 1, \ldots, P\,, \quad \forall a, b = 1, \ldots, k \quad \text{s.t.} \quad a < b\,. \tag{4.13}$$

The constraint that the overlaps equal $\rho^{a,b}$ fixes the geometry of the k-plets. Notice that $-1 \leqslant \rho^{a,b} \leqslant 1$ for all pairs of points in the k-plet. The model is further defined by restricting the set of admissible labellings to those that equally classify samples in the same k-plet, i.e.

$$y^{\mu,a} = y^{\mu,b} \qquad \forall \mu = 1, \ldots, P\,, \quad \forall a, b = 1, \ldots, k \quad \text{s.t.} \quad a < b\,. \tag{4.14}$$

Given this constraint, the labels are randomly generated for each k-plet as in the unstructured case. See Figure 4.2 for an example.

In this model, structure is imposed in multiple ways. Each k-plet has a fixed geometry, introducing correlations between groups of samples. Moreover, the constraint on the admissible labellings introduces a correlation between samples and labels, namely that samples in the same k-plet will have the same label.

A variant of the model is given by margin learning. In this case, samples are not k-plets, but hyperspheres of fixed radius κ. An admissible classifier labels all points in the interior of one of the spheres coherently.

[4] See [CLS16; CLS18; CCL+20] for a similar model defined without the spherical constraint.

4.1.4 Predictor functions: measuring expressiveness

In this Chapter, I want to focus on a particular aspect of supervised learning, that is measuring the expressivity of simple families of predictor functions, in particular linear classifiers. In order to do this we have to define what do we mean by expressivity (limiting the discussion to binary classification tasks as anticipated).

The qualitative idea is that a more expressive set of predictors will be able to represent more relations between samples and labels than a less expressive one. More formally, consider a fixed set of samples $\{x^\mu\}_{\mu=1}^P$ and a set of predictor functions f_w. For every possible binary labelling of the samples $\{y^\mu\}_{\mu=1}^P$ (there are 2^P of them), we can determine if there is at least a value w^* of the parameters such that

$$f_{w*}(x^\mu) = y^\mu \qquad \forall \mu = 1, \dots, P\,, \tag{4.15}$$

that is equivalent to saying that the predictor function is able to represent the given labelling of the sample set. A more expressive architecture will be characterized by a higher number of representable labellings. Of course, at this point this definition is ill-posed, as it depends in the specific set of samples $\{x^\mu\}$; we will define it more precisely shortly.

Beyond intuition, the number of classifiable labellings is a central object in classical SLT, the branch of statistics concerned with providing rigorous bounds on the generalization properties of learning architectures (see [MRT18; Wol18] for a good introduction). In the following paragraphs, I will recall some measures of expressivity related to the number of classifiable labellings without entering too much into the details.

The fraction of classifiable labellings and the critical capacity The first proper measure of expressivity is the average fraction of classifiable labellings $C_{N,P}/2^P$, where 2^P is trivially the total number of unconstrained binary labellings over P points, and $C_{N,P}$ is the number of classifiable labellings of P points in N dimensions averaged over some distribution of the P inputs P_X. We expect this function to be monotone decreasing in P at fixed N, as it is harder and harder to separate more and more points at fixed ambient dimension and predictor function class. The value $P(N)$ at which $C_{N,P}/2^P = \frac{1}{2}$ pinpoints the so-called *critical capacity* of the learning architecture. More precisely, the critical capacity is defined as

$$\alpha_c = \lim_{N\to\infty} \frac{P(N)}{N}\,, \tag{4.16}$$

as we expect the threshold $P(N)$ to scale to infinity as N grows larger[5]. The ratio $\alpha = \frac{P}{N}$ is often call load. Thus, a first measure of (average) expressivity is given by the critical capacity: a high critical capacity means that one can correctly classify, on average, larger training sets.

The critical capacity, and more in general the average fraction of classifiable labellings, hides a fundamental connection between SLT and Statistical Mechanics. The first connection is that it can be experimentally observed that α_c defines a *phase transition* between

[5] Notice that more complex architectures may require more complex scaling behaviours than the simple linear one [DOS99], but the linear scaling is often verified in practice.

a satisfiable phase $0 < \alpha < \alpha_c$, where the classification problem is typically solvable, and an unsatisfiable phase $\alpha > \alpha_c$ where the classification problem is typically not-solvable to zero training error. Thus, the critical capacity can be studied using methods from the statistical mechanics of disordered systems (as the training set can be thought of as a form of quenched disorder) just by identifying the correct order parameter describing the transition. It turns out [Gar88; EV01] that the right order parameter is given by the volume in the space of parameters of the predictor function class spanned by functions that correctly classify a random training set. In the case of linear classifiers, this is given by

$$V(\{x^\mu\},\{y^\mu\}) = \int dw\, p(w) \prod_{\mu=1}^{P} \theta\left(y^\mu \left(w \cdot x^\mu\right)\right) \tag{4.17}$$

where x^μ are the position of the samples, y^μ a binary labelling and $p(w)$ a prior or regularizing distribution over the parameter space. Samples and labels are considered to be a form of quenched disorder to be averaged over. The expected behaviour is that:

- for $\alpha < \alpha_c$, the volume is composed by exponentially many (in the dimension N) vectors w that, on average, are able to correctly classify a typical sample/label assignment;

- for $\alpha > \alpha_c$, the volume shrinks and becomes sub-exponential in N, meaning that a smaller and smaller portion of the space of parameters is able to correctly classify a typical sample/label assignment.

Thus, the transition threshold can be found by computing at which value of the load $\alpha = \frac{P}{N}$ at which the average free-entropy $\frac{1}{N}\mathbb{E}\log(V)$ vanishes[6].

For more general predictor functions, the Gardner's volume can be defined as

$$V(\{x^\mu\},\{y^\mu\}) = \int dw\, p(w) \prod_{\mu=1}^{P} \delta_{y^\mu, f_w(x^\mu)}\,. \tag{4.18}$$

Analogous expressions can be useful to characterize regression problems; see for example [CBP21].

The second connection between SLT and Statistical Mechanics was noted recently in the literature [AAK+20], and relates the average number of admissible labellings to the Rademacher complexity, which is a crucial object in SLT that appears in rigorous bounds to the generalization error. We will not delve deeper here; see the cited reference for more details.

The Vapnik-Chervonenkis entropy Another relevant quantity is given by the Vapnik-Chervonenkis (VC) entropy, which is defined as

$$H_{N,P} = \log\left(C_{N,P}\right)\,, \tag{4.19}$$

[6] This is unprecise, but conveys the correct picture. In problems with continuous parameters the free-entropy goes to $-\infty$, while in problems with discrete parameters it vanishes.

where again $C_{N,P}$ is the average number of classifiable labellings of P points in N dimension. Its relevance stems from a rigorous upper-bound to the generalization error that a given learning architecture can achieve. This upper-bound depends on the specific distribution of the samples and their labels only through the implicit average in the definition of $C_{N,P}$, and is uniform with respect to the particular class of predictor functions.

Let's have a look at this upper-bound. It can be proven that (see for example [Wol18]), in the case of binary classification with the error counting cost (and the same generalization cost), for any $\delta > 0$, with probability at least $1 - \delta$,

$$|\epsilon_{\text{gen}} - \epsilon_{\text{train}}| \leqslant 2\sqrt{2\frac{H_{N,2P} + \log\frac{2}{\delta}}{P}}\,, \tag{4.20}$$

where $|\epsilon_{\text{gen}} - \epsilon_{\text{train}}|$ is the difference between the training and the generalization cost. Notice that $C_{N,P}$, entering the bound through $H_{N,P}$, is averaged over the sample/label distribution *before* taking the log. For this reason, H is often called annealed VC entropy.

Notice that the VC entropy is defined as above only in strictly binary classification problems. In the following I will try not to abuse the nomenclature. Indeed, it is not completely correct to call VC entropy the logarithm of $C_{N,P}$ in the case of k-plet classification, which is not a strictly binary classification problem due to the fact that an hyperplane that cuts through a k-plet labels it ambiguously. In other words, the logarithm of $C_{N,P}$ computed for structured data is just a proxy for the real VC entropy. For this reason, the generalization bound above cannot benefit directly from any consideration regarding $\log(C_{N,P})$ on structured data.

The Vapnik-Chervonenkis dimension A final[7] measure of expressivity is given by the Vapnick-Chervonenkis dimension d_{VC}, i.e. the maximum value of P such that there exists at least one assignment of the positions of the samples for which every labelling is correctly classifiable by the class of predictor functions. Notice that here there is no average to be taken. The VC-dimension is a worst-case, and not typical-case, expressivity measure.

Again, this quantity enters into an upper-bound to the generalization cost thanks to a provable inequality (see [Wol18]) stating that

$$H_{N,2P} \leqslant d_{\text{VC}} \log\left(\frac{e\,P}{d_{\text{VC}}}\right)\,. \tag{4.21}$$

This allows to further relax the upper-bound in Equation (4.20), obtaining a architecture-and-data-independent bound on the generalization error.

Notice that, while powerful and rigorously-derived, SLT bounds on the generalization error are mostly too loose to be of any usefulness in the case of modern over-parametrized

[7] For the purposes of this Thesis, of course!

classes of predictor functions [Bot15]. For example, in deep fully-connected neural networks the VC dimension scales as $\sim Q \log Q$ where $Q \sim 10^6 - 10^9$ is the total number of parameters, and typical datasets are composed of $P \sim 10^4 - 10^6$ samples.

The main aim of this Chapter will be to study the observable $C_{N,P}$ for linear classifiers, that is

$$f_w(x) = \mathrm{sign}\,(w \cdot x) \tag{4.22}$$

under some additional assumption on the structure of the sample set $\{x^\mu\}_{\mu=1}^P$, namely that is generated by the perceptual manifold prescription and that the classifier coherently labels samples in the same k-plet.

4.2 Expressivity of linear classifiers on unstructured data

The first step to understand how data structure alters the expressivity of linear classifiers is to study their expressivity in the unstructured case. Thus, we would like to compute and study $C_{N,P}$, that is the number of binary labellings of P N-dimensional points that can be realized with no errors by a linear classifier

$$f_w = \mathrm{sign}\,(w \cdot x)\ . \tag{4.23}$$

First of all, let's get a good mental picture of what a linear classifier looks like. A linear classifier is nothing more than an hyperplane passing through the origin, identified by its normal direction w. It assigns label $+1$ to all points in the half-space with positive projection on w, and label -1 otherwise. The case in which a points falls exactly onto the separating hyperplane is a rare event[8]. In what follows I will provide a mild technical condition that allows to treat with no headaches this edge case, but for the moment let us forget about this.

Given a certain dataset of P labelled samples, we may ask if there exists a linear classifier that can assign the correct label to each of the points. It may not exists, but if it does, there exists infinitely many of them. In fact, if a separating hyperplane w^* that labels correctly each sample exists, i.e.

$$\mathrm{sign}(w \cdot x^\mu) = y^\mu \tag{4.24}$$

for all samples $\mu = 1, \dots, P$, then at least all of its infinitesimal deformations $w^* + \delta w$ perform equally good, as the function $w \to w \cdot x$ is continuous, and small perturbations on the input do not alter the sign of the output. Here δw is some perturbation with arbitrary small norm.

Thus, for a given linearly separable dataset and a corresponding separating hyperplane w, we can identify three regions of the ambient space $\mathbb{R}^N$:

[8] For example, if samples are i.i.d. according to a well-behaved - say absolutely continuous w.r.t. the Lebesgue measure - probability distribution in the ambient space $\mathbb{R}^N$, then the probability of having a point falling exactly onto the hyperplane is zero.

- a first region defined by

$$\{x \in \mathbb{R}^N \mid w \cdot x > \min_{\substack{\mu=1,\dots,P \\ y^\mu=1}} w \cdot x^\mu\}\,. \tag{4.25}$$

  This is the region of points that are deep in the region where the hyperplane w and all other possible separating hyperplanes that correctly classify the dataset assign label $+1$;

- a second region conversely defined as

$$\{x \in \mathbb{R}^N \mid w \cdot x < \max_{\substack{\mu=1,\dots,P \\ y^\mu=-1}} w \cdot x^\mu\}\,. \tag{4.26}$$

  This is the region of points that are deep in the region where the hyperplane w and all the other good hyperplanes assign label -1;

- the complementary region defined by

$$\{x \in \mathbb{R}^N \mid \max_{\substack{\mu=1,\dots,P \\ y^\mu=-1}} w \cdot x^\mu < w \cdot x < \min_{\substack{\mu=1,\dots,P \\ y^\mu=1}} w \cdot x^\mu\}\,. \tag{4.27}$$

  This is sort of a "grey zone". In this region, each of the good separating hyperplanes may behave differently from the others.

See Figure 4.3, as a figure is better than a thousand words here. In particular, notice that if we add a new point to the dataset in one of the first two regions, its label must agree with that of its neighbours in order to preserve the linear classifiability of the dataset. On the other hand, if the new point is added to the grey zone, than a non-empty portion of the good separating hyperplanes will guarantee the linear separability of the enhanced dataset irrespectively of the label of the added sample.

Equipped with this intuitive notions, we are ready to see how expressive linear classifiers are.

4.2.1 The number of classifiable labellings

It turns out that, under a very mild assumption on the particular instance of the sample set, $C_{N,P}$ is independent on the actual positions of the samples, and thus equals its average value with respect to basically all probability distributions P_X over samples. This is a result due to Cover [Cov65], who proved that

$$C_{N,P} = 2\sum_{i=0}^{N-1} \binom{P-1}{i}\,. \tag{4.28}$$

Figure 4.4 shows $C_{N,\alpha N}$ as a function of the load $\alpha = \frac{P}{N}$ for growing values of N.

The mysterious mild assumption is that the P samples must be in *general position*, i.e. all points in a subset X' of cardinality $|X'| \leqslant N$ of the sample set X are linearly independent. In other words, no subset of points is accidentally aligned, spanning an hyperplane[9]. This is a rather weak condition on the sample set. For example, all sample

[9] This is the technical assumption promised in the previous section. This condition ensures that there is always some space to slightly rotate the hyperplane in order to eliminate the edge case.

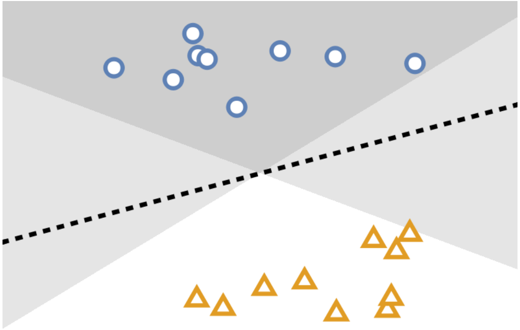


Figure 4.3: Labelled sets partition the ambient space into different regions. The plot represents a training set (blue circles have label +1, orange triangle -1) correctly classified by the dotted hyperplane (in $N = 2$, an hyperplane is just a line). The training set partitions the ambient space into 3 regions. In the dark grey region, all hyperplanes that correctly classify the datasets assign uniformly the label +1 to every point. The same is true in the white region, with label -1. In other words, all the hyperplane that correctly classify the training set are contained in the light grey region. As a consequence, if both w and w' are hyperplanes that correctly classify the dataset, they will label equally all points in the dark grey and white areas, and they will possibly disagree on the points in the light grey region.

sets generated by sampling P i.i.d. points from a probability density are almost always in general position, as d-dimensional linear subspaces ($d < N$) have null N-dimensional Lebesgue measure. This means also that $C_{N,P}$ is a deterministic quantity if the P samples are randomly generated with simple laws.

Let's prove Cover's result. Suppose that you have a set of P points in N dimension, and that you know that there are $C_{N,P}$ binary labellings that can be linearly classified with no error over this set of points. To be explicit, this means that for the given set of points $\{x^\mu\}_{\mu=1}^P$, and for each one of the $C_{N,P}$ classifiable labellings $\{y^\mu\}_{\mu=1}^P$, there exists a separating hyperplane identified by its normal direction $w \in \mathbb{R}^D$ s.t. $\text{sign}(w \cdot x^\mu) = y^\mu$ for all $\mu = 1, \ldots, P$.

What happens to the $C_{N,P}$ classifiable labellings when we add a new unlabelled point? Let's consider a particular classifiable labelling ϕ, and see if it can be promoted to a labelling that correctly classifies also the newly added point. There are two possibilities (see Figure 4.5 for a graphical representation of the proof):

1. the labelling ϕ can be realized by an hyperplane that passes through the position of the new point x (i.e. $w \cdot x = 0$). This is equivalent to asking that the new point is added to the grey zone (see panel b of Figure 4.5). As already argued, in this case the hyperplane w can be infinitesimally rotated in order for the new point to be classified correctly irrespectively of the choice of its label. Thanks to the assumption of general position, this infinitesimal rotation does not alter the classification of all other points. Thus, for each of these M labellings on P points, there exist two classifiable labellings of $P + 1$ points, one for each label of the newly added point.

   How many of these ϕ's exist? There are $M = C_{N-1,P}$ of such labellings, as we

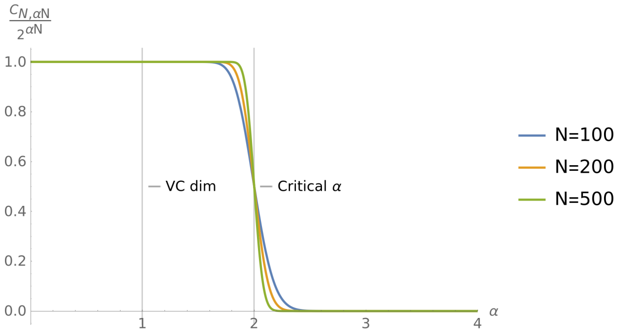


Figure 4.4: Fraction of classifiable labellings of αN points in N dimensions. For growing value of N, we show the number of classifiable labellings normalized over the total number of labellings, $2^{\alpha N}$. At $0 < \alpha < 1$, all labellings are linearly classifiable (excluding pathological sets of samples that are not in general position). At $\alpha = \alpha_c = 2$, the curves at various values of N intersect, a typical behaviour found in phase transitions. In this case, the phase transition separates an easily separable phase from an hard phase. In the easy phase, the volume spanned by parameters that can separate a given dataset is exponentially large in N, while in the hard phase it grows only sub-exponentially.

are now asking how many classifiable labellings of P points can be realized by an hyperplane in N dimension, constrained to pass through x (lowering the effective number of available dimensions by 1);

2. the labelling ϕ cannot be realized by an hyperplane passing through x (see panel e of Figure 4.5). Equivalently, x falls in one of the two zones where all separating hyperplanes associated to ϕ assign the same label to all points, either $+1$ or -1. In this case, the label of the newly added points matters, and only one choice for it allows to extend ϕ to a classifiable labelling of $P + 1$ samples.

   By subtraction, there are $C_{N,P} - M$ of these labellings, each of which contributes with a single classifiable labelling to $C_{N,P+1}$.

Thus,

$$C_{N,P+1} = 2M + (C_{N,P} - M) = C_{N,P} + C_{N-1,P}\,. \tag{4.29}$$

The initial condition for the recursion is given by $C_{N,1} = 2(1 - \delta_{N,0})$, as a single point can always be classified irrespectively of its labels (the Kronecker's delta takes care of the non-physical zero-dimensional case $N = 0$).

The recursion can be solved by noticing that $C_{N,P}$ is a linear combination of the initial values $C_{N-i,1} = 2$ for all $i = 0, \ldots, N-1$. If we could compute the coefficients of this linear expansion, the recursion would be solved. It's easy to see that the coefficient of $C_{N-i,1}$ is given by the number of directed paths $\{\gamma_j\}_{j=1,\ldots,P}$ such that (i) $\gamma_1 = N-i$, (ii) $\gamma_P = N$ and (iii) γ_{j+1} equals either γ_j or $\gamma_j + 1$. The number of these paths is given by the binomial coefficient $\binom{P-1}{i}$, as each path is determined by the choice of i among the $P - 1$ transitions such that γ increases by one unit. Thus,

$$C_{N,P} = 2\sum_{i=0}^{N-1}\binom{P-1}{i}\,. \tag{4.30}$$

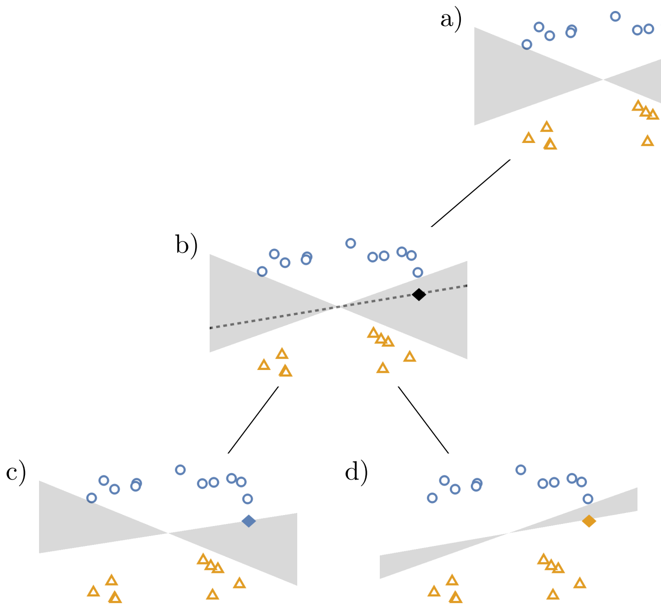


Figure 4.5: Graphical representation of Cover's argument. We start by considering a fixed classifiable labelling ϕ of $P = 19$ points in $N = 2$ dimensions (panel a). In all panels, the grey shaded zone denotes by all possible hyperplanes that correctly separate blue circles and orange triangles. Next, we add a new unlabelled point x. The new point either falls into the grey zone (panel b), i.e. there exists a separating hyperplane w of the original labelling such that $w \cdot x = 0$ (dashed line in panel b), or does not (panel e). In the former case, there are two distinct ways of extending ϕ to the new point so that the new dataset is still linearly separable: either x gets labelled as a blue circle (panel c), or as a orange triangle (panel d). In the latter case, there is only one possibility of extending ϕ to a classifiable labelling of the enlarged dataset, namely by assigning to x the label of its neighbours. This justifies Equation (4.29).

Cover's result is powerful not only because it is independent on the position of the P samples, but also because it applies to more general classes of predictor functions. In fact, suppose that before applying the linear classifier of Equation (4.23) each sample is mapped to a high-dimensional space $\mathbb{R}^{N'}$ with $N' > N$ through a so-called feature map ψ. Suppose also that ψ preserves the general position of the samples. Then, we can apply Cover's result in the feature space $\mathbb{R}^{N'}$, obtaining that the classifiable labellings grow from $C_{N,P}$ to $C_{N',P}$.

Again, asking that ψ preserves the general position of its inputs turns out to be a very mild requirement. For example, polynomial feature maps of the form

$$\psi(x)_{i_1,i_2,\ldots,i_d} = \psi_{i_1,i_2,\ldots,i_d} x_{i_1} x_{i_2} \ldots x_{i_d} \,, \tag{4.31}$$

where d is the degree of the map and $\psi_{i,j,\ldots}$ are coefficients, preserve general position. Neural network with non-linear activation function with fixed intra-layer weights, as in Equation (4.2), do the trick too.

4.2.2 Expressivity of linear classifiers

Inspired by applications and theoretical interest, and keeping in mind the discussion of Section 4.1.4, it is tempting to look at the high-dimensional limit $N \to \infty$ to compute the critical capacity and the VC dimension of linear classifiers. We let P and N go to infinity while keeping their ratio $\alpha = \frac{P}{N}$ fixed, as we expect that linear classification becomes harder when we try to separate a number of points comparable with the ambient dimension.

VC dimension The VC dimension can be easily computed by noticing that, if $P \leqslant N$,

$$C_{N,P} = 2\sum_{i=0}^{N-1}\binom{P-1}{i} = 2\sum_{i=0}^{P-1}\binom{P-1}{i} = 2^P \tag{4.32}$$

as all the binomials with $i > P-1$ are zero by definition, and where we used the Newton's binomial theorem. Instead, if $P = N+1$,

$$C_{P-1,P} = 2\sum_{i=0}^{P-2}\binom{P-1}{i} = 2\left(2^{P-1}-1\right) = 2^P - 2 \tag{4.33}$$

where we used again Newton's binomial theorem. Thus, for linear classifiers $d_{\mathrm{VC}} = N$[10] (see Figure 4.4).

Critical capacity To compute the critical capacity α_c, we need to compute the value $P(N)$ such that

$$\lim_{N\to\infty}\frac{C_{N,P(N)}}{2^{P(N)}} = \frac{1}{2}\,. \tag{4.34}$$

We notice that, if $P(N) = 2N+1$,

$$\begin{aligned} C_{N,2N+1} &= 2\sum_{i=0}^{N-1}\binom{2N}{i} = 2\sum_{i=0}^{N-1}\frac{1}{2}\left[\binom{2N}{i} + \binom{2N}{2N-i}\right] \\ &= \sum_{i=0}^{2N}\binom{2N}{i} - \binom{2N}{N} = 2^{2N} - \binom{2N}{N}\,. \end{aligned} \tag{4.35}$$

This implies that

$$\frac{C_{N,2N+1}}{2^{2N+1}} = \frac{1}{2} - \frac{2}{\sqrt{\pi N}}\left(1 + \mathcal{O}\left(\frac{1}{N}\right)\right) \tag{4.36}$$

[10] This result may differ by one unit with respect to other references. This is usually due to the fact that here we consider strictly a linear classifier, while other references often consider linear classifiers with bias, i.e. with corresponding hyperplane not passing from the origin. In the limit $N \to \infty$, this difference vanishes.

where we used Striling's formula to approximate the central binomial $\binom{2N}{N}$ for large N. Thus, we see that

$$\alpha_c = \lim_{N\to\infty} \frac{P(N)}{N} = 2\,. \tag{4.37}$$

Figure 4.4 confirms this result, and clearly shows that α_c pinpoints a phase transition, as all curves at growing values of N intersect (modulo finite-size corrections).

As already mentioned in Section 4.1.4, the capacity can also be computed by studying the behaviour of the Gardner's volume

$$V = \int dw\, p(w) \prod_{\mu=1}^{P} \theta\left(y^\mu\left(w\cdot x^\mu\right)\right)\,, \tag{4.38}$$

and more precisely the corresponding quenched entropy $\langle \log V\rangle$, where angular brackets denote the average with respect to the sample/label distribution. Here, $p(w)$ can be taken to be either the uniform distribution on the sphere, or equivalently (in high dimension N) a multivariate Gaussian with null mean and unit covariance, as in both case the two distributions weight equally all possible unit vectors (that identify the separating hyperplanes).

A detailed computation of Gardner's volume in the replica-symmetric ansatz can be found in [Gar88; EV01].

Logarithm of $C_{N,P}$ It would be extremely useful to obtain an explicit expression for the whole curve $C_{N,\alpha N}$, without any summation lurking around. This would provide us with more manageable and intuitively analysable information about the number of classifiable labellings, and thus about $\log(C_{N,P})$. While it is not possible to simplify Equation (4.28) more, we can provide a more explicit form for some parameter regions.

As $d_{\mathrm{VC}} = N$, we have that

$$C_{N,P} = 2^P \tag{4.39}$$

for $0 < P \leqslant N$. Thus, $C_{N,P}/2^P = 1$ and $\log(C_{N,P}) = P\log 2$ for all $0 < P \leqslant N$ trivially.

A less trivial information is obtained by studying the limit of large load $\alpha = P/N$ for either fixed or diverging N. I will state the result first, and prove it just afterwards. We have that, for fixed N and large load $\alpha \to \infty$,

$$C_{N,\alpha N} \sim \frac{2(\alpha N)^{N-1}}{\Gamma(N)}\,. \tag{4.40}$$

Notice, in particular, that for fixed N this is a monotone increasing function of α, and that while for $\alpha < 1$ the number of classifiable labellings grows exponentially in α, for large α it only grows polynomially fast (see Figure 4.6). Qualitatively, this is due to the fact that adding a new point to the dataset generates a combinatorial proliferation of possible labellings, that are totally unconstrained for $\alpha < 1$, and more and more constrained as α grows larger and larger. Consequently, for large α, also $\log(C_{N,P})$ is monotone increasing.

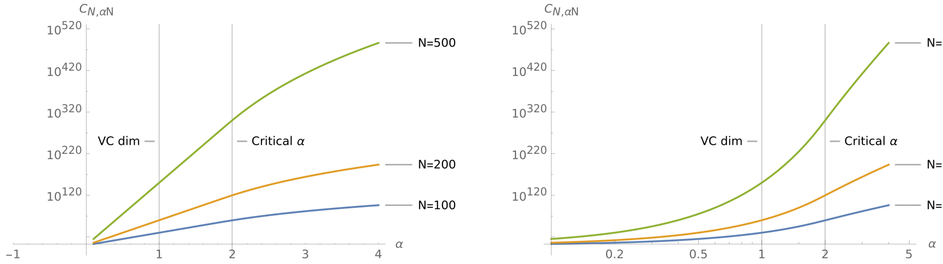


Figure 4.6: Behaviour of $\log(C_{N,P})$ as a function of the load α. Log-lin plot (left) and log-log plot (right) of $C_{N,\alpha N}$ as a function of α for growing values of N. We clearly see in the log-lin plot that for $0 < \alpha < 1$, and approximately for $1 < \alpha < 2$ as N grows, the number of classifiable labellings grows exponentially fast. In the log-log plot, the polynomial growth for large α is visible instead.

In order to motivate Equation (4.40) we will set up a computational technique that may seem cumbersome at this level of complexity, but that will be crucial to generalize this analysis to structured data later on. We start from Cover's recursion in Equation (4.29), and we introduce the family of generating functions (with complex variable)

$$g_N(z) = \sum_{P\geqslant 1} C_{N,P} z^P \,. \tag{4.41}$$

The idea is that the asymptotics behaviour of $g_N(z)$ around its singularity of smallest module z_0 is in bijection with the asymptotics for large P of the coefficients of $g_N(z)$, i.e. $C_{N,P}$[a]. More explicitly, it can be proven that

$$g(z) \sim \gamma (z_0 - z)^{-a} \ \text{ for } z \sim z_0 \iff [g]_P \sim \gamma |z_0|^{-P} \frac{P^{\,a-1}}{\Gamma\,(a)} \ \text{ for } P \sim +\infty\,, \tag{4.42}$$

where, just for this example, I suppressed the dependence of $g_N(z)$ of N, and I denoted the P-th coefficient of its power-series expansion as $[g]_P$. This is a central result in analytic combinatorics, see [FS09]. Thus, to obtain the asymptotic form of $C_{N,P}$ for large N, P and large load α, we just need to study the asymptotic behaviour of $g_N(z)$ around its leading singularity, and this can be done explicitly by using Equation (4.29). In fact, at the level of generating functions, Equation (4.29) can be rewritten as (for $N \geqslant 1$):

$$\sum_{P\geqslant 1} C_{N,P+1} z^P = \sum_{P\geqslant 2} C_{N,P} z^{P-1} = \sum_{P\geqslant 1} C_{N,P} z^{P-1} - C_{N,1} = \frac{1}{z} g_N(z) - 2\,, \tag{4.43}$$

$$\sum_{P\geqslant 1} \left(C_{N,P} + C_{N-1,P}\right) z^P = g_N(z) - g_{N-1}(z) \tag{4.44}$$

giving the functional recursion

$$\begin{aligned} g_N(z) &= \frac{z}{1-z}\left(g_{N-1}(z)+2\right)\,, \\ g_0(z) &= 0\,. \end{aligned} \tag{4.45}$$

This recursion can be treated in two ways: exactly and asymptotically. Indeed, on one side this is a linear non-homegeneous first-order recursion, whose explicit solutions can be computed with standard methods, giving

$$g_N(z) = \frac{2z}{2z-1}\left[\left(\frac{z}{1-z}\right)^N - 1\right] \tag{4.46}$$

which has a single pole at $z_0 = 1$ of order N, and asymptotic expansion

$$g_N(z) \sim 2(1-z)^{-N} \quad \text{for } z \sim 1 \tag{4.47}$$

for $N \geqslant 1$. On the other side, given Equation (4.45), it's very easy to obtain the asymptotic behaviour of $g_N(z)$ around its dominant singularity without explicitly solving the recursion. In fact, given the ansatz $g_N(z) = \gamma_N(z_0 - z)^{-a_N}$ for $z \to z_0$, it's immediate to see that $z_0 = 1$, $a_N = N$ and $\gamma_N = 2$ in order for the recursion to be self-consistent.

Equation (4.47) and Equation (4.42) readily imply Equation (4.40). Notice that this asymptotic expansion is valid only at fixed N and large P. In out context, this means that this expansion is valid only for diverging load α, either at fixed or diverging N. It would be interesting to study the bivariate generating function of $C_{N,P}$ in order to obtain the asymptotic behaviour for all values of α.

[a] Obviously, under some requirements on the function $g_N(z)$, namely that it is analytic in an open disk centered at the origin, with radius $|z_0|$, and with one of the half-lines (z_0, ∞) excluded. See [FS09] for the details.

4.3 Expressivity of linear classifiers on structured data: segments

We are now ready to extend Cover's argument to structured data, and in particular to the perceptual manifolds model described in Section 4.1.3. More concretely, we wish to compute the number of admissible labellings of P k-plets of points on the $N-1$ dimensional hypersphere that can be correctly labelled by a linear classifier. We shall call this number $C^{(k)}_{N,P}$. We will start from the simplest case, $k = 2$.

4.3.1 The number of classifiable labellings of segments

We start with the simplest case. We will show in a moment that the number of classifiable labellings $C^{(2)}_{N,P}$ satisfies a mean-field recursion equation given by

$$C^{(2)}_{N,P+1} = \Psi_2(\rho) C^{(2)}_{N,P} + C^{(2)}_{N-1,P} + \left(1 - \Psi_2(\rho)\right) C^{(2)}_{N-2,P} \tag{4.48}$$

with initial condition $C^{(2)}_{N,1} = 2\left[1 - (1 - \Psi_2(\rho))\delta_{N,1}\right](1 - \delta_{N,0})$. Here $\Psi_2(\rho)$ is the probability that two points on the sphere with overlap ρ fall on the same side of a uniformly-draw random hyperplane, i.e.

$$\Psi_2(\rho) = \frac{1}{Z}\int dw\, \delta(||w|| - 1)\theta(w \cdot x)\,\theta(w \cdot y) = \frac{2}{\pi}\arctan\sqrt{\frac{1+\rho}{1-\rho}}\,, \tag{4.49}$$

where Z is the normalization constant

$$Z = \int dw\, \delta(||w|| - 1) = \Omega_{N-1}\,. \tag{4.50}$$

Notice that $\Psi_2(\rho)$ depends only on the overlap between the two fixed points, and it is thus constant in our model. Notice also that when $\rho = 1$, i.e. each doublet collapses to a single point, $\Psi_2(1) = 1$ and the recursion reduces to Cover's one, Equation (4.29), as expected.

The recursion can be solved using a technique similar to the unstructured case, leading to

$$C^{(2)}_{N,P} = 2\sum_{i=0}^{N-2} K_{i,P} + 2\Psi_2(\rho)K_{N-1,P} \tag{4.51}$$

where

$$K_{i,P} = \sum_{m=0}^{P-1}\binom{P-1}{m, i-2m, P-1-i+m}\Psi_2(\rho)^{P-1-i+m}\left(1-\Psi_2(\rho)\right)^m \tag{4.52}$$

and

$$\binom{a}{b, c, d} = \frac{a!}{b!\,c!\,d!} \tag{4.53}$$

is the usual multinomial coefficient.

Before proving the recursion and the form of the explicit solution, let me stress that this recursion is not exact. It is a mean-field approximation, in which when we have to estimate the probability that two constrained points fall on the same side of an hyperplane, we estimate it as if the points were totally unconstrained. Moreover, $C^{(2)}_{N,P}$ is now an average number over the uniform distribution of doublets on the sphere. Figure 4.8, panel (a), compares the mean field value of $C^{(2)}_{N,P}$ to numerical simulations, showing perfect agreement.

Let's start by proving the recursion. Again, we suppose that there are, on average[a], $C^{(2)}_{N,P}$ classifiable labellings of P doublets in dimension N with fixed overlap ρ, and we study how these labellings behave when adding a new doublet (x, y).

We start by adding the first point x. Repeating Cover's argument, there are exactly $Q_{N,P} = C^{(2)}_{N,P} + C^{(2)}_{N-1,P}$ classifiable labellings of the original dataset with x added. Notice that in each of these labellings, x has a well-defined label. When adding y, we again have two possibility for each classifiable labellings ϕ among these $Q_{N,P}$ (see Figure 4.7 for some graphical help):

1. ϕ can be realized by an hyperplane passing through y, i.e. y falls in the grey zone of the labelling (panel b and c of Figure 4.7). In this case, all of these labellings give rise to one classifiable labelling of the enlarged dataset by deforming the

hyperplane to correctly classify y.

How many of these labellings exists? There are $R_{N,P} = C^{(2)}_{N-1,P} + C^{(2)}_{N-2,P}$ of such labellings. In fact, $R_{N,P} = Q_{N-1,P}$ is precisely the number of classifiable labellings of P doublets and one point in N dimension satisfying a single linear constraint, leading to an effective dimension of $N-1$.

2. ϕ cannot be realized by an hyperplane passing through y, i.e. y does not fall into the grey zone of ϕ. In this case, ϕ can be promoted to a classifiable labelling over the enlarged dataset only if x and y fall on the same side of the original separating hyperplane, which on average happens with probability $\Psi_2(\rho)$[b] (panel d and e of Figure 4.7).

   By subtraction, there are $Q_{N,P} - R_{N,P}$ of such labellings.

Thus, combining all cases, we obtain

$$C^{(2)}_{N,P+1} = R_{N,P} + \Psi_2(\rho)\,(Q_{N,P} - R_{N,P}) \tag{4.54}$$

leading to Equation (4.48).

The initial conditions at $P = 1$ for the recursion are the same as in the unstructured case, except in the $N = 1$ case. When $N = 1$ in fact, if both points in the doublet fall on the same half-line, which happens with probability $\Psi_2(\rho)$, then both labellings are classifiable, otherwise none of them is. On average, $C^{(2)}_{1,1} = 2\Psi_2(\rho)$, giving the initial condition specified in Equation (4.48).

To solve the recursion, we notice once again that $C^{(2)}_{N,P}$ is a linear combination of the initial conditions $C^{(2)}_{N-i,1}$ for $i = 0, \ldots, N-1$. The coefficients of this expansion, let's call them $K_{i,P}$, can be computed again by using the paths analogy. In fact, $K_{i,P}$ is the total weight of all directed paths $\{\gamma_j\}_{j=1,\ldots,P}$ such that $\gamma_1 = N - i$, $\gamma_P = N$ and $\gamma_{j+1} - \gamma_j = 0, 1$ or 2. Each transition $\gamma_j \to \gamma_{j+1}$ is weighted according to the recursion, i.e. with weight $\Psi_2(\rho), 1$ or $1 - \Psi_2(\rho)$ if $\gamma_{j+1} - \gamma_j = 0, 1$ or 2 respectively. Combining all this recovers Equation (4.52), where one should think that m counts the number of transitions with step-difference equal to 2, and $i - 2m$ counts the number of transitions with step-difference equal to 1 (giving a total height excursion equal to i as needed).

[a] Here we have the first mean-field hand-weaving

[b] Here we have the second mean-field hand-weaving

4.3.2 Expressivity of linear classifiers on segments

As in the unstructured case, we can now look at the various measures of expressivity.

Capacity In this structured case, obtaining analytically the critical load α_c corresponding to the capacity is not possible. An approximate value for the capacity can be obtained by approximating Equation (4.51) with

$$C^{(2)}_{N,P} \approx 2 \sum_{i=0}^{N-1} K_{i,P} \tag{4.55}$$

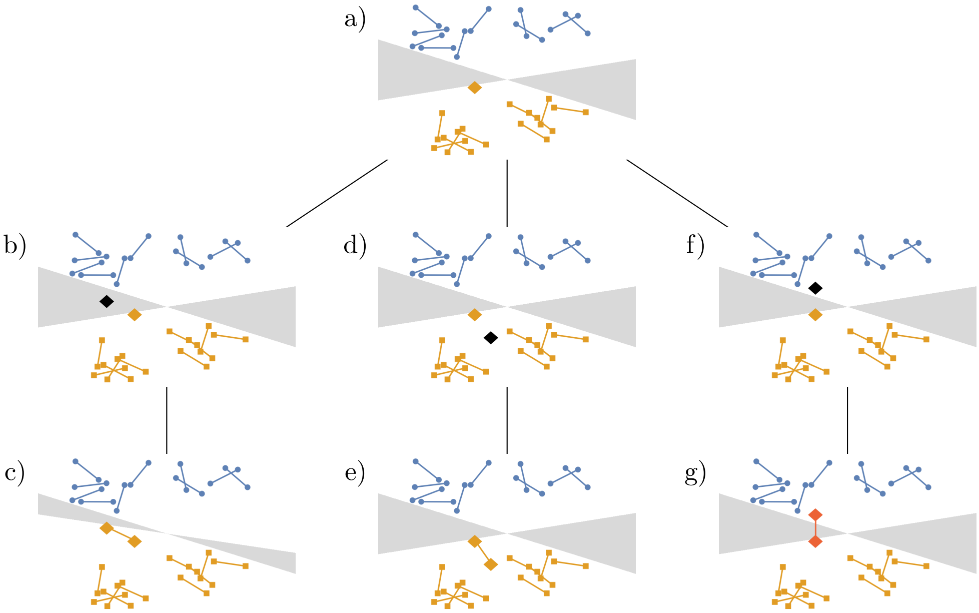


Figure 4.7: Graphical representation of Cover's argument for segments. We start by considering a fixed classifiable labelling of $P = 20$ segments in $N = 2$ dimension (panel a). The orange diamond is the point x (refer to the detail box for the notation), i.e. one of the two extremes of a new segments that we want to add to the dataset. The addition of the point x follows the original Cover's argument, and its location is not important for what follows. We now complete the new segment by adding the point y, at fixed overlap with x (i.e. at fixed distance). Depending on the overlap, a subset of the following three options can be realized. Option 1: y falls into the grey zone (panel b). In this case, point y can be labelled as orange (as its companion) without breaking the linear separability of the dataset (panel c). Option 2: y falls in the middle of the other orange segments (panel d). In this case, again y can be labelled orange without braking the linear separability (panel e). Option 3: y falls in the middle of the blue segments (panel f). In this case, there is no way of labelling y with the same color as x without breaking the linear separability of the dataset. The last option never happens in the unstructured case (see Figure 4.5. Notice that for the sake of clarity, in this graphical representation I moved from points on the sphere and overlaps to normal points and distances.

for large N. This is equivalent to modifying one among the N initial conditions of Equation (4.48), and in the limit of large N we expect that this change generates vanishingly small corrections. We will see that the value of the capacity that this approximation predicts matches with independent replica computations and agrees with the numerical experiments, justifying further the procedure.

Leaving aside the validity of the approximation, we now want to compute for which value of the load α

$$C^{(2)}_{N,\alpha N} = 2^{P-1} \,. \tag{4.56}$$

i.e., the value for which the sum $\sum_{i=0}^{N-1} K_{i,P}$ equals half of its maximum value. We observe that each summand is the total weight of all directed paths of P steps that perform a vertical excursion of i units, each step counted with weight $\Psi_2(\rho), 1, 1 - \Psi_2(\rho)$ if it gains

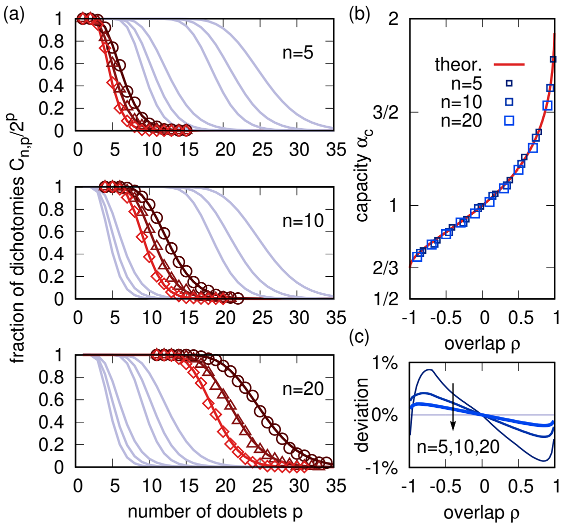


Figure 4.8: Theory vs numerical results for structured data $k = 2$. Numerical simulations (open markers in the plots) are obtained by training a linear classifier on 1000 randomly labelled dataset of 2-plets using the perceptron algorithm, and computing the fraction of perfectly classifiable trials. Circles (○), triangles (△) and diamonds (◇) denotes different values of $\rho = 0.6, 0.2, -0.2$. Theory (solid lines) is given by Equation (4.48). (a) For different values of N and ρ, numerical simulations are in perfect agreement with the mean field theory. (b) The theoretical capacity (Equation (4.58)) agrees with numerical simulations for values of N as small as $N = 5$. The numerical capacity was computed by linearly interpolating data such as those shown in panel (a). (c) Finite size deviation between the approximate capacity given in Equation (4.58) and the theoretical capacity obtained by numerically solving $C^{(2)}_{N,P} = 2^{P-1}$ using Equation (4.52). As N increases, the approximation get better and better. (Reprinted from [RLG20]).

0,1 or 2 vertical units, and 0 otherwise. If we normalize the weights by dividing them by their sum, i.e. 2, this is also proportional to the probability that a random directed path (with transition probabilities given by the weights) of P steps reaches a final height of i units. In this picture, $C^{(2)}_{N,P}$ is the probability that in P steps a random walk reaches height at most $N-1$. Thus, when we ask for which value of α $C^{(2)}_{N,\alpha N}$ equals half of its maximum value, we are equivalently asking at which value of the step number αN the total probability that a walk ends at height lesser than $N-1$ equals 1/2. That is, we would like to ask at which value of α the median of the distribution of the final height of the random walks is exactly N. Approximating the median with the mean, this gives

$$N = (P-1)\sum_{\ell=0}^{2} \ell P(\gamma_j \to \gamma_j + \ell) = \left(\frac{3}{2} - \Psi_2(\rho)\right)(P-1) \tag{4.57}$$

where the transition probabilities equal the transition weights normalized with a factor 2, and the factor $P-1$ accounts for the independence of the $P-1$ transitions of a path. Finally we obtain, in the large N, P limit,

$$\alpha_c^{(2)} = \frac{2}{3 - 2\Psi_2(\rho)}\,. \tag{4.58}$$

Figure 4.8, panel (b-c), compares Equation (4.58) with numerical simulations, and with the numerical solution of $C^{(2)}_{N,P} = 2^{P-1}$ using the exact form given in Equation (4.52). There is a very good agreement between the three estimates of the capacity. This justifies *a posteriori* our mean-field approach and the approximation on the initial condition of the recursion. Moreover, Figure 4.8, panel (b) shows that, as expected, the capacity gets smaller as the geometry of the 2-plets gets more and more extended, i.e. as ρ goes from 1 (unstructured) to -1 (maximally structured).

As already mentioned in Section 4.1.4 and in Equation (4.38), the capacity can also be computed by studying the behaviour of the Gardner's volume. In this case, the relevant volume is

$$V = \int dw\, p(w) \prod_{\mu=1}^{P} \prod_{a=1}^{2} \theta\left(y^\mu \left(w \cdot x^{\mu,a}\right)\right) . \tag{4.59}$$

Again, $p(w)$ can be taken to be either the uniform distribution on the sphere, or equivalently (in high dimension N) a multivariate Gaussian with null mean and unit covariance, as in both case the two distributions weight equally all possible unit vectors (that identify the separating hyperplanes). Here the samples and labels are considered to be quenched disorder. Each 2-plet is given independently by a random pair of points on the sphere with fixed overlap ρ and random label y.

A detailed computation of this Gardner's volume can be found in [BLR+19].

VC dimension Following the same line of thought (and the same approximation) as that of the previous section, the VC dimension corresponds to the maximum value of P such that $C^{(2)}_{N,P}$ takes its maximum value. Equivalently, we ask that the probability that a random walk with P steps ends at height lesser than $N-1$ is 1. This happens for $N = 2(P-1)$, as the maximum height that a random walk can reach in P steps is exactly $2(P-1)$, giving that (at least for large N, P)

$$d^{(2)}_{\text{VC}} = \frac{N}{2} . \tag{4.60}$$

Logarithm of $C_{N,P}$ To obtain qualitative informations on the full curve $C^{(2)}_{N,\alpha N}$ for large values of α, we resort again to the generating functions technique already used in the unstructured case. Again, let me give the result first. For large N and α, we have that

$$C^{(2)}_{N,\alpha N} \sim \frac{2(\alpha N)^{N-1}}{\Gamma\left(N\right)} \left[\Psi_2(\rho)\right]^{(\alpha-1)N} . \tag{4.61}$$

Notice the similarity with Equation (4.40). The only crucial difference is the factor $\Psi_2(\rho)$, accounting for the geometry of the data, that modifies the monotonicity of $C^{(2)}_{N,\alpha N}$ at large load α. In fact, in the unstructured case the number of classifiable labellings grows polynomially with the load, while in the structured case it goes to zero exponentially fast. This effect can be qualitatively understood as a competition between two effect. The first one is the combinatorial growth of possible labellings as new samples are added. The second one is the fact that existing classifiable labellings can be broken by the addition of

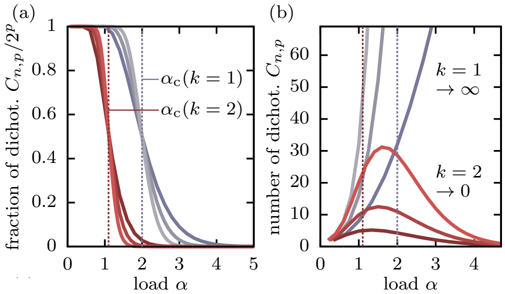


Figure 4.9: Comparison between the number of classifiable labellings of unstructured and structure data. Fraction (a) and number (b) of classifiable labellings of unstructured ($k = 1$, grey) and structured ($k = 2$, red, $\rho \neq 1$) data, for growing N (small N corresponding to dark colors, big N to lighter colors). N equals $5, 10, 20$ in (a) and $3, 4, 5$ in (b). We see that, while in the case of the fraction (a) of classifiable labellings the behaviour of unstructured and structured data is quite similar, with just a shift in the value of the critical capacity, at the level of the full $C_{N,P}$ (b) the difference is much more evident. The number of classifiable labellings switches from being increasing to being decreasing at large loads. (Reprinted from [PRE$^+$20]).

new samples, and this effect is greatly enhanced by the extended geometry of structured samples.

The fact that $\log(C_{N,P})$ is monotone decreasing in the large load phase is crucial, as it could provide more meaningful bounds to the generalization error in the large-dataset ($P \gg N$) limit. Figure 4.9 compares the number of classifiable labellings of unstructured and structured $k = 2$ data, focusing on the change of monotonicity of $C_{N,P}$ as data structures enters into play.

Let's prove Equation (4.61). For the sake of simplicity, I will drop all superscipts (2) during the next few lines of computation. We again start from the recursion in Equation (4.48), and we introduce the family of generating functions (with complex variable)

$$g_N(z) = \sum_{P \geqslant 1} C_{N,P} z^P \,. \tag{4.62}$$

We are interested in the behaviour of $g_N(z)$ around its singularity of smallest module, i.e. its dominant singularity.

At the level of generating functions, Equation (4.48) can be rewritten as (for $N \geqslant 1$):

$$\begin{aligned} g_N(z) &= \frac{z}{1 - \Psi_2(\rho) z} \Big(g_{N-1}(z) - \big(1 - \Psi_2(\rho)\big) g_{N-2}(z) + 2 \Big) \,, \\ g_0(z) &= 0 \,. \end{aligned} \tag{4.63}$$

Notice that, in the same spirit as the previous sections, we are assuming that for large N the initial conditions of Equation (4.48) provide a negligible overall contribution. This allows to use the same initial conditions as those of Equation (4.29).

This recursion can be treated again in two ways: exactly and asymptotically. Indeed, on one side this is a linear non-homegeneous second-order recursion, whose explicit solutions can be computed with standard methods, giving a quite complicated and uninspiring result that it is not worth reporting here; see [PRE$^+$20]. On the other side, given Equation (4.63), it's very easy to obtain the asymptotic behaviour of $g_N(z)$ around its dominant singularity without explicitly solving the recursion. In fact, given the ansatz $g_N(z) = \gamma_N(z_0 - z)^{-a_N}$ for $z \to z_0$, it's immediate to see that $z_0 = \Psi_2(\rho)$, $a_N = N$ and $\gamma_N = 2[\Psi_2(\rho)]^{-N}$. This, combined with Equation (4.42), readily implies Equation (4.61).

Again, notice that this asymptotic expansion is valid only at fixed N and large P.

4.4 Expressivity of linear classifiers on structured data: polytopes

We are finally ready to study the most general case of structured dataset with k-polytope geometry.

4.4.1 The number of classifiable labellings of polytopes

Looking at the unstructured and $k = 2$ casees, we may guess what to expect. If we strongly believe in inducing patterns out of a couple of cases, we expect the average number of classifiable labellings $C^{(k)}_{N,P}$ to satisfy a mean-field recursion relation that generalizes Equation (4.29) and Equation (4.48), depending on $C^{(k)}_{N-l,p}$ for $l = 0, 1, \ldots, k$, i.e.

$$C^{(k)}_{N,P} = \sum_{l=0}^{k} \theta^k_l C^{(k)}_{N-l,P} \tag{4.64}$$

for some coefficients $\theta^{(k)}_l$. The coefficients will depend on the geometry of the polytopes through the $\frac{k(k-1)}{2}$ fixed overlaps $\rho^{a,b}$ (I will call them collectively ρ), but guessing their explicit form at this stage would be nothing less than pure magic. It turns out that all of the above it's actually true, and the coefficients $\theta^{(k)}_l$ are recursively determined by

$$\theta^{(k)}_l = \tilde{\Psi}_k(\rho)\theta^{(k-1)}_l + (1 - \tilde{\Psi}_k(\rho))\theta^{(k-1)}_{l-1} \tag{4.65}$$

with initial condition $\theta^{(1)}_0 = \theta^{(1)}_1 = 1$ and $\theta^{(k)}_{-1} = \theta^{(k)}_{k+1} = 0$. The geometrical coefficients $\tilde{\Psi}_m(\{\rho^{a,b}\}^k_{a,b=1})$ (notice that $m \leqslant k$ in general) are the conditional probability that a random hyperplane that does not separate m vertices of a fixed k-plet does not separate the whole set of k vertices of the same k-plet, symmetrized on the choice of the m excluded vertices. This can be written as

$$\begin{aligned}\tilde{\Psi}_m(\{\rho^{a,b}\}^k_{a,b=1}) &= \left\langle \frac{\Psi_m(\{\rho^{a,b}\}^k_{a,b=1})}{\Psi_{m-1}(\{\rho^{a,b}\}^m_{a,b=1})} \right\rangle_{\text{sym}} \\ &= \left\langle \frac{\int_{\mathcal{S}_{N-1}} dw \prod^k_{a,b=1} \theta\left((w \cdot x^a)(w \cdot x^b)\right)}{\int_{\mathcal{S}_{N-1}} dw \prod^m_{a,b=1} \theta\left((w \cdot x^a)(w \cdot x^b)\right)} \right\rangle_{\text{sym}}\end{aligned} \tag{4.66}$$

where angular bracket denote symmetrization with respect to the excluded vertex, and $\{x^a\}_{a=1}^k$ are the vertices of the k-plet. A non-trivial fact is that Ψ_m is a function of the $\frac{k(k-1)}{2}$ overlaps $\rho^{a,b}$ only, and thus it is independent on the choice of the k-plet, as long as its geometry is fixed by the model. This can be justified by noticing that Ψ_m is invariant if we rigidly rotate the m points x^a.

Notice that $\tilde{\Psi}_m(\{\rho^{a,b}\}_{a,b=1}^k)$ depends both on m explicitly, and on k implicitly. In the rest of the section, we will abuse the notation by writing $\tilde{\Psi}_m(\{\rho^{a,b}\}_{a,b=1}^k) = \tilde{\Psi}_m(\rho)$, assuming that the dimension of the polytopes k is fixed.

In this more general case, writing the initial conditions for Equation (4.64), and solving explicitly Equation (4.64) and Equation (4.65) is much more cumbersome than in the $k = 1, 2$ cases. While both elements could be obtained explicitly, we notice that (i) for $k \ll N$ we expect that modifying the $k-1$ non-trivial initial conditions into those of the $k = 1$ case should not alter the asymptotic results and (ii) the explicit form of $C_{N,P}^{(k)}$ is not needed in order to generalize the expressivity results of the previous section. For these reasons, we will not explore these elements further.

Figure 4.10, panel (a), compares the mean field value of $C_{N,P}^{(3)}$ to numerical simulations, showing again perfect agreement.

Let's prove Equation (4.64) and its companions. We generalize carefully the $k = 2$ case.

We suppose that $C_{N,P}^{(k)}$ is given, and add a new random k-plet to the sample set. Among the $C_{N,P}^{(k)}$ classifiable labellings of P k-plets, we have $Q_{k,N,P}$ labellings that correctly classify the original dataset enlarged with the last $k-1$ points of the new k-plet. $Q_{k,N,P}$ will depend on the set $\{C_{N-l,P}^{(k)}\}_{l=0}^{k-1}$. This will be clear in the following, and it's due to the fact that the logic that we will follow can be applied recursively for lesser values of k. The subscript k is a reminder that $Q_{k,N,P}$ depends on k variables.

Among these $Q_{k,N,P}$ we have $R_{k,N,P}$ labellings that can be realized by an hyperplane passing through the first point of the new k-plet. The $R_{k,N,P}$ labellings are all realizable thanks to the possibility of slightly deforming the separating hyperplane, the other $Q_{k,N,P} - R_{k,N,P}$ are realizable only if the first point of the k-plet is not separated by its companions by the separating hyperplane, which happens with probability $\tilde{\Psi}_k(\rho)$. Thus

$$C_{N,P+1}^{(k)} = \tilde{\Psi}_k(\rho) Q_{k,N,P} + (1 - \tilde{\Psi}_k(\rho)) R_{k,N,P} \,. \tag{4.67}$$

But $R_{k,N,P} = Q_{k,N-1,P}$, as we are just constraining the separating hyperplane with a single one-dimensional linear constraint. Finally, we notice that $Q_{k,N,P}$ is functionally determined by the same equation as $C_{N,P}^{(k)}$, but with $k \to k-1$. Thus,

$$C_{N,P+1}^{(k)} = f_k(C_{N,P}^{(k)}, \dots, C_{N-k,P}^{(k)}) \tag{4.68}$$

where

$$\begin{aligned} f_k(x_N, \dots, x_{N-k}) = {} & \tilde{\Psi}_k(\rho) f_{k-1}(x_N, \dots, x_{N-k+1}) \\ & + (1 - \tilde{\Psi}_k(\rho)) f_{k-1}(x_{N-1}, \dots, x_{N-k}) \end{aligned} \tag{4.69}$$

with initial condition $f_2(x_N, x_{N-1}) = x_N + x_{N-1}$. Inserting the ansatz of Equation (4.64) allows to obtain the recursion for the θ coefficient given in Equation (4.65).

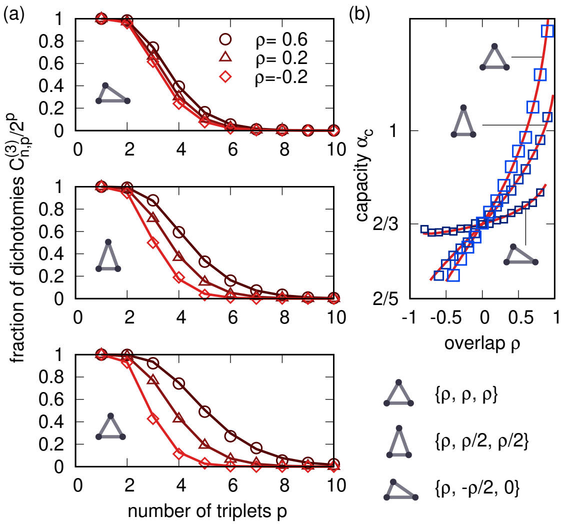


Figure 4.10: Theory vs numerical results for structured data $k = 3$. Numerical simulations (open markers in the plots) are obtained by training a linear classifier on 1000 randomly labelled dataset of 3-plets using the perceptron algorithm, and computing the fraction of perfectly classifiable trials. Circles ($\circ$), triangles ($\triangle$) and diamonds ($\diamond$) denotes different values of $\rho = 0.6, 0.2, -0.2$. Theory (solid lines) is given by Equation (4.64). (a) For different kinds of geometry (see legend in the figure) and values of ρ, numerical simulations are in perfect agreement with the mean field theory. (b) The theoretical capacity (Equation (4.58)) agrees with numerical simulations. The numerical capacity was computed by linearly interpolating data such as those shown in panel (a). Notice that the range of ρ in the three different geometries is restricted differently by the spherical constraint. (Reprinted from [RLG20]).

4.4.2 Expressivity of linear classifiers on polytopes

Capacity and VC dimension To access the capacity and the VC dimension we can again leverage the random path analogy, if we approximate the initial conditions to those of the $k = 1$ case as already discussed. Indeed, Equation (4.64) is in the same form of Equation (4.48). Thus, $C^{(k)}_{N,P}$ is the total weight of all walks $\{\gamma_j\}_{j=1}^{P}$ such that $\gamma_1 = 0$, $\gamma_{j+1} - \gamma_j = 0, 1, \ldots, k$ and $\gamma_P - \gamma_1 < N$. Each step of the path is weighted such that if $\gamma_{j+1} - \gamma_j = l$, then the weight equals $\theta^{(k)}_l$, and the total weight of a path is the product of the weights of its steps. Equivalently, if the weights $\theta^{(k)}_l$ are properly normalized, $C^{(k)}_{N,P}$ equals the probability that the corresponding random walk ends at height lesser than N after $P - 1$ steps.

Thus, the VC dimension is given by the value of P such that the probability associated to $C^{(k)}_{N,P}$ equals 1. But the maximum height of a random walk is $k(P - 1)$, giving (at leading order in $P, N \to \infty$)

$$d_{\mathrm{VC}} = \frac{N}{k}\,. \tag{4.70}$$

The critical load α_c is instead given by the value of $P = \alpha_c N$ such that the median of the distribution of the final height of the random walk equals N. Approximating the median with the mean, this gives

$$N = (\alpha_c N - 1)\frac{\sum_{l=0}^{k} l\theta_l^{(k)}}{\sum_{l=0}^{k} \theta_l^{(k)}} \tag{4.71}$$

that is (at leading order in $P, N \to \infty$)

$$\alpha_c = \frac{\sum_{l=0}^{k} \theta_l^{(k)}}{\sum_{l=0}^{k} l\theta_l^{(k)}} = \left(k - \frac{1}{2} - \sum_{l=2}^{k} \tilde{\Psi}_l(\rho)\right)^{-1} . \tag{4.72}$$

Figure 4.10, panel (b), compares Equation (4.72) to numerical simulations for $k = 3$ and three different families of one-parameter geometries (equilateral, isosceles and general triangles). The mean field theory agrees perfectly with the numerical simulations, validating *a posteriori* all the approximations.

Notice that the critical capacity can be computed by studying the associated Gardner's volume, which is the same as that given in Equation (4.59), with the substitution 2 → k.

The last equality can be derived as follows. If we define $\lambda_m^{(k)} = \sum_{l=0}^{k} l^m \theta_l^{(k)}$, Equation (4.65) implies that

$$\lambda_0^{(k)} = \lambda_0^{(k-1)} \tag{4.73}$$

i.e. $\lambda_0^{(k)} = 2$ for all values of k as the initial condition is given by $\lambda_0^{(1)} = 2$. Instead

$$\lambda_1^{(k)} = \lambda_1^{(k-1)} + (1 - \tilde{\Psi}_k(\rho))\lambda_0^{(k-1)} \tag{4.74}$$

with initial condition $\lambda_1^{(1)} = 1$, giving

$$\lambda_1^{(k)} = 2k - 1 - 2\sum_{l=2}^{k} \tilde{\Psi}_l(\rho) . \tag{4.75}$$

This recovers the last equality in Equation (4.72).

Logarithm of $C_{N,P}$ Again, the behaviour of $\log(C_{N,P})$ entropy for large load α can be derived by generating functions techniques. This gives that, for large P, N and large load α,

$$C_{N,\alpha N}^{(k)} \sim \frac{2(\alpha N)^{N-1}}{\Gamma(N)} \left[\theta_1^{(k)}\right]^{N-1} \left[\theta_0^{(k)}\right]^{(\alpha-1)N} . \tag{4.76}$$

Notice that $\theta_1^{(2)} = 1$ and $\theta_0^{(2)} = \tilde{\Psi}_2(\{\rho^{a,b}\}_{a,b=1}^2) = \Psi_2(\{\rho^{a,b}\}_{a,b=1}^2)$, so that we obtain the $k = 2$ case as a special case of the last equation. Again, for large α the asymptotic behaviour is an exponential decay induced by the geometry (notice that $\theta_0^{(k)} < 1$ if not all overlaps are equal to 1, as it is a product of the $\tilde{\Psi}_k(\rho)$ coefficients).

Equation (4.76) is again implied by Equation (4.64) by (i) defining the corresponding generating functions, (ii) obtaining a recursion for the generating function and (iii) studying the recursion at leading order in the dominant singularity. For the details, see Section 4.3.2 or [PRE$^+$20].

It turns out that the dominant singularity of $g_N(z)$ is a order N pole located at $z_0 = 1/\theta_0^{(k)}$, with coefficient $2[\theta_1^{(k)}]^{N-1}[\theta_0^{(k)}]^{-N}$. Singularity analysis then implies Equation (4.76).

Table 4.1 compares the results obtained up to this point for the expressivity of linear classifiers on structured and unstructured data.

Model	d_{VC}	Capacity	$C_{N,P}$ at large load
$k=1$	N	2	$\sim \frac{2(\alpha N)^{N-1}}{\Gamma(N)}$
$k=2$	$\frac{N}{2}$	$\frac{2}{3-2\Psi_2(\rho)}$	$\sim \frac{2(\alpha N)^{N-1}}{\Gamma(N)}\left[\Psi_2(\rho)\right]^{(\alpha-1)N}$
$k>2$	$\frac{N}{k}$	$\left(k-\frac{1}{2}-\sum_{l=2}^{k}\tilde{\Psi}_l(\{\rho^{a,b}\}_{a,b=1}^{k})\right)^{-1}$	$\sim \frac{2(\alpha N)^{N-1}}{\Gamma(N)}\left[\theta_1^{(k)}\right]^{N-1}\left[\theta_0^{(k)}\right]^{(\alpha-1)N}$

Table 4.1: Summary of expressivity measures for structured and unstructured data. The main qualitative difference between structured and unstructure data is the behaviour of $\log(C_{N,P})$ for large load α. In the structured case, $\log(C_{N,P})$ converges to zero, while in the unstructured case it diverges logarithmically.

4.5 Geometric structure may forbid separability

Figure 4.9, and its logarithmic version Figure 4.11 suggest to look more closely at the large load regime of $\log(C_{N,P})$. Both figures show that at large load α polytopes are much more difficult to separate than points. Qualitatively, this can be easily understood. If α is large, there may be so many polytopes in the dataset that all hyperplanes intersect at least one of them, meaning that there is no admissible linear classifier. In this sense, geometric structure forbids separability even before knowing anything about the labelling of the samples.

This remark closely resembles the definition of a constraint satisfaction problem, suggesting that the change in behaviour may be sharp and due to a phase transition. This is further supported by the fact that in Figure 4.11 the curves at different values of N intersect at the same value of the load.

4.5.1 Locating the new phase transition

To pinpoint the value α_* that defines the transition, we follow Figure 4.11 and we look for the (smallest) value of the load such that

$$\lim_{N\to\infty} \partial_N \log C_{N,\alpha_* N} = 0\,. \tag{4.77}$$

To compute explicitly α_*, we resort to the asymptotic expression of $C_{N,\alpha N}$ for large load α that we derived in the previous sections, and that are summarized in Table 4.1. Notice that these expressions are valid for fixed large N and large α, while Equation (4.77) would need an asymptotic expansion for fixed α and large N. We will validate this approximation *a posteriori* by showing that the estimates of α_* that we will obtain match with the numerical simulations.

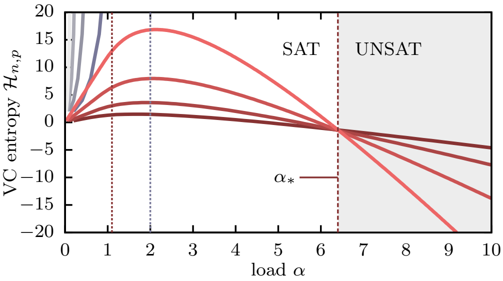


Figure 4.11: Comparison between the $\log(C_{N,P})$ of unstructured and structure data. The plot shows the logarithm of the number of classifiable labellings of unstructured ($k = 1$, grey) and structured ($k = 2$, red, $\rho \neq 1$) data, for $N = 5, 10, 20, 40$ (small N corresponding to dark colors, big N to lighter colors). The number of classifiable labellings switches from being increasing to being decreasing. For $\alpha = \alpha_*$, all curves intersect. This is a typical sign of a phase transition. In this case, the phase transition can be interpreted as a SAT/UNSAT transition between a low α regime in which at least a small number of configurations of labelled polytopes is linearly separable, and a high α regime in which the extended geometry of polytopes completely prevents linear separability, irrespective of the actual labelling. (Reprinted from [PRE+20]).

According to Equation (4.76),

$$\log C_{N,\alpha N} \sim \log\left[2\frac{\Gamma\left(\alpha N+N\right)}{\Gamma\left(N\right)\Gamma\left(\alpha N+1\right)}\right] + (N-1)\log\theta_1^{(k)} + (\alpha-1)N\log\theta_0^{(k)}\,. \tag{4.78}$$

Deriving with respect to N, and imposing Equation (4.77), we find

$$\log\theta_1^{(k)} + (\alpha-1)\log\theta_0^{(k)} - \psi(N) - \alpha\psi(\alpha N+1) + (\alpha+1)\psi(N+\alpha N) = 0\,, \tag{4.79}$$

where $\psi(x) = \partial_x \log\Gamma(x)$ is the digamma function, whose behaviour for large argument id given by $\psi(x) \sim \log(x)$. Thus, taking the leading order in N, we obtain that α_* is determined by

$$\log\theta_1^{(k)} + (\alpha_*-1)\log\theta_0^{(k)} + (\alpha_*+1)\log(\alpha_*+1) - \alpha_*\log(\alpha_*) = 0\,. \tag{4.80}$$

Figure 4.12 compares numerical and theoretical estimates of α_*, showing good agreement and verifying the goodness of our approximations. Notice also that when the geometry is trivial, i.e. if all overlaps tend to $\rho^{a,b} = 1$, α_* diverges to infinity as, in the unstructured case, there is no transition.

It may be worth now to determine explicitly the geometric coefficients $\theta_j^{(k)}$, $j = 0, 1$, using Equation (4.65). For $j = 0$, we see that the second term of Equation (4.65) is null, so that the recursion can be easily solved. Thus

$$\theta_0^{(k)} = \prod_{l=2}^{k}\tilde{\Psi}_l(\{\rho^{a,b}\}_{a,b=1}^{k})\,. \tag{4.81}$$

For $j = 1$, Equation (4.65) simplifies to a non-homogeneous linear recursion when the explicit form of $\theta_0^{(k)}$ is used. The recursion can be solved with standard techniques,

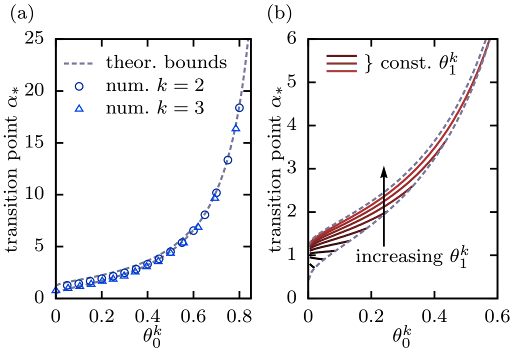


Figure 4.12: Comparison between numerical and theoretical values of α_*. (a) Numerical estimates of α_* at varying $\theta_0^{(k)}$ for two different geometries: $k=2$ (where $\theta_0^{(2)}$ is just Ψ_2) and $k=3$. In the latter case we fix $\{\rho^{a,b}\}_{a,b=1}^3$ by requiring that the three points in the k-plet form an equilateral triangle of varying size. (b) Theoretical results (red curves) for α_* as a function of $\theta_0^{(k)}$ for increasing values of $\theta_1^{(k)}$, within its allowed ranges given by Equations (4.88) and (4.89). Dashed lines in both panels are the k-independent upper and lower bounds for α_* obtained from Equation (4.88) and Equation (4.89). (Reprinted from [PRE+20]).

giving

$$\theta_1^{(k)} = \left(2 - k + \sum_{l=2}^{k} \left[\tilde{\Psi}_l(\{\rho^{a,b}\}_{a,b=1}^k)\right]^{-1}\right) \prod_{l=2}^{k} \tilde{\Psi}_l(\{\rho^{a,b}\}_{a,b=1}^k) \tag{4.82}$$

Specializing to $k=3$, for instance, yields

$$\begin{aligned} \theta_0^{(3)} &= \tilde{\Psi}_3 \tilde{\Psi}_2 \,, \\ \theta_1^{(3)} &= \tilde{\Psi}_3 + \tilde{\Psi}_2 - \tilde{\Psi}_3 \tilde{\Psi}_2 \,, \end{aligned} \tag{4.83}$$

where all $\tilde{\Psi}$ depend on $\{\rho^{a,b}\}_{a,b=1}^3$.

It is also useful to study the interdependence between $\theta_0^{(k)}$ and $\theta_1^{(k)}$ as the geometry of the k-plets varies, as the two quantities are not independent. While it is not possible to do this in general, one can provide interdependent bounds on both quantities under the simplifying assumptions that the geometric quantities $\tilde{\Psi}_k$ are independent variables ranging in the interval $[0,1]$. It's easy to see than that the range of $\theta_0^{(k)}$ is the interval $[0,1]$. To compute strict upper and lower bound on $\theta_1^{(k)}$ at fixed $\theta_0^{(k)}$, we resort to Lagrange constrained optimization. First of all, let us simplify the notation. Define:

$$\begin{aligned} x_m &:= \tilde{\Psi}_{m+1} \,, \quad \forall m \geqslant 1 \\ f_{(k)}(x_1 \dots x_k) &:= \theta_1^{(k+1)}(\tilde{\Psi}_2 \dots \tilde{\Psi}_{k+1}) \,, \quad \forall k \geqslant 1 \,. \end{aligned} \tag{4.84}$$

Explicitly:

$$f_{(k)}(\vec{x}) = \left(1 - k + \sum_{m=1}^{k} \frac{1}{x_m}\right) \prod_{m=1}^{k} x_m \,, \tag{4.85}$$

where $\vec{x} = (x_1 \dots x_k)$.

Our problem is to optimize (i.e., to find the infimum and the supremum) $f_{(k)}(\vec{x})$ in the hypercube $\vec{x} \in [0,1]^k$, subject to the constraint

$$\prod_{m=1}^{k} x_m = t \in [0,1]\,. \tag{4.86}$$

We will prove by induction that

$$\begin{aligned} \sup_{\vec{x}\in[0,1]^k} f_{(k)}(\vec{x}) &= 1 - \delta_{t,0}\,, \quad \forall k \geqslant 1 \\ \inf_{\vec{x}\in[0,1]^k} f_{(k)}(\vec{x}) &= \phi(k,t)\,, \quad \forall k \geqslant 1 \end{aligned} \tag{4.87}$$

where $\phi(k,t) = (1-k)t + kt^{1-\frac{1}{k}}$. Notice that $\phi(k,t)$ is a monotone decreasing function of k, and it is always lesser than 1.

The case $t = 0$ is special, as the constraint restricts the domain to the origin and $f_{(k)}$ is null; in the following, suppose that $t > 0$.

If $k = 1$, the constraint implies that $x_1 = t$, so that $f_{(1)}(x_1) = f_{(1)}(t) = 1$. The fact that $\phi(1,t) = 1$ proves that the proposed bounds are indeed true.

If $k > 1$, we first look for critical points inside $[0,1]^k$ using Lagrange's theorem for constrained optimization; then, we optimize our function on the boundary of $[0,1]^k$ to look for non-critical extrema:

- inside the domain, Lagrange's theorem gives that $\vec{x}_* = (t^{\frac{1}{k}} \dots t^{\frac{1}{k}})$ is the only critical point, and $f_{(k)}(\vec{x}_*) = \phi(k,t)$;
- on the boundary, we have that at least one of the x variables (without loss of generality, let us take x_k to be this boundary variable) must be either 0 or 1; the former is not compatible with the constraint as $t > 0$, so $x_k = 1$. But $f_{(k)}(x_1 \dots x_{k-1}, 1) = f_{(k-1)}(x_1 \dots x_{k-1})$, and $t = \prod_{m=1}^{k} x_m = \prod_{m=1}^{k-1} x_m$, so that the constrained optimization of $f_{(k)}(\vec{x})$ on the boundary of the domain is equivalent to the constrained optimization of $f_{(k-1)}(\vec{x})$ on the full domain $[0,1]^{k-1}$.

Thus, the candidates for the infimum and the supremum of $f_{(k)}(\vec{x})$ are given by $\phi(k,t)$ (inside the domain, by Lagrange's theorem) and 1 or $\phi(k-1,t)$ (on the boundary of the domain, by induction hypothesis). The properties of ϕ imply that 1 is the supremum and $\phi(k,t)$ is the infimum of $f_{(k)}(\vec{x})$.

Finally, again by induction, we see that the supremum is realized on the point $(t, 1, \dots)$ and by all the distinct permutations of its coordinates, and that the infimum is realized by $(t^{\frac{1}{k}} \dots t^{\frac{1}{k}})$.

Thus, we have shown that

$$\begin{aligned} &\text{(i) } \theta_1^{(k)} \leqslant 1, \\ &\text{(ii) } \theta_1^{(k)} \geqslant (k-1)\left(\theta_0^{(k)}\right)^{1-\frac{1}{k-1}} + (2-k)\theta_0^{(k)}. \end{aligned} \tag{4.88}$$

The lower-bound is monotonically decreasing with k at fixed $\theta_0^{(k)}$; therefore, by letting $k \to \infty$ one obtains a global lower bound that depends on k only implicitly through

$\theta_0^{(k)}$:

$$\theta_1^{(k)} \geqslant \theta_1^{(\infty)} = \theta_0^{(k)} \left[1 - \log \theta_0^{(k)}\right] . \tag{4.89}$$

4.5.2 The new transition is a SAT/UNSAT transition

It's now useful to rationalize the new phase transition as a SAT/UNSAT transition, akin to that happening at the critical load α_c.

The constraint satisfaction problem (CSP) related to the critical load transition can be defined as follows:

Constraint satisfaction problem 1. *Given a set of P input-label pairs $\{\{x^{\mu,a}\}_{a=1}^k, y^\mu\}_{\mu=1}^P$, find a unit vector w such that* $\mathrm{sign}(w \cdot x^{\mu,a}) = y^\mu$ *for all μ and a.*

In other words, CSP 1 is the training problem of a linear classifier: we look for a parameter vector w that defines predictor function that can correctly classify the dataset. A random version of this CSP can be defined by specifying a probability distribution over the dataset. For example, for $k = 1$ we could study the random CSP defined by sampling i.i.d. sample/label pairs uniformly on the sphere and with random label, i.e. with probability density

$$P_{X,Y}(x, y) = \frac{1}{Z}\delta(||x|| - 1) \left[\delta_{y,1} + \delta_{y,-1}\right] , \tag{4.90}$$

where Z is the proper normalization. For $k > 1$ and for fixed set of $k(k-1)/2$ overlaps $\{\rho^{a,b}\}_{a,b=1}^k$, this can be generalized to k-plet/label pairs as follows

$$\begin{aligned} P_{X,Y}(\{x^a\}_{a=1}^k, y) = \frac{1}{Z} &\left[\prod_{a=1}^k \delta(||x^a|| - 1)\right] \\ &\times \left[\prod_{1 \leqslant a < b \leqslant k} \delta(x^a \cdot x^b - \rho^{a,b})\right] \left[\delta_{y,1} + \delta_{y,-1}\right] , \end{aligned} \tag{4.91}$$

where again Z is the proper normalization, now depending on the overlaps. The random CSP is then said to be in the satisfiable (SAT) phase if the deterministic CSP admits solution with probability that tends to one in the thermodynamic limit $P, N \to \infty$ at fixed load $\alpha = P/N$. *Viceversa*, the random CSP is in the unsatisfiable (UNSAT) phase if the deterministic CSP admits solution with probability tending to zero in the thermodynamic limit.

The critical load α_c identifies precisely the threshold of this phase transition. The corresponding Gardner's volume

$$V = \int dw \, p(w) \prod_{\mu=1}^P \prod_{a=1}^k \theta\left(y^\mu \left(w \cdot x^{\mu,a}\right)\right) , \tag{4.92}$$

translates the probability of existence of a solution of the CSP to the volume that solutions of the CSP occupy in parameter space: the SAT phase corresponds to an exponentially large volume of solutions, and the UNSAT phase to a sub-exponentially large, or vanishing, one. The probability density on the dataset translates in this picture to a form of quenched disorder.

At this point, it is useful to remind that making the link between the critical load, the corresponding random CSP and the relevant Gardner's volume is extremely useful. Indeed, in many problems the critical load may be accessible only through Gardner's approach, which is more general than the combinatorial approach described before at the cost of describing only the transition point and not the full $C_{N,P}$ curve. Thus, obtaining intuition on the nature of the phase transition happening at α_*, i.e. rationalizing it as a SAT/UNSAT transition and deriving the corresponding Gardner's volume opens up the possibility of computing α_* in other models, where the combinatorial approach may not be readily applied.

The correct CSP is obtained by promoting the label variables from quenched disorder to dynamical variables, i.e.

Constraint satisfaction problem 2. *Given a set of P inputs $\{x^{\mu,a}\}_{\mu,a=1}^{P,k}$, find a set of labels $\{y^\mu\}$ and a vector w such that* $\mathrm{sign}(w \cdot x^{\mu,a}) = y^\mu$ *for all μ and a.*

In this modified CSP, we are just asking whether there exists a labelling of the given inputs that is linearly separable. The corresponding UNSAT phase is thus given by the values of α such that $C_{N,\alpha N} \to 0$ in the thermodynamic limit. This is precisely what is happening in the structured case where $C_{N,\alpha N}$ is monotone decreasing to zero at large values of the load α. On the other hand, in the unstructured case, the CSP is always trivially solved by almost all hyperplanes w[11] choosing $y^\mu = \mathrm{sign}(w \cdot x^\mu)$, and this is reflected in the fact that $C_{N,\alpha N}$ is monotone increasing for all values of α. The relevant Gardner's volume is given by

$$V = \sum_{\{y^\mu=\pm1\}} \int dw\, p(w) \prod_{\mu=1}^{P} \prod_{a=1}^{k} \theta\left(y^\mu \left(w \cdot x^{\mu,a}\right)\right) , \tag{4.93}$$

where the summation over the labels states that the labels are now regarded as dynamical variables.

In [PRE+20; RPG20], this redefinition of the Gardner's volume was used to compute α_* for the polytope $k = 2$ model of data structure treated here, and for the margin learning problem. The details of the replica computation used lay outside of the scope of this Thesis, and can be found in [PRE+20]. Here I report the results.

Polytope model The value of α_* can be obtained by averaging $\log V$ over the random realizations of the sample set, and looking for the value of α at which the entropy goes to $-\infty$. As it usually happens with replica computations, different approximation schemes of increasing computational complexity can be used[12]: annealed, replica symmetric (RS), one-step replica symmetry breaking (1RSB), etc.

At the annealed level, i.e. assuming that $\langle \log V \rangle \sim \log \langle V \rangle$, the transition threshold equals

$$\alpha_*^A(\rho) = -\frac{1 + \log 2\pi}{2 \log\left(\frac{1}{2} + \frac{1}{\pi} \arcsin \rho\right)} . \tag{4.94}$$

[11] Almost because there may be hyperplanes intersecting exactly a point. These have null measure in parameter space, and can be disregarded.

[12] Notice that the right scheme for a replica computation is determined by the problem, and more complex schemes do not necessarily mean higher accuracy.

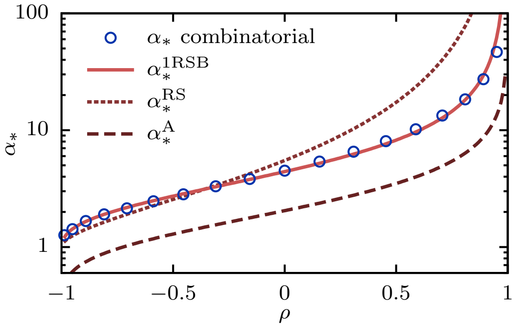


Figure 4.13: Comparison between combinatorial and replica estimates of α_* for $k = 2$. Circles represent the combinatorial result, which is in agreement with numerical simulations. All the different approximation schemes used for the replica computations display the same qualitative shape. However the annealed and RS ansatz fail in reproducing quantitatively the combinatorial result. Using a 1-RSB ansatz it is possible to obtain a one-parameter expression for α_* in Equation (4.96) that fits the combinatorial result tightly. (Reprinted from [PRE+20]).

A comparison of the annealed approximation and of the result obtained with combinatorics in Equation (4.80) is shown in Fig. 4.13. Although the annealed approximation fails in reproducing quantitatively the behavior of $\alpha_*(\rho)$, it bounds the combinatorial result from below, and qualitatively recovers the expected divergence for $\Psi_2(\rho) \to 1$ when the geometry trivializes ($\rho \to 1$).

At the RS level, the transition threshold equals

$$\alpha_*^{\text{RS}}(\rho) = \frac{\pi}{2\arctan\sqrt{(1-\rho)/(1+\rho)} - \sqrt{1-\rho^2}} . \tag{4.95}$$

The result is reported in Fig. 4.13: the RS curve presents the expected limits for $\rho \to \pm 1$, but again we do not observe quantitative agreement with the combinatorial curve. This leads to conjecture that one needs at least one step of replica symmetry breaking (1RSB), obtaining

$$\alpha_*^{1\text{RSB}}(\rho; q_0 = 0, w) = \frac{-\log[1+w]}{2\log\left[\frac{1}{2} + \frac{1}{\pi}\arcsin(\rho) + \frac{2\arctan\left(\sqrt{(1+w)\frac{1-\rho}{1+\rho}}\right)}{\pi\sqrt{1+w}}\right]} , \tag{4.96}$$

where q_0 and w are 1RSB parameters to be optimized. This last result is not the optimal 1RSB solution: in principle we should consider the full expression of $\alpha_*^{1\text{RSB}}(\rho; q_0, w)$ and optimize upon the remaining parameters q_0 and w. Here, the objective was only to verify that the functional form $\alpha_*^{1\text{RSB}}(\rho; q_0 = 0, w)$ allows to fit nicely the combinatorial result, by adjusting the parameter w (see Fig. 4.13). This simple observation strongly supports our conjecture that this SAT-UNSAT transition exhibits at least one step of RSB, but it does not rule out a full-RSB scenario.

Margin learning To show the versatility of Gardner's formulation, we report the results for the value of α_* in the case of margin learning, which cannot be treated with the combinatorial approach.

The synaptic volume relevant to this case is

$$V_\kappa = \sum_{\{y^\mu = \pm 1\}} \int dw\, p(w) \prod_{\mu=1}^{P} \theta(y^\mu (w \cdot x^\mu) - \kappa) \,, \tag{4.97}$$

where κ is the margin. Note again that here, as in the case of Equation (4.93), the outputs y^μ are dynamical variables, at variance with the usual Gardner's volume. The annealed approximation leads to

$$\alpha_*^{\mathrm{A}}(\kappa) = -\frac{1 + \log(2\pi)}{2 \log[2\, \mathrm{erfc}(\kappa)]} \,. \tag{4.98}$$

In the RS approximation one obtains the critical threshold

$$\alpha_*^{\mathrm{RS}}(\kappa) = \frac{1}{2} \left[\int_0^\kappa Dy\, (\kappa - y)^2 \right]^{-1} \,, \tag{4.99}$$

where Dy is the standard Gaussian measure. Finally, the 1RSB ansatz again depends on the parameters q_0 and w, which should be investigated numerically. However, in the special case $q_0 = 0$ one finds the simpler expression

$$\alpha_*^{\mathrm{1RSB}}(\kappa; q_0 = 0, w) = \frac{-\log[1 + w]}{2 \log \left\{ 2 \left[\mathrm{erfc}(\kappa) + \int_0^\kappa Dz\, e^{-w \frac{(z-\kappa)^2}{2}} \right] \right\}} \,. \tag{4.100}$$

These results essentially share the same features of those for the k-plets computed above: in particular, at variance with the usual storage capacity, α_* computed in all the different approximation schemes diverges in the limit $\kappa \to 0^+$, when the problem reduces to a standard classification of unstructured points.

Even in absence of a closed expression for $\log(C_{N,P})$ of margin classification, the existence of the phase transition at a finite load is a clear indication of its non-monotonicity.

4.6 Perspectives

Understanding how data specificities impact the performance of machine learning models and algorithms can be considered one of the major challenges for contemporary Statistical Physics. In this Chapter, we have explored how to deal with a simple model of data structure, and how to characterize the expressive power of linear classifiers over such datasets. The presence of input-output correlations in a dataset constraints the class of admissible predictor functions under consideration.

For simple models of data structure we summarized two phenomena that take place above the VC dimension. First, $\log(C_{N,P})$, i.e. a proxy of the VC entropy, becomes non-monotonic. This is a strong indication that the rigorous bounds in SLT may be substantially improved by taking data structure into account, and this is the first direction in which the work presented in this Chapter may be expanded. Second, a novel transition appears beyond the well-known storage capacity, at the onset of unsatisfiability for a data-related constraint satisfaction problem. When available, a combinatorial theory *à la Cover* allows one to compute $\log(C_{N,P})$ of a finite-size system, and to reveal explicitly its nonmonotonic behavior. However, this is not always feasible, such as for spherical object manifolds and margin learning. In these cases, the phase transition can be probed with the standard tools of statistical physics, thus allowing an indirect quantification of the data-dependent behavior.

The new satisfiability transition is due to a competition between the combinatorial expansion, with sample size, of the space of possible functions and the reduction due to the constraints [RPG20]. This observation suggests that the emergence of the data-driven transition, as well as the nonmonotonic proxy of the VC entropy it entails, is not specific to the two models of data that we have studied here, but is more generally present whenever the constraints imposed on the set of predictor functions by data structure are strong enough.

Other possible directions of improvement here are mainly given by the possibility of extending the analysis to more complex models of geometrically structured data and learning architectures. As for the storage capacity, the description in terms of a Gardner's volume suggests that a form of universality holds for the novel data-driven transition. It remains to be seen whether this hold in practice or not.

On a more technical level, the combinatorial mean-field approach could be improved by computing the asymptotic behaviour of $C_{N,P}$ not only at large load, but also at fixed load. This should be possible within the techniques provided by analytic combinatorics, but it is a path that remains to be explored.

4.7 Chapter bibliography

[AAK+20] Alia Abbaras, Benjamin Aubin, Florent Krzakala, and Lenka Zdeborová. Rademacher complexity and spin glasses: A link between the replica and statistical theories of learning. In Jianfeng Lu and Rachel Ward, editors, *Proceedings of The First Mathematical and Scientific Machine Learning Conference*, volume 107 of *Proceedings of Machine Learning Research*, pages 27–54, Princeton University, Princeton, NJ, USA. PMLR, 20–24 Jul 2020. URL: http://proceedings.mlr.press/v107/abbaras20a.html.

[ALR+16] Fabio Anselmi, Joel Z. Leibo, Lorenzo Rosasco, Jim Mutch, Andrea Tacchetti, and Tomaso Poggio. Unsupervised learning of invariant representations. *Theoretical Computer Science*, 633:112–121, 2016. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2015.06.048. Biologically Inspired Processes in Neural Computation.

[BLR+19] Francesco Borra, Marco Cosentino Lagomarsino, Pietro Rotondo, and Marco Gherardi. Generalization from correlated sets of patterns in the perceptron. *Journal of Physics A: Mathematical and Theoretical*, 52(38):384004, August 2019. DOI: 10.1088/1751-8121/ab3709.

[Bot15] Léon Bottou. Making vapnik–chervonenkis bounds accurate. In *Measures of Complexity*, pages 143–155. Springer, 2015. DOI: 10.1007/978-3-319-21852-6_9.

[BR92] Avrim L Blum and Ronald L Rivest. Training a 3-node neural network is np-complete. *Neural Networks*, 5(1):117–127, 1992.

[Bre95] Leo Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995. DOI: 10.1080/00401706.1995.10484371.

[CBP21] Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 12(1):2914, May 2021. ISSN: 2041-1723. DOI: 10.1038/s41467-021-23103-1.

[CCL$^+$20] Uri Cohen, SueYeon Chung, Daniel D. Lee, and Haim Sompolinsky. Separability and geometry of object manifolds in deep neural networks. *Nature Communications*, 11(1):746, February 2020. ISSN: 2041-1723. DOI: 10.1038/s41467-020-14578-5.

[CLS16] SueYeon Chung, Daniel D. Lee, and Haim Sompolinsky. Linear readout of object manifolds. *Phys. Rev. E*, 93:060301, 6, June 2016. DOI: 10.1103/PhysRevE.93.060301.

[CLS18] SueYeon Chung, Daniel D. Lee, and Haim Sompolinsky. Classification and geometry of general perceptual manifolds. *Phys. Rev. X*, 8:031003, 3, July 2018. DOI: 10.1103/PhysRevX.8.031003.

[Cov65] Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334, June 1965. DOI: 10.1109/pgec.1965.264137.

[DOS99] Rainer Dietrich, Manfred Opper, and Haim Sompolinsky. Statistical Mechanics of Support Vector Networks. en. *Physical Review Letters*, 82(14):2975–2978, April 1999. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.82.2975.

[DRB$^+$20] Stéphane D'Ascoli, Maria Refinetti, Giulio Biroli, and Florent Krzakala. Double trouble in double descent: bias and variance(s) in the lazy regime. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2280–2290. PMLR, 13–18 Jul 2020. URL: http://proceedings.mlr.press/v119/d-ascoli20a.html.

[EV01] Andreas Engel and Christian Van den Broeck. *Statistical Mechanics of Learning*. Cambridge University Press, 2001.

[FS09] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. en. Cambridge University Press, Cambridge ; New York, 2009. ISBN: 978-0-521-89806-5.

[Gar88] E Gardner. The space of interactions in neural network models. *Journal of Physics A: Mathematical and General*, 21(1):257–270, January 1988. DOI: 10.1088/0305-4470/21/1/030.

[GMK$^+$20] Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová. Modeling the influence of data structure on learning in neural networks: the hidden manifold model. *Phys. Rev. X*, 10:041044, 4, December 2020. DOI: 10.1103/PhysRevX.10.041044.

[HK70] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. DOI: 10.1080/00401706.1970.10488634.

[HS$^+$99] Geoffrey E Hinton, Terrence Joseph Sejnowski, et al. *Unsupervised learning: foundations of neural computation*. MIT press, 1999.

[KW+96] Ron Kohavi, David H Wolpert, et al. Bias plus variance decomposition for zero-one loss functions. In *ICML*, volume 96, pages 275–83, 1996.

[MM20] Song Mei and Andrea Montanari. The generalization error of random features regression: precise asymptotics and double descent curve, 2020. arXiv: 1908.05355 [math.ST].

[MRT18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[NKB+20] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020. URL: https://openreview.net/forum?id=B1g5sA4twr.

[PRE+20] Mauro Pastore, Pietro Rotondo, Vittorio Erba, and Marco Gherardi. Statistical learning theory of structured data. *Phys. Rev. E*, 102:032119, 3, September 2020. DOI: 10.1103/PhysRevE.102.032119.

[RLG20] Pietro Rotondo, Marco Cosentino Lagomarsino, and Marco Gherardi. Counting the learnable functions of geometrically structured data. *Phys. Rev. Research*, 2:023169, 2, May 2020. DOI: 10.1103/PhysRevResearch.2.023169.

[RPG20] Pietro Rotondo, Mauro Pastore, and Marco Gherardi. Beyond the storage capacity: data-driven satisfiability transition. *Phys. Rev. Lett.*, 125:120601, 12, September 2020. DOI: 10.1103/PhysRevLett.125.120601.

[SB18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Sej20] Terrence J. Sejnowski. The unreasonable effectiveness of deep learning in artificial intelligence. *Proceedings of the National Academy of Sciences*, 117(48), 2020. ISSN: 0027-8424. DOI: 10.1073/pnas.1907373117.

[SL00] H. Sebastian Seung and Daniel D. Lee. The manifold ways of perception. *Science*, 290(5500):2268–2269, 2000. ISSN: 0036-8075. DOI: 10.1126/science.290.5500.2268.

[SMG19] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, 2019. ISSN: 0027-8424. DOI: 10.1073/pnas.1820226116.

[Wol18] Michael M. Wolf. Mathematical foundations of supervised learning, July 2018. URL: http://131.159.69.70/foswiki/pub/M5/Allgemeines/MA4801_2018S/ML_notes_main.pdf.

CHAPTER 5

Bibliography

[AAK+20] Alia Abbaras, Benjamin Aubin, Florent Krzakala, and Lenka Zdeborová. Rademacher complexity and spin glasses: A link between the replica and statistical theories of learning. In Jianfeng Lu and Rachel Ward, editors, *Proceedings of The First Mathematical and Scientific Machine Learning Conference*, volume 107 of *Proceedings of Machine Learning Research*, pages 27–54, Princeton University, Princeton, NJ, USA. PMLR, 20–24 Jul 2020. URL: http://proceedings.mlr.press/v107/abbaras20a.html.

[AB02] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, 1, January 2002. DOI: 10.1103/RevModPhys.74.47.

[AB20] Konstantin E. Avrachenkov and Andrei V. Bobu. Cliques in high-dimensional random geometric graphs. *Applied Network Science*, 5(1):92, November 2020. ISSN: 2364-8228. DOI: 10.1007/s41109-020-00335-6.

[AFD+20] Michele Allegra, Elena Facco, Francesco Denti, Alessandro Laio, and Antonietta Mira. Data segmentation based on the local intrinsic dimension. *Scientific Reports*, 10(1):16449, October 2020. ISSN: 2045-2322. DOI: 10.1038/s41598-020-72222-0.

[All18] Alfonso Allen-Perkins. Random spherical graphs. *Physical Review E*, 98(3), September 2018. DOI: 10.1103/PhysRevE.98.032310.

[ALM+19] Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/cfcce0621b49c983991ead4c3d4d3b6b-Paper.pdf.

[ALR+16] Fabio Anselmi, Joel Z. Leibo, Lorenzo Rosasco, Jim Mutch, Andrea Tacchetti, and Tomaso Poggio. Unsupervised learning of invariant representations. *Theoretical Computer Science*, 633:112–121, 2016. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2015.06.048. Biologically Inspired Processes in Neural Computation.

[Bar11] Marc Barthélemy. Spatial networks. *Physics Reports*, 499(1):1–101, 2011. ISSN: 0370-1573. DOI: https://doi.org/10.1016/j.physrep.2010.11.002.

[BBC+21] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: grids, groups, graphs, geodesics, and gauges, 2021. arXiv: 2104.13478 [cs.LG].

[BCD+18] Alessandro Benfenati, Emilie Chouzenoux, Laurent Duval, Jean-Christophe Pesquet, and Aurélie Pirayre. A review on graph optimization and algorithmic frameworks. Research Report, LIGM - Laboratoire d'Informatique Gaspard-Monge, October 2018. URL: https://hal.archives-ouvertes.fr/hal-01901499.

[BDE+16] Sébastien Bubeck, Jian Ding, Ronen Eldan, and Miklós Z. Rácz. Testing for high-dimensional geometry in random graphs. en. *Random Structures & Algorithms*, 49(3):503–532, October 2016. ISSN: 1098-2418. DOI: 10.1002/rsa.20633.

[BLG17] Arianna Bottinelli, Rémi Louf, and Marco Gherardi. Balancing building and maintenance costs in growing transport networks. *Phys. Rev. E*, 96:032316, 3, September 2017. DOI: 10.1103/PhysRevE.96.032316.

[BLR+19] Francesco Borra, Marco Cosentino Lagomarsino, Pietro Rotondo, and Marco Gherardi. Generalization from correlated sets of patterns in the perceptron. *Journal of Physics A: Mathematical and Theoretical*, 52(38):384004, August 2019. DOI: 10.1088/1751-8121/ab3709.

[BM08] J Adrian Bondy and Uppaluri SR Murty. Graph theory. In Springer, 2008.

[BML06] Yoshua Bengio, Martin Monperrus, and Hugo Larochelle. Nonlocal estimation of manifold structure. *Neural Computation*, 18(10):2509–2528, 2006. DOI: 10.1162/neco.2006.18.10.2509.

[Bot15] Léon Bottou. Making vapnik–chervonenkis bounds accurate. In *Measures of Complexity*, pages 143–155. Springer, 2015. DOI: 10.1007/978-3-319-21852-6_9.

[BR92] Avrim L Blum and Ronald L Rivest. Training a 3-node neural network is np-complete. *Neural Networks*, 5(1):117–127, 1992.

[Bre95] Leo Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995. DOI: 10.1080/00401706.1995.10484371.

[Bur17] Amanda Burcroff. Johnson schemes and certain matrices with integral eigenvalues. *University of Michigan, Tech. Rep*, 2017. URL: http://math.uchicago.edu/~may/REU2017/REUPapers/Burcroff.pdf.

[Cam03] Francesco Camastra. Data dimensionality estimation methods: a survey. en. *Pattern Recognition*, 36(12):2945–2954, December 2003. ISSN: 00313203. DOI: 10.1016/S0031-3203(03)00176-6.

[Cao97] Liangyue Cao. Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D: Nonlinear Phenomena*, 110(1-2):43–50, December 1997. DOI: 10.1016/s0167-2789(97)00118-8.

[CBP21] Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 12(1):2914, May 2021. ISSN: 2041-1723. DOI: 10.1038/s41467-021-23103-1.

[CBR+14] Claudio Ceruti, Simone Bassis, Alessandro Rozza, Gabriele Lombardi, Elena Casiraghi, and Paola Campadelli. Danco: an intrinsic dimensionality estimator exploiting angle and norm concentration. *Pattern recognition*, 47(8):2569–2581, August 2014. DOI: 10.1016/j.patcog.2014.02.013.

[CCL+20] Uri Cohen, SueYeon Chung, Daniel D. Lee, and Haim Sompolinsky. Separability and geometry of object manifolds in deep neural networks. *Nature Communications*, 11(1):746, February 2020. ISSN: 2041-1723. DOI: 10.1038/s41467-020-14578-5.

[CDE+20] Sergio Caracciolo, Matteo P. D'Achille, Vittorio Erba, and Andrea Sportiello. The dyck bound in the concave 1-dimensional random assignment model. *Journal of Physics A: Mathematical and Theoretical*, 53(6):064001, January 2020. DOI: 10.1088/1751-8121/ab4a34.

[Cer14] Claudio Ceruti. *Novel techniques for intrinsic dimension estimation*. PhD thesis, Scuola di Dottorato in Matematica e Statistica per le Scienze Computazionali - XXVII ciclo - Dipartimento di Matematica "Federigo Enriques", 2014. DOI: 10.13130/ceruti-claudio_phd2014-12-16.

[CES20] Sergio Caracciolo, Vittorio Erba, and Andrea Sportiello. The p-airy distribution, 2020. arXiv: 2010.14468 [math.CO].

[CES21] Sergio Caracciolo, Vittorio Erba, and Andrea Sportiello. The number of optimal matchings for euclidean assignment on the line. *Journal of Statistical Physics*, 183(1):3, March 2021. ISSN: 1572-9613. DOI: 10.1007/s10955-021-02741-1.

[Cho05] André-Louis Cholesky. Sur la résolution numérique des systèmes d'équations linéaires. *Bulletin de la Sabix. Société des amis de la Bibliothèque et de l'Histoire de l'École polytechnique*, (39):81–95, 2005.

[CLS16] SueYeon Chung, Daniel D. Lee, and Haim Sompolinsky. Linear readout of object manifolds. *Phys. Rev. E*, 93:060301, 6, June 2016. DOI: 10.1103/PhysRevE.93.060301.

[CLS18] SueYeon Chung, Daniel D. Lee, and Haim Sompolinsky. Classification and geometry of general perceptual manifolds. *Phys. Rev. X*, 8:031003, 3, July 2018. DOI: 10.1103/PhysRevX.8.031003.

[Cov65] Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334, June 1965. DOI: 10.1109/pgec.1965.264137.

[CRH10] K.M. Carter, R. Raich, and A.O. Hero. On local intrinsic dimension estimation and its applications. *IEEE Transactions on Signal Processing*, 58(2):650–663, February 2010. DOI: 10.1109/tsp.2009.2031722.

[CSS+19] Giulio Cimini, Tiziano Squartini, Fabio Saracco, Diego Garlaschelli, Andrea Gabrielli, and Guido Caldarelli. The statistical physics of real-world networks. *Nature Reviews Physics*, 1(1):58–71, January 2019. ISSN: 2522-5820. DOI: 10.1038/s42254-018-0002-6.

[CV01] Francesco Camastra and Alessandro Vinciarelli. Intrinsic dimension estimation of data: an approach based on grassberger–procaccia's algorithm. *Neural Processing Letters*, 14(1):27–34, August 2001. ISSN: 1573-773X. DOI: 10.1023/A:1011326007550.

[CV02] F. Camastra and A. Vinciarelli. Estimating the intrinsic dimension of data with a fractal-based method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(10):1404–1407, October 2002. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2002.1039212.

[DC02] Jesper Dall and Michael Christensen. Random geometric graphs. en. *Physical Review E*, 66(1):016121, July 2002. ISSN: 1063-651X, 1095-3787. DOI: 10.1103/PhysRevE.66.016121.

[DGL+11] Luc Devroye, András György, Gábor Lugosi, and Frederic Udina. High-Dimensional Random Geometric Graphs and their Clique Number. en. *Electronic Journal of Probability*, 16(0):2481–2508, 2011. ISSN: 1083-6489. DOI: 10.1214/EJP.v16-967.

[DOS99] Rainer Dietrich, Manfred Opper, and Haim Sompolinsky. Statistical Mechanics of Support Vector Networks. en. *Physical Review Letters*, 82(14):2975–2978, April 1999. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.82.2975.

[DQV19] Mateo Díaz, Adolfo J. Quiroz, and Mauricio Velasco. Local angles and dimension estimation from data on manifolds. *Journal of Multivariate Analysis*, 173:229–247, September 2019. DOI: 10.1016/j.jmva.2019.02.014.

[DRB+20] Stéphane D'Ascoli, Maria Refinetti, Giulio Biroli, and Florent Krzakala. Double trouble in double descent: bias and variance(s) in the lazy regime. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2280–2290. PMLR, 13–18 Jul 2020. URL: http://proceedings.mlr.press/v119/d-ascoli20a.html.

[EAG+20] Vittorio Erba, Sebastiano Ariosto, Marco Gherardi, and Pietro Rotondo. Random geometric graphs in high dimension. *Phys. Rev. E*, 102:012306, 1, July 2020. DOI: 10.1103/PhysRevE.102.012306.

[EGR19] Vittorio Erba, Marco Gherardi, and Pietro Rotondo. Intrinsic dimension estimation for locally undersampled data. *Scientific Reports*, 9(1):17133, November 2019. ISSN: 2045-2322. DOI: 10.1038/s41598-019-53549-9.

[EPR21] Vittorio Erba, Mauro Pastore, and Pietro Rotondo. Self-induced glassy phase in multimodal cavity quantum electrodynamics. *Phys. Rev. Lett.*, 126:183601, 18, May 2021. DOI: 10.1103/PhysRevLett.126.183601.

[ER60] Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

[ER92] J.-P. Eckmann and D. Ruelle. Fundamental limitations for estimating dimensions and Lyapunov exponents in dynamical systems. en. *Physica D: Nonlinear Phenomena*, 56(2-3):185–187, May 1992. ISSN: 01672789. DOI: 10.1016/0167-2789(92)90023-G.

[ERO20] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis, 2020. arXiv: 2012.09841 [cs.CV].

[ES15] Ernesto Estrada and Matthew Sheerin. Random rectangular graphs. *Physical Review E*, 91(4):042805, April 2015. DOI: 10.1103/PhysRevE.91.042805.

[EV01] Andreas Engel and Christian Van den Broeck. *Statistical Mechanics of Learning*. Cambridge University Press, 2001.

[FdR+17] Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7(1), September 2017. DOI: 10.1038/s41598-017-11873-y.

[FK15] Alan Frieze and Michał Karoński. *Introduction to Random Graphs*. Cambridge University Press, 2015. DOI: 10.1017/CBO9781316339831.

[FO71] K. Fukunaga and D.R. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, C-20(2):176–183, February 1971. DOI: 10.1109/t-c.1971.223208.

[FPR+19] Elena Facco, Andrea Pagnani, Elena Tea Russo, and Alessandro Laio. The intrinsic dimension of protein sequence evolution. *PLOS Computational Biology*, 15(4):1–16, April 2019. DOI: 10.1371/journal.pcbi.1006767.

[FS09] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. en. Cambridge University Press, Cambridge ; New York, 2009. ISBN: 978-0-521-89806-5.

[FWV07] Damien Francois, Vincent Wertz, and Michel Verleysen. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19(7):873–886, 2007. DOI: 10.1109/TKDE.2007.1037.

[Gar88] E Gardner. The space of interactions in neural network models. *Journal of Physics A: Mathematical and General*, 21(1):257–270, January 1988. DOI: 10.1088/0305-4470/21/1/030.

[GC16] Daniele Granata and Vincenzo Carnevale. Accurate Estimation of the Intrinsic Dimension Using Graph Distances: Unraveling the Geometric Complexity of Datasets. en. *Scientific Reports*, 6(1):31377, November 2016. ISSN: 2045-2322. DOI: 10.1038/srep31377.

[Gen92] Alan Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1(2):141–149, 1992. DOI: 10.1080/10618600.1992.10477010.

[GGD15] Alexander P. Giles, Orestis Georgiou, and Carl P. Dettmann. Betweenness centrality in dense random geometric networks. en. In *2015 IEEE International Conference on Communications (ICC)*, pages 6450–6455, London. IEEE, June 2015. ISBN: 978-1-4673-6432-4. DOI: 10.1109/ICC.2015.7249352.

[GMK$^+$20] Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová. Modeling the influence of data structure on learning in neural networks: the hidden manifold model. *Phys. Rev. X*, 10:041044, 4, December 2020. DOI: 10.1103/PhysRevX.10.041044.

[GP83a] Peter Grassberger and Itamar Procaccia. Characterization of strange attractors. *Physical Review Letters*, 50(5):346–349, January 1983. DOI: 10.1103/physrevlett.50.346.

[GP83b] Peter Grassberger and Itamar Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1-2):189–208, October 1983. ISSN: 0167-2789. DOI: 10.1016/0167-2789(83)90298-1.

[GTR$^+$18] Robert Geirhos, Carlos R. M. Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. Generalisation in humans and deep neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/0937fb5864ed06ffb59ae5f9b5ed67a9-Paper.pdf.

[HA05] Matthias Hein and Jean-Yves Audibert. Intrinsic dimensionality estimation of submanifolds in rd. In *Proceedings of the 22nd international conference on Machine learning -ICML 05*. ACM Press, 2005. DOI: 10.1145/1102351.1102388.

[HK70] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. DOI: 10.1080/00401706.1970.10488634.

[HL81] Paul W. Holland and Samuel Leinhardt. An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33–50, 1981. DOI: 10.1080/01621459.1981.10477598.

[HS$^+$99] Geoffrey E Hinton, Terrence Joseph Sejnowski, et al. *Unsupervised learning: foundations of neural computation*. MIT press, 1999.

[KBD19] Alexander P. Kartun-Giles, Marc Barthelemy, and Carl P. Dettmann. The shape of shortest paths in random spatial networks. en. *Physical Review E*, 100(3):032315, September 2019. ISSN: 2470-0045, 2470-0053. DOI: 10.1103/PhysRevE.100.032315.

[Kég02] Balázs Kégl. Intrinsic dimension estimation using packing numbers. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, pages 697–704, Cambridge, MA, USA. MIT Press, 2002. URL: http://dl.acm.org/citation.cfm?id=2968618.2968705.

[KW$^+$96] Ron Kohavi, David H Wolpert, et al. Bias plus variance decomposition for zero-one loss functions. In *ICML*, volume 96, pages 275–83, 1996.

[LB04] Elizaveta Levina and Peter J. Bickel. Maximum likelihood estimation of intrinsic dimension. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, pages 777–784, Vancouver, British Columbia, Canada. MIT Press, 2004. URL: http://dl.acm.org/citation.cfm?id=2976040.2976138.

[LCB10] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[LLJ+09] Anna V. Little, Jason Lee, Yoon-Mo Jung, and Mauro Maggioni. Estimation of intrinsic dimensionality of samples from noisy low-dimensional manifolds in high dimensions with multiscale SVD. In *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*. IEEE, August 2009. DOI: 10.1109/ssp.2009.5278634.

[Llo82] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. DOI: 10.1109/TIT.1982.1056489.

[LMR17] Anna V. Little, Mauro Maggioni, and Lorenzo Rosasco. Multiscale geometric methods for data sets I: Multiscale SVD, noise and curvature. en. *Applied and Computational Harmonic Analysis*, 43(3):504–567, November 2017. ISSN: 10635203. DOI: 10.1016/j.acha.2015.09.009.

[LRC+11] Gabriele Lombardi, Alessandro Rozza, Claudio Ceruti, Elena Casiraghi, and Paola Campadelli. Minimum neighbor distance estimators of intrinsic dimension. In *Machine Learning and Knowledge Discovery in Databases*, pages 374–389. Springer Berlin Heidelberg, 2011. DOI: 10.1007/978-3-642-23783-6_24.

[LV07] John A. Lee and Michel Verleysen, editors. *Nonlinear Dimensionality Reduction*. Springer New York, 2007. DOI: 10.1007/978-0-387-39351-3.

[MAR+21] Tiago Mendes-Santos, Adriano Angelone, Alex Rodriguez, Rosario Fazio, and Marcello Dalmonte. Intrinsic dimension of path integrals: data mining quantum criticality and emergent simplicity, 2021. arXiv: 2103.02640 [cond-mat.stat-mech].

[MM20] Song Mei and Andrea Montanari. The generalization error of random features regression: precise asymptotics and double descent curve, 2020. arXiv: 1908.05355 [math.ST].

[MRT18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[MTD+21] T. Mendes-Santos, X. Turkeshi, M. Dalmonte, and Alex Rodriguez. Unsupervised learning universal critical behavior via the intrinsic dimension. *Phys. Rev. X*, 11:011040, 1, February 2021. DOI: 10.1103/PhysRevX.11.011040.

[Mur21] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2021. URL: probml.ai.

[Net13] David F. Nettleton. Data mining of social networks represented as graphs. *Computer Science Review*, 7:1–34, 2013. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2012.12.001.

[NKB+20] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020. URL: https://openreview.net/forum?id=B1g5sA4twr.

[Pen03] Mathew Penrose. *Random Geometric Graphs*. Oxford University Press, May 2003.

[PM13] Dominique Perraul-Joncas and Marina Meila. Non-linear dimensionality reduction: riemannian metric estimation and the problem of geometric discovery, 2013. arXiv: 1305.7255 [stat.ML].

[PRE+20] Mauro Pastore, Pietro Rotondo, Vittorio Erba, and Marco Gherardi. Statistical learning theory of structured data. *Phys. Rev. E*, 102:032119, 3, September 2020. DOI: 10.1103/PhysRevE.102.032119.

[RLG20] Pietro Rotondo, Marco Cosentino Lagomarsino, and Marco Gherardi. Counting the learnable functions of geometrically structured data. *Phys. Rev. Research*, 2:023169, 2, May 2020. DOI: 10.1103/PhysRevResearch.2.023169.

[Row00] S. T. Roweis. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000. DOI: 10.1126/science.290.5500.2323.

[RPG20] Pietro Rotondo, Mauro Pastore, and Marco Gherardi. Beyond the storage capacity: data-driven satisfiability transition. *Phys. Rev. Lett.*, 125:120601, 12, September 2020. DOI: 10.1103/PhysRevLett.125.120601.

[RSK+21] Alec Radford, Ilya Sutskever, Jong Wook Kim, Gretchen Krueger, and Sandhini Agarwal. Clip: connecting text and images, 2021.

[SB18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Sej20] Terrence J. Sejnowski. The unreasonable effectiveness of deep learning in artificial intelligence. *Proceedings of the National Academy of Sciences*, 117(48), 2020. ISSN: 0027-8424. DOI: 10.1073/pnas.1907373117.

[SL00] H. Sebastian Seung and Daniel D. Lee. The manifold ways of perception. *Science*, 290(5500):2268–2269, 2000. ISSN: 0036-8075. DOI: 10.1126/science.290.5500.2268.

[SM90] George Sugihara and Robert M. May. Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature*, 344(6268):734–741, April 1990. DOI: 10.1038/344734a0.

[SMG19] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, 2019. ISSN: 0027-8424. DOI: 10.1073/pnas.1820226116.

[SPG+17] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin. A review of clustering techniques and developments. *Neurocomputing*, 267:664–681, 2017. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.06.053.

[Ste56] Hugo Steinhaus. Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci*, 1(804):801, 1956.

[Str94] Steven H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. en. Studies in Nonlinearity. Addison-Wesley Pub, Reading, Mass, 1994. ISBN: 978-0-201-54344-5.

[Tak81] Floris Takens. Detecting strange attractors in turbulence. In *Lecture Notes in Mathematics*, pages 366–381. Springer Berlin Heidelberg, 1981. DOI: 10.1007/bfb0091924.

[Tak85] F. Takens. On the numerical determination of the dimension of an attractor. In *Lecture Notes in Mathematics*, pages 99–106. Springer Berlin Heidelberg, 1985. DOI: 10.1007/bfb0075637.

[Ten00] J. B. Tenenbaum. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000. DOI: 10.1126/science.290.5500.2319.

[Tru68] G.V. Trunk. Statistical estimation of the intrinsic dimensionality of data collections. *Information and Control*, 12(5):508–525, May 1968. DOI: 10.1016/s0019-9958(68)90591-3.

[Van00] Aad W Van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge university press, 2000.

[vdMH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

[Wol18] Michael M. Wolf. Mathematical foundations of supervised learning, July 2018. URL: http://131.159.69.70/foswiki/pub/M5/Allgemeines/MA4801_2018S/ML_notes_main.pdf.