

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
FACULTÉ DES SCIENCES APPLIQUÉES
DÉPARTEMENT D'INGÉNIERIE MATHÉMATIQUE

CENTER FOR SYSTEMS ENGINEERING AND APPLIED MECHANICS

High-dimensional data analysis:

optimal metrics and feature selection

DAMIEN FRANCOIS

Thèse soutenue
en vue de l'obtention du grade
de Docteur en Sciences Appliquées

Membres du Jury:

Prof. **L. Vandendorpe**, Président (UCL-Laboratoire de Télécomunications)
Prof. **P. Dupont** (UCL-Département d'Ingénierie Informatique)
Prof. **J. Hérault** (Institut National Polytechnique de Grenoble)
Dr. **A. Lendasse** (Helsinki University of Technology)
Dr. **F. Rossi** (Inst. Nat. de Recherche en Informatique et Automatique, Paris)
Prof. **M. Verleysen**, Promoteur (UCL-Laboratoire de Microélectronique)
Prof. **V. Wertz**, Promoteur (UCL-Département d'Ingénierie Mathématique)

Janvier 2007

Remerciements

Ce travail a été rendu possible grâce au soutien financier du Fonds pour la formation à la Recherche dans l'Industrie et dans l'Agriculture (FRIA) et du Pôle d'Attraction Inter-universitaire IAP V/22 (2002 - 2006) "Dynamical Systems and Control: Computation, Identification and Modelling".

Par ailleurs, il est évident que j'ai bénéficié, tout au long de mon travail, des conseils, de l'appui, et du soutien de beaucoup ; je suis donc redevable de nombreux remerciements plus personnels.

Tout d'abord, il est évident que cette thèse n'aurait pu voir le jour sans mes promoteurs, les Prof. Michel Verleysen et Vincent Wertz. Ils m'ont appris énormément ; je les remercie de m'avoir guidé tout au long de ces quatre années.

Mes remerciements vont également aux les membres du jury pour leur apport et leurs idées, qui ont permis d'améliorer ce travail. Je dois remercier tout particulièrement le Prof. Luc Vandendorpe, qui a accepté de présider ce jury, le Prof. Jeanny Hérault, qui m'a accueilli une semaine dans son laboratoire, le Prof. Pierre Dupont pour sa lecture approfondie et ses nombreuses suggestions, Fabrice Rossi avec qui j'ai découvert les joies de la chimiométrie, et Amaury Lendasse, qui a grandement facilité mon intégration dans l'équipe du CESAME.

La fameuse équipe du CESAME, j'y ai reçu un accueil chaleureux, de la part tant des académiques, des secrétaires, des informaticiens que des techniciens. Mais de tous les peuples du CESAME, les plus fous sont sans conteste les assistants et assimilés, amis de l'humour, des tartes, du foot et des échecs. Ils me rappellent l'équipe des premiers jours du Machine Learn-

ing Group, qui rendait l'ambiance de l'ESANN incomparable.

J'ai beaucoup apprécié la liberté de travail dont on bénéficie dans cette université, ce qui m'a permis d'aller jouer au volley, sur le temps de midi, avec une chouette équipe ASPU. Cette liberté d'horaire est surtout appréciable les lendemains de whists, à quatre ou cinq, autour d'une kriek.

Je remercie du fond du coeur mes parents pour leur soutien inconditionnel depuis ces années ; ils m'ont permis de m'épanouir dans le sport et la musique puis ont guidé efficacement mes premiers pas d'adultes. Je pense également à mes grands parents et à ma soeur, qui, eux aussi, ont toujours cru en moi.

Enfin, je remercie Sophie d'avoir eu beaucoup de patience lorsque les deadlines se rapprochaient, d'avoir su m'imposer des vacances bénéfiques, mais surtout d'être ma compagne de tous les jours, de m'aider à progresser et de rendre ma vie heureuse, tout simplement.

Abstract

High-dimensional data are everywhere: texts, sounds, log, spectra, images, biomedical data, financial data, are described by hundreds or thousands of attributes. If we want to extract from those data meaningful information to help us in making decisions, we need to analyse these data.

Many data analysis tools (coming from statistics, artificial intelligence, etc.) were designed for low-dimensional data. Unfortunately, **high-dimensional data escape from our natural intuition**; many of the explicit or implicit assumptions made while developing the classical data analysis tools are not transposable to high-dimensional data.

For instance, many tools rely on the Euclidean distance, with which we are familiar in our three-dimensional world, to compare data elements. But the Euclidean distance concentrates in high-dimensional spaces: all distances between data elements seem identical. The Euclidean distance is furthermore incapable of identifying important attributes from irrelevant ones. **The Euclidean distance might thus not be the most relevant one for high-dimensional data**.

This thesis therefore focuses on two different (though related) problems: (i) the choice of a relevant distance function (metric) to compare high-dimensional data and (ii) the selection of the relevant attributes.

In **Part One** of the thesis, the phenomenon of the **concentration of the distances** is considered, and its consequences

on data analysis tools are studied. It is shown that for nearest neighbours search, the concentration phenomenon is not as problematic as it seems at first sight.

However, for kernel-based methods, and more particularly for those based on the Gaussian kernel, the concentration phenomenon leads to a loss of the 'local' property of the kernels. This problem is addressed by considering **Generalised Gaussian kernels**, from which the Gaussian kernel is a particular case. Part One also considers generalisations of the Euclidean norm (namely the Minkowski norms and **fractional norms**) and illustrates their usefulness in cases where the noise affecting the data is different from the usually-considered Gaussian white noise.

Part Two of this thesis focuses on the problem of **feature selection** in the case of a large number of initial features. With high-dimensional data, even methods with a reasonable (e.g. polynomial) complexity become very demanding from the computational point of view. Furthermore, as high-dimensional data often consist of highly-correlated features, the feature selection process can be unstable and very sensitive to the choice of its parameters.

Two methods are proposed to cope with those difficulties. The first one uses **BSplines** to effectively reduce the number of initial features while preserving a lot of the information content of the data. This approach is motivated by the fact that many high-dimensional data are in fact disctretised functions (of the time for sounds, of the wavelength for spectra, of the space for images, etc.) The other method uses **resampling techniques** to choose the parameters of the feature selection method so as to best discriminate between relevant and irrelevant features.

Most of the concepts studied and presented in this thesis are illustrated on **chemometric data**, and more particularly on spectral data. Spectra are obtained from the analysis of the absorbance or reflectance of the light at different wavelengths on physical or chemical materials, like a sample of wine for instance. The objective of spectral analysis it to infer a physical or chemical property of the material, for instance the alcohol content of the wine sample, without performing any costly chemical analysis.

Contents

Introduction 1

High-dimensional data analysis 7
2.1 High-dimensional data 7
2.2 High-dimensional spaces 9
2.3 Impact on data analysis tools 13
2.4 High dimensionality and Support Vector Machines 19
2.5 Other paths to remedies 21

Part I Seeking the right metric

Concentration effect and learning 27
3.1 Distance-based learning tools 28
3.2 The concentration of distances 30
3.3 The stability of nearest-neighbour search 37
3.4 The locality of kernel-based models 44

Alternative measures of similarity 53
4.1 Alternatives to the Gaussian kernel 54
4.2 Alternatives to the Euclidean distances 63

Focus on the concentration of the norm 73
5.1 State of the Art 73
5.2 Further theoretical results 78
5.3 Proofs and Illustrations 84

Part II Selecting the right features

The problem of feature selection 101
6.1 What is feature selection ? 101
6.2 Subset relevance assessment 104
6.3 Optimal subset search 111
6.4 Discussion 115

Practical Feature Selection 121
7.1 Problems specific to feature selection from high-dimensional data 122
7.2 Grouping correlated features 123
7.3 Using resampling methods 128
7.4 Mutual information or Gamma test? 143

Conclusions 147

References 151

Index 161

1

Introduction

> *The most exciting phrase to hear in science -the one that heralds new discoveries- is not "Eureka!" but "That's funny...". – I. Asimov*

Data are nowadays ubiquitous and overwhelming, mainly because they have become so easy to acquire, transfer, store and copy. They may come from a large variety of sources, such as sensors, microphones, cameras, surveys, studies, logs, etc. Consequently, data can take many forms: image, sound, text, spreadsheet (table), sequence, digital signal, connection graph, and others.

Those data are stored and acquired for a purpose. This purpose can be to extract patterns, identify trends, make predictions, identify natural groups, visualize, infer rules, find relationships between attributes, automatically classify, etc. Ultimately, the goal of data analysis is to gain information and knowledge from the data in order to make decisions.

This work primarily focuses on clustering, pattern recognition, and prediction analysis. *Clustering* is the task of finding, in the dataset, groups of similar data elements, called *clusters*. This is useful for instance to identify groups of customers with similar behaviour and target each group specifically in the marketing strategy.

Regression

Prediction analysis, or *regression*, is the task of inferring, from a data element, some response value associated to it. A particular example is prediction of the consumption of electricity at a given hour of the day from both the observed consumption earlier in time (a few hours before, the day before, the year before, etc) and meteorological observations (temperature, etc).

This work particularly focusses on regression.

Pattern recognition, or *classification*, consists in finding the category (or *class*) to which a given data element belongs. An every-day example is the problem of spam that has to be distinguished from regular email and moved to the junk folder while regular emails are directed to the inbox.

To clusterise, classify or predict data, we use mathematical tools, mainly borrowed from the following fields:

- learning theory (support vector machines, neural networks, etc.)
- statistics (statistical tests, re-sampling theory, etc.)
- information theory (entropy, mutual information, etc.),
- optimisation theory (combinatorial optimisation, gradient descent, etc.)
- evolutionary computation (genetic algorithms.)

All those methods use concepts and tools from linear algebra (vector space theory, matrix theory, etc.), calculus (integration, differentiation, etc.), geometry (metric spaces, norm theory, etc.), probability (probability density, random variable, etc.), and others. Consequently, data have to be mathematically described.

Data dimensionality

The attributes that describe the data are usually transformed into a set of *numerical features*. For instance, images can be seen as vectors whose components correspond to the various pixels. Texts can be transformed into a numerical vector where each entry counts the number of occurrences of a given word. Those features are assembled into a feature vector living in a vector space called the *feature space*. The number of features is the *dimension* of the feature space. Most of the time, this feature space is equipped with a *metric*, i.e. a function that measures the similarity between data elements.

An example of tool used for classification is the nearest neighbours classifier (`k-NN`). It consists in setting the class label (category) of a data element according to the majority class among its `k` nearest neighbours, most of the time using the Euclidean metric on the feature space. Geometry is used to describe its functioning, concepts from probability to prove its soundness,

and tools from statistics to choose an adequate value for k.

The use of the Euclidean distance in the `k-NN` is motivated by the fact that the Euclidean metric corresponds to the notion of distance as we think of it in our real, three-dimensional world. As for the `k-NN`, all the tools have been designed having in mind low-dimensional settings, according to schemes that the human mind can represent and interpret.

The "curses" of dimensionality

However, far more than two or three features are necessary to describe data that we encounter nowadays. For instance, texts, sounds, images, consumer records, web logs, etc. are made up of hundreds or thousands of attributes. They are said to be *high-dimensional*.

Obviously, high-dimensional data have to be described in a high-dimensional space. However, high-dimensional spaces have rather counter-intuitive properties, as will be shown in Chapter 2. Many concepts that are commonly admitted in low dimension cannot be extrapolated in higher dimension. Some of these counter-intuitive properties are known as the *empty space phenomenon* or the *concentration of distances phenomenon*. Moreover, many machine learning tools do not scale well with dimensionality. Their performances degrade as the number of features grows. This is often referred to as the *curse of dimensionality*.

Goal of the thesis

The aim of this thesis is to study the link between the properties of high-dimensional spaces and the problems that are witnessed when using machine learning tools with high-dimensional data. The thesis focuses on two different problems, both related to high dimensionality.

Alternative metrics

The first one concerns the phenomenon of the concentration of distances, namely all distances in a high-dimensional space seem identical. This fact leads us to question the relevance of the notion of distance, most often the Euclidean distance, to measure similarity between high-dimensional data elements. However, distances are used in many data analysis tools! Part One of this thesis will study the phenomenon of distance concentration, and analyse to which extent this phenomenon is

harmful to machine learning tools. We will see that for analysing high-dimensional data, the so widely used Euclidean distance and Gaussian kernels are not the most appropriate ones.

Feature selection

The second problem concerns feature selection methods. Because even if the concentration phenomenon is taken care of, distance-based machine learning tools are not able to detect and discard irrelevant features, nor to address the other aspects of the curse of dimensionality. Part Two will present two methods that were developed to address the problems encountered while using feature selection methods on high-dimensional data, namely large computation time and instability of the procedures.

Main Contributions of the thesis

The main contributions of this thesis are, besides a literature review of the properties of high-dimensional dimensional spaces:

- A theoretical analysis of the phenomenon of the concentration of Minkowski norms and fractional norms
- A classification of the machine learning tools that are affected by the concentration of distances and those that are not.
- Theoretical motivation and empirical evidence for the use of fractional norms in high-dimensional spaces when the noise that affects the data is strongly non-white or non-Gaussian.
- Theoretical motivation and empirical evidence for the use of Generalised Gaussian Kernels in kernel-based methods used on high-dimensional data.
- A method for choosing a priori the parameters of the Generalised Gaussian kernels with respect to the distribution of the distances, effectively reducing the number of meta-parameters in Radial-Basis Function Networks and others.

In the field of feature selection, the main contributions are:

- A method for choosing the number of neighbours in mutual information estimation so as to best discriminate between informative attributes and useless ones.
- An automatic stopping criterion for the forward feature selection method that is based on a nonparametric statistical test.

Organisation of the thesis

This thesis is organised in two parts. The first one is dedicated to the study of a particular counter-intuitive phenomenon appearing in high-dimensional spaces: all the Euclidean distances between data seem to be identical. Whether or not this phenomenon is an issue for the data analysis tools that are based on the Euclidean distance is the main object of this first part. Alternative metrics are studied, too. The second part concerns feature selection methods and how they behave when the initial number of features is large. Two practical improvements of a classical method are presented. Before these two parts, Chapter 2 gives a background introduction to high-dimensional data and high-dimensional spaces.

2

High-dimensional data analysis

In order to emphasize the fact that analysing high-dimensional data is a task demanding much care and caution, this chapter proposes, without any attempt to be exhaustive, some counter-intuitive properties of high-dimensional data spaces. It then reviews some of the problems that are often encountered while analysing high-dimensional data by explaining the origins of the "Curse of dimensionality", "Empty space phenomenon", "Concentration of distances", and "Peaking phenomenon". The chapter ends with some information on how to cope with the dimensionality.

The times, they are changing, Bob Dylan was right. In the past, high dimensionality used to mean four or five attributes [Rao, 1948]. Nowadays, we have to deal with, comparatively, "super-high-dimensional" data, described by thousands of attributes. How this enormous number of attributes can affect machine learning tools is the focus of this chapter. Three real-life examples of (very) high-dimensional data are given in Section 2.1, and some counter-intuitive facts about high-dimensional spaces are presented (Section 2.2). The consequences on machine learning tools are then illustrated in Section 2.3.

2.1 High-dimensional data

Data are nowadays complex and high-dimensional. In scene analysis, face recognition, document categorization, speech recognition, optical character recognition, spectral and hyperspectral analysis, financial time series analysis, customer log analysis, gene expression analysis, and many others, one has to deal with data that are described by many attributes, and consequently described with high-dimensional (from ten to thousand attributes) feature vectors [Donoho, 2000; Guyon & Elisseeff, 2003; Verleysen, 2003]. Each data element is thus viewed as a point in a vector space whose dimension is the number of numerical features necessary to describe each data element. Fol-

lowing are three examples illustrative of the context and needs of high-dimensional data analysis.

The DELVE-CENSUS dataset, available from the University of Toronto[1], comprises data collected during the 1990 US Census. Each of the 22,784 data elements concerns a small survey region and is described by no less than 139 features measuring demographic information like the total population in the region, the proportion of males, the percentage of people aged between 25 and 64, etc. The aim is to predict the median price of the houses in each survey region.

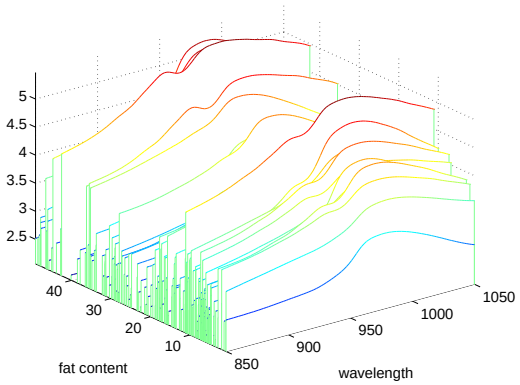


Figure 2.1: The Tecator meat spectra dataset. Each curve is a spectrum of absorbance. The spectra are ordered from front to back in increasing order of fat content.

The TECATOR dataset (figure 2.1) is a set of meat spectra. Samples of meat are lit with near infrared light, and the absorbance level of the light is recorded as a function of the wavelength, in the 850–1050 nm range. The spectra are digitalized into 100 scalar values, each of which representing the proportion of photons of a given frequency that were absorbed by the meat sample. The goal is to estimate, from the meat spectra, the fat content of the meat. In addition to the fact that the problem is mathematically challenging, it is both cheaper and faster to

[1] `http://www.cs.toronto.edu/~delve/data/census-house/desc.html`

perform spectral analysis than performing a chemical analysis. Knowing the fat content of the meat is important in food regulation and food quality control. The database contains 240 samples. The dataset is described in [Borggaard & Thodberg, 1992] and can be downloaded from Statlib [2].

The XM2VTS datasets contains, in addition to voice and video data, face images from 295 subjects with 8 images per person. A subset of the database is shown in Figure 2.2. Each image has 720x576 pixels for each RGB channel (that is 1244160 numerical values to describe one image.) The goal is to perform face authentication to grant or deny access to facilities, documents, etc. The dataset is described in [Messer *et al.* , 1999] and the data can be ordered at the Department of Electrical Engineering of University of Surrey [3].

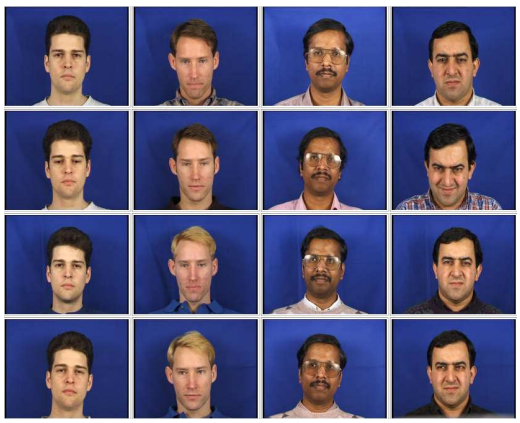


Figure 2.2: Some face images from the XM2VTS database.

2.2 High-dimensional spaces

Very few are those who claimed the ability to think in four dimension [Poincaré, 1963; Rucker & Povilaitis, 1984]. The hu-

[2] `http://lib.stat.cmu.edu/datasets/tecator`
[3] `http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb/`

man mind is indeed set up to think in two or three dimensions. The human mind is able to mentally represent objects in a three-dimensional space, and to figure out two-dimensional projections of those objects. It is however not trained to think about four-dimensional objects and their 2D or 3D representations. Even picturing the simplest four-dimensional volume, the *tesseract* (4D-hypercube) is not easy and can be done in many different ways, as depicted in Figure 2.3.

The geometry of high-dimensional spaces, in all its rigour, can be deceptive too. The following section introduces some counter-intuitive properties of generalizations of objects we are familiar with in two and three dimension.

2.2.1 About the geometry of high-dimensional spaces

The volume of the hypercube

We will first focus on the *hypercube* in dimension d, with side length l. If $l = 1$, then the (hyper-) volume of the hypercube is independent from the dimension: it always values one. However, whenever $l > 1$, the volume diverges to infinity as dimension increases while when $l < 1$, it converges to zero! The volume of a cube whose edge length is 0.9 in dimension 100, is more than twenty thousands times smaller than the volume of a cube with length 1.1...

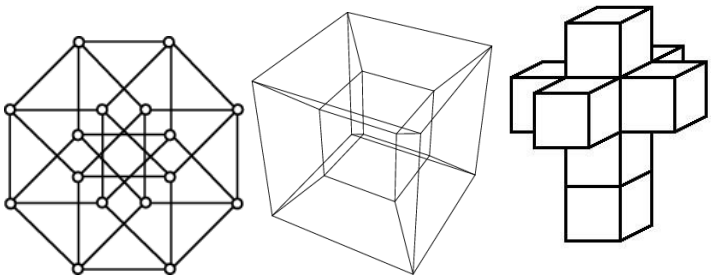


Figure 2.3: Three different ways of representing a two-dimensional view of the four-dimensional cube (tesseract)

The shape of the hypercube

In a d-dimensional hypercube of side length 1, the distance from the centre to any side is one half, while the distance to any

corner is $\sqrt{d}/2$. Therefore, when the dimension reaches 100, for instance, the distance from the centre to the corners is ten times larger than the distance to the sides. This fact has lead Hecht-Nielsen to describe a hypercube as a spiky shape (see Figure 2.4): "The surfaces of cubes are so horribly jagged that they might even be thought of as being almost fractal."[Hecht-Nielsen, 1991, p. 42].

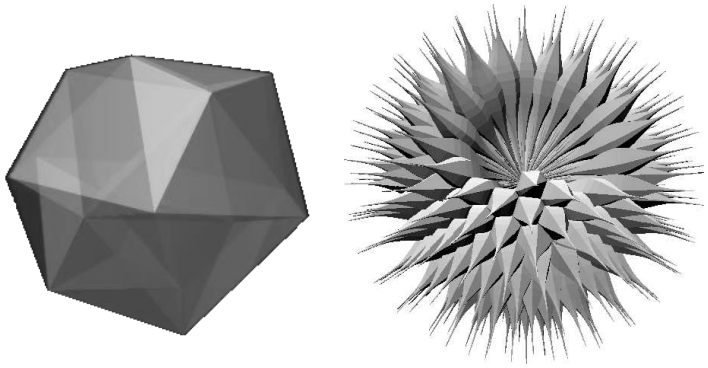


Figure 2.4: Three-dimensional representations of a hypercube in dimension 4 (left) and 8 (right)

The volume of the hypersphere

The volume of the *hypersphere* is also worth investigating. The hypersphere in dimension 1 is a straight line, in dimension 2 it is a circle and in 3 dimension it is a ball. Its volume for unit radius in dimensions one to three is thus respectively 2, $\pi \simeq 3.14159$, and $4/3\,\pi \simeq 4.1887$. In higher dimension, the sphere with centre c and radius r is defined as the locus of the points whose distances to r is less than or equal to r. The general formula for dimension d and radius r is [Kendall, 2004]

$$V_r(d) = \frac{\pi^{d/2}}{\Gamma(d/2+1)} \cdot r^d. \tag{2.1}$$

The shape of (2.1) is depicted in Figure 2.5. Although the volume increases with dimension up to dimension 5 it then decreases to finally converge to zero as dimension grows! Note that, as a consequence, whereas in dimensions two and three, a hypersphere nearly fills the volume of its bounding hypercube, the same is not true anymore in high dimension, where

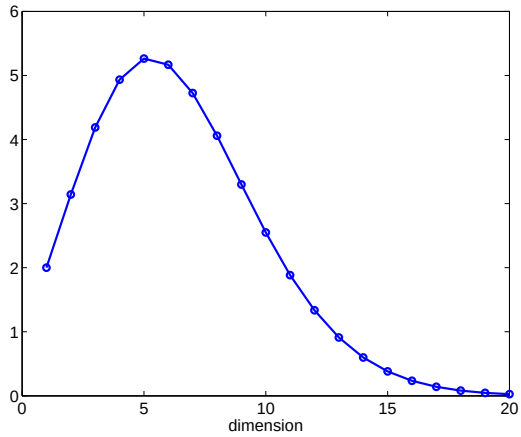


Figure 2.5: The volume of a hypersphere of radius one as a function of the dimension

the volume of the sphere is negligible compared with that of the hypercube!

2.2.2 Data lost in space

High-dimensional spaces have counter-intuitive geometric properties. Representing data points in those spaces can thus sometimes be misleading. It is easy to represent 1000 points in two dimension: the scatter plot will probably highlight the structure of the data, if there exists one: separate clouds, relationship between variables, etc. To the contrary, 100 points in a 50-dimensional space will hardly represent anything: they will just appear scattered over the space in a random fashion; even if there is an internal structure, it will most probably not appear. It is just like trying to infer something from four points in three dimension.

Normal distribution

Let us consider normally distributed data that are drawn from a multivariate normal distribution. In two and three dimension, many points will lie near the centre of the distribution, and few others will get far from the centre. The probability to see a point at distance r from the centre is very high near the centre of the distribution. As the dimension increases, however,

this probability actually becomes very close to zero near the centre. There will indeed be a peak at some distance from the centre that increases with dimension (Figure 2.6). The probability density function corresponding to finding a point at distance r from the centre of the distribution is given by:

$$f = \frac{r^{d-1}}{2^{d/2-1}} \cdot \frac{e^{-r^2/2}}{\Gamma(d/2)} \tag{2.2}$$

which is maximum for $r = \sqrt{d-1}$. As d grows, the probability to find a point whose distance to the centre is close to zero is very low.

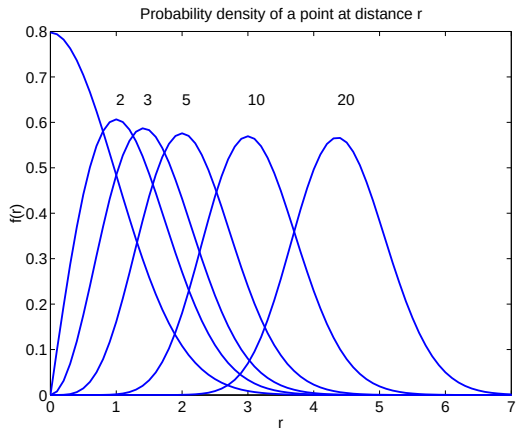


Figure 2.6: Probability density function to find a point at distance r from the centre of a multivariate d-dimensional normal distribution

2.3 Impact on data analysis tools

As illustrated in the preceding section, it is not uncommon for high-dimensional data to have more features than observations ($d > N$). This sharply contrasts with classical data analysis that consider the dimension to be fixed and low, and the number of observations to be large ($N >> d$). Many theoretical results

even consider the case where N goes to infinity.

Obviously, with high-dimensional data, these assumptions are not plausible. For *functional data*, moreover, the sample size is fixed while the dimensionality goes to infinity! [Ramsay & Silverman, 1997]

A glimpse on the consequences

This has consequences on the tools and methods designed for (low-) dimensional data analysis. When the dimensionality of the data is high, one never has enough data samples to cover the input space entirely. Data are thus sparsely distributed in the data space, which actually looks nearly empty, and all distances between two data elements seem roughly equal. Another misleading intuition is the fact that prediction should be easier with more information than with little, and hence prediction should be easier with high-dimensional data than with low-dimensional data. Unfortunately, what is really observed is different: the performances of the data analysis tools decrease, as dimensionality grows large.

2.3.1 The curse of dimensionality

Richard Bellman introduced the terms "*curse of dimensionality*" in [Bellman, 1961], to refer to the fact that, in order to optimise a function of d binary variables by exhaustive search over the input domain, one needs to perform 2^d evaluations of the function. This of course becomes quickly computationally intractable.

More generally, in a d-dimensional space, to be able to cover (for instance fill with points according to a regular Cartesian grid) the portion of the space within the unit hypercube, we need $O((1/\epsilon)^d)$ points where ϵ is the spacing of the grid. If the dimension is six and ϵ values $1/10$, no less than 1,000,000 points are required.

From a machine learning viewpoint

The consequence of that exponential "need for points" is that in practical cases, when the dimension of the data is above 10 or 20, one never has enough points available to entirely cover the input space. This can be a real problem for supervised learning as well as for unsupervised learning. The smaller the dataset, the less accurate the model can be. Ideally, the dataset should

even span the whole input space of interest, in order to ensure that any predicted value (i.e. output of the model) is the result of an interpolation process and that no hazardous extrapolation is done (as in Figure 2.7). When the number of learning points is too small with respect to the dimension, the model runs chances of being too specific to the data and lacking generalization.

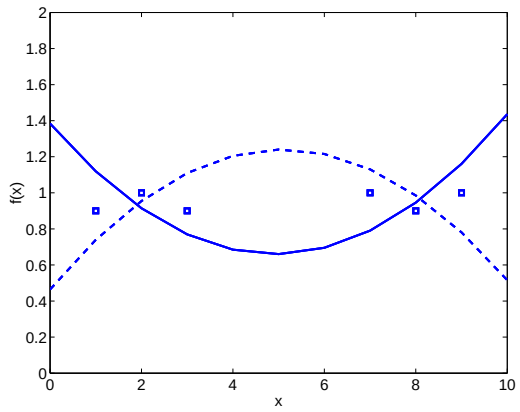


Figure 2.7: Two models (plain line and dashed line) approximating the same data with not enough learning points. Both perform equally well on the learning (known) data but it is impossible to decide which one is best at $x = 5$ because the data do not cover that part of the space.

The example of Silverman

Silverman [Silverman, 1986] addressed the problem of finding the necessary number of training points (samples) to approximate a Gaussian distribution with fixed Gaussian kernels. He found that the required number of samples growths exponentially with dimension. Although the example is purely theoretical (in practice one Gaussian kernel is sufficient to approximate a Gaussian distribution) it shows that in order to learn correctly the distribution of data, one needs a number of examples that grows exponentially with the dimension.

The terms 'Curse of dimensionality' have since been used to designate all kind of problems people face when dealing with high-dimensional data.

2.3.2 The empty space phenomenon

The *empty space shenomenon* denotes the same situation as the curse of dimensionality but from another point of view. It was first introduced by Scott and Thompson [Scott & Thompson, 1983].

Suppose that we have N points described in a d-dimensional bounded space. If that space is spanned (covered) by a regular ϵ-grid, $O(1/\epsilon^d)$ cells are built and the N points are spread among the cells according to their position. When dimension grows, more and more cells will be empty, simply because there will be much more cells than available points.

From a machine learning viewpoint

High-dimensional spaces thus seem "*empty*"; very few cells contain several points, some contain only one point, and actually most cells do not contain any point at all. The inherent sparsity of such spaces renders the estimation a probability density function (pdf) very difficult. When histograms are built for example, many of their bins are empty, or have small values, and thus lead to poor and non-smooth approximation of the true pdf. Figure 2.8 shows an histogram approximation of a normal probability density function with a small number of samples. Some bins are empty, and the estimation is sharp (non-smooth) and far from the true distribution.

As dense regions are very rare in empty spaces, it is very difficult to perform a sound cluster analysis [Steinbach *et al.* , 2004]; the clusters that are found by the algorithms often have no interpretation.

2.3.3 Concentration of distances

Beyer showed that distances in high-dimensional spaces concentrate, that is, the distances to nearest neighbours are large, closer to the distances to farthest neighbours than to zero. Beyer consequently questions the relevance of nearest neighbours search in high dimension [Beyer *et al.* , 1999].

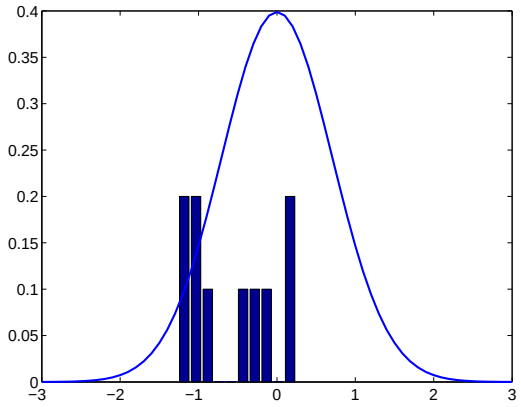


Figure 2.8: Histogram approximation of the normal pdf with not enough points.

In a high-dimensional space, all points are most of the time distant from one another. Furthermore, distances seem to be all the same, as can be seen on Figure 2.9. On the left, the histogram of distances between points randomly drawn from a uniform distribution in dimension 2 is spread over the whole domain of the distances (i.e. from 0 to $\sqrt{2}$). By contrast, in dimension 1000, only a small portion of the possible values (from 0 to 10) is encountered in practice. No single distance is even close to zero.

The metric defined on the data space is designed to act as a measure of similarity between data elements. If the distance between two data elements is small, these can be considered to be similar. Many machine learning tools are based on this concept that is sometimes called *smoothness prior*. If all distance are roughly equal, what consequence can it have on machine learning tools?

Beyer concludes from the concentration of the norms that all nearest-neighbour searches in high-dimensional spaces are unstable, while Friedman shows that it renders the choice of an appropriate value of the number of neighbours in a `k-NN` much more difficult than in lower dimension [Friedman, 1997]. It is also very difficult to spot outliers because nearly all points can

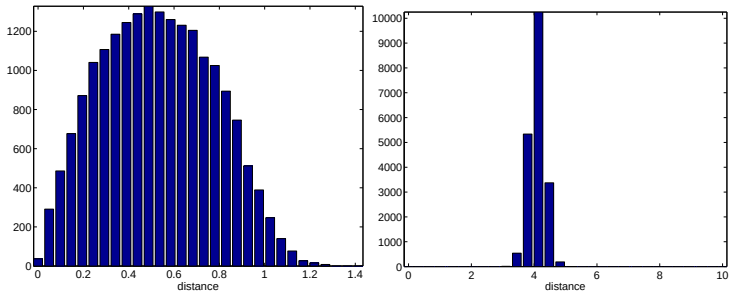


Figure 2.9: Histogram of distances in a 2 -(left) and 100- (right) dimensional data spaces filled with uniformly distributed data ($N = 250$).

be considered outliers by many of the usual definitions in such situations [Aggarwal & Yu, 2001].

Similar observations can be made about the scalar product; the scalar product between any two feature vectors is near to zero with high probability, if the data are normalised. This was found to be problematic for the Support Vector Machine (SVM) with linear kernels [Evangelista *et al.* , 2006].

2.3.4 The peaking phenomenon

Hughes was the first to describe the "Peaking phenomenon". The peaking phenomenon relates to the number of parameters of a model, the dimension d of the dataspace, and the number of samples N used to build the model. Hughes shows that the classification accuracy of pattern classifiers depends, for fixed sample size, on the dimensionality of the data (or, measurement complexity), i.e. how many features are extracted from the data. With a finite sample, the accuracy of the classifier, as a function of the dimensionality of the data, first increases as more attributes are measured, but then reaches a peak and then decreases to the performances of a random classifier as yet more attributes are measured [Hughes, 1968]. The fact that the accuracy increases as more features are added in a first time is normal since more information is fed to the classifier. However, at some point, the classifier is not able anymore to deal with that much information, and performs badly. The number

of examples is not sufficient for the classifier to set appropriate values for all its attributes.

To set a parameter (e.g. a mean) many observations are needed. Indeed computing the mean of three elements cannot be seen as a statistically relevant estimation... The same is true for the parameters of the regression or classification models. The ratio between the number of observations and the number of parameters should be as large as possible.

From a machine learning viewpoint

When the number of parameters becomes large compared with the sample size, those parameters cannot be correctly estimated anymore, and the variance of the models increases; their performances decrease. Furthermore, even for a simple linear classifier, we need at least as many observations as dimensions. This is of course an even more dramatic problem for nonlinear models, especially for those whose number of parameters grows exponentially with the dimension.

Fukunga [Fukunaga & Hayes, 1989] showed that the required number of points to build a linear classifier grows linearly with the dimension while it grows quadratically for a quadratic classifier. Raudys showed that accuracy is exponential for Parzen classifier, linear for the linear one, quadratic for the quadratic one and linear for the k-NN [Raudys & Jain, 1991]

2.4 High dimensionality and Support Vector Machines

Support Vector Machines (SVM) have been attracting the attention in the last few years, because they rely on solid theoretical bases, and fast and efficient learning algorithms have been designed, implemented, and popularized.

Curse or blessing of dimensionality?

It is often claimed that, amongst their many virtues, SVMs are free from the curse of dimensionality. It is important to note that we are interested here in the actual dimension of the data and not in the dimension of the "kernel"[4] space of the SVM. The high dimensionality of the induced kernel space is even considered as a *blessing* as it allows building more powerful

[4] The classical name of that space is 'feature space'; we will however not use it here to avoid confusion.

classifiers.

From Vapnik's theory [Vapnik, 2000], it follows that, in terms of risks of overfitting due to a large number of parameters, the important concept is not the dimensionality of the data, but the capacity of the model. The notion of capacity of a model is a measure of the complexity, or expression power of the model, more general than the number of parameters. Hence, the *curse of dimensionality* is often traded with the *curse of capacity*, that involves the *Vapnik-Chervonenkis dimension* (or *VC-dimension*) of the model. The theory indeed provides bounds on the generalization error that depend not on the dimension of the data, but on the VC-dimension of the model. A model with too large a capacity will poorly generalise. Another argument often presented is that the number of parameters in a SVM depends on the number of samples and not on the dimension of the space, by contrast with a Multi-Layer Perceptron (MLP) for instance.

Dimensionality vs. capacity

The notion of capacity is broader than the number of parameters to measure the complexity of a model. A model with few parameters can have a large VC-dimension: the one-dimensional sine model $\mathrm{Y} = a \cdot \sin(\omega \mathrm{X})$ for instance can approximate nearly any function $\mathrm{Y} = f(\mathrm{X})$ with only two parameters. Conversely, a model with many parameters can have a small VC-dimension if constraints are imposed on the values of those parameters, e.g. all values must be equal. But a model with many parameters and few constraints will have a large VC-dimension. Interestingly, for linear models, the VC-dimension is exactly the number of parameters, that is one plus the dimensionality of the data.

Support Vector Machines with kernels, soft margin, and structural risk minimization are thus a good option to reduce overfitting, compared with controlling the number of parameters with a meta-parameter. Not only do SVM control overfitting with models with many parameters, but also with models that have few parameters but nevertheless high expression power like the sine model presented above.

Nevertheless, SVM are not the only methods that control overfitting in the presence of many parameters. See for instance [Moody, 1992] for a review of other regularization methods, where the expression power of models is measured, not by

the number of parameters, but rather by a notion which they call the *effective number of parameters* that resemble the VC-dimension. This effective number of parameters is used to estimated (and minimize) the *final prediction error*. Furthermore, SVM are not the only family of methods whose number of parameters does not depend on the dimensionality of the data. Examples are RBFN trained with the OLS method [Orr, 1995] and, by definition, nearly all Lazy Learning methods [Aha, 1997]. Was the curse of dimensionality solved in the sixties when the k-NN was invented ? No: it appears that choosing the optimal value of the number of neighbours for high dimensional data is both critical and extremely difficult [Friedman, 1997]. The *curse of dimensionality* has many faces.

SVM: good but not perfect

The reason is that the regularisation and complexity management approaches only address one specific aspect of the problems with high-dimensional data. Indeed, the distance concentration phenomenon leads to kernel matrices that are close to a diagonal matrix (see [Evangelista *et al.* , 2006] and Section 3.4), which renders generalisation difficult. Furthermore, SVM with Gaussian kernels or polynomial kernels are not able to cope with irrelevant features because they consider each feature to have the same "weight" in the computation of the kernel matrix. Finally, as any other method, a SVM is not able to generalise in portions of the space where no examples were given; to cover the whole dataspace, a number of examples that is exponential in the dimension are needed, no matter which model is used.

2.5 Other paths to remedies

The properties of high-dimensional spaces that we have reviewed in a previous section all rely on the implicit hypothesis that all numerical attributes are statistically independent. If two features are identical, the data should be considered, obviously, to be $(d-1)$-dimensional. In real data, identical features are of course rarely encountered. Nevertheless, truly independent high-dimensional data are also rare. Often, there will be a "hidden structure" in the data in the sense that some attributes depend, in a subtle way, on some others. Consequently, data are not spread over the whole data space; they rather lie on a *submanifold* in the dataspace. That "self-similarity" property of

many high-dimensional data is a clue that dimensionality reduction can be performed without loosing too much information.

Reducing dimension

In front of those problems particular to high-dimensional data, the obvious solution of reducing dimensionality comes to mind. Many different methods exist, which can be distinguished as either *feature selection* methods or *projection* techniques. While the former methods select among the original variables those that are better suited for the problem at hand, the latter build new variables that are combinations of the original ones. Feature selection can be seen as a special case of projection that preserve the interpretation of the variables.

Parsimonious models and regularisation

Beside the use of any dimensionality reduction technique, it is good practice to use models that are *parsimonious*. A parsimonious model is a model whose number of parameters increases at most linearly with the dimension. Clearly, a Radial-Basis Function Network (RBFN) is parsimonious by contrast with splines for instance that also have universal approximation abilities but at the same time require a number of parameter that increases in a combinatorial manner with the dimension. Moreover, as discussed above, regularised models are a good alternative to further reduce the risks of overfitting.

Alternative metrics

Beside the "complexity management" approaches described here above, another path can be explored. We have seen that the Euclidean distance concentrates in high dimension. It may not be the case for all metrics. There is thus, for this particular aspect, another alternative: to find other metrics that are more relevant than the Euclidean distance in high-dimensional spaces. Furthermore, the real impacts of concentration on machine learning tools still have to be studied.

Chapter summary

Building a model in a high-dimensional space is far from being a trivial task because the number of observations is seldom large enough both compared with the volume of the input space and with the number of parameters of the model, dramatically increasing the risks of overfitting. High-dimensional spaces are often sparsely populated with data. Applying a

model on any new data will thus most probably be extrapolation; the model is estimated in an area of the input space where it has no examples to rely on. Furthermore, all distances look very similar one to another. The relevance of the distance is consequently questioned, as it does not seem to have a discriminating power to distinguish between similar and dissimilar data elements. One option to address those problems is to reduce the dimensionality. Another option is to study and use alternative metrics.

Organization of the remaining of the thesis

The remaining of the thesis is consequently divided into two parts. The first part studies the phenomenon of concentration of the distances and its impact on machine learning tools. It comprises three chapters. Chapter 3 identifies the machines learning tools that suffer from the concentration of the distances. Chapter 4 proposes solutions by means of alternative distance measures to the commonly used Euclidean distance and Gaussian kernel. Chapter 5 contains all the proofs and technical discussions.

The second part is about the application of feature selection methods to reduce the dimensionality of real high-dimensional data and the challenges that it rises. Chapter 6 presents a general introduction to feature selection. In Chapter 7, tools and methods for efficient feature selection in the case of a large number of initial features are given.

Part I

Seeking the right metric

There certainly is no consensus in the literature about how the concentration of distances can impact on data analysis tools. Some ignore it, others propose alternative metrics to reduce concentration. The following three chapters will consolidate and unify several independent results that can be found in the literature, identify the methods that suffer from the concentration, and propose alternative solutions to the rarely-questioned Euclidean norm and Gaussian kernel.

”

That is the essence of science: ask an impertinent question, and you are on the way to a pertinent answer.” – J. Bronowski

3

Concentration effect and learning

The impacts of distance concentration on data analysis tools are investigated in this chapter. First, the difference between rank-based methods and kernel-based methods is explicated. Then, arguments are given, which show that rank-based methods do not suffer from the concentration while kernel-based methods face problems when distances concentrate, especially because the distances to nearest neighbours become significantly larger than zero. The shape of the kernel in high-dimensional spaces must be properly chosen; the commonly-used Gaussian kernel is inappropriate.

Many machine learning tools for supervised and unsupervised learning rely on the definition of a metric (usually the Euclidean distance) over the data space. These tools use the distance defined by the metric as a measure of (dis)similarity between data elements. This principle applies for instance to the `k-NN` [Cover & Hart, 1967], the `K-MEANS` [Hartigan & Wong, 1979], hierarchical clustering [WardJr, 1963], Radial-Basis Function Networks (`RBFN`) [Powell, 1987], Support Vector Machines (`SVM`) [Boser *et al.* , 1992], Self-Organizing Maps (`SOM`) [Kohonen, 1995], Neural Gas [Martinetz & Schulten, 1991], Learning Vector Quantization (`LVQ`) [Kohonen, 1995], Takagi-Sugeno fuzzy models [Takagi & Sugeno, 1985], as well as many Lazy Learning methods [Aha, 1997].

We have seen however in Chapter 2 that the Euclidean distance concentrates as dimensionality increases. This chapter investigates the impact of this concentration on distance-based data analysis tools. While Section 3.1 presents some distance-based learning tools, Section 3.2 illustrates the concentration phenomenon. Section 3.3 and 3.4 study the impacts of concentration respectively on rank-based tools and on kernel-based tools.

3.1 Distance-based learning tools

There is a whole class of Machine Learning tools that use a measure of distance between data elements. The rationale for using distances is the rather sensible hypothesis that there exists a metric space in which the data can be described, such that data elements that are perceived to be similar will be close in the space, while dissimilar elements will be distant from one another. The hypothesis that the Euclidean space is the right metric space to be used is, however, less evidenced; it is nevertheless very rarely questioned for numerical data.

Many machine learning tools use distances

Along with the notion of distance between data objects, we can define a *neighbourhood* around each data element so that all data elements lying in that neighbourhood can be considered similar one to another. The notion of neighbourhood is exploited as well by supervised methods for classification or function approximation/regression as by unsupervised methods for clustering.

Supervised methods can be split up into two main categories: *ranking* methods against *kernel* methods. Moreover, we will distinguish between methods that operate on *feature vectors* and methods that operate on *dissimilarity matrices*. Sometimes indeed, data are only known through their pairwise dissimilarities; the attributes of the data are not directly measured.

3.1.1 Rank vs Kernel

The rank-based machine learning tools only compare distances to produce response values. Once the model is built, the actual values of the distances are not used for computing the estimate of the response value or class label for a new data element; they are only used to rank elements according to their similarity/distance to some other data. Examples of rank-based tools are the `k-NN` and the `LVQ` classifier.

Kernels measure similarity

By contrast, kernel-based methods use the actual value of the distance in the estimation of the response value. Kernels are functions of two arguments $\mathbf{x}$ and $\mathbf{z}$, whose value decreases with the distance between $\mathbf{x}$ and $\mathbf{z}$. Kernels are parameterized by a *width*, denoted here θ: $K(\mathbf{x}, \mathbf{z}; \theta)$. If the distance between $\mathbf{x}$ and $\mathbf{z}$ is small compared with the width of the kernel, the

kernel value will be close to one (or another maximum value). Large distances by contrast are mapped to values close to zero.

Kernels use the distance in the prediction

In kernel-based methods, the estimation of the response value for a new data element $\mathbf{x}'$ is estimated as a sum of products between a kernel functions and some parameter of the model (which one "weighs" the other depends on the interpretation of the model). Each kernel is computed with $\mathbf{x}'$ as first argument, and a parameter of the method as second argument. Those parameters live in the data space; in Lazy Learning tools they are nearest neighbours to $\mathbf{x}'$, in `RBFN` they are centroids, in `SVM` they are support vectors, etc. The kernel function actually defines a neighbourhood of similarity around $\mathbf{x}'$.

The notion of kernel considered here is very general. However, Support Vector machines are well-defined for *Mercer kernels* only [Schölkopf & Smola, 2002] while the universal approximation properties of the Radial-Basis Function Networks was proved for *radial-basis* kernels [Park & Sandberg, 1991]. The widely used Gaussian kernel is both a Mercer kernel and a radial-basis function.

3.1.2 Attribute measurement vs similarity measurement

Some tools are designed to work with feature vectors while other consider distance, or dissimilarity matrices. In the former case, the attributes of the data are measured, and 'stored' in a *feature vector*. By contrast, in the latter, only the distance measurements are known.

The `K-MEANS`, `SOM`, `LVQ`, etc. compute the distances between data elements 'by themselves', from their feature vectors. They need the values of the attributes because at some point in their respective algorithms, they compute 'averages' of the data, called *centroids*.

Feature vector vs. distance matrix

This contrasts with, for instance the `k-NN`, the `K-MEDOIDS` [Kaufmann & Rousseeuw, 1987], the `SOM` adapted to dissimilarity data [Golli *et al.* , 2004], and the hierarchical clustering methods that can be used when only pairwise distances or dissimilarities are known. In some applications, data cannot be described in a vector space in a meaningful manner. Rather,

they are described in a metric space and the metric is assessed by humans who score the similarity between data elements, or measured by very specific algorithms (time wrapping, edit distance, etc).

From features to similarities

Obviously, the methods that were designed for dissimilarity data can be used with feature vectors, provided an appropriate norm (or distance) is defined, by simply first computing the matrix of all pairwise distances.

3.2 The concentration of distances

This section presents the phenomenon of the concentration of the Euclidean norm, and extends it to the Minkowski norms family.

The terms 'concentration of distances' refer to the fact that, in a high-dimensional space, all pairwise distances between data elements seem to be very similar. The distances to the nearest and furthest neighbours even look nearly identical. Most of the observed distances are actually 'concentrated' in a small interval of the set of all possible values of the distances.

The concepts of distances and norms are strongly tied. The Euclidean distance between two vectors is the Euclidean norm of the difference between those vectors. Many norms can be defined; the most encountered ones being the Minkowski norms.

Minkowski norms

Minkowski norms, called l_p norms are defined for a vector $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ as follows:

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{d} |x_i|^p \right)^{\frac{1}{p}} \tag{3.1}$$

with $p \in I\!N \backslash \{0\}$. The limit case is defined as

$$\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq d} |x_i|. \tag{3.2}$$

Norms are by definition positive: $\|\mathbf{x}\| > 0$ when $\mathbf{x} \neq 0$ and $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = 0$, homogeneous: $\|\alpha \cdot \mathbf{x}\| = |\alpha| \|\mathbf{x}\|$ for

any $\alpha \in \mathbb{R}$ and obey the triangle inequality: $\|\mathbf{x}+\mathbf{z}\| \leq \|\mathbf{x}\|+\|\mathbf{z}\|$.

Fractional norms

Fractional norms are defined similarly to Minkowski norms except that p can take any positive real value. The norms with $p < 1$ are very different from the classical Minkowski norms. The balls defined using such norms are not convex sets and, the triangle inequality is actually reversed [Beckenbach & Bellman, 1971]; the name 'norm' does not formally apply anymore. In the following, we will reserve the term 'fractional' for norms with $p < 1$.

The Euclidean distance is the most used one. Most of the time, indeed, the use of the Euclidean norm is not questioned because it intuitively corresponds to the distance as we define it in our three-dimensional world. Unfortunately, the Euclidean norm has a counter-intuitive behaviour in higher-dimensional spaces: it concentrates, and so do other norms, as will be shown in the remaining of this section.

3.2.1 Illustrations

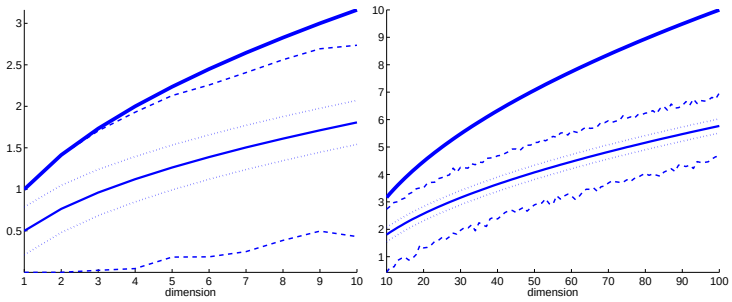


Figure 3.1: From bottom to top: minimum observed value, average minus standard deviation, average value, average plus standard deviation, maximum observed value, maximum possible value of the Euclidean norm of a random vector uniformly distributed over $[0, 1]^d$ (the sample size is 10000).

Let us consider the distribution of the norms $\|\mathrm{X}\|_2$ where $\mathrm{X} = (X_1, \cdots, X_d) \in \mathbb{R}^d$ is a random vector uniformly distributed over $[0,1]^d$, for growing dimension d.

Illustration on synthetic data

Figure 3.1 shows the maximum and minimum observed values (over 10000 draws) of the norm along with the average value and the estimated standard deviation. We can see that the growth of the minimum value is roughly proportional to the square root of the dimension. The same observation can be made for the maximum observed values. However, the difference between the extreme values seems constant. As the dimension grows large, both extreme values tend to be very similar; their relative difference becomes negligible (10% for $d = 10^3$ and 3% for $d = 10^4$).

For a given dimension, the domain of the norm of such a vector is the interval $[0, \sqrt{d}]$. We see on Figure 3.1 that in small dimension, the observed extrema are close to their theoretical values. When the dimension grows, however, the maximum observed values is much lower than $\sqrt{d}$ and the minimum observed value is much larger than zero. The observed minimum is even closer to the observed maximum than it is to zero. This illustrates the concentration of the Euclidean norm: most values of the norm seem to belong to a small interval of the domain of the norm.

Illustration with real data

Nevertheless this experiment is illustrative: the more points are drawn, the wider the observed bounds will be. Hence, it is not obvious that the concentration is an intrinsic property of the norm; it might well be an artefact of finite sample.

Figure 3.2 shows the result of the same experiment but with a number of samples growing exponentially with the dimension. The observed minimum is close to zero, while the observed maximum is close to the theoretical one. However, it appears that the variance remains constant. This experiment shows that most norms are condensed in a small sub interval of the domain of the norm, but that the maximum and minimum depend on the sample size. We will therefore focus on distributions rather than on samples. By comparison, with real data, whose intrinsic dimensionality is often smaller than the dimension of the data space, both the range and the variance of the norms increase with dimension (see Figure 3.3), indicat-

ing that we should focus on the intrinsic dimension of the data. This will be detailed in Section 5.2.3.

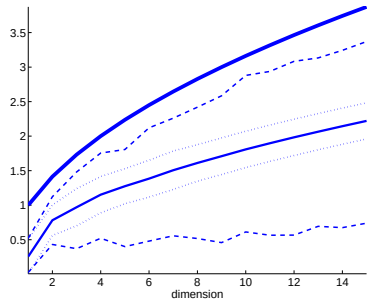


Figure 3.2: From bottom to top: minimum observed value, average minus standard deviation, average value, average plus standard deviation, maximum observed value, maximum possible value of the Euclidean norm of a random vector uniformly distributed over $[0,1]^d$ (the sample size grows exponentially with the dimension).

Theoretical illustration

Let us now consider X to be a random vector with independently normally distributed components. The norm of such a vector is itself a random variable whose distribution is known; it is a χ distribution with d degrees of freedom. The moments of this distribution have analytical expressions [Evans *et al.* , 2000] that show that the expectation increases with dimension while the variance remains constant. Most norms are thus concentrated in a small portion of the domain around the expected value, and this is indeed an intrinsic property of the norms.

3.2.2 Theoretical results

The concentration of norms can be observed; it can also be measured. We consider two ways of estimating how concentrated the distances, or the norms, are.

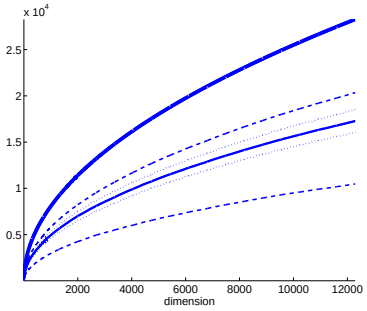


Figure 3.3: From bottom to top: minimum observed value, average minus standard deviation, average value, average plus standard deviation, maximum observed value, maximum possible value of the Euclidean norm of the images from the XM2VTS dataset. Vectors of increasing dimension are created by randomly subsampling the images.

The *relative contrast* [Beyer *et al.* , 1999; Hinneburg *et al.* , 2000] concerns a specific dataset $\mathcal{X}$ and is defined according to the maximum and minimum values of the distances:

Relative contrast

$$\mathcal{K}_{\mathcal{X},p} = \frac{\max_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x}\|_p - \min_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x}\|_p}{\min_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x}\|_p}.$$

The numerator of the latter equation is called the *absolute contrast*.

However we have seen from the above experiments that the maximum and minimum depend on the number of points. Therefore, we propose to use the *relative variance*; i.e. the ratio between the standard deviation of the norm and its expectation:

Relative variance

$$\mathcal{V}_{\mathrm{X},p} = \frac{\sqrt{\mathrm{Var}\left(\|\mathrm{X}\|_p\right)}}{\mathrm{E}\left(\|\mathrm{X}\|_p\right)}.$$

Both the relative contrast and the relative variance are the ratio between a measure of spread of the data (the range or the standard deviation) and a measure of location (the minimum or the expectation). Both are close to zero when the norms or distances are concentrated.

The following two theorems show that the concentration increases with dimension.

Theorem 1 (Beyer [Beyer *et al.* , 1999], adapted). *Let* $\mathrm{X}^{(j)} : 1 \leq j \leq N$ *be* N *d-dimensional i.i.d. random vectors. If*

$$\lim_{d \to \infty} \mathrm{Var}\left(\frac{\|\mathrm{X}^{(j)}\|}{\mathrm{E}\left(\|\mathrm{X}^{(j)}\|\right)} \right) = 0 \tag{3.3}$$

then, for any $\epsilon > 0$*,*

$$\lim_{d \to \infty} \mathbf{P}\left[\frac{\max_j \|\mathrm{X}_j^{(j)}\| - \min_j \|\mathrm{X}_j^{(j)}\|}{\min_j \|\mathrm{X}_j^{(j)}\|} \leq \epsilon \right] = 1$$

The theorem is interpreted as follows. Suppose a set of N data points, randomly distributed in the d-dimensional space. Some query point is supposed to be located at the origin, without loss of generality. Then, if hypothesis (3.3) is satisfied, independently from the distribution of the components of the $\mathrm{X}^{(j)}$, the difference between the largest and smallest distance to the query point becomes smaller and smaller when compared with the smallest distance, when the dimension increases.

New result

Theorem 2 (from [Francois *et al.* , tba]). *Let* $\mathrm{X} = (X_1, \cdots, X_d)$ *be a random vector with i.i.d. components :* $X_i \sim \mathcal{F}$*. Then,*

$$\lim_{d \to \infty} \frac{\sqrt{\mathrm{Var}\left(\|\mathrm{X}\|_p\right)}}{\mathrm{E}\left(\|\mathrm{X}\|_p\right)} = 0;$$

i.e. the relative variance decreases towards zero when the dimension grows.

This theorem is proved in Section 5.3.1. It shows that Minkowski and fractional norms concentrate as dimensionality increases but considers distribution by contrast with Beyer's result that consider finite datasets.

In [Aggarwal *et al.* , 2001] and in [Francois *et al.* , tba], it has been shown that, if the data are *i.i.d.* and uniformly distributed (a strong hypothesis which unfortunately seldom appears in real cases), the distances are less concentrated as p is chosen smaller. Aggarwal even advocates the use of fractional metrics (i.e. with $p < 1$) to reduce the concentration and to make

nearest-neighbour searches more robust in practice.

It appears that all Minkowski and fractional norms concentrate when the dimension is large; the concentration depends on the distribution of the data and on the type of norm. For some distributions, like the uniform one, fractional norms are always less concentrated than higher-order norms for a given sample of finite size in a fixed dimension. For other distributions it is not the case; however, all norms with $p > 2$ concentrate so much that the variance of the norms tends to zero and that the maximum and minimum values tend towards the same value, as d infinitely increases [Aggarwal *et al.* , 2001; Francois *et al.* , tba]. This will be exposed in details in Chapter 5.

3.2.3 What are the consequences of concentration?

The fact that norms concentrate in high-dimensional spaces has some consequences that bring us to question the relevance of the norms in such spaces.

As Demartines notes [Demartines, 1994], the relative error made by considering the expectation of the norm instead of the real value of the norm for a given vector is very small and gets smaller and smaller as the dimension increases. For sufficiently large dimensions, nearly all distances seem identical to the expected value.

All distances seem identical

The 'information content' of the norm function, measured by the entropy of the distribution [Cover & Thomas, 1991] of the norm decreases as the dimension grows because a set of few values are much more likely than all others. It then nearly has the same "uncertainty", and variability, as a constant function, carrying no information other than its mean.

All distances seem thus nearly the same, the maximum distances and the minimum distances are very similar, and the discriminative power of the distances in terms of similarities is less and less obvious.

In the database community, this has been acknowledged for some time, and the impact of the concentration on the indexing methods for similarity search in high-dimensional data spaces

have been studied empirically and theoretically (see for instance [Beyer *et al.* , 1999; Korn *et al.* , 2001]).

The fundamental question

For machine learning tools, however, the question is still pending: does the concentration of the norm have an impact on the distance-based learning tools? Section 4 will provide the answer for rank-based methods and Section 5 for kernel-based methods. As we will observe, these answers are different.

3.3 The stability of nearest-neighbour search

This section investigates the effects of distance concentration on methods and tools that use distances only for similarity search in the dataset.

3.3.1 Ranking the distances

In this section, we focus on the stability of the search for nearest neighbours in vector spaces. The search for nearest neighbours is central in many learning algorithms : `k-NN`, `K-MEANS`, Vector quantization, `SOM`, some Lazy Learning algorithms, etc.

Although searching for nearest neighbours can be done in sublinear time cost using indexing techniques, we will consider here only the simple linear scan through all data to find the most similar ones. We are indeed here interested in the *stability* of the nearest neighbour search and not in the performances. The consequences of the concentration on indexing techniques to accelerate the ranking and search have been studied, among others in [Berchtold *et al.* , 1997; Katayama & Satoh, 1997; Lin *et al.* , 1994].

3.3.2 The concept of stability

In [Beyer *et al.* , 1999], concerns have been raised about the stability of the search for nearest neighbours in high-dimensional datasets. Intuitively, if all distances are nearly identical, very similar query elements could have very different neighbours, and small perturbations of the query point could potentially lead to very different answers. It would be the case for instance in two dimensions if all data were distributed on a circle and the query point would be the centre of the circle, as depicted on

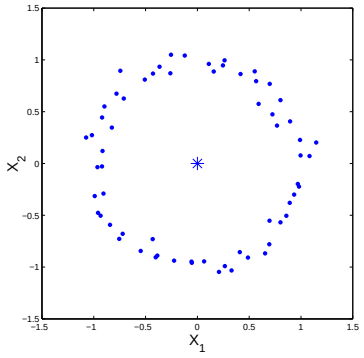


Figure 3.4: Scatter plot of an artificial two-dimensional dataset for which all distances to the query point (the star) are roughly equal. This represents a situation where all distances to the query point are concentrated. A small perturbation of the query point can lead to very different results. While this is illustrative of the notion of concentration of distances around one specific point, in high-dimensional datasets, the distances are concentrated with respect to all points!

Figure 3.4. A small perturbation of the query point (identified by a star) can lead to the retrieval of elements that are very far away one from another on the circle. Although this situation is very rarely encountered in practice in two dimension, it illustrates how unstable a query can become when all distances to the query point are roughly equal.

The definition of stability

We consider that a search for nearest neighbour is *stable* in a dataset if, on average, its outcome is robust to small perturbations : when searching for $\mathbf{z}$ the nearest neighbour of $\mathbf{x}$ an element of the dataset, we still recover $\mathbf{z}$ when we search for the nearest neighbour of $\mathbf{x} + \epsilon$ where ϵ is a small perturbation of the query point. In other words, we ask that a measurement error on the query point does not lead to the retrieval of another point than the actual nearest neighbour of the original (unaltered) query point. Intuitively, similar queries should produce similar responses.

3.3.3 Concentration and stability

In [Beyer *et al.* , 1999; Hinneburg *et al.* , 2000; Aggarwal *et al.* , 2001], it is implicitly assumed that stability and concentration (i.e. small relative contrast) are linked together. The rationale is in a sense that a distance measure that is highly concentrated brings very little relevant discriminative information and consequently makes the search for nearest neighbours unstable.

Our claim: concentration does not imply instability

Although this seems a very natural reasoning, we claim that : *The fact that the relative contrast decreases to zero as the dimensionality increases does NOT imply (that comparing distances in that space is meaningless and) that the stability of the search for nearest neighbour in a vector space decreases.* However, if the absolute contrast decreases, as it is the case for Minkowski norms with $p > 2$, precision can be lost. Following are four arguments supporting this view.

Argument 1 The important concept is the absolute contrast and not the relative one. In all distance-based tools, the *absolute* distances are used, not the distances divided by some statistics (like the minimum of distances) over the distribution of distances. There exist situations where the relative contrast decreases with dimension while the absolute contrast does not. For the Euclidean norm for instance, the absolute contrast remains constant with dimension. Therefore, the numerical precision when comparing distances remains constant too.

Argument 2 One could think that as distances concentrate, nearest neighbour classifiers will perform badly. This is not necessarily the case however. Let us consider two normally distributed, unit-variance, classes with centres respectively located at $(-1, -1, ..., -1)$ and $(1, 1, ..., 1)$. As the dimension increases, the centres 'move' away from one another, while the variances of the clusters remain constant: the classification task becomes thus easier, even though the relative variance decreases as dimension grows.

Argument 3 Even if the Euclidean distances concentrate, squared Euclidean distances do not concentrate. Even though no information is lost nor gained by squaring, the concentration phenomenon disappears. Indeed, both the expectation and the variance increase linearly with the dimension. The relative

variance for squared distances is thus constant with respect to dimension!

Argument 4 Finally, the distribution of all distances and distribution of distances to nearest neighbours do not overlap more in high-dimensional spaces than in lower-dimensional ones. Figure 3.5 shows the result of the following experiment. Some data points are drawn from a uniform distribution over the unit cube in a d-dimensional space and all pairwise distances are computed. For each dimension d, the mean and standard deviation of all distances are estimated, and the same is done for the distances to the nearest neighbours only. The upper plain line is the average value of the distances, while the lower plain line is the average distance to nearest neighbours. The dashed lines represent the standard deviation around both means. What can be observed is that as the dimension grows, the distributions of distances to nearest neighbours do not tend to overlap more with the distribution of other distances, even though all distances are concentrating. The fact that the relative contrast decreases to zero may lead us to think that the distances to nearest neighbours tend to become very similar to the distances to others neighbours, or even furthest neighbours. This is correct only in terms of order of magnitude. The average difference between distances which correspond to nearest neighbours and distances which corresponds to other data points remains constant with dimension.

The four arguments presented above are illustrative of the fact that the concentration cannot be used to estimate the relevance of the Euclidean norm. The reasoning that a concentrated situation would lead to an unstable search for nearest neighbours is flawed.

A real example

To further illustrate that concentration does not imply instability, that is loss of robustness to noise on the coordinates, we consider a real situation in a 110-dimensional space, where less concentrated situations lead to less robust queries. Minkowski and Fractional norms are used, as proposed in [Aggarwal *et al.* , 2001], with several values of the exponent p to vary the concentration levels and estimate the robustness to noise in terms of nearest neighbour retrieval. The robustness is measured using the following experiment. One element is copied from the dataset and altered with a Gaussian white noise (signal to noise

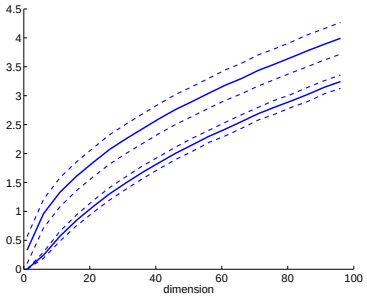


Figure 3.5: Some $N = 10000$ points are drawn in the d-dimensional hypercube $[0,1]^d$. We are interested in the distribution of all N^2 pairwise distances and in the distribution of all N distances to the nearest neighbours of every data point. The upper plain line is the average of distances to the other points while the lower is the average of distances to nearest neighbours. The dotted lines represent the standard deviation around the average. We observe that, on average, as dimension increases, and hence as concentration increases, the distances to nearest neighbours are still significantly smaller than other distances.

ratio (SNR) $\simeq 20$). This altered data is then considered as query point and the nearest neighbour is retrieved. If the retrieved element happens to be the original data from which the query point was constructed, the query is considered successful. This procedure is iterated for each data element and the robustness level is taken as the proportion of successful queries.

The Apple dataset

The dataset that we consider here is the APPLE database: a set of near-infrared absorbance spectra of apples [Benoudjit & Verleysen, 2003]. Each of the 337 data elements is a 110-dimensional vector whose coordinates correspond to wavelengths from 1100 to 2190 nm. Figure 3.6 shows both the concentration level and the robustness to noise for several p-norms. For this dataset, the more concentrated situations actually lead to the best performances, showing that concentration does not imply loss of robustness.

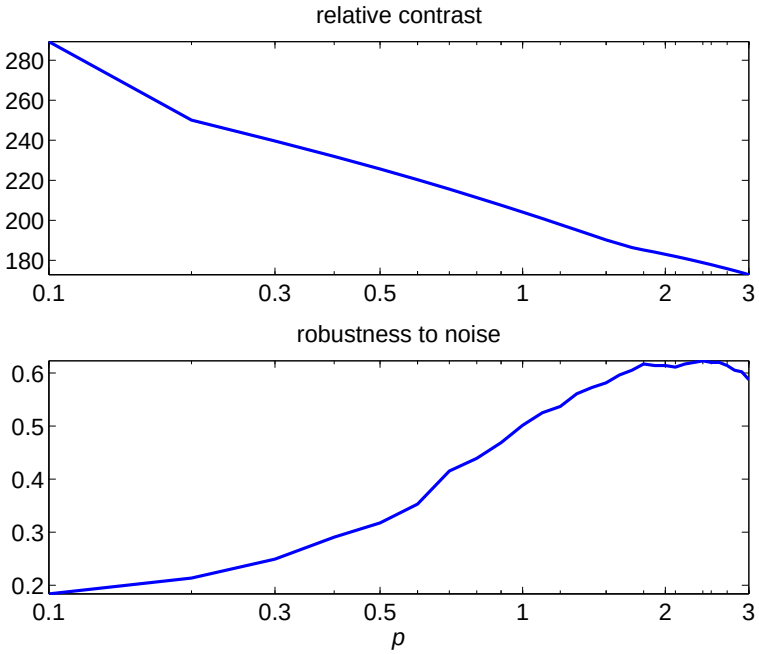


Figure 3.6: Relative contrast (top) and robustness to noise (bottom) of the nearest-neighbour search with several p-norms in the Apple dataset. Each data element, one at a time, is altered with noise (SNR=20) and then considered as the query point for nearest neighbour search. The robustness is measured as the proportion of queries that actually retrieves the original, unaltered, element). More concentrated situations actually lead to more stable searches.

Nevertheless, the Euclidean norm is not always the most adequate to measure similarity as proved by the experiments reported in [Aggarwal *et al.* , 2001] and [François *et al.* , 2005].

3.3.4 The particular case of dissimilarity data

In some cases, the attributes of the data are not measured directly; rather, all pairwise distances are. This is sometimes called dissimilarity data.

The concentration phenomenon can appear in those cases too. The concentration can be estimated, either using the relative contrast or the relative variance, on the basis of the distance matrix, as suggested in [Chavez *et al.* , 2001].

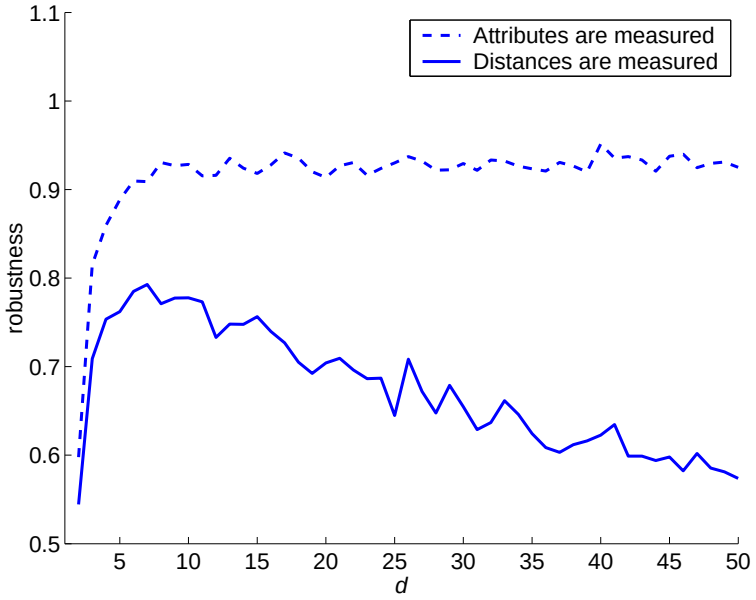


Figure 3.7: Robustness to noise, as measured as described in the previous experiment except that absolute figures are given instead of proportions, for the Euclidean norm under Gaussian White noise with SNR of 100. For each dimension, a random, uniformly distributed, dataset is drawn over $[0, 1]^d$. The dashed line corresponds to noise being added to coordinates while the plain line corresponds to noise being added to the distance measurements. As dimension increases, the robustness decreases when distance measurements are affected while it is not the case when attributes are subject to noise.

Figure 3.7 shows the robustness to noise of the nearest neighbours search in a uniformly distributed dataset, for growing dimension. A 1% measurement error is added with Gaussian noise, once to the coordinates, the other time to the distance measurement. The distances get more and more concentrated as the dimension increases. Figure 3.7 illustrates that the loss of robustness is much more dramatic when the distances are measured directly (with noise) than when the distances are computed (exactly) from the (noisy) attributes. This can be explained by the fact that the measurement noise on the coordinates (attributes) is averaged out over all coordinates while it is not the case when the measurement of the distance is noisy.

3.4 The locality of kernel-based models

In the previous section, we focused on methods that rank distances. In this section, we will consider methods that actually use the values of the distances in the estimation of the response value.

3.4.1 Using the value of the distance in kernel functions

NExamples of tools using the actual values of the distances in the computation of the response values rather than their ranks only, are the `SVM`, `RBFN`, `Lazy Learning` methods and some Fuzzy models. The value of the distance is taken explicitly into account for the prediction of the response value.

Most if not all methods that process the actual values of the distances do it through so-called *kernel* functions. Intuitively, a kernel is a function $K(\mathbf{x}, \mathbf{z}; \theta)$ that measures the similarity between its arguments $\mathbf{x}$ and $\mathbf{z}$. Kernel functions often have an additional parameter called the *width*, noted here θ. If $\mathbf{x}$ and $\mathbf{z}$ are identical, the value of the kernel is maximum; this value decreases as $\mathbf{x}$ and $\mathbf{z}$ are more and more dissimilar.

Kernel definition(s)

The name 'kernel' actually refers to slightly different functions, depending on the method. For instance, RBF Networks deal with kernels that are radially symmetric [Park & Sandberg, 1991]. Support Vector Machines need kernels that satisfy the *Mercer conditions*, that is they correspond to a scalar product in a Hilbert space [Cortes & Vapnik, 1995]. The definition of

kernel in Lazy Learning is the following: a kernel is a function between 0 and 1 such that $K(\mathbf{x}, \mathbf{x}; \theta) = 1$ and that decays smoothly with the distance between its first two arguments [Atkeson *et al.* , 1997]. This is the definition we will consider here (polynomial kernels often used in SVM are thus not considered here.)

The extra thing with kernels: the width

Kernel functions do not only transform distances into similarity measurements. Their width parameter brings a qualitative interpretation of the similarity measurement. The width defines a distance up to which two data elements are to be considered as roughly similar. This allows taking into account the variance in the data. Two very close elements (say distance 0.1 for instance) may actually be very different, if all other distances lie between 0 and 0.01. The width of the kernel can be set appropriately to cope with the distribution of the distances.

When used in a prediction model, the kernels take as first argument the new element $\mathbf{x}'$ for which the response value has to be estimated. The other argument is a parameter of the model and is called the *centre* of the kernel. The centre lives in the dataspace and may be a *centroid* [Moody & Darken, 1989], or a *support vector* [Cortes & Vapnik, 1995], or another data point [Aha, 1997]. Around this centre, the width parameter θ defines a *neighbourhood* of influence. Models based on such neighbourhoods in the data space are said to be *local*.

Gaussian kernel

Although many kernel functions can be constructed many of them have a bell-like shape to match the concept of similarity. The most representative, and most used kernel function is the Gaussian kernel (defined here up to a constant) :

$$K(\mathbf{x}, \mathbf{z}; \theta) = \exp - \left(\frac{\|\mathbf{x} - \mathbf{z}\|}{\theta} \right)^2$$

The Gaussian shape is indeed a 'natural' one : it is the distribution to which any infinite sum of i.i.d. variables tends (Central Limit Theorem.) It furthermore corresponds to the sole distribution that can be described completely by its first two moments. It is moreover indefinitely differentiable (and hence smooth). It is radial and corresponds to a scalar product in an infinite-dimensional space.

3.4.2 Tuning locality versus covering

What is expected from a kernel-based model is two-fold: *locality* and *covering*. Locality means that the kernel functions must have a high value for specific areas of the *manifold* on which the data lie, and have a low (near zero) value everywhere else. Covering means that the sum of all kernel has a high value over the whole data manifold.

While the locality property seems important in many algorithms, few papers address the reasons why it is indeed important. In the following, some intuitive arguments in the favour of local kernels are developed, without any attempt to be exhaustive.

Locality

Because kernels may be used to express in a numerical form the intuitive notion of similarity, they facilitate the interpretation of the models that use them. Indeed, regions spanned by kernels up to the limits defined by their width define groups of data that are similar and that 'participate' in a sense in a similar manner to the estimation of the response value. The intuition is clearly that the elements that contribute to the computation of the predicted response value are the elements of the dataset that are most similar to the new element.

Another argument in favour of local kernels is numerical stability. For RBFN as for SVM, the values of each kernel at each data point is gathered into an $N \times M$ matrix which is used to formulate the corresponding optimisation problem. It is known that the numerical stability of the system depends on the condition number of the matrix, which is defined as the ratio between its largest and smallest singular values [Golub & van Loan, 1996]. Local kernels lead to matrices with diagonal dominance, a property that leads to good conditioning. Intuitively, if the standard deviation is too large, the Gaussian kernels will be very flat, covering large portions of the dataspace ; consequently all the elements of the kernel matrix are large (near 1), leading to a poorly conditioned situation. Approximating a smooth but non-constant function with large kernels results in very large, both positive and negative, model coefficients.

Covering

However, kernel functions must *cover* the data space of interest. Indeed, the prediction is possible only for new data elements

lying in the space spanned by the kernel functions. The kernels must be wide enough so that their union covers the whole data manifold. If some parts of the space are not covered, the model will not be able to generalize well in those portions of the space.

Furthermore, if the widths of the kernels are too small, they will not reach (with a significant value) the data points, even those that are close to the centroids. Very large coefficients will thus result from the system solution in the fitting of the model, both in positive and negative values, in order to both include all data into the radius of attraction of at least one Gaussian kernel, and at the same time keeping a weighted sum into a small range (corresponding to a smooth function to approximate). Such situation should be avoided, as it is highly sensitive (not robust) to small perturbations in the data, in the parameters, etc.

The width to tune locality/covering

In conclusion, a model constructed with the kernels must make the right balance between covering and locality to ensure interpretability, generalization ability, numerical stability and robustness. The width of the kernels is thus often considered in kernel-based methods as a structural parameter, which has to be optimised with respect to (an estimation of) the generalization error of the model.

3.4.3 Concentration and locality

At first sight, the objective, i.e. measuring the similarity between two vectors, and the way to reach the goal, i.e. using a Gaussian kernel, perfectly match. Before we see what happens in high-dimensional settings, let us review what is expected from a kernel function to adequately serve as a similarity measure. For example, among a finite distribution, the closest data elements to a query should be evaluated as similar to the latter, while vectors that are far from the query should be evaluated as non-similar. In other words, in a finite distribution, the selected similarity measure should be able to identify in an acceptable proportion both similar and non-similar vectors to a query point. In this section, it will be shown that Gaussian kernels fit with this definition in low-dimensional spaces, while they do not in high-dimensional ones.

Tuning impossible with concentrated distances

To illustrate this problem, let us imagine that data have a Gaussian distribution centred at $\mathcal{C}$ (the following is qualitatively valid for any distribution though). We will compare the distribution of distances between all points and $\mathcal{C}$, to the shape of a kernel centred on $\mathcal{C}$ too. As the kernel will be used to assess if points are close or not from $\mathcal{C}$, this experiment allows us to verify that the kernel is discriminative in the effective range of the distance distribution, that is if it adequately gives a value close to one for the points closest to the centre and close to zero for those furthest away.

On Figure 3.8, the thick lines represent the value of the kernel as a function of the distance to $\mathcal{C}$, while the thin line (and greyed area) represent the distance distribution. One easily sees on graphs (a) and (b) that, in low dimension, for a well-chosen kernel width value (typically the mean of the distances [Benoudjit & Verleysen, 2003]), the small (resp. large) distances in the distribution will be mapped onto kernel values close to one (resp. zero). This matches the definition of an ideal kernel as detailed in the previous paragraph. On the graph, this is seen by the fact that the 5th (resp. 95th) percentile of the distribution of the distances is mapped to high (resp. low) value of the kernel function.

Tuning is impossible in high dimension

However when the dimension increases, the correspondence between the range of distances in the histogram, and the range of the decreasing slope in the Gaussian kernel cannot be guaranteed anymore. Graphs (c) and (d) refer to space dimensions 10 and 100 respectively, for several kernel width values. It is more difficult to adjust the value of the kernel width in order to cope with the ideal kernel definition: in all cases, there is a large part of the Gaussian kernel decreasing slope that falls out of the effective range of distances in the histogram. This means that close distances (left queue of the distribution) and large distances (right queue of the distribution) are hardly distinguishable from their kernel values; the notion of similarity itself (are data close or distant from one another) looses its significance.

The contrast in distances

Another view of the same phenomenon comes from the following experiment. Let us imagine a d-dimensional uniform distribution, quantized into a predefined number M of centroids. A Gaussian kernel is centred on each centroid. For each of those

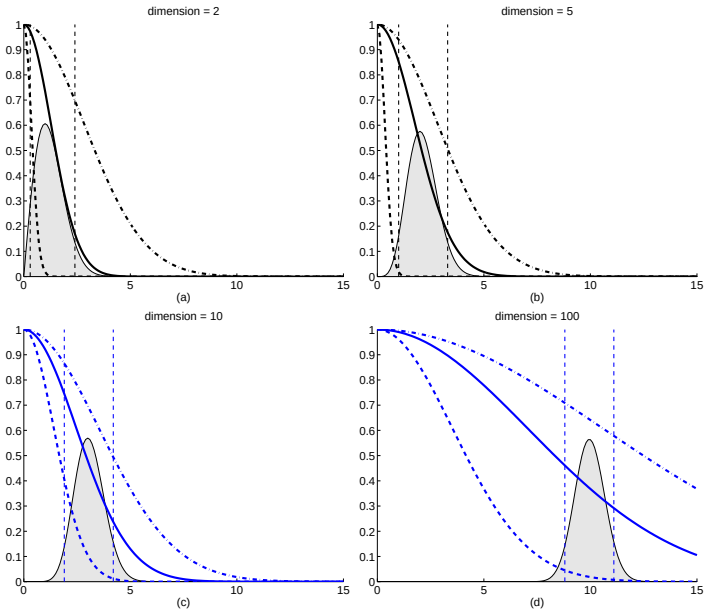


Figure 3.8: Kernel values as a function of the distance to their centres for several width values and space dimensions, along with the distribution of distances between normally distributed data and the centre (greyed area). Vertical lines correspond to 5 and 95 percentile respectively. Plain and dotted thick lines represent the kernel shape for several values of the width. We observe that in lower dimensions, it is possible to find a value of the width of the kernel such that elements close to the centre correspond to a high value (near one) of the kernel and elements further away correspond to low values of the kernel (near zero). When dimension increases, however, this property is lost.

kernels, the value of the kernel computed at the closest point to the centre and the value computed at the other centroid furthest from the centre are computed. The latter is subtracted form the former and the difference is averaged over all centroids. The result is representative of the contrast between the similarity of a point to its closest neighbour and to the furthest

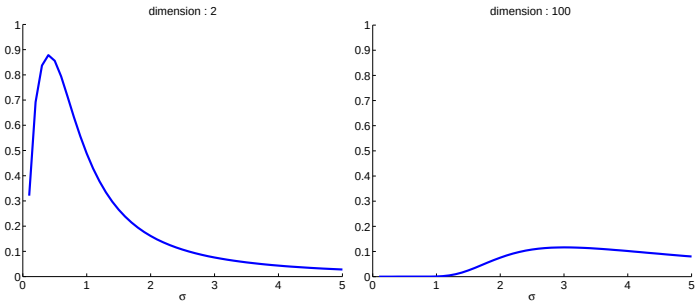


Figure 3.9: Contrast (difference between the highest and lowest observed values of the kernel in the dataset) in a 2-dimensional (left) and a 100-dimensional (right) random dataset drawn from a uniform distribution, with respect to the kernel standard deviation. In dimension 100, all kernel values are very similar, whatever width is chosen, while in dimension 2, a width chosen near 0.5 allows a contrast whose value nearly equals one.

away centroids; if the contrast is large, a model built with such a kernel can be considered as 'local'; if it is small, the notion of locality looses its significance.

Figure 3.9 shows this contrast, computed for 10000 points, with respect to the width of the kernels. In dimension 2 (left), the contrast is close to 1 for a well-chosen value of the kernel width; distances are easily distinguishable. Note that the ideal kernel standard deviation is relatively small, which corresponds to a kernel having a local character. In dimension 100, the contrast hardly reaches 0.2; distances are far less distinguishable, whatever the kernel standard deviation is.

Kernels are either too narrow or too wide

Consequently, when distances are concentrated, it becomes impossible to fine-tune the width parameter; kernels are either too local (only identical elements are considered as similar) or too global (any two data elements can be considered as similar). No 'mid-range' value for the width parameter is satisfactory, leading to loss of interpretability, of conditioning and of generalization ability. The kernels are indeed too local to cover the space or too wide to provide the necessary expression power

to the model, which is then near to a constant.

Chapter summary

Most of the methods for analysing data that are based on the definition of a metric on the dataspace can be classified as either 'rank-based' or 'kernel-based'. Most of the time, the Euclidean distance is used because it intuitively corresponds to the natural notion of distance in the three-dimensional space. However, the Euclidean norm has the counter-intuitive property to concentrate in high-dimensional spaces.

This concentration phenomenon, however, does dot prejudice the machine learning methods that only compare distances, i.e. rank data element according to their distances to some given point. The broadly used `k-NN` classifier or Kohonen's Self Organzing Maps do not suffer from the phenomenon because the variance of the distribution of distances is quite constant, and distances to nearest neighbours are still distinguishable from distances to other points. This however does not mean that the Euclidean distance is always the most suitable for measuring similarity.

By contrast to rank-based methods, Kernel-based methods, like Support Vector Machines and Radial-Basis Function Networks, will suffer from the concentration phenomenon. The fact that all distances concentrate leads to nearest neighbours distances being rather large. The Gaussian kernel shape is then not able to be 'local', i.e. to map high values (near 1: perfect similarity) to nearest distances and at the same time low values (near zero: total dissimilarity) for furthest distances. This holds whatever width of the kernel is considered.

4

Alternative measures of similarity

This chapter proposes to depart from the common usage of considering the Euclidean distance and the Gaussian kernel by default. Optimal kernels and metrics can be chosen from the respective families that generalize those notions, i.e. the Generalised Gaussian kernel family, and the Minkowski and fractional norms. This approach adds one parameter to the respective definitions of the distance and the kernel. This extra parameter helps coping with the particularities of high-dimensional spaces; methods for choosing the extra parameter at reasonable cost are proposed.

The previous chapter provides clues that for high-dimensional data, the widely-used Gaussian kernel and Euclidean distance are not necessarily appropriate functions to quantify similarity. This chapter proposes to adapt those with respect to the observations made in the previous chapter about distances in high-dimensional spaces.

Although the ideas developed in the present chapter are very close to what is called *metric learning*, we will explore other paths than those usually encountered in that field. Classically, in metric learning, the Euclidean distance is extended with a weighting matrix A:

$$d(\mathbf{x}, \mathbf{z}) = \sqrt{(\mathbf{x} - \mathbf{z})^T \cdot \mathsf{A} \cdot (\mathbf{x} - \mathbf{z})}, \tag{4.1}$$

and the problem is to find the optimal A [Atkeson *et al.* , 1997; Hastie & Tibshirani, 1996]. This kind of metric is still 'Euclidean' in the sense it is still in quadratic form; it corresponds to the scalar product in the Euclidean space. Intuitively, the "exponent" p still has the value 2. By contrast, we are interested in fractional and Minkowski metrics where the parameter p is not necessarily two:

$$d(\mathbf{x}, \mathbf{z}) = \sum_i \left(|x_i - z_i|^p\right)^{\frac{1}{p}}.$$

The same kind of extension has been developed for the Gaussian kernel so that the kernel width is different for each direction in the space [Lowe, 1995]. Learning the kernel matrix directly has also been studied [Lanckriet *et al.* , 2004] but, in the latter reference, this is restricted to *transductive learning*, that is the elements of the test set must be known at the learning stage.

The approaches presented in this chapter aim at designing a similarity measure that takes into account the particularities of high-dimensional data. Section 4.1 presents alternatives to the Gaussian kernel, which are capable of regaining a local character even in high-dimensional spaces. Section 4.2 then shows how rank-based methods can benefit from using alternatives to the Euclidean distance in high-dimensional spaces independently from concentration considerations.

4.1 Alternatives to the Gaussian kernel

In high-dimensional spaces were the distances are concentrated, the Gaussian kernel looses its interpretation in terms of locality around its centre, as we saw in the previous chapter.

4.1.1 The Generalised Gaussian kernel

The necessity to more or less span the effective range of distances between data in a real distribution setting, by the effective part of the kernel (i.e. the part with the decreasing slope), requires to add a parameter with respect to the Gaussian kernel. Besides the width that controls the slope of the kernel, there is a need for a supplementary parameter that controls the smallest distance corresponding to the decreasing part of the kernel. An example of kernel that fulfils this requirement is the *Generalised Gaussian kernel*:

$$K(\mathbf{x}, \mathbf{z}; \theta) = \exp\left(-\left(\frac{\|\mathbf{x} - \mathbf{z}\|}{\theta}\right)^{\tau}\right), \tag{4.2}$$

where τ and θ are the two parameters. The effect of the τ parameter is depicted on Figure 4.1. Figure 4.2 (left) shows an example of Generalised Gaussian kernel, with $\tau = 11$ and $\theta = 4.3$. It is seen that the kernel slope effectively covers the range of distances, according to the definition of "ideal" kernel: the smallest distances are mapper to a high value (near

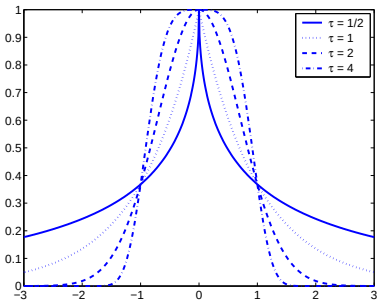


Figure 4.1: Shape of Generalised Gaussian kernels for several values of τ

one) of the kernel while the largest distances are mapped to low a value (near zero.)

One more parameter for higher contrast

Figure 4.2 (right) shows the results of the same experiment as in Figure 3.9, with respectively, the Gaussian kernel and a kernel with optimised τ, as a function of the θ. The contrast is higher with the Generalised Gaussian kernel than with the Gaussian kernel. The difference between the values of the kernel for close neighbours are much larger than those for points lying further away, which is not the case with the Gaussian kernel.

4.1.2 Choosing the kernel

Generalised Gaussian kernels allow a better matching between the shape of the kernel and the distribution of the distances. Theirs parameters θ and τ intimately relate to the distribution of the distances and thus can be set according to the specificities of that distribution directly. This means that these parameters no longer need to be optimized by costly resampling methods in Radial-Basis Function Networks (RBFN) or Support Vector Machines (SVM).

Setting parameters θ and τ

The methods to choose adequate values for τ and θ with respect to the distribution of the distances is the following. Two requirements must be fulfilled: (i) smallest distances in the distribution (which we will note d_N) are to be mapped to high values (for instance 0.95) of the kernel and (ii) largest distances

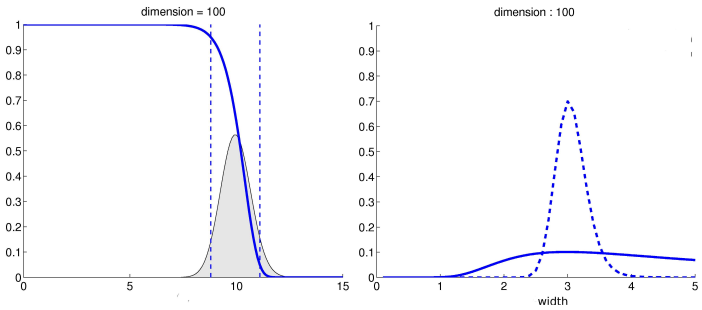


Figure 4.2: Using the same dataset as the previous experiment: (left) Kernel values along with distance distribution for the ideal kernel, $\tau = 11$; (right) contrast for the Gaussian kernel (plain line) and for the τ-kernel (dashed line) with τ optimised using (4.4) and (4.5) in dimension 100, as a function of θ.

(d_F) must be mapped to lower values (i.e. 0.05). Once translated into equations, we get the following:

$$\exp\left(-\left(\frac{d_N}{\theta}\right)^{\tau}\right) = 0.95 \quad ; \quad \exp\left(-\left(\frac{d_F}{\theta}\right)^{\tau}\right) = 0.05 \tag{4.3}$$

Solving the equations gives:

$$\tau = \frac{\ln\left(\frac{\ln(0.05)}{\ln(0.95)}\right)}{\ln\frac{d_F}{d_N}} \tag{4.4}$$

and

$$\theta = \frac{d_F}{(-\ln(0.05))^{1/\tau}} = \frac{d_N}{(-\ln(0.95))^{1/\tau}}. \tag{4.5}$$

Once the respective definitions of d_N and d_F are set (for instance respectively to the 5th and 95th percentiles of the distribution), τ is found by (4.4) and then θ is determined using one of the relations in (4.5).

4.1.3 Applicability

The Generalised Gaussian kernel can be plugged blindly into many kernel-based prediction models, with the notable exceptions of the SVM and the Kernelised k-Nearest Neighbours

(KkNN).

Restrictions for SVMs

Generalised Gaussian kernels are not Mercer kernels, i.e. they are not *positive semi-definite* (PSD) for all values of their τ parameter [Vapnik, 1998]. Great care must therefore be taken when using such kernels in Support Vector Machines.

A theorem by Schoenberg [Schoenberg, 1938] states that a function of the form $F(\|\mathbf{x}-\mathbf{z}\|)$ is positive definite if **and only if** $G(u) = F(\sqrt{\|(x)-\mathbf{z}\|})$ is continuous and completely monotonic, that is G is infinitely differentiable and satisfies

$$(-1)^a \frac{\mathrm{d}^a G}{\mathrm{d}u^a}(u) \geq 0, u \in (0,\infty), a = 0,1,2,\ldots \tag{4.6}$$

This condition implies that $F(\sqrt{\|\mathbf{x}-\mathbf{z}\|})$ is non negative, its derivative is negative and its second derivative is non negative. This last condition holds for Generalised Gaussian kernels only if $0 \leq \tau \leq 2$. As can be seen on Figure 4.1, the second derivative (curvature of the function) of $\exp(-\|.\|^\tau)$ is positive only for $0 \leq \tau \leq 1$, hence it is positive for $F(\sqrt{\|.\|}) = \exp\left(-\|.\|^{\tau/2}\right)$ when $0 \leq \tau \leq 2$. Unfortunately, if distances are large and concentrated, d_N will be large, and the corresponding value of τ will be greater than two.

However, this should not discourage us from using such kernels in SVMs. Indeed the kernel may not be PSD but the *Gram matrix* involved in the learning of the SVM may anyhow be positive semi-definite, depending on the data. In that case, the solution of the optimisation problem corresponding to the learning of the SVM will still be unique and global; however no relation will exist between the induced space and the (non-Mercer) kernel used [Burges, 1998]. For instance, the sigmoid kernel is known to be a Mercer kernel for only certain values of its parameter; it is nevertheless successfully used in the context of SVMs [Lin & Lin, 2003].

When the matrix itself is not PSD, and hence corresponds to an 'indefinite' kernel, the optimisation problem will generally not be convex, and the geometrical interpretation in terms of scalar product in an implicit Hilbert space is lost. However, indefinite kernels can be used provided that appropriate methods for solving the optimisation problem are used, for instance a

branch and bound algorithm [Pardalos & J.H. Glick, 1987]. Indefinite kernels still have a geometrical interpretation both in terms of distances between convex hulls in a pseudo-Euclidean space [Haasdonk, 2005] and as a scalar product in a Reproducing Kernel Kreĭn Space [Ong *et al.* , 2004].

Finally let us note that the matrix can also be transformed, using a spectral decomposition approach, into a 'regularised' kernel matrix that is positive semi-definite and very similar to the original matrix [Pekalska *et al.* , 2002].

Restrictions for the KkNN

It is also worth noting that a kernelised k-nearest neighbours model with a Generalised Gaussian kernel reduces to the classical k-NN in the original data space. Indeed, the squared distance between the images $\psi(\mathbf{x}), \psi(\mathbf{z})$ of the elements $\mathbf{x}, \mathbf{z}$ in the implicit space defined by the kernel function $K(\cdot, \cdot; \theta)$ is:

$$d^2(\psi(\mathbf{x}), \psi(\mathbf{z})) = K(\mathbf{x}, \mathbf{x}; \theta) - 2K(\mathbf{x}, \mathbf{z}; \theta) + K(\mathbf{z}, \mathbf{z}; \theta), \quad (4.7)$$

where θ is a parameter. If $K(\mathbf{x}, \mathbf{x}; \theta) = F(\|\mathbf{x} - \mathbf{z}\|; \theta)$ then $K(\mathbf{x}, \mathbf{x}; \theta) = K(\mathbf{z}, \mathbf{z}; \theta) = 1$. Furthermore, for the Gaussian, and for the Generalised Gaussian kernel, the term $K(\mathbf{x}, \mathbf{z}; \theta)$ is monotonically decreasing with $d(\mathbf{x}, \mathbf{z})$; hence the nearest neighbours in the implicit space are exactly the same as those identified in the data space. Using Generalised Gaussian kernels, or Gaussian kernels is consequently not interesting as it will produce the same results while demanding more computations. A more detailed proof is described in [Yu *et al.* , 2002].

4.1.4 Generalised Gaussian kernels in learning models

The usefulness of the Generalised Gaussian kernel in prediction models is tested with four kernel-based models:

1. LL: A Lazy Learning algorithm [Aha, 1997]. Each prediction for test examples are computed as a convex combination the response values associates to the k elements of the learning set that are closest to the test data. The coefficients are taken to be the values of the kernel evaluated between the test data and its nearest neighbours.
2. RBFN: The Radial Basis Function Network [Powell, 1987]. Clusters in the training data are first found by vector quantization. Then, each cluster is associated to a kernel; the prediction is based on evaluations of the kernel between the test data and the cluster representatives.

3. SVM: The Support Vector Machine [Boser *et al.* , 1992], where critical points known as *support vectors* are identified. The prediction is a combinations of kernel evaluations between those support vectors and the test data.
4. LSSVM: The Least-Squares Support Vector Machine [Suykens *et al.* , 2002], a variant of the SVM based on a quadratic loss function.

The Generalised Gaussian kernel was plugged into each of these models and the parameters of the kernel were chosen according to heuristic (4.4). For the sake of comparison, classical Gaussian kernels were also used with (i) a heuristic consisting in setting the width of the kernel according to the median of the distance distribution [Chapelle *et al.* , 2006] and (ii) a cross-validated (optimised) value of the width.

The choice of d_N and d_F

The most critical part in choosing the parameters of the kernel is in choosing the values of the "near" distances (d_N) and "far" distances (d_F). In the case a single kernel type is used for the model (like in the SVM for instance), the distribution of all pairwise distances in the training can be considered. The values of "near" and "far" are set respectively to the $5th$ and $95th$ percentiles of the distance distribution.

In practice, when the number of pairwise distances is small, it is wise to consider means instead of percentiles. Especially, in case the kernels in the model have different parameter values (e.g. in the RBFN and the LL model). In the RBFN, the "far" parameters are set to the distances to the centres of the other kernels; d_N are set to the average distance from the centre of the kernel to each data point in its cluster. For the LL tool, a similar scheme is used; the "far" parameters are set to the distances to the $(k+1)$th neighbour; d_N are set to the average distance from the test data to its k nearest neighbours in the training set.

4.1.5 Experiments

Five datasets were used for testing: the APPLE dataset, the TECATOR dataset (described in Section 2.1), the WINE, the HOUSING and DELVE datasets.

Datasets

From the APPLE spectra described in the previous experiments, an interesting, though difficult, task is to predict the crushing

force that is necessary to penetrate the skin of the apple [Benoudjit & Verleysen, 2003]. The necessary crushing force is a measure of the ripeness of the apple. Predicting this value from the near infrared spectrum of the apple brings an estimation of the ripeness of the apple in a non-destructive way. The WINE dataset also contains spectral data; the aim is to predict the content of alcohol in the wine sample; it is further described in [Benoudjit & Verleysen, 2003].

The objective in the HOUSING dataset is to predict the value of houses (in k$) described by 13 attributes representing demographic statistics of the area around each house. The dataset contains 506 instances split into 338 learning examples and 169 for testing. The DELVE Census dataset, available directly from the University of Toronto[1], comprises data collected during the 1990 US Census. Each of the 22,784 data elements concerns a small survey region and is described by 139 features measuring demographic information like the total person count in the region, the proportion of males, the percentage of people aged between 25 and 64, etc. The aim is to predict the median price of the houses in each survey region. This problem can be considered as a large scale version of the Housing dataset.

Results

Table 4.1 gives the Root Mean Square Error of each model for each dataset. The meta parameters such as the number of neighbours, the number of centroids and the "regularisation" parameters are set by 4-fold cross-validation. The additional ϵ parameter of the ϵ-insensitive Support Vector Machine for regression is chosen according to the variance of the output [Smola *et al.* , 1998].

For the `RBFN` and the Lazy Learning Model, the Generalised Gaussian kernel performs equally or better in almost all cases. In the `RBFN` each cluster corresponds to a different kernel, designed to cope with the distribution of distances in the cluster, and in the `LL` model, each data point has a different kernel, whose parameters are set according to the distances of each training point to the test point. Figure 4.3 represents the histograms of the values of τ for each kernel in the `RBFN` and the `LL` for the APPLE dataset. Most values are above 2 leading to

[1] `http://www.cs.toronto.edu/~delve/data/census-house/desc.html`

kernels that fit the shape of the distance distribution around each centroid (RBFN) or test sample (LL). For the other datasets, the tendency is confirmed for the LL model for all datasets except the TECATOR. For the RBFN model, the proportion of values of τ above 2 varies between 16% and 95%.

By contrast, for the SVM and LSSVM, the kernel is common for all support vectors. For all the datasets, the values found by heuristic (4.4) are close to two, i.e. the Gaussian kernel. Therefore, the prediction errors of the models with and without the Generalised Gaussian kernel are rather similar.

	LL		RBFN		LSSVM		SVM	
dataset	Gauss.	Gener.	Gauss.	Gener.	Gauss.	Gener.	Gauss.	Gener.
apple	4.827	**4.806**	4.407	**4.306**	3.360	**3.349**	**3.859**	3.882
housing	7.985	**6.306**	7.133	**6.192**	**9.189**	12.568	7.580	**7.346**
tecator	7.488	7.488	10.329	**7.949**	**3.249**	3.995	**5.520**	5.537
wine	0.416	**0.413**	0.776	**0.189**	**0.101**	0.102	0.218	**0.123**
delve	60.634	**57.647**	**53.407**	53.562	62.351	**53.488**	**55.129**	55.168

Table 4.1: Test RMSE for each model on each dataset. The Generalised Gaussian Kernel performs slightly better than the Gaussian kernel whose width was set using an heuristic in 14 of the 20 experiments.

Table 4.2 compares the results of the prediction using the Generalised Gaussian kernel and using a Gaussian kernel whose width has been optimised by cross validation. The results of the experiments using the optimised Gaussian kernel are not strikingly better than those using the Generalised Gaussian kernel. The latter is even better in 10 out of 20 of the experiments, mostly those involving the RBFN or the LL model. This is even more interesting considering that the optimisation of the width of the Gaussian kernel requires large amounts of computations compared with using heuristic (4.4).

4.1.6 Conclusions

The Generalised Gaussian kernels, along with heuristics (4.4) and (4.5) allows building kernel functions that fit the distance distribution between data elements. They can be used in many

	LL		RBFN		LSSVM		SVM	
dataset	OptG.	Gener.	OptG.	Gener.	OptG.	Gener.	OptG.	Gener.
apple	**4.801**	4.806	4.781	**4.306**	**3.259**	3.349	4.028	**3.882**
housing	6.812	**6.306**	**5.576**	6.192	**9.018**	12.568	**5.658**	7.346
tecator	7.488	7.488	8.854	**7.949**	**3.413**	3.995	7.957	**5.537**
wine	0.413	0.413	0.379	**0.189**	0.104	**0.102**	**0.099**	0.123
delve	58.947	**57.647**	53.896	**53.562**	61.562	**53.488**	**55.153**	55.168

Table 4.2: Test RMSE for each model on each dataset. The Generalised Gaussian Kernel performs slightly better than the Gaussian kernel whose width was optimised by cross-validation, in 10 of the 20 experiments.

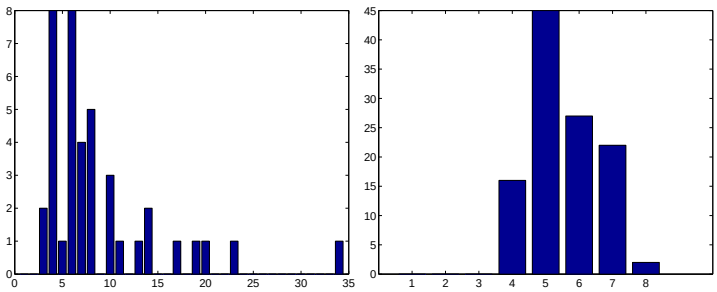


Figure 4.3: Distribution of the values of τ for all kernels in the Generalised Gaussian RBFN (left) and LL (right) built on the Apple dataset (see text for details.)

kernel-based prediction models, except for the kernelised k-nearest neighbour; and with restrictions in the Support Vector Machine. They allow better performances than the Euclidean kernel, especially in models that combine kernels with different parameters (like here the RBFN and the LL model.) With the latter models, the performances using the Generalised Gaussian kernel are close to those obtained with a Gaussian kernel whose width was optimised through cross-validation, at a much larger computational cost than for the Generalised Gaussian-based models.

4.2 Alternatives to the Euclidean distances

The conclusions of the previous chapter concerning the concentration of the Euclidean distance are that the concentration of distances does not really have a negative influence on rank-based methods. This, however, contrasts with the often observed fact that rank-based methods, no less than others, suffer from the 'curse of dimensionality'; their performances degrade as dimensionality increase. The Euclidean distance may have to be questioned, but not necessarily because it concentrates.

4.2.1 Alternative metrics to regain robustness to noise

The Euclidean norm may not necessarily be the most relevant one to measure similarity, not because of concentration but because the metric is not suitable with respect to the noise altering the data.

Two different measurements of the same object altered with noise will probably not be identical; the distance between them is then not zero. An ideal metric should take this fact into account and be able to distinguish between distances that are due to noise and distances that are due to a real dissimilarity. A relevant metric should actually measure whether some data element is likely or not to be a 'noisy version' of another. The distance should be small if the probability that one is a noisy version of the other and large if this is not likely, because they are really dissimilar.

Let us consider that, when a 'pure' d-dimensional object $\mathbf{x}$ is observed, it is altered with noise: $\mathbf{x}' = \mathbf{x} + \epsilon$, where $\epsilon = (\eta_1, \ldots, \eta_d)$. The η_i are here supposed to be independent and identically distributed random variables with a generalised Gaussian distribution function

$$f(\xi) = c^{-1} \cdot e^{-|\xi|^\tau}, \tag{4.8}$$

where c is a normalizing constant.

The link between metric and noise

Considering two different observations $\mathbf{x} = (x_1, \ldots, x_d)$ and $\mathbf{z} = (z_1, \ldots, z_d)$, the question is thus to know whether $\mathbf{x}$ and $\mathbf{z}$ are completely independent realizations or whether they are

related by noise. In other words: how likely is $\mathbf{x}$ a 'noisy version' of $\mathbf{z}$?

Let us call Δ the vector of (absolute) differences between the elements of $\mathbf{x}$ and those of $\mathbf{z}$ and δ_i its components : $\delta_i = |x_i - z_i|$. The answer to the last question is

$$\mathbf{P}\left[\eta_1 = \delta_1, \ldots, \eta_i = \delta_i, \ldots, \eta_d = \delta_d\right].$$

By the independent hypothesis, we have

$$\mathbf{P}\left[\eta_1 = \delta_1, \ldots, \eta_i = \delta_i, \ldots, \eta_d = \delta_d\right] = \frac{1}{c^d} \prod_{i=1}^{d} e^{-|\delta_i|^\tau}.$$

The negative loglikelihood is then expressed as

$$-\log \prod_{i=1}^{d} e^{-|\delta_i|^\tau} = \sum_{i=1}^{d} |\delta_i|^\tau = \|\Delta\|_\tau^\tau.$$

A small distance indicates thus a high likelihood that the differences between the elements are due to noise, while a large distance indicates a small likelihood. Therefore, the p-norm is appropriate to distinguish between distances due to noises and distances due to real dissimilarity under the hypothesis of p-Gaussian noise. This shows how the choice of the distance definition and the noise altering the data should be related.

Stability depends on the norm

The Euclidean norm is the most often used because it is most relevant when the noise is Gaussian white, a widely accepted and common noise scheme. Indeed, choosing as most similar element the element that is closest according to the Euclidean norm is equivalent to maximizing the likelihood that the retrieved data corresponds to identical data altered with white Gaussian noise. By 'white' we mean that all coordinates are potentially equally altered by the noise. By analogy with the name 'coloured' noise used, in signal analysis, to denote a noise whose intensity is not evenly distributed over the frequency spectrum, we will call *coloured* noise a noise that is not evenly distributed over the coordinates, in other words that alters some of the coordinates more than others. In this case, it is important to note that the coordinates that are altered may change from observation to observation.

While the Gaussian white noise hypothesis is acceptable in low-dimensional spaces, it may not be the case in higher-dimensional ones. The Gaussian white noise is a model which takes into account small alterations gently distributed over the coordinates. Obviously, in low-dimensional spaces, this is the only kind of noise we can cope with. Imagine now a noise scheme that models large alterations of some of the coordinates instead of small alteration of all coordinates. This kind of noise will generate so-called 'outliers' in low-dimensional spaces. In higher dimension, however, they will most probably not be considered to be outlier, but merely noisy data, altered with a specific noise, called burst noise, or salt and pepper, etc. In [Aggarwal *et al.* , 2001] and in [François *et al.* , 2005], it was shown that fractional norms are well suited when such 'coloured' noise schemes are considered.

Experiment

Figure 4.4 shows the robustness (see definition in Section 3.3) of nearest-neighbour search (for $p = 1/2, 1$ and 2) in the APPLE spectra dataset with white noise of increasing variance. Figure 4.5 shows the same experiment but with a more 'coloured' noise scheme: each coordinate of the data vectors is altered with a given probability. If it is altered, the value is randomly changed to another value in the same range as the original data. We observe that for a white noise, the Euclidean distance is the most robust while for a noise scheme that does not alter all coordinates in the same way; the fractional norm is the most robust one. More experiments leading to similar observations are described in [François *et al.* , 2005]. In the next section, we will see that, under non-Gaussian noise, the fractional norms perform better than the Euclidean one in prediction models.

4.2.2 Fractional norms are quasi-norms

As mentioned in Section 3.2, fractional norms do not obey the triangle inequality and hence are often called quasi-norms. Although it may not seem obvious at first sight, the triangle inequality plays a crucial role in many clustering methods.

The triangle inequality is important for clustering

Indeed, many of them, like the `K-MEANS` of the Self-Organizing Maps (`SOM`) use a centroid representation for the clusters. These algorithms aim at minimizing the sum of distances between each $\mathbf{x}^{(j)}$ point in the dataset and the centroid of the cluster to which it belongs:

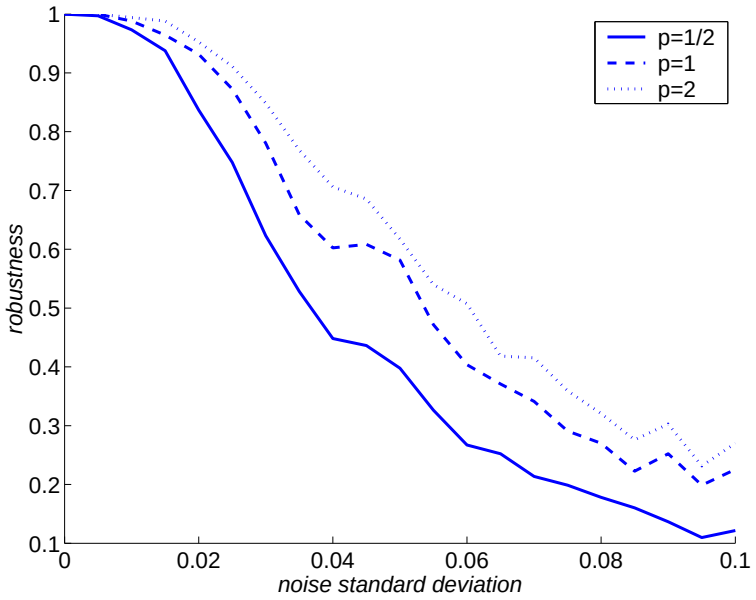


Figure 4.4: Robustness to noise, as measured as described in Section 3.3, for several p-norms under Gaussian White noise in the Apple dataset. The Euclidean norm is always more robust.

$$minimize \sum_{j=1}^{N} \|\mathbf{x}^{(j)} - C(\mathbf{x}^{(j)})\|,$$

where $C(\mathbf{x}^{(j)})$ is the function that maps an element of the dataset to the centroid of the cluster to which it belongs. However, the actual objective of the clustering is to build groups of similar data: each data element in a cluster is supposed to be similar to every other element in that cluster. Actually, the algorithm should minimize the sum of all $O(N^2)$ pairwise distances between points in each cluster:

$$minimize \sum_{\mathbf{x}^{(j)},\mathbf{x}^{(j')} \mid C(\mathbf{x}^{(j)})=C(\mathbf{x}^{(j')})} \sum \|\mathbf{x}^{(j)} - \mathbf{x}^{(j')})\|.$$

If $\|.\|$ obeys the triangle inequality, the distance between any two points in a cluster is bounded by twice the maximum

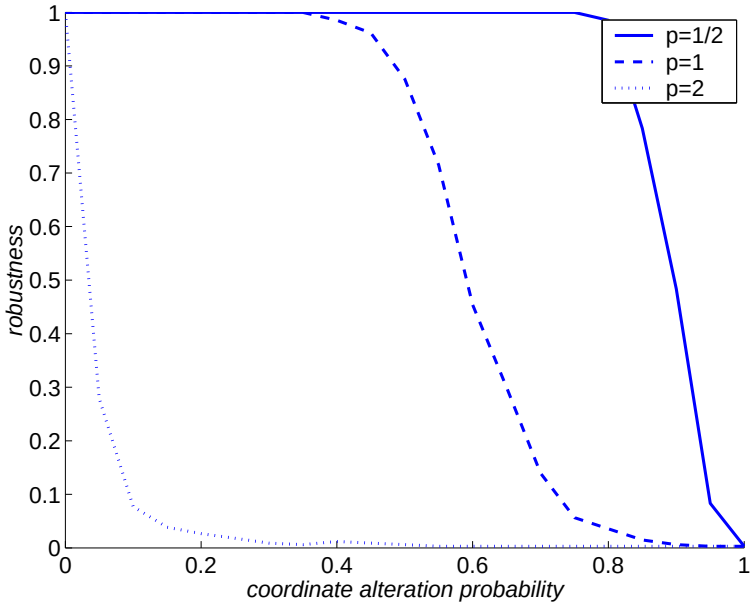


Figure 4.5: Robustness to noise, as measured as described in Section 3.3, for several p-norms under a masking noise. The fractional norm is always more robust.

distance between each of those two points and the centroid: $\forall \mathbf{x}^{(j)}, \mathbf{x}^{(j')} \,|\, C(\mathbf{x}^{(j)}) = C(\mathbf{x}^{(j')})$

$$\begin{aligned}\|\mathbf{x}^{(j)} - \mathbf{x}^{(j')}\| &\leq \|\mathbf{x}^{(j)} - C(\mathbf{x}^{(j)})\| + \|\mathbf{x}^{(j')} - C(\mathbf{x}^{(j)})\| \\ &\leq 2\max\left(\|\mathbf{x}^{(j)} - C(\mathbf{x}^{(j)})\|, \|\mathbf{x}^{(j')} - C(\mathbf{x}^{(j)})\|\right).\end{aligned}$$

Therefore, minimizing all distances to centroids ensures that all pairwise distances in a cluster are kept small as well.

If the distance used in the clustering algorithm does not obey the triangle inequality, the resulting clusters may be such that distances between points in that cluster are large even though each point is close to the centroid. But this should not discourage us from using fractional distances in clustering algorithms. Indeed, although the classical triangle inequality is not valid, another, wider, bound exists [Carothers, 2004]: for $0 < p \leq 1$,

A relaxed inequality

$$\|\mathbf{x}^{(j)}-\mathbf{x}^{(j')}\|_p \leq 2^{\frac{1}{p}-1}\left(\|\mathbf{x}^{(j)}-C(\mathbf{x}^{(j)})\|_p + \|\mathbf{x}^{(j')}-C(\mathbf{x}^{(j)})\|_p\right). \tag{4.9}$$

If $p = 1$ the inequality corresponds to the classical triangle inequality. For $p = 1/2$, the coefficient $2^{\frac{1}{p}-1}$ equals two, while if $p = 1/4$, it equals 8. The lower the value of p the larger the coefficient in the inequality. Therefore, clustering with fractional norms does not make sense for very small values of p but it will provide sound results if p is close enough to one.

4.2.3 The choice of p

The previous subsection illustrated the link between the noise scheme altering the data and the optimal metric with respect to that noise, clearly showing that the 'optimal' value of p is highly application-dependent.

In a prediction or classification problem, the value of p could be chosen so as to get the best model performances, according to the expected error in predicting the response value or the class label. It would thus be considered as an additional parameter to the model: the norm that is chosen is the norm that leads to a model with best prediction performances. However, this would necessitate multiplying the computation times by the number of different values of p that are tested.

Choosing the norm *a priori*

An elegant alternative is to choose the value prior to the building of the model. In this case, the norm is chosen before any prediction model is constructed. It is chosen according to a statistical measure of relevance for each p-norm that is considered. This statistical measure gives each p-norm a score based on how well similar data elements according to the p-norm relate to similar response values or class labels. If data elements that are close in the data space, i.e. similar according to the norm, are also close in terms of associated response values, then the norm is considered as a relevant measure of similarity for those data. The leave-one-out performance of a 1-nearest-neighbour model is fast and suitable for this purpose. Nonparametric noise estimators as the Gamma test [Stefánsson *et al.* , 1997] or the Differogram [Pelckmans *et al.* , 2005] could be used too (we will talk about the Gamma test further in Section 6.2.)

The LOO error of the `1-NN` for prediction

In the following experiments, we added a sort of *burst noise* to the data: some coordinates were altered with a multiplicative noise of high amplitude. With such a noise scheme, strongly non-Gaussian, fractional norms are better suited to measure similarity. The leave-one-out error of a `1-NN` is used to measure the relevance of each norm and to choose the most relevant one. Then, a `RBFN` with generalised Gaussian is built with the chosen norm. The fact that generalised Gaussian kernels are used here is important because as fractional norms often have large expectation (the expectation of the norm is the pth root of the dimension see Lemma 1 page 80) the shape of the kernel really needs to be adapted. Table 4.3 reports the Root Mean Square Error – RMSE of prediction on an independent test set for the HOUSING, WINE, TECATOR and APPLE datasets.

Dataset	d	Learning/test	Euclidean norm	Fractional norm
HOUSING	13	400/106	$p = 2 \rightarrow 9.737$;	$p = 1/8 \rightarrow 7.056$
WINE	256	94/34	$p = 2 \rightarrow 0.388$;	$p = 1 \rightarrow 0.106$
TECATOR	100	172/43	$p = 2 \rightarrow 8.220$;	$p = 1/8 \rightarrow 6.768$
APPLE	110	250/87	$p = 2 \rightarrow 1.638$;	$p = 1/2 \rightarrow 1.519$

Table 4.3: Performances of the RBFN (with generalised Gaussian kernels) on several datasets with the Euclidean norm and with fractional norms. The kernel parameters are chosen according to the heuristic described in this section and the number of centres is chosen by 4-fold cross validation. The data are altered with burst noise. The norm is chosen according to the leave-one-out performances of a `1-NN`. For each of these datasets, the fractional norm, chosen prior to the building of the model, gives better results than the Euclidean norm.

Relevance feedback

In many multimedia database systems, the user feedback can be used to estimate the relevance of the result of the search for similar elements. This can be translated into the relevance of the metric used to get similar elements. For instance, in image and text retrieval, the relevance measure is used to weight the coordinates in the computation of the distance to better reflect the percieved (subjective) similarity between objects[Rui *et al.* , 1998; Salton & Buckley, 1990].

A relevance feedback algorithm is then the following : suppose that all elements are described by a feature vector ; given a query Q, the system retrieves the nearest or the 2 nearest neighbours according to several p-norms (with p=0.1, 0.5, 1, 2, 4 for instance). The user then identifies the most relevant results, and the score of the p-norms corresponding to those results is increased. After several iterations, the p-norm with highest score is chosen.

To illustrate this procedure, the XM2VTS database is used. This database comprises 600 photographs, altered with Salt and Pepper noise (some pixels are set to black or white randomly) of 200 individuals (i.e. 3 pictures per individual.) At each iteration a picture from the dataset is considered as query point. Its nearest neighbour according to several p-norms are retrieved, and the score of the norms for which the retrieved image corresponds to the same individual is increased by one. Figure 4.6 shows the evolution of the score for each p-norm at iterations 1 to 20 and iteration 580 to 600. We can see that the 1/2-norm is the one with the highest score after 20 iterations and still the one after all 600 iterations. The 1/4- and 1/8-norms also have high scores by contrast to higher-order norms that perform poorly. The same experiment was repeated 100 times with only 10 iterations. In 86% of these experiments was the 1/2-norm identified as the most relevant. The 1/8-norm was chosen as the most relevant 3 times, the 1/4-norms only once, the 1-norm ten times and the other (higher-order) norms were never chosen.

The leave-one-out performances of the 1-nearest neighbour classifier with the 1/2-norm is 75% good classification while it is only 10% for the Euclidean norm.

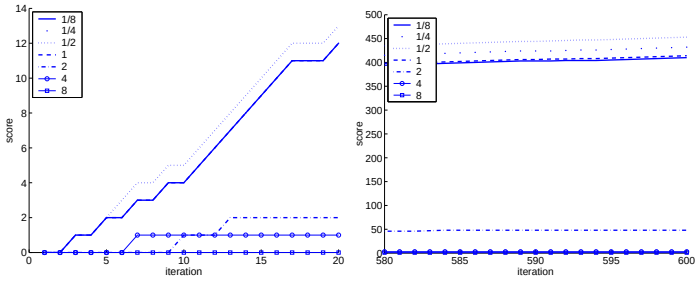


Figure 4.6: The evolution of the score of each norm as a function of the number of iterations for the face image database. The 1/2-norm appears as the most relevant from iteration 10 on.

Chapter summary

The Gaussian kernel and the Euclidean distance, event though being widely used and for good reasons in low dimensions, may not be suitable in high-dimensional spaces. The Generalised Gaussian kernel, along with a simple and intuitive heuristic, can be used to build kernels that are adapted to the specific distance distribution observed in high-dimensional datasets. Not only does the use of Generalized Gaussian kernels allow recovering the intuition of locality of kernels; it also leads to a very simple and effective choice of the parameter of the kernel. Consequently, heavy cross-validation techniques for choosing those parameters accurately in prediction models can be avoided.

Furthermore, the Euclidean distance has been recognized as relevant when the noise on the data is Gaussian and white. However, high-dimensional data can be subject to many different noise schemes that are not accurately modelled with the Gaussian distribution whose variance is equal for all attributes. The family of Minkowski distances, augmented with fractional distances, can be interesting to use in such situation. The parameter of the norm can be optimised using the leave-one-out error of a simple `1-NN` model.

5

Focus on the concentration of the norm

The relevance of the Euclidean distance has been questioned in the past and fractional norms were introduced to fight the concentration phenomenon. Nevertheless, the literature is far from being complete. This chapter shows that all norms concentrate when dimensionality is high, even if an infinite number of data elements are considered and even if those data elements are drawn from distributions with infinite support (e.g. the Gaussian distribution). Furthermore, an estimation of the concentration as a function of the exponent of the norm and of the distribution of the data is given. It leads to the conclusion that, contrarily to what is generally admitted, fractional norms are not always less concentrated than the Euclidean norm; a counter-example is given to prove this claim. Theoretical arguments are presented which show that the concentration phenomenon can appear for real data that do not match the hypotheses of the theorems, in particular the assumption of independent and identically distributed variables.

This chapter proposes some insights into the phenomenon of concentration of the distances, more particularly the Minkowski distances and fractional distances, details and proves Theorem 2 presented in Chapter 3. Section 5.1 presents the most important results that can be found in the literature about the concentration phenomenon. Those results are then discussed and extended in Section 5.2. For the sake of clarity, all proofs and illustrations are delayed to Section 5.3.

5.1 State of the Art

Literature on the subject of concentration come from various and independent sources, the main results are described.

5.1.1 Concentration of the Euclidean norm

Demartines' result

In the experiment described in Figure 3.1 (page 31), we observed that the expectation of the norm of a random vector increases with the dimension while its standard deviation (and hence its variance) remains rather constant. Demartines [Demartines, 1994] has theoretically confirmed this fact.

Theorem 3 (Demartines). *Let* $\mathrm{X} \in \mathbb{R}^d$ *be a random vector with independent and identically distributed (i.i.d.) components :* $X_i \sim \mathcal{F}$. *Then*

$$\mathrm{E}\left(\|\mathrm{X}\|_2\right) = \sqrt{\alpha \cdot d - \beta} + O(1/d) \text{ and} \tag{5.1}$$

$$\mathrm{Var}\left(\|\mathrm{X}\|_2\right) = \beta + O(1/\sqrt{d}), \tag{5.2}$$

where α *and* β *are constants that depend on the distribution of* X *but do not depend on its dimension.*

The theorem is valid whatever the distribution $\mathcal{F}$ of the X_i might be. Different distributions will lead to different values for a and b, but the asymptotic results remain.

The theorem proves that the expectation of the Euclidean norm of random vectors increases as the square root of the dimension, while its variance is constant and independent from the dimension. Therefore when the dimension is large, the variance of the norm is very small compared with its expected value.

Vector seem normalised

Demartines concludes that when the dimension is large, vectors seem normalised: the relative error made while considering $\mathrm{E}\left(\|\mathrm{X}\|_2\right)$ instead of the real value of $\|\mathrm{X}\|_2$ becomes negligible. As a consequence, high-dimensional vectors appear to be distributed on a sphere of radius $\mathrm{E}\left(\|\mathrm{X}\|_2\right)$. Demartines also notes that since the Euclidean distance is the norm of the difference between two random vectors, its expectation and variance follow laws (5.1) and (5.2); pairwise distances between points in high-dimensional spaces seem to be all identical.

Demartines mentions that if the X_i are not independent, the results are still valid provided we replace d with the actual number of *degrees of freedom*, but he does not explain nor illustrate this claim.

The result from Demartines is interesting in that it confirms the experimental results, but it is restricted to the Euclidean norm, and makes the rather strong hypothesis of independence and identical distributions.

5.1.2 Concentration of arbitrary norms

Independently from the results of Demartines, Beyer explored the effect of dimensionality on the nearest neighbour problem [Beyer *et al.* , 1999].

Beyer's result

While Demartines defines a dataset $\left\{x^{(j)}\right\}_{j=1}^{N}$ as consisting of N independent draws from a single random vector X, Beyer considers N random vectors $\mathrm{P}^{(j)}$; a dataset is then made of one realization of each random vector. The main result of Beyer is the following theorem, recalled from Chapter 3. The original theorem is stated for an arbitrary distance measure; it is rewritten here with norms for illustration purposes.

Theorem 4 (Beyer, adapted). *Let $\mathrm{P}^{(j)} : 1 \leq j \leq N$ be N d-dimensional i.i.d. random vectors and $\|\cdot\|$ denote an arbitrary norm. If*

$$\lim_{d\to\infty} \mathrm{Var}\left(\frac{\|\mathrm{P}^{(j)}\|}{\mathrm{E}\left(\|\mathrm{P}^{(j)}\|\right)}\right) = 0 \tag{5.3}$$

then, for any $\epsilon > 0$,

$$\lim_{d\to\infty} \mathbf{P}\left[\frac{\max_j \|\mathrm{P}_j^{(j)}\| - \min_j \|\mathrm{P}_j^{(j)}\|}{\min_j \|\mathrm{P}_j^{(j)}\|} \leq \epsilon\right] = 1$$

The theorem is interpreted as follows. Suppose a set of N data points, randomly distributed in the d-dimensional space. Some query point is supposed to be located at the origin, without loss of generality. Then, if hypothesis (5.3) is satisfied, independently from the distribution of the components of the $\mathrm{P}^{(j)}$, the difference between the largest and smallest distance to the query point becomes smaller and smaller when compared with the smallest distance when the dimension increases. The ratio

$$\frac{\max_j \|\mathrm{P}_j^{(j)}\| - \min_j \|\mathrm{P}_j^{(j)}\|}{\min_j \|\mathrm{P}_j^{(j)}\|}$$

is called the *relative contrast*.

All points seem located at the same distance from the query

Beyer concludes that all points seem to be located at approximately the same distance from the query point; the concept of nearest neighbour in a high-dimensional space is then less intuitive than in a lower-dimensional one.

Beyer explores some scenarios which satisfy (5.3) and some which do not. In particular, he asserts that Minkowski norms satisfy the hypothesis but does not prove his claim. A proof will be given in Section 5.3.

5.1.3 Concentration of Minkowski norms

Hinneburg focuses on the search for nearest neighbour too [Hinneburg *et al.* , 2000]. He produced the following theorem relative to Minkowski norms.

Hinnerbug's result

Theorem 5 (Hinneburg). *Let* $\mathrm{P}^{(j)} : 1 \leq j \leq N$ *be* N *d-dimensional i.i.d. random vectors, and* $\|.\|_p$ *be the Minkowski norm with exponent* p*. If the* $\mathrm{P}^{(j)}$ *are distributed (not necessarily uniformly) over* $[0,1]^d$*, then there exists a constant* C_p *independent from the distribution of the* $\mathrm{P}^{(j)}$ *such that*

$$C_p \leq \lim_{d \to \infty} \mathrm{E}\left(\frac{\max_j \|\mathrm{P}^{(j)}\|_p - \min_j \|\mathrm{P}^{(j)}\|_p}{d^{\frac{1}{p}-\frac{1}{2}}} \right) \leq (N-1) \cdot C_p. \tag{5.4}$$

Hinneburg notes that a consequence of (5.4) is the surprising fact that, on average, the *contrast*

$$\max_j \|\mathrm{P}^{(j)}\|_p - \min_j \|\mathrm{P}^{(j)}\|_p, \tag{5.5}$$

grows as $d^{1/p-1/2}$.

The 1-norm seems to concentrate less than the others

He further concludes that, as a result, the contrast converges to a constant when the dimension increases and when the Euclidean distance is used. For the L_1 norm, it increases as $\sqrt{d}$, for the Euclidean norm ($p = 2$), it remains constant, and for norms with $p \geq 3$, it tends toward zero. Hinneburg concludes that for L_p metrics with $p \geq 3$, the nearest-neighbour search in a

high-dimensional space tends to be meaningless for Minkowski norms with exponent larger than, or equal to, 3, since the maximum observed distance tends toward the minimum one. The distance has lost its discriminative power between the notions of 'close' and 'far'.

However, monotonic decrease for $\max_j \|\mathrm{P}^{(j)}\|_p - \min_j \|\mathrm{P}^{(j)}\|_p$ as a function of p is not ensured by (5.4). The conclusion we can draw from this theorem is weaker than as claimed by Hinneburg, and is the following: on average, the ratio between the contrast and $d^{1/p-1/2}$ is bounded. But the bounds themselves depend on the value of p. Furthermore, if the number of points n is large, the upper bound may be very large too. The conclusion that the relative contrast converges to a constant when p is 2, for example, may thus not be valid anymore. In practice, though, it may appear that the value of the ratio is much closer to the lower bound than to the upper one. A result independent from the number of points is presented in the next section.

Yianilos' result

Yianilos [Yianilos, 1999] mentions that the standard deviation of the p-norm, $p \geq 1$, of a uniformly distributed random vector is $\Theta(d^{1/p-1/2})$ and sketches a proof. By contrast to the approaches by Beyer and Hinneburg, Yianilos considers random vector distributions rather than realizations of random vectors. He also asserts that the *i.i.d.* hypothesis can be weakened. He then makes the same conclusions as Hinneburg does and investigates the consequences for advanced methods for similarity search in databases, i.e. methods based on an index that perform the search in sub-linear time. We will see a similar result in Section 5.2 without the restriction of the uniform distribution and on the values of p.

5.1.4 Concentration of fractional norms

Aggarwal's result

Aggarwal extends Hinneburg's result to fractional p-norms [Aggarwal *et al.* , 2001]. In a sense, if $p = 2$ is "better" than $p = 3$ and $p = 1$ is "better" than $p = 2$, why not to look at $p = \frac{1}{2}$ to see if it is "better" than $p = 1$? Aggarwal produced the following theorem, where fractional p-norms are defined as in (3.1), with fractional values of p.

Theorem 6 (Aggarwal). *Let* $\mathrm{P}^{(j)} : 1 \leq j \leq N$ *be* N *d-dimensional independent random vectors, uniformly distributed over* $[0, 1]^d$.

There exists a constant C independent from p and from d such that

$$C\sqrt{\frac{1}{2p+1}} \leq \lim_{d\to\infty} \mathrm{E}\left(\frac{\max_j \|\mathrm{P}^{(j)}\|_p - \min_j \|\mathrm{P}^{(j)}\|_p}{\min_j \|\mathrm{P}^{(j)}\|_p}\right)\sqrt{d} \leq (N-1)C\sqrt{\frac{1}{2p+1}}$$

Fractional norms concentrate less then others

Aggarwal notes that the constant $\sqrt{1/(2p+1)}$ may play a valuable role in affecting the relative contrast and confirmed it experimentally with synthetic datasets. He then concludes that, on average, fractional norms provide better contrast than Minkowski norms in terms of relative distances.

Once again, the result of Theorem 6 does not imply that the relative contrast will evolve as $\sqrt{d}$: Aggarwal's theorem results in an interval whose upper bound diverges as the number of points grows. Finally, the result is valid only for uniform distributions of $\mathrm{P}^{(j)}$, and nothing is said about other distributions.

Skala's result

Independently from Aggarwal, Skala [Skala, 2005] has shown that the ratio

$$\rho_p(d) = \frac{\mathrm{E}\left(\|\mathrm{X}\|_p\right)^2}{2\mathrm{Var}\left(\|\mathrm{X}\|_p\right)}, \tag{5.6}$$

increases linearly with dimension d. Here, X is a random vector whose components are *i.i.d.* distributed. The theorem does not require uniform densities; however, it is exact only in the $p = 1$ case and gives an approximate result for other values of p.

5.2 Further theoretical results

Section 5.1 reviewed major results from the literature, and briefly pointed out some of their weaknesses. This section will describe new results in order to better understand the phenomenon of norm concentration in high-dimensional spaces. All proofs and examples are postponed to Section 5.3 for the ease of readability.

5.2.1 The finite and bounded sample

A finite number of points will most probably be sparsely distributed in a high-dimensional space. Most points will be far

away from each other, and the density will be very low over the whole space. This is referred to as the *empty Space Phenomenon* [Bellman, 1961]. If furthermore the points live in a closed (bounded) region of the space, for instance the $[0, 1]^d$ hypercube, then the maximum distance is bounded too. In such a situation, it may happen that the relative contrast is very low. The question is then: Is the Concentration Phenomenon a side effect of the Empty Space Phenomenon, just because we consider a finite number of points in a bounded portion of a high-dimensional space? Would the conclusions still hold if an infinite number of points (in other words a distribution) spanning the whole space were considered?

Unfortunately, the results from Beyer, Hinneburg and Aggarwal cannot be extended to the case where the number of data points is arbitrarily large. Indeed, the bounds on the relative contrast depend on the number of points. Furthermore, if the values the data points $\mathrm{P}^{(j)}$ can take are unbounded, the notion of relative contrast may not be relevant anymore since it relies on maximum and minimum values.

By contrast to Beyer, Hinneburg and Aggarwal's results, Demartines and Yianilos' ones do not refer to a finite number of points, but rather to a distribution. An interesting result would be to extend their results to fractional p-norms.

The relative variance

In that purpose, it is proposed to use the ratio

$$RV_{\mathcal{F},p} = \frac{\sqrt{\mathrm{Var}\left(\|\mathrm{X}\|_p\right)}}{\mathrm{E}\left(\|\mathrm{X}\|_p\right)} \tag{5.7}$$

as a measure of the concentration; $RV_{\mathcal{F},p}$ will be called the *relative variance* of the norm. It is non-negative; a small value indicates that the distribution of the norm is concentrated, while a larger one indicates a wide effective range of norm values. Intuitively, we can see that $RV_{\mathcal{F},p}$ measures the concentration by relating a measure of spread (standard deviation) to a measure of location (expectation). In that sense, it is similar to the relative contrast that also relates a measure of spread (range) to a measure of location (minimum).

The main result of this section is that regarding the relative variance, all p-norms concentrate as the dimensionality of the

space increases. This is stated more precisely in Theorem 2, deduced from two lemmas that respectively characterize the individual behaviours of the variance and of the expectation of the norm. Those lemmas are presented here while their respective proofs can be found in Section 5.3.

Lemma 1. *Let* $\mathrm{X} = (X_1, \cdots, X_d)$ *be a random vector with i.i.d. components :* $X_i \sim \mathcal{F}$*. Then,*

$$\lim_{d\to\infty} \frac{\mathrm{E}\left(\|\mathrm{X}\|_p\right)}{d^{1/p}} = c \tag{5.8}$$

where c *is a constant independent from the dimension* d *but related to* p *and to the distribution* $\mathcal{F}$*.*

The expectation of the norms

Therefore, it appears that the expectation of the p-norm of a random vector grows as the p^{th} root of the dimension. Note that this result is consistent with (5.1) in the Euclidean case ($p = 2$).
The second lemma concerns the variance of the norm.

Lemma 2. *Let* $\mathrm{X} = (X_1, \cdots, X_d)$ *be a random vector with i.i.d. components :* $X_i \sim \mathcal{F}$*. Then,*

$$\lim_{d\to\infty} \frac{\mathrm{Var}\left(\|\mathrm{X}\|_p\right)}{d^{\frac{2}{p}-1}} = c', \tag{5.9}$$

where c' *is a constant independent from the dimension* d *but related to* p *and to the distribution* $\mathcal{F}$*.*

The variance of the norms

The dependency on $d^{\frac{2}{p}-1}$ indicates that the variance remains constant with the dimension for the Euclidean distance. We can also see that the variance decreases when the dimension increases for p-norms with $p > 2$ and increases for p-norms with $p < 2$. This is consistent with Demartines' and Hinneburg's results.

From these Lemmas it can be shown that all p-norms concentrate as the dimension increases

Theorem 2 *(recalled from Chapter 3) Let* $\mathrm{X} = (X_1, \cdots, X_d)$ *be a random vector with i.i.d. components :* $X_i \sim \mathcal{F}$*. Then,*

$$\lim_{d\to\infty} \frac{\sqrt{\mathrm{Var}\left(\|\mathrm{X}\|_p\right)}}{\mathrm{E}\left(\|\mathrm{X}\|_p\right)} = 0;$$

i.e. the relative variance decreases towards zero when the dimension grows.

The proof can be found in Section 5.3. From Theorem 2 it can be concluded that the concentration of the norms in high-dimensional spaces is really an intrinsic concentration property of the norms and not a side effect of the finite sample size or from the empty space phenomenon. This result extends Demartines' one to all p-norms and is more general than the result of Hinneburg. Indeed, Hinneburg considers finite samples and bounded distribution supports, which makes it unclear whether the concentration phenomenon is really an intrinsic property of the norm. However, (5.4) is only an inequality where the upper bound grows infinitely with the size of the dataset.

Moreover, while all p-norms concentrate as the dimension increases, they do not concentrate in the same way. Characterizing these differences with respect to p is the topic of the next subsection.

5.2.2 Impact of the value of p on the concentration

In the previous section, it has been shown that all p-norms concentrate, whatever is the value of p ($p > 0$). In this section, the relationship between the relative variance and the value of p for a given dimension will be made explicit.

The relative variance depends on p

Theorem 7. *Let* $\mathrm{X} = (X_1, \cdots, X_d)$ *be a random vector with i.i.d. components :* $X_i \sim \mathcal{F}$*. Then,*

$$\lim_{d\to\infty} \sqrt{d} \cdot \frac{\sqrt{\mathrm{Var}\left(\|\mathrm{X}\|_p\right)}}{\mathrm{E}\left(\|\mathrm{X}\|_p\right)} = \frac{1}{p}\frac{\sigma_{\mathcal{F},p}}{\mu_{\mathcal{F},p}}$$

where $\mu_{\mathcal{F},p} = \mathrm{E}\left(|X_i|^p\right)$ *and* $\sigma^2_{\mathcal{F},p} = \mathrm{Var}\left(|X_i|^p\right)$.

If d is large, we can thus approximate the relative variance by

$$RV_{\mathcal{F},p} = \frac{\sqrt{\mathrm{Var}\left(\|\mathrm{X}\|_p\right)}}{\mathrm{E}\left(\|\mathrm{X}\|_p\right)} \simeq \frac{1}{\sqrt{d}} \cdot \left(\frac{1}{p} \cdot \frac{\sigma_{\mathcal{F},p}}{\mu_{\mathcal{F},p}}\right);$$

this means that for a fixed (large) dimension, the relative variance evolves with p as

$$\mathcal{K}_{\mathcal{F},p} = \frac{1}{p} \cdot \frac{\sigma_{\mathcal{F},p}}{\mu_{\mathcal{F},p}}. \tag{5.10}$$

The evolution of $\mathcal{K}_{\mathcal{F},p}$ with p determines the value of the relative variance for a fixed dimension d.

Aggarwal's claim

Aggarwal has claimed that if the points are uniformly distributed, the relative contrast at fixed dimension increases as p decreases. He concludes that fractional norms, with $0 < p < 1$, are in general less concentrated than higher order norms ($p \geq 2$). We will show below and prove in Section 5.3.4 that, under the uniform distribution hypothesis, the relative variance increases as p decreases too.

Aggarwal takes this result as a strong clue that fractional norms are always less concentrated and present a larger relative contrast than higher-order norms [Aggarwal *et al.* , 2001]. Nevertheless, in general, the function described by (5.10) is not strictly decreasing with p as stated in the following Proposition.

Proposition 1. *The relative variance* (5.10) *is (a) a strictly decreasing function of p when the variables are distributed according to a uniform distribution $\mathcal{F}$ over the interval $[0, 1]$, but (b) increases locally with p for other distributions $\mathcal{F}$. Fractional norms are not always less concentrated than higher-order norms.*

Fractional norms are not always less concentrated

The proof is given in Section 5.3. There are thus distributions $\mathcal{F}$ for which the 1-norm is more concentrated than the 2-norm; in general, a higher-order norm can be less concentrated than a fractional norm, by having a higher relative contrast or relative variance. Generalizing the result of Theorem 6 to arbitrary distributions is simply wrong. In conclusion, it may be useless to use fractional norms to fight concentration effects, when the component distributions are not known.

5.2.3 The *i.i.d.* hypothesis

All theorems presented in the Section 5.2 rely on the fact that data are supposed to be independent and identically distributed. What happens if it is not the case in practice (actually it will hardly ever be)? Are these assumptions really needed, or do they merely make the proofs easier, or even feasible? This section addresses these issues.

First the *identically distributed* assumption is considered. In the previous sections, we have supposed that all X_i are distributed according to the same distribution: $X_i \sim \mathcal{F}$. If variables X_i are not identically distributed, it means that each X_i is distributed according to some distribution noted $\mathcal{F}_i : \forall i : X_i \sim \mathcal{F}_i$.

Proposition 2. *If data are not identically distributed, then the conclusion of Theorem 2 still holds provided that data are normalised.*

Normalised data need not be identically distributed

The proof is given in Section 5.3. Normalizing means here to subtract the mean from the variables and divide them by their standard deviation so that $\forall i : \mathrm{E}(X_i) = 0$ and $\mathrm{Var}(X_i) = 1$. Normalizing data is often considered as important when using norms and distances, because it ensures that all variables X_i will have equal influence on the computation of the norm. If this it is not the case, the variables with largest variances will have the largest influence on the distance value, while the variables with low variance will have little or no influence on the computation of the norm.

Actually norms will concentrate for non-identically distributed data if they are normalised. But if the data are not normalised, some variables may have too little effect on the distance value.

If data are not independent, the intrinsic dimension matters

The *independent* assumption may not be encountered in practice. If X has components X_i that are not independent, it means that the joint distribution F of $\mathrm{X} = (X_1, \cdots, X_d)$ differs from the product of the marginal distributions $\mathcal{F}_i$. The existence of relations between the elements of X means that X lies on a (possibly nonlinear) sub-manifold of $\mathbb{R}^d$. If the sub-manifold is nonlinear, the Euclidean norm, for instance, measures the distances between data points using "short cuts", that is through a straight line in the space; this can be very different from a geodesic distance measured along the manifold [Tenenbaum *et al.* , 2000; Lee *et al.* , 2003]. However, the manifold can be represented in a vector space $\mathbb{R}^{d_{int}}$ whose dimension is the number of degrees of freedom of the manifold. Dimension d_{int} is called the *intrinsic dimension* and d the *embedding* dimension. Each realization of X in the embedding space may then be mapped to a unique realization in a d_{int}-dimensional projection space. One may thus expect that the asymptotic properties of the norms behave similarly in both spaces. Actually, the measure of intrinsic dimensionality ρ proposed by Chavez and Navarro [Chavez *et al.* , 2001] is precisely the inverse of the

square of the relative variance.

The fact that the concentration depends on the intrinsic dimension of the data is often admitted; it has however never been shown clearly, due to the lack of consensus on the actual definition of the intrinsic dimension.

Conjecture 1. If data are identically distributed but not independent, the value of the relative variance does not depend on the embedding dimension but rather on the intrinsic dimension. The dimension d in Theorem 2 and in Lemmas 1 and 2 must be understood as the intrinsic dimension of the data and not as its embedding dimension.

As a consequence, high-dimensional data that present a lot of correlation or dependencies between variables will be much less concentrated than if all variables are independent. The conclusion we can draw is that the concentration phenomenon depends on the intrinsic dimension of the data more than on the dimension of the embedding space.

5.3 Proofs and Illustrations

In this section, proofs are given for Lemmas 1 and 2, for Theorems 2 and 7 and for Propositions 1, 2. Motivations for Conjecture 1 are given.

5.3.1 Proof of Theorem 2

The proof of Theorem 2 is based on Lemmas 1 and 2. Therefore, the proofs of the lemmas are given first.

Proof of Lemma 1

Lemma 1 (recalled) *Let* $\mathrm{X} = (X_1, \cdots, X_d)$ *be a random vector with i.i.d. components :* $X_i \sim \mathcal{F}$*. Then,*

$$\lim_{d \to \infty} \frac{\mathrm{E}\left(\|\mathrm{X}\|_p\right)}{d^{1/p}} = c \tag{5.11}$$

where c *is a constant independent from the dimension* d *but related to* p *and to the distribution* $\mathcal{F}$*.*

The proof requires two steps. First it will be shown that under the assumptions of Lemma 1, we have

$$\mathbf{P}\left[\lim_{d\to\infty}\frac{\|\mathrm{X}\|_p}{d^{1/p}} \quad = \quad c\right] = 1, \tag{5.12}$$

where c is a constant independent from the dimension d of X (Step 1). Then this result will be extended to the expectation of the norm (Step 2).

Step 1
Let $S_i = |X_i|^p$ for $i = 1, \ldots, d$. The S_i are independent and identically distributed too ; let $\mu_{\mathcal{F},p}$ be their expectation.
The Strong Law of Large Numbers (SLLN) allows us to write

$$\mathbf{P}\left[\lim_{d\to\infty}\frac{1}{d}\cdot\sum_{i=1}^{d}S_i = \mu_{\mathcal{F},p}\right] = 1.$$

Then

$$\mathbf{P}\left[\lim_{d\to\infty}\left(\frac{1}{d}\cdot\sum_{i=1}^{d}S_i\right)^{1/p} = {\mu_{\mathcal{F},p}}^{1/p}\right] = 1$$

or, by definition of S_i,

$$\mathbf{P}\left[\lim_{d\to\infty}\frac{1}{d^{1/p}}\cdot\left(\sum_{i=1}^{d}|X_i|^p\right)^{1/p} = {\mu_{\mathcal{F},p}}^{1/p}\right] = 1.$$

This is nothing else than

$$\mathbf{P}\left[\lim_{d\to\infty}\frac{\|\mathrm{X}\|_p}{d^{1/p}} = {\mu_{\mathcal{F},p}}^{1/p}\right] = 1$$

which concludes the proof with $c = {\mu_{\mathcal{F},p}}^{1/p}$.
Step 2
By (5.12), for any realization ξ of X except some $\xi \in \Omega$ a subset of $\mathbb{R}^d$ with measure 0, we have

$$\lim_{d\to\infty}\frac{\|\xi\|_p}{d^{1/p}} = {\mu_{\mathcal{F},p}}^{1/p}.$$

Thus

$$\int_{\mathbb{R}^d\setminus\Omega}\mathcal{F}(\xi)\cdot\lim_{d\to\infty}\frac{\|\xi\|_p}{d^{1/p}}\,\mathrm{d}\xi = \int_{\mathbb{R}^d\setminus\Omega}\mathcal{F}(\xi)\cdot{\mu_{\mathcal{F},p}}^{1/p}\,\mathrm{d}\xi = {\mu_{\mathcal{F},p}}^{1/p}, \tag{5.13}$$

the last equality coming from the fact that ${\mu_{\mathcal{F},p}}^{1/p}$ is bounded. In addittion, we have

$$\int_{\mathbb{R}^d \setminus \Omega} \mathcal{F}(\xi) \cdot \lim_{d\to\infty} \frac{\|\xi\|_p}{d^{1/p}} \, \mathrm{d}\xi = \int_{\mathbb{R}^d} \mathcal{F}(\xi) \cdot \lim_{d\to\infty} \frac{\|\xi\|_p}{d^{1/p}} \, \mathrm{d}\xi = \lim_{d\to\infty} \int_{\mathbb{R}^d} \mathcal{F}(\xi) \cdot \frac{\|\xi\|_p}{d^{1/p}} \, \mathrm{d}\xi \tag{5.14}$$

since Ω is of measure 0. Combining (5.13) and (5.14) gives

$$\lim_{d\to\infty} \int_{\mathbb{R}^d} \frac{\|\xi\|_p}{d^{1/p}} \, \mathrm{d}\xi = \lim_{d\to\infty} \frac{\mathrm{E}\left(\|\mathrm{X}\|_p\right)}{d^{1/p}} = {\mu_{\mathcal{F},p}}^{1/p} \tag{5.15}$$

which concludes the proof since the right-hand side of (5.15) is independent from the dimension d.

Proof of Lemma 2

Lemma 2 (recalled) *Let* $\mathrm{X} = (X_1, \cdots, X_d)$ *be a random vector with i.i.d. components :* $X_i \sim \mathcal{F}$*. Then,*

$$\lim_{d\to\infty} \frac{\mathrm{Var}\left(\|\mathrm{X}\|_p\right)}{d^{\frac{2}{p}-1}} = c', \tag{5.16}$$

where c' *is a constant independent from the dimension* d *but related to* p *and to the distribution* $\mathcal{F}$*.*

Let $\mu_{\mathcal{F},p}$ be the expectation of variable $|X_i|^p$ and $\sigma_{\mathcal{F},p}$ its variance. For random vectors X of dimension d and a given p-norm, we have

$$\begin{aligned} \frac{Var\|X\|_p}{d^{2/p-1}} &= \frac{\mathrm{E}\left[\left(\|X\|_p - E\|X\|_p\right)^2\right]}{d^{2/p-1}} \\ &= \mathrm{E}\left[\left(\frac{\|X\|_p - E\|X\|_p}{d^{1/p-1/2}}\right)^2\right] \end{aligned}$$

Since

$$\|X\|_p - E\|X\|_p = \frac{\|X\|_p^p - \left(E\|X\|_p\right)^p}{\sum_{r=0}^{p-1} \|X\|_p^{p-r-1} \cdot \left(E\|X\|_p\right)^r}$$

we have

$$\frac{\|X\|_p - E\|X\|_p}{d^{1/p-1/2}} = \frac{\left(\|X\|_p^p - \left(E\|X\|_p\right)^p\right)/\sqrt{d}}{\sum_{r=0}^{p-1} \left(\frac{\|X\|_p}{d^{1/p}}\right)^{p-r-1} \cdot \left(\frac{E\|X\|_p}{d^{1/p}}\right)^r}.$$

Therefore,

$$\frac{Var\|X\|_p}{d^{2/p-1}} = \mathrm{E}\left[\left(\frac{(\|X\|_p^p - (E\|X\|_p)^p)/\sqrt{d}}{\sum_{r=0}^{p-1}\left(\frac{(\|X\|_p}{d^{1/p}}\right)^{p-r-1}\cdot\left(\frac{E\|X\|_p}{d^{1/p}}\right)^r}\right)^2\right].$$

Using the theorem by Lebesgue and Fatou about the convergence of integrable functions, we can swap the limit and the expectation operators :

$$\lim_{d\to\infty}\frac{Var\|X\|_p}{d^{2/p-1}} = \mathrm{E}\left[\lim_{d\to\infty}\frac{\left((\|X\|_p^p - (E\|X\|_p)^p)/\sqrt{d}\right)^2}{\left(\sum_{r=0}^{p-1}\left(\frac{(\|X\|_p}{d^{1/p}}\right)^{p-r-1}\cdot\left(\frac{E\|X\|_p}{d^{1/p}}\right)^r\right)^2}\right] \quad (5.17)$$

$$= \mathrm{E}\left[\frac{\lim_{d\to\infty}\left((\|X\|_p^p - (E\|X\|_p)^p)/\sqrt{d}\right)^2}{\lim_{d\to\infty}\left(\sum_{r=0}^{p-1}\left(\frac{(\|X\|_p}{d^{1/p}}\right)^{p-r-1}\cdot\left(\frac{E\|X\|_p}{d^{1/p}}\right)^r\right)^2}\right] \quad (5.18)$$

The transition from (5.17) to (5.18) is allowed since the denominator of (5.18) does not tend towards zero.

Because of Lemma 1, we know that $\frac{\mathrm{E}\|X\|_p}{d^{1/p}}$ tends towards a constant as d increases. Furthermore, we have seen that *almost surely*, i.e. with probability 1, $\frac{\|X\|_p}{d^{1/p}}$ also tends towards a constant.

Therefore, the denominator of (5.18) almost surely tends towards a constant as the dimension grows :

$$\mathbf{P}\left[\lim_{d\to\infty}\sum_{r=0}^{p-1}\left(\frac{\|X\|_p}{d^{1/p}}\right)^{p-r-1}\cdot\left(\frac{E\|X\|_p}{d^{1/p}}\right)^r = p\cdot\mu_{\mathcal{F},p}^{(p-1)/p}\right] = 1 \quad (5.19)$$

which means that, with probability 1 :

$$\lim_{d\to\infty}\frac{Var\|X\|_p}{d^{2/p-1}} = \frac{\mathrm{E}\left[\lim_{d\to\infty}\left((\|X\|_p^p - (E\|X\|_p)^p)/\sqrt{d}\right)^2\right]}{\left(p\cdot\mu_{\mathcal{F},p}^{(p-1)/p}\right)^2}$$

If we focus on the numerator of (5.18) now, we notice that

$$\begin{aligned}\|X\|_p^p - (E\|X\|_p)^p &= \sum_{i=1}^{d} |X_i|^p - (E\|X\|_p)^p \\ &= \sum_{i=1}^{d} \left(|X_i|^p - \frac{(E\|X\|_p)^p}{d} \right)\end{aligned}$$

Using the result from Lemma 1, we write

$$\lim_{d\to\infty} \left(\|X\|_p^p - (E\|X\|_p)^p \right) = \lim_{d\to\infty} \sum_{i=1}^{d} \left(|X_i|^p - \mu_{\mathcal{F},p} \right);$$

the numerator of (5.18) can now be written

$$\mathrm{E}\left[\left(\sum_{i=1}^{d} \left(|X_i|^p - \mu_{\mathcal{F},p} \right) \right)^2 / d \right].$$

Since

$$\begin{aligned}\mathrm{E}\left[\sum_{i=1}^{d} \left(|X_i|^p - \mu_{\mathcal{F},p} \right) \right] &= \sum_{i=1}^{d} \mathrm{E}\left[\left(|X_i|^p - \mu_{\mathcal{F},p} \right) \right] \\ &= \sum_{i=1}^{d} \left(\mathrm{E}\left[|X_i|^p \right] - \mu_{\mathcal{F},p} \right) \\ &= \sum_{i=1}^{d} \left(\mu_{\mathcal{F},p} - \mu_{\mathcal{F},p} \right) \\ &= 0,\end{aligned}$$

we have

$$\mathrm{E}\left[\left(\sum_{i=1}^{d} \left(|X_i|^p - \mu_{\mathcal{F},p} \right) \right)^2 \right] = \mathrm{Var}\left(\sum_{i=1}^{d} \left(|X_i|^p - \mu_{\mathcal{F},p} \right) \right).$$

However,

$$\begin{aligned}\mathrm{Var}\left(\sum_{i=1}^{d} \left(|X_i|^p - \mu_{\mathcal{F},p} \right) \right) &= \sum_{i=1}^{d} \mathrm{Var}\left(|X_i|^p - \mu_{\mathcal{F},p} \right) \\ &= \sum_{i=1}^{d} \mathrm{Var}\left(|X_i|^p \right) \\ &= d \cdot \sigma_{\mathcal{F},p}^2,\end{aligned}$$

leading to the conclusion that

$$\mathrm{E}\left[\left(\sum_{i=1}^{d}\left(|X_i|^p-\mu_{\mathcal{F},p}\right)\right)^2/d\right]=\sigma^2_{\mathcal{F},p}. \tag{5.20}$$

Using expressions (5.20) for the numerator and (5.19) for the denominator, it results that :

$$\mathbf{P}\left[\lim_{d\to\infty}\frac{Var\|X\|_p}{d^{2/p-1}}=\frac{\sigma^2_{\mathcal{F},p}}{(p\cdot\mu^{(p-1)/p}_{\mathcal{F},p})^2}\right]=1. \tag{5.21}$$

The same arguments as in **Step 2** from the proof of Lemma 1 can be used to get

$$\lim_{d\to\infty}\frac{Var\|X\|_p}{d^{2/p-1}}=\frac{\sigma^2_{\mathcal{F},p}}{(p\cdot\mu^{(p-1)/p}_{\mathcal{F},p})^2}. \tag{5.22}$$

which proves the Lemma since the right-hand side of (5.22) is independent from the dimension d.

Proof of Theorem 2

Lemmas 1 and 2 are used to prove Theorem 2:

Theorem 2 *(recalled from Chapter 3) Let* $\mathrm{X}=(X_1,\cdots,X_d)$ *be a random vector with i.i.d. components :* $X_i\sim\mathcal{F}$*. Then,*

$$\lim_{d\to\infty}\frac{\sqrt{\mathrm{Var}\left(\|\mathrm{X}\|_p\right)}}{\mathrm{E}\left(\|\mathrm{X}\|_p\right)}=0;$$

i.e. the relative variance decreases towards zero when the dimension grows.

Writing

$$\frac{\sqrt{\mathrm{Var}\|X\|_p}}{\mathrm{E}\|X\|_p}=\frac{\frac{\sqrt{\mathrm{Var}\|X\|_p}}{d^{1/p-1/2}}}{\frac{\mathrm{E}\|X\|_p}{d^{1/p}}}\cdot d^{-1/2}$$

and taking the limit

$$\lim_{d\to\infty}\frac{\sqrt{\mathrm{Var}\|X\|_p}}{\mathrm{E}\|X\|_p}=\lim_{d\to\infty}\frac{\frac{\sqrt{\mathrm{Var}\|X\|_p}}{d^{1/p-1/2}}}{\frac{\mathrm{E}\|X\|_p}{d^{1/p}}}\cdot d^{-1/2},$$

we have by Lemma 1 and Lemma 2

$$\lim_{d\to\infty} \frac{\sqrt{\mathrm{Var}\|X\|_p}}{\mathrm{E}\|X\|_p} = \frac{\sqrt{c'}}{c} \cdot \lim_{d\to\infty} d^{-1/2} = 0.$$

5.3.2 Proof of Theorem 7

Similarly to Theorem 2, the proof of Theorem 7 is based on Lemmas 1 and 2.

Theorem 7 (recalled) *Let* $\mathrm{X} = (X_1, \cdots, X_d)$ *be a random vector with i.i.d. components :* $X_i \sim \mathcal{F}$*. Then,*

$$\lim_{d\to\infty} \sqrt{d} \cdot \frac{\sqrt{\mathrm{Var}\left(\|\mathrm{X}\|_p\right)}}{\mathrm{E}\left(\|\mathrm{X}\|_p\right)} = \frac{1}{p}\frac{\sigma_{\mathcal{F},p}}{\mu_{\mathcal{F},p}}$$

where $\mu_{\mathcal{F},p} = \mathrm{E}\left(|X_i|^p\right)$ *and* $\sigma^2_{\mathcal{F},p} = \mathrm{Var}\left(|X_i|^p\right)$.

The theorem is proven as follows.
From Lemmas 1 and 2,

$$\lim_{d\to\infty} \sqrt{d} \cdot \frac{\sqrt{\mathrm{Var}\|X\|_p}}{\mathrm{E}\|X\|_p} = \lim_{d\to\infty} \frac{\frac{\sqrt{\mathrm{Var}\|X\|_p}}{d^{1/p-1/2}}}{\frac{\mathrm{E}\|X\|_p}{d^{1/p}}} = \frac{\sqrt{c'}}{c}. \tag{5.23}$$

Using the values of c and c' respectively from equations (5.15) and (5.22), we have

$$\frac{\sqrt{c'}}{c} = \frac{\sqrt{\frac{\sigma_{\mathcal{F},p}}{\left(p\cdot\mu_{\mathcal{F},p}^{\frac{p-1}{p}}\right)^2}}}{\mu_{\mathcal{F},p}^{\frac{1}{p}}} = \frac{1}{p} \cdot \frac{\sigma_{\mathcal{F},p}}{\mu_{\mathcal{F},p}} \tag{5.24}$$

5.3.3 Proof of Proposition 1

Proposition 1 (recalled) *The relative variance* (5.10) *is (a) a strictly decreasing function of* p *when the variables are distributed according to a uniform distribution* $\mathcal{F}$ *over the interval* $[0, 1]$*, but (b) increases locally with* p *for other distributions* $\mathcal{F}$*. Fractional norms are not always less concentrated than higher-order norms.*

Proposition 1 contains two claims ; the proofs are respectively given in Part (a) and Part (b).

Part (a)
If $\mathcal{F}$ is the uniform distribution over the interval $[0, 1]$, $\mu_{\mathcal{F},p}$ is given by :

$$\mu_{\mathcal{F},p} = \frac{1}{p+1}.$$

Since $\sigma^2_{\mathcal{F},p} = \mu_{\mathcal{F},2p} - {\mu_{\mathcal{F},p}}^2$, we have

$$\sigma_{\mathcal{F},p} = \frac{p}{p+1} \cdot \sqrt{\frac{1}{2p+1}}.$$

Therefore,

$$\frac{1}{p} \cdot \frac{\sigma_{\mathcal{F},p}}{\mu_{\mathcal{F},p}} = \sqrt{\frac{1}{2p+1}}. \tag{5.25}$$

We can conclude that, under the uniform distribution and large dimension d hypotheses, the relative variance decreases with p. The concentration of the norm thus increases as p grows.

Part (b)
A counter example is provided to prove assertion (b) of the proposition. Let us consider a situation where data are dispatched into two Gaussian clusters with variance $\sigma^2 = 1$ and respectively centered on 1 and -1. The marginal distribution of each X_i is then

$$\mathcal{F}^*(\xi) = \frac{1}{cst} \cdot \left(e^{-\left(\frac{\xi-1}{\sigma}\right)^2} + e^{-\left(\frac{\xi+1}{\sigma}\right)^2} \right) \tag{5.26}$$

In this example, the relative variance is higher for higher-order norms than for fractional norms as illustrated on Figure 5.1; consequently, fractional norms are more concentrated than higher-order norms with values of $p \in [8, 30]$.

The next examples are taken from real datasets used by Aggarwal in [Aggarwal *et al.* , 2001] : the Segmentation dataset and the Wisconsin Diagnostic Breast Cancer (WDBC) dataset from the UCI Machine Learning Repository [D.J. Newman & Merz, 1998]. Figure 5.2 shows a plot of the relative contrast as a function of p. Between $p = 1$ and $p = 2$, the relative contrast is actually increasing. This is a real example where the Euclidean norm is actually less concentrated than the 1-norm. The second example (Figure 5.3) is even more interesting. Here, the relative contrast is consistently better for higher-order norms than for fractional norms.

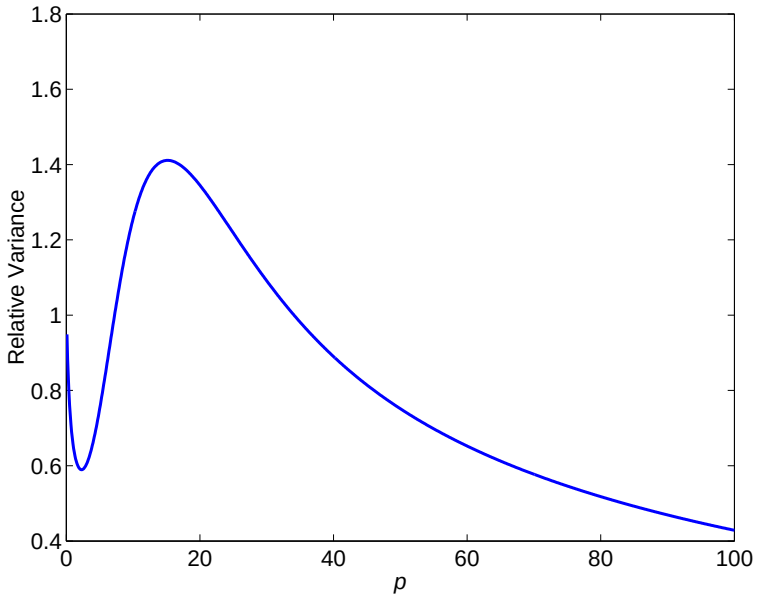


Figure 5.1: Relative variance for data distributed as $\mathcal{F}^*$, as a function of p. We can see that a maximum is obtained for a rather large value of p, far from 1. This shows that there exist distribution for which the least concentrated situation do not correspond to fractional norms.

5.3.4 Proofs for Propositions 2

The proof of Theorem 2 was built using the Law of Large Numbers. Although this Law is often stated with *i.i.d.* hypothesis, the *identically distributed* one is sufficient but not necessary. Actually, if data are normalised, the assumption of identical distributions is not necessary.

Proposition 2 (recalled) *If data are not identically distributed, then the conclusion of Theorem 2 still holds provided that data are normalised.*

It has been shown that a less restrictive sufficient condition for the Law of Large Numbers to hold is that the set of X_i must be

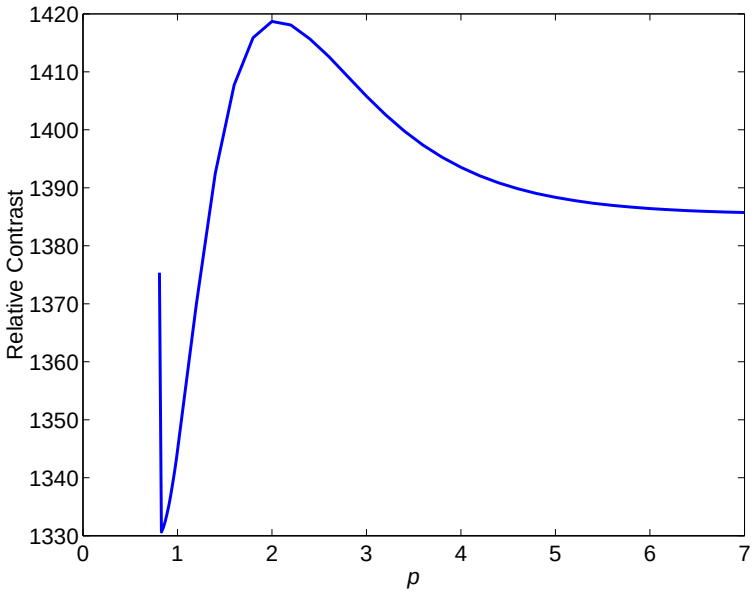


Figure 5.2: Relative contrast for segmentation dataset. The relative contrast is higher for higer-order norms that for the fractional norms.

uniformly integrable [Landers & Rogge, 1987]. This means that for any $\epsilon > 0$ there exists some $K < \infty$ such that for every X_i,

$$\int_{|X_i| \geq K} \mathcal{F}_i(\xi)|\xi| d\xi < \epsilon. \tag{5.27}$$

Note that (5.27) is actually the expectation of $|X_i|$ restricted to the values $|X_i| \geq K$. It is equivalent to [Landers & Rogge, 1987]

$$\exists k > 1 : \sup_i \mathrm{E}\left(|X_i|^k\right) < \infty. \tag{5.28}$$

If the data are normalised, $\mathrm{E}(X_i) = 0$ and $\mathrm{Var}(X_i) = 1$ for all i. We then have

$$1 = \mathrm{Var}(X_i) = \mathrm{E}\left(X_i^2\right) - \mathrm{E}(X_i)^2 = \mathrm{E}\left(X_i^2\right) = \mathrm{E}\left(|X_i|^2\right)$$

Taking $k = 2$ in (5.28) leads to the conclusion that if the data are centred and reduced, as often done in practice, then they

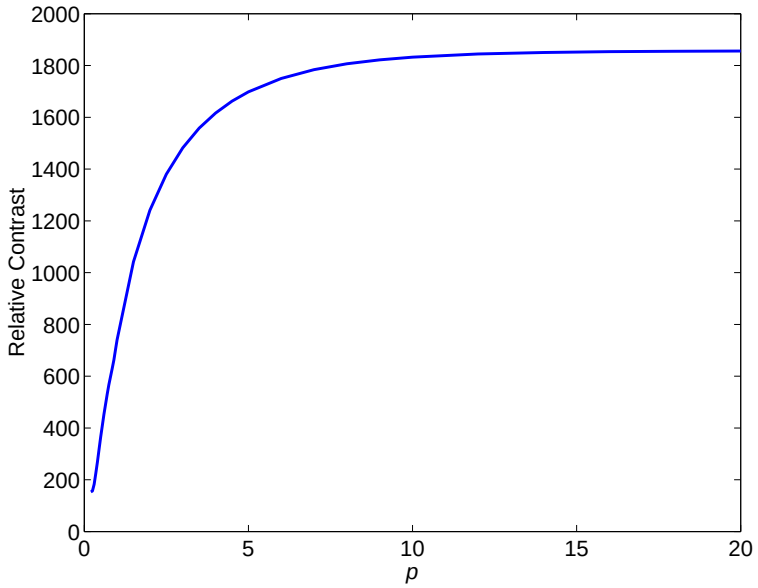


Figure 5.3: Relative contrast for WDBC dataset. The relative contrast is higher for higher-order norms that for the fractional norms.

are uniformly integrable. In this case, the law of large numbers still holds without the assumption of equal distributions. Therefore, all subsequent results are valid provided the data are normalised; this proves Proposition 2.

5.3.5 Motivations for Conjecture 1

Conjecture 1 (recalled) *If data are identically distributed but not independent, the value of the relative variance does not depend on the embedding dimension but rather on the intrinsic dimension. The dimension d in Theorem 2 and in Lemmas 1 and 2 must be understood as the intrinsic dimension of the data and not as its embedding dimension.*

To motivate Conjecture 1, we will give examples of datasets with identically distributed variables, having the same embedding dimensionality but a different intrinsic dimensionality;

we will then observe that the respective relative variances estimated on both datasets are different.

Let us begin with a simple example. Suppose that

$$\mathrm{X} = (X_1, \ldots, X_{d_{int}}, 0, \ldots, 0) \in \mathbb{R}^d.$$

Then, it is obvious that $\|\mathrm{X}\| = \|(X_1, \ldots, X_{d_{int}})\|$ and thus that the norm of X is distributed as the norm of the d_{int}-dimensional vector $(X_1, \ldots, X_{d_{int}})$; the concentration of X is then equal to the concentration of $(X_1, \ldots, X_{d_{int}})$ which is of dimension $d_{int} < d$.

As a second example, let us take

$$\mathrm{X} = (X_1, X_1, X_2, X_2, \cdots, X_{d_{int}}, X_{d_{int}}) \in \mathbb{R}^d$$

with $d = 2d_{int}$. A simple derivation shows that

$$\frac{\sqrt{\mathrm{Var}\|\mathrm{X}\|}}{\mathrm{E}\|\mathrm{X}\|} = \frac{\sqrt{\mathrm{Var}\|(X_1, X_2, ..., X_{d_{int}})\|}}{\mathrm{E}\|(X_1, X2, ..., X_{d_{int}})\|}$$

which indicates that the concentration of X is the same as the concentration of $(X_1, X2, ..., X_{d_{int}})$.

A further illustration comes from the study of real datasets. We will compare the concentration of norms in datasets with same embedding dimensions but different intrinsic dimensions.

Suppose we have some data described in a d-dimensional space, for which we know that the intrinsic dimension d_{int} is lower. Such situation occurs for example when the variables are significantly correlated. We will see that for such data, the value of the relative variance is much more similar to the one of a d_{int}-dimensional dataset with independent variables than of a d-dimensional dataset with independent variables. In that purpose, we will ensure that marginal distributions, i.e. is the distribution of each variable taken separately, are identical in both datasets.

Suppose we have $\mathbf{x} = \left\{x^{(j)}\right\}_{j=1}^{N}$ a sample drawn from $\mathrm{X} = (X_1, \cdots, X_d) \sim F(\xi_1, \xi_2, \ldots, \xi_d)$ where F is the multivariate probability density function of X. The marginal distributions of X_i are

$$\mathcal{F}_i(\xi_i) = \iint F(\xi_1, \dots \xi_i, \dots, \xi_d)\ \mathrm{d}\xi_1 \dots\ \mathrm{d}\xi_{i-1}\ \mathrm{d}\xi_{i+1} \dots\ \mathrm{d}\xi_d$$

To produce a dataset $\mathbf{x}'$ which is marginally identically distributed as $\mathbf{x}$, we propose the following. If we consider a matrix where each row corresponds to a data element $x^{(j)}$ and each column to a variable X_i, the values in each column are randomly permuted. As a consequence, the marginal distributions of each variable will not change. By contrast, all relationships between variables are destroyed in the process. Therefore, we obtain a sample $\mathbf{x}'$ which is marginally distributed like $\mathbf{x}$ but for which the components are independent; the intrinsic dimension of $\mathbf{x}'$ is thus equal to its embedding dimension.

Let us denote $\widehat{RV_{\mathbf{x},2}}$ the estimation of $RV_{\mathcal{F},p}$ with the Euclidean norm given the dataset $\mathbf{x}$ with high embedding dimension and low intrinsic dimension. We will compare this value to the value of the relative variance of $\mathbf{x}'$ ($\widehat{RV_{\mathbf{x}',2}}$) that has a high intrinsic dimension. Moreover, we will compare those values to the value of the relative variance for a dataset made of a small number (typically 20 times lower than the original number of variables) of low-correlated variables from $\mathbf{x}$ ($\widehat{RV_{\mathbf{x}s,2}}$) (small embedding dimension). The relative variance of a uniformly distributed synthetic dataset of dimensionality d : ($\widehat{RV_{rand,2}}$) (high intrinsic dimension) is also computed. It is expected that the relative variances for datasets with low intrinsic dimension will be similar while being much lower than the relative variances of the datasets with high intrinsic dimension.

The relative variance for a dataset $\mathbf{x} = \left\{x^{(j)}\right\}_{j=1}^{N}$ is estimated as follow :

$$\widehat{RV_{\mathbf{x},2}} = \frac{\sqrt{\sum_{j=1}^{n}\left(\|x^{(j)}\|_2 - \sum_{j=1}^{n}\|x^{(j)}\|_2\right)^2}}{\sum_{j=1}^{n}\|x^{(j)}\|_2}$$

The datasets mentioned in Table 5.1 are high-dimensional data coming from various domains. The first two datasets are Near Infrared spectra of APPLE and meat (TECATOR) respectively[1], the next two datasets are recorded sounds of 'yes' and 'no' (YESNO), and of 'boat' and goat' (BOATGOAT) [Biau *et al.* ,

[1] available from `http://www.ucl.ac.be/mlg`

2005] [2]. The IONOSPHERE and the MUSK dataset come from the UCI machine learning repository[3]. The dimensionality goes from 34 to more than 8000.

As expected, it is seen in Table 5.1 that the relative variance of the original dataset (column 1) is very similar to the relative variance of a subset of its variables (column 3). In contrast, the relative variance for the crafted dataset $\mathbf{x}'$ (column 2) is very similar to the relative variance of a random, uniformly distributed, dataset of the same dimension (column 4). It is clear that two datasets with identical distributions and embedding dimensionality can lead to different relative variances.

dataset	d	$\widehat{RV_{\mathbf{x},2}}$	$\widehat{RV_{\mathbf{x}',2}}$	$\widehat{RV_{\mathbf{x}_s,2}}$	$\widehat{RV_{rand,2}}$
APPLE	110	0.6061	0.0585	0.6331 (5)	0.0708
TECATOR	100	0.7616	0.0779	0.7741 (5)	0.0738
YESNO	8192	0.3459	0.0305	0.3656 (400)	0.0077
BOATGOAT	8192	0.1832	0.0171	0.2021(400)	0.0083
MUSK	167	0.1740	0.0437	0.2757 (8)	0.0561
IONOSPHERE	34	0.2855	0.0816	0.2170 (5)	0.1214

Table 5.1: Relative variance for several real datasets compared with relative variance of artificial *i.i.d.* datasets. The number between brackets is the number of variables used to build $\mathbf{x}_s$

Other experiments are presented in [Korn *et al.* , 2001]. The authors consider analytically the case of uniform distribution under the Euclidean and infinite norms. They propose nearest-neighbour search experiments and show that the average search time for a query depends on the *fractal* dimension of the data, and not on the embedding dimension. Computing the fractal dimensionality of a dataset is a way to estimate its intrinsic dimension. Therefore, though these results refer to retrieval times and not concentration directly, they are further arguments in favour of the thesis that the concentration depends more on

[2] available from `http://nathalie.vialaneix.free.fr/maths/article.php3?id_article=20`

[3] available from `http://www.ics.uci.edu/~mlearn/MLRepository.html`

the intrinsic dimension of the data than on the embedding dimension.

Chapter summary

This chapter fills some gaps in the existing literature concerning the mathematical properties and consequences on the analysis of real data, of the concentration of the norm phenomenon. Indeed, in the literature, the precise impact on the concentration of the parameter p of the norm was only known for the uniform distribution, the results depend on the number of data points, and the theorems consider independent and identically distributed data. This chapter shows that:

- all p-norms concentrate, even when an infinite number of data (i.e. a distribution) is considered, including when the distribution is unbounded;
- the value of p and the shape of the distribution of high-dimensional data influence the value of the relative variance, which is derived for any distribution and used as a measure of concentration;
- consequently, the exponent p of the norm can be adjusted to fight the concentration phenomenon;
- there exist distributions for which the relative contrast does not increase when the exponent p of the norm decreases;
- the identically distributed hypothesis behind the concentration of the norm theorems is not necessary as soon as the variables are normalised;
- the concentration phenomenon is more related to the intrinsic dimension of data than to their embedding dimension, which makes its consequences in practical situations less severe than mathematically expected;

Part II

Selecting the right features

Part Two focusses on feature selection. Indeed, the concentration phenomenon is not the only problem we are facing with high-dimensional spaces; reducing dimensionality is always important. Furthermore, searching for the optimal metric defined on irrelevant attributes is meaningless. Chapter 6 presents the state of the art. Chapter 7 focuses on feature selection for high-dimensional data, and proposes two ways of improving the efficiency and the stability of a mutual information-based forward subset search procedure.

”

It makes research more interesting to know there is no one universally best method. [...] We need to learn more about what works best where. But emphasis on theory often distracts us from doing good engineering and living with data.”

– L. Breiman

6

The problem of feature selection

This chapter is intended to be a general introduction to the problem of feature selection for regression. It focuses on methods that are suitable when no strong hypothesis (normality, linearity, etc.) can be made and on method that do not depend on a specific prediction model. The general recipe for feature selection is given first in an abstract manner, and then the ingredients at disposal are presented. While this chapter mainly reviews literature, it concludes with a proposition of a detailed recipe, general enough to cope with a large majority of problems.

Part One of this Thesis shed some light on how to choose a relevant measure of similarity for high-dimensional data. But we did not investigate yet which attributes (features) should be taken into account in the definition of the similarity measure. This chapter presents an overview of feature selection methods that can be used (i) to define a more relevant measure of similarity by excluding irrelevant features and (ii) to reduce dimensionality and hence trying to escape from the many other aspects of the curse of dimensionality. Section 6.1 first presents the general framework of feature selection while Section 6.2 and Section 6.3 respectively present feature relevance measures and feature subset search procedures. Section 6.4 proposes some more practical information about feature selection.

6.1 What is feature selection ?

This section presents and motivates the problem of feature selection and briefly reviews some of the fundamental ideas behind feature selection methods.

6.1.1 Problem statement

We consider a random vector of input features $\mathrm{X} = (X_1, \ldots, X_d)$ and an output random response variable Y. The aim is to build

a function (model) f that approximates the mapping between X and Y, i.e. such that

$$\mathrm{Y} = f(\mathrm{X}) + \epsilon, \tag{6.1}$$

where ϵ is a random prediction or classification error. Feature selection consists in finding a subset of the X_i that has maximal predictive power.

Feature selection pointless?

Theoretically, feature selection is useless for prediction: the more information available, the easier is the prediction. The model simply has to ignore the features that are not relevant for the prediction. For instance, a `C4.5` classification tree model would simply never use a useless feature as split point [Quinlan, 1993], a Multi-Layer Perceptron (`MLP`) [Fausett, 1994] model would set to zero the weights (coefficients) corresponding to useless features, etc. Nevertheless, dimensionality reduction is very important in practice, for two reasons.

First, not all families of models are able to distinguish between relevant and irrelevant features. A `RBFN` or a `SVM` with Gaussian kernels, for instance, consider each feature to have equal importance in the computation of the predicted value.

Feature selection crucial!

Second, the model f is built in practice from finite size samples; input-output pairs $\left(\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\right)$, $1 \leq j \leq N$ are observed, drawn from the joint probability density of X and Y. Because of the limited size of the sample, having a model with too many inputs, and hence too many parameters to set, will often lead to overfitting and poor generalization abilities, induce large computation times, or even prevent the algorithms from converging.

Definition

The *feature selection* thus consists in choosing, among the coordinates of X, those that are the most useful to predict Y, or, in other words, such that the model has highest possible generalization ability.

6.1.2 A general framework for feature selection

Many different feature selection procedures can be designed. Nevertheless, all of them rely on the conjunction of two independent elements:

1. a *relevance criterion* to score a feature or a group of features according to its/their predictive power with respect to the response variable Y. The relevance criterion is often based either on a statistical measure or on (an estimation of) the prediction error of an actual model.
2. a *search procedure* for the optimal feature subset according to the chosen criterion. The search procedure consists itself of three components: an initialisation procedure, a subset exploration algorithm and a stopping criterion.

Any relevance criterion, provided that it applies to a group of features as well as to a single feature, can be used with any search procedure.

1. The relevance criterion

The optimal relevance criterion for a subset of features to predict a variable using a model chosen from a family of models (for instance `RBFN` or `SVM`) is obviously the generalization error of the best model built with those features. However, estimating this generalization error requires costly cross-validation or bootstrap procedures to choose the optimal meta-parameters of the model (number of neurons, regularization coefficient, etc.)

In order to avoid going through the whole model optimisation procedure, statistical measures of relevance can be considered, e.g. the correlation coefficient or the mutual information. They are often cheaper to evaluate, because they do not build an actual model. However, their reliability drops when the number of features becomes large because they become very difficult to estimate correctly.

2. Subset search

For data dimensionality d, there are 2^d possible subsets. Obviously, for large values of d, it can be prohibitive to estimate the relevance of all those subsets. Instead, simple ranking of the features according to their individual relevance can be considered, as well as all techniques developed for combinatorial optimisation.

The algorithms for combinatorial optimisation trade optimality of the result with computational burden. Two classes of algorithms can be distinguished: incremental algorithms that reduce complexity by selecting one feature at a time and never question the relevance of that feature again, and randomised algorithms that look for the optimal results among a well-

crafted subset of all the possible subsets.

Table 6.1 presents some of the most used relevance criteria and subset search procedures. Most of the elements in the left column can be associated with any element listed in the right column. Some important relevance criteria are detailed in Section 6.2 while the search methods are presented in Section 6.3.

Relevance criterion	**Subset search procedure**
Mean-difference test	Ranking
(Canonical) Correlation	Forward
Mutual information	Backward
Gamma test	Genetic algorithm
Cross-validation error	Simulated Annealing
Bootstrap error	Backtracking
...	...

Table 6.1: The two key elements to design a feature selection method. Pick a relevance criterion and a subset search procedure and combine them to select features.

6.2 Subset relevance assessment

This section presents some interesting relevance criteria. It first focusses on criteria reserved for classification and then presents two general-purpose ones.

Classification can be seen as the approximation of a binary function; hence methods used for regression can be used for classification by simply enhancing them with a threshold to binarise the output. However, for feature selection, many criteria have been proposed that are specific to classification. These criteria are not theoretically better than those presented in the remaining of the section; they are however easier to estimate and, consequently, may give better results on real, noisy, datasets. A rather complete list is given in Chapter 3 of [Guyon *et al.* , 2006].

One popular way to estimate the strength the interactions between two variables is the *correlation*. It is defined for random variables X and Y as

$$\rho(\mathrm{X},\mathrm{Y}) = \frac{\mathrm{E}\left(\mathrm{XY}\right) - \mathrm{E}\left(\mathrm{X}\right)\mathrm{E}\left(\mathrm{Y}\right)}{\sqrt{\mathrm{Var}\left(\mathrm{X}\right)\mathrm{Var}\left(\mathrm{Y}\right)}}. \tag{6.2}$$

The correlation however only measures how linear the relationship between two variables is.

The following two methods are said to be nonparametric in the sense that they make assumptions neither about the prior distribution of the variables nor about the shape of the mapping (if any exists) between the variables, contrarily to the correlation coefficient for instance. They can be used for classification as well as for regression.

6.2.1 Mutual information

The *mutual information* of two variables is the amount of information that one variable brings about the other, and *vice versa*. The notion of information is defined from the notion of uncertainty of the random variable.

Uncertainty

The uncertainty of a variable can be estimated through its entropy. The entropy of a real-valued random vector Y is a nonnegative value given by [Shannon & Weaver, 1949]

$$H(\mathrm{Y}) = -\int \mathcal{F}_{\mathrm{Y}}(\mathrm{y})\,\log\,\mathcal{F}_{\mathrm{Y}}(\mathrm{y})\,\mathrm{dy} \tag{6.3}$$

where $\mathcal{F}_Y$ is the (multivariate) probability density function of Y. The joint entropy of Y and X,

$$H(\mathrm{Y},\mathrm{X}) = -\int \mathcal{F}_{\mathrm{Y},\mathrm{X}}\left(\mathrm{y},\mathrm{x}\right)\,\log\,\mathcal{F}_{\mathrm{Y},\mathrm{X}}\left(\mathrm{y},\mathrm{x}\right)\,\mathrm{dydx}, \tag{6.4}$$

denotes the total amount of uncertainty about both variables. It is equivalent to the sum of each variable's entropy only if they are statistically independent.

Conditional entropy

The entropy of Y when the value of X is known is the *conditional entropy*:

$$H(\mathrm{Y}|\mathrm{X}) = H(\mathrm{Y},\mathrm{X}) - H(\mathrm{X}). \tag{6.5}$$

The difference between the entropy of Y and the entropy of Y conditioned on X is called the mutual information between Y and X:

$$I(\mathrm{Y},\mathrm{X}) = H(\mathrm{Y}) - H(\mathrm{Y}|\mathrm{X}). \tag{6.6}$$

The mutual information can be rewritten as the Kulback-Leibler divergence between the joint probability density function and the product of the marginal probability density functions:

$$I(\mathrm{Y},\mathrm{X}) = \int \mathcal{F}_{\mathrm{X},\mathrm{Y}}(\mathsf{x},\mathsf{y}) \log \frac{\mathcal{F}_{\mathrm{X},\mathrm{Y}}(\mathsf{x},\mathsf{y})}{\mathcal{F}_{\mathrm{X}}(\mathsf{x}) \cdot \mathcal{F}_{\mathrm{Y}}(\mathsf{y})} \mathrm{dxdy} \tag{6.7}$$

The mutual information therefore measures how different the actual joint density is from the density that would be observed if the variables were independent, as illustrated on Figure 6.1.

The mutual information is symmetric and always comprised between 0 and $\max\{H(\mathrm{Y}), H(\mathrm{X})\}$. It takes value zero if and only if the variables are independent. The mutual information is thus well suited to measure the relevance of X to predict the values of Y [Battiti, 1994]. A feature subset that has a high mutual information with the response value can thus be considered as useful for prediction.

Estimation of the mutual information

In practice, the mutual information has to be estimated from the dataset, as the exact probability density functions in the above equations are not known. The estimation of the multi-dimensional mutual information can be performed with multivariate density estimators (for a review see [Scott, 1992]) when the dimensionality does not exceed 10 [Bonnlander & Weigend, 1994; Kwak & Choi, 2002a], because of the empty space phenomenon explained in Chapter 2. It can also be approximated by using combinations of 2-dimensional [Battiti, 1994; Kwak & Choi, 2002b] or 3-dimensional [Fleuret, 2004] mutual information measurements. But it can also be estimated using nearest neighbours-based methods.

Kraskov's estimator

Kraskov's estimator is based on nearest neighbours in the joint space of attributes X and response values Y. It is defined, for k neighbours and a sample of size N as:

$$I(\mathrm{X},\mathrm{Y}) = \psi(k) - \frac{1}{k} - \frac{1}{N}\sum_{j=1}^{N}\left(\psi(n_{\mathrm{X}}(j)) + \psi(n_{\mathrm{Y}}(j))\right) + \psi(N)\,. \tag{6.8}$$

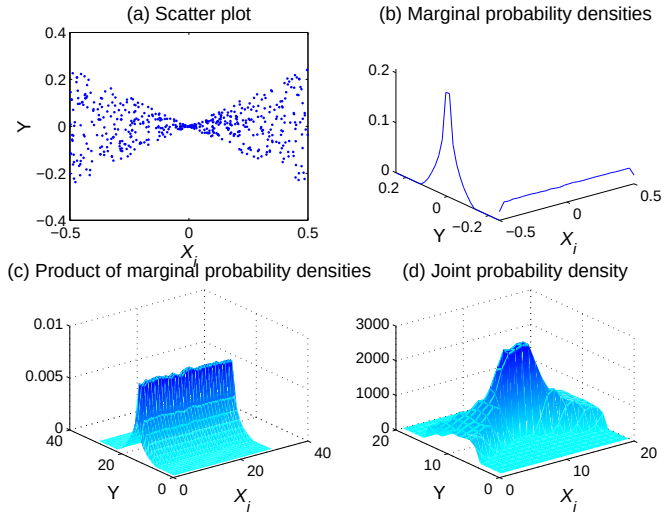


Figure 6.1: The intuition behind mutual information. X_i is a uniformly distributed random variable and Y is the response variable. (a) plots Y as a function of X_i. (b) shows the marginal probability densities of X_i and Y. (c) shows the product of the marginal densities, i.e. the joint probability density as if both variables were independent. (d) presents the actual joint probability density. We can see that (d) is different from (c), indicating that X_i and Y are not independent. The mutual information measure quantifies how dependent they are.

Here, $\psi(k)$ is the digamma function, satisfying $\psi(k+1) = \psi(k)+1/k$ and $\psi(1) = -0.577215...$ (the Euler-Mascheroni constant.) The values $n_{\mathrm{X}}(j)$ are defined, with respect to $\mathbf{x}^{(j)}$, the jth observation in the sample, as the number of other points whose distance to $\mathbf{x}^{(j)}$ is smaller than the distance, in the input space, to the projection of the kth neighbour of the pair $(\mathbf{x}^{(j)}, \mathbf{y}^{(j)})$ in the joint space of X and Y, onto the space of X. The value of $n_{\mathrm{Y}}(j)$ is found the same way in the space of Y. A more detailed description is available in [Kraskov *et al.* , 2004]. This estimator has been shown to be less sensitive to the empty space phenomenon than the other nonparametric methods [Kraskov

et al. , 2004; Rossi *et al.* , 2005], and should be recommended whenever the dimension of X exceeds 10.

6.2.2 Noise variance estimation: the Gamma test

First, it is important to note that the name "Gamma test" refers to a method for estimating the noise variance in a attribute-response values dataset. It is not a test in the statistical sense of the meaning. The relevance of a set of features $\mathrm{X} = (X_1, \ldots, X_d)$ is estimated by considering the noise of the mapping in the Euclidean space between the input features $(X_1, \ldots, X_d)$ and the output response value Y. The noise of the mapping can be estimated by using nearest neighbours techniques. Intuitively, those techniques measure how well data elements that are similar according to the features correspond to similar response values (See Figure 6.2.) A feature subset that violates this principle is a not a relevant feature subset for predicting the values of Y. This criterion is very similar to the leave-one-out error of a `1-NN` approximator.

Considering a dataset $\mathcal{X} = \left\{\mathbf{x}^{(j)}\right\}_{j=1}^{N}$ and $\mathcal{Y} = \left\{\mathbf{y}^{(j)}\right\}_{j=1}^{N}$ such that $\mathbf{y}^{(j)} = f\left(\mathbf{x}^{(j)}\right) + \epsilon$, the goal is to estimate the variance of ϵ, assuming that its mean is close to zero.

Estimated using nearest-neighbour estimators

The Gamma test estimates the limit, for distances in the input space decreasing to zero, of the average distance in the output space. When there exists a mapping between X and Y, similar inputs are mapped onto similar outputs, and hence, the distances in Y for small distances in X are smaller than if there is no mapping. Let us denote NN_j the index of the nearest neighbour of $\mathbf{x}^{(j)}$; then, we define γ as:

$$\gamma = \frac{1}{2N} \sum_{j=1}^{N} \left(\mathbf{y}^{(j)} - \mathbf{y}^{(NN_j)}\right)^2 . \tag{6.9}$$

This quantity is proven to converge to $\mathrm{Var}\,(\epsilon)$ as the number of samples goes to infinity. In practice, it is possible to approach $\mathrm{Var}\,(\epsilon)$ faster (i.e. with less observations) by considering more than one neighbour [Stefánsson *et al.* , 1997]. Let us denote ${}^{k}NN_j$ the kth neighbour of $\mathbf{x}^{(j)}$. We consider the following two quantities:

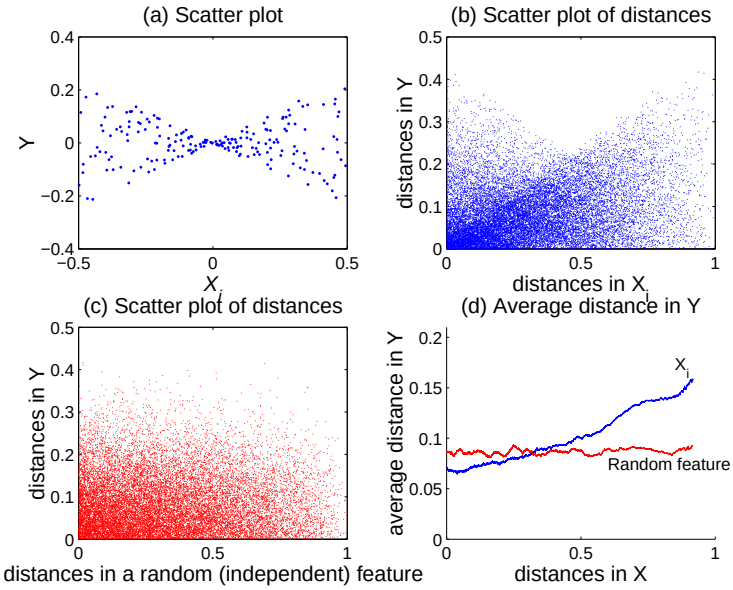


Figure 6.2: The intuition behind the Gamma test. X_i is a uniformly distributed random variable and Y is the response variable. (a) plots Y as a function of X_i. (b) shows the scatter plot of the distances in the space of Y as a function of the distances of the corresponding inputs in the space of X_i. (c) shows the same scatter plot if both variables were independent. (d) shows an averaged distance in the response variable space as a function of the distance in the input space. The Gamma test estimates the intersection of this function with the vertical axis.

$$\Delta(l) = \frac{1}{l} \cdot \sum_{k=1}^{l} \frac{1}{N} \sum_{j=1}^{N} \|\mathbf{x}^{(j)} - \mathbf{x}^{(^kNN_j)}\|^2 \tag{6.10}$$

and

$$\Gamma(l) = \frac{1}{l} \cdot \sum_{k=1}^{l} \frac{1}{2N} \sum_{j=1}^{N} \left(\mathbf{y}^{(j)} - \mathbf{y}^{(^kNN_j)}\right)^2 \tag{6.11}$$

Obviously, $\Delta(l)$ is the average mean square distance to the l-nearest neighbours. The $\Gamma(l)$ is an estimate of γ based on all l-nearest neighbours. Both are often highly correlated for small distances; the intercept of the regression line between

them has been found to be a better estimate of $\mathrm{Var}\,(\epsilon)$ than (6.9) [Stefánsson *et al.* , 1997]. Hence, the values of (6.10) and (6.11) are estimated for an increasing number l of neighbours, say from one to k, and the vector

$$\left(\begin{pmatrix} \Delta(1) & \Delta(2) & \cdots & \Delta(k) \\ 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} \Delta(1) & 1 \\ \Delta(2) & 1 \\ \vdots & \vdots \\ \Delta(k) & 1 \end{pmatrix} \right)^{-1} \begin{pmatrix} \Delta(1) & \Delta(2) & \cdots & \Delta(k) \\ 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} \Gamma(1) \\ \Gamma(2) \\ \vdots \\ \Gamma(k) \end{pmatrix} \tag{6.12}$$

is computed. It results in a two-dimensional column vector whose second entry is precisely the estimation of $\mathrm{Var}\,(\epsilon)$.

6.2.3 Generalization error of a model

Using an estimation of the generalization error of the (nonlinear) prediction model was advocated by Kohavi and John [John *et al.* , 1994]. The idea that the optimal feature subset is the one that allows building the best model is of course not new, especially for linear models. It was however seldom considered for nonlinear models due to huge computational load. Kohavi and John noticed that, since computers become more and more powerful, it is worth spending large amounts of time optimising nonlinear models to find the optimal feature subset. Selecting features in this way takes into account the possible bias of the model; it will reject features that are relevant but that the model could not use by lack of expression power for instance.

The best model is the model for which the structure (i.e. all meta parameters) are optimised with respect to its generalization error. The generalization error of a model is the average error the model would make on all possible data on which it could be tested.

Resampling methods

The estimation of the generalization error is done using resampling techniques such as the bootstrap and the cross-validation. The cross-validation estimate of the generalization error is computed by segmenting the dataset into K subsets. Each subset is iteratively used as validation set while the remaining of the dataset is used for building the model. Let us denote $E^{\mathcal{X}_v,\mathcal{Y}_v}_{\mathcal{X}_t,\mathcal{Y}_t}$ the total error that a model built on $(\mathcal{X}_t, \mathcal{Y}_t)$ does when it is applied to $(\mathcal{X}_v, \mathcal{Y}_v)$. The K-fold cross validation error is thus

$$\frac{1}{K} \cdot \sum_{l=1}^{K} E^{\mathcal{X}_l, \mathcal{Y}_l}_{\mathcal{X} \setminus \mathcal{X}_l, \mathcal{Y} \setminus \mathcal{Y}_t} \tag{6.13}$$

where $\mathcal{X}_l, \mathcal{Y}_l$ are the K partitions of the dataset. Each data is used once to test the model.

Note that a bootstrap methodology, based on random resampling with replacement, could also be used. The bootstrap estimation of the prediction error is made of two terms; the empirical error made on the learning set, and an *optimism* term that estimates difference between the empirical error and the true generalisation error. This method has some statistical benefits compared with cross-validation, namely a smaller variance [Efron & Tibshirani, 1993]. The bootstrap is however more computationally demanding, even if some tricks can be used to accelerate it for certain classes of models, by interpolating the optimism instead of estimating it directly for each different model hyper-parameter value [Lendasse *et al.* , 2005].

6.3 Optimal subset search

Once a feature relevance criterion has been chosen in order to score each feature or feature subset, a search procedure is needed to find the optimal feature subset according to that criterion. Searching for the optimal subset can be done in different ways, depending on the time that can be allocated to the feature selection procedure, from the least demanding simple ranking to the most often intractable exhaustive search. Many 'intelligent' methods have been devised in the Artificial Intelligence literature (see [Russell & Norvig., 2003] for a review and Figure 6.3 for an illustration of some of them.)

6.3.1 Ranking

The simplest procedure is to score all features individually and to rank them accordingly. Then, a threshold is set and all features that score more than the threshold are kept; the others are discarded.

This method is of course the simplest and fastest of all, performing in linear time with respect to the initial number of

features d, but at the cost of an over-simplified solution to the problem.

The most often used approach is to plot the graph of the ordered scores of the features and use one's experience to set the threshold.

The permutation test

However, two methods can be used to set the threshold automatically. The permutation test allows performing a statistical test on the significance of the score obtained by a feature. The permutation test considers the distribution of possible scores given the sample size and the marginal distributions of the feature and of the response value. Intuitively, the permutation test actually estimates how likely the given score is assuming that the features are independent. The permutation test will be explained in more details in Section 7.3 of the next chapter.

A gauge (probe) variable

Another method for setting a threshold automatically is the use of a "probe" variable [Stoppiglia *et al.* , 2003]. A new, independent, feature is created and added to the set of features. All features whose score is lower than the score of the probe are considered irrelevant.

6.3.2 Exhaustive search

While simple ranking is the cheapest (least time consuming) method, the exhaustive search is the most expensive. It consists in scoring each possible subset of features. Of course, the subset which is optimal according to the chosen criterion will be found, but at the cost of examining 2^d subsets.

Sometimes tractable

Exhaustive search is not tractable for large numbers of initial features, but for small numbers like 10, it can still be tractable (2^{10} is only 1024) provided that the relevance criterion is fast to compute.

The exhaustive approach will allow scoring every possible subset of features and ranking them accordingly. This way, it is possible to identify the most relevant subset, but also feature subsets with similar predictive power and possibly different features. This brings interesting information about the process being modelled [Krier *et al.* , 2006a].

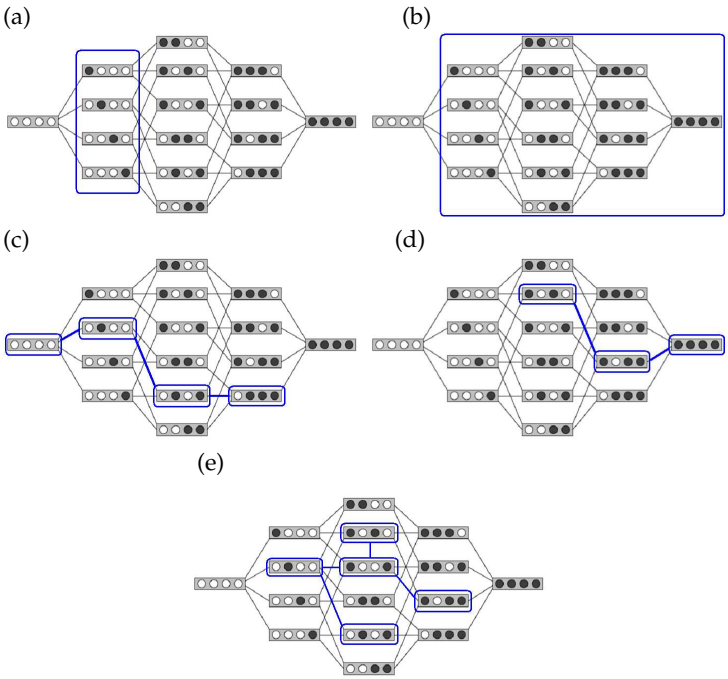


Figure 6.3: Several search algorithms for the optimal feature subset. There are four initial features; each possible subset is depicted as a rectangle. The ith circle represents the ith feature; a black (resp. white) circle means the feature does (resp. not) belong to the subset. The subsets that are tested by the method are identified with a rectangle. The methods are the simple ranking (a), the exhaustive search (b), the forward selection (c), the backward elimination (d) and a randomised algorithm (e).

6.3.3 Incremental

Incremental methods add or remove one feature at each iteration of the search algorithm. The procedure is halted when adding (or removing) any feature does not improve the criterion anymore, or when a pre-defined number of features have been selected.

Greedy algorithms

The *greedy methods* make final decisions about features at each iteration. Once a feature is chosen or discarded, it is never considered again. The forward and the backward algorithms will therefore perform at most $O\left(d^2\right)$ estimations of the criterion. The *forward feature selection* begins with an empty set of features and adds, at each iteration, the feature that has the most positive influence on the criterion. The backward feature elimination starts with the full set of features and removes, at each iteration, the feature that allows the best improvement of the criterion (or the least degradation) when it is not used. The backward procedure is more computationally demanding because it begins with the full set of features, and can even be impossible, but the forward search can be sensitive to the choice of the first feature [Benoudjit & Verleysen, 2003]. Both procedures can be merged into a forward-backward procedure by first performing a forward feature subset and then finishing by initialising a backward elimination with the subset found by the forward stage. Even if these procedures may reach a suboptimal solution, they often perform well in the context of feature selection [Deng & Moore, 1998].

Backtracking algorithms

Backtracking algorithms are also incremental. The *backtracking* methods are variants of the forward-backward procedure. They explore the space of possible subsets by performing a kind of forward feature selection but where a feature that has been selected at one iteration can be discarded at another. It is equivalent to a procedure where forward and backward phases are interleaved. The worst-case complexity of such algorithms is $O\left(2^d\right)$, except if the relevance criterion has some monotonicity properties [Land & Doig, 1960]. The Branch and Bound algorithm (see for instance [Hamamoto *et al.* , 1990]) is an example of backtracking procedure.

6.3.4 Randomised algorithms

Randomised methods were developed for non-convex optimisation. Genetic algorithms are based on the notion of biological evolution while simulated annealing takes its name from a procedure used in metallurgy to increase the size of crystals in a material and improve its quality.

Genetic algorithms

The *genetic algorithm* starts with a random set of initial subsets. It then extracts the most relevant ones and randomly performs

mutations and cross-overs. A mutation is a random deletion or addition of a feature from/to the subset. A cross-over is a procedure by which half (for example) of the features of a subset are exchanged with another. Once those mutations and cross-overs have occurred, the relevance criteria are evaluated for all subsets that were built and only the most relevant are kept. The process is then iterated [Yang & Honavar, 1998].

Simulated annealing

For *simulated annealing*, the process is initialised with a random subset. At each iteration, the 'neighbour' subsets, i.e. the subsets that have either one more, one less or one different feature are considered. The probability to move to another subset is controlled by the *temperature* T. If T is small, only neighbouring subsets that have better relevance criterion are considered, while if T is large, subsets with worse score are also considered to avoid local minima. The temperature decreases at each iteration, reducing to a forward selection procedure in the few last iterations [Stephen P. Brooks & King, 2003].

6.4 Discussion

The remaining part of this chapter is dedicated to more practical issues concerning feature selection, namely implementation details, things to keep in mind when interpreting the results, and a proposed methodology useful in many practical cases.

6.4.1 Implementation issues

Both the mutual information estimator and the Gamma test implementations are available respectively from Caltech[1] and from the Gamma Archive[2]. Both implementations are efficient, but sometimes at the cost of numerical instability for the mutual information package. For a model-based approach, any implementation of any model can be used, but of course, the faster the better.

The search procedures, by contrast, will most often have to be re-implemented according to the relevance criterion that

[1] `http://www.klab.caltech.edu/~kraskov/MILCA/`
[2] `http://users.cs.cf.ac.uk/Antonia.J.Jones/GammaArchive/IndexPage.htm`

is used. The forward-backward scheme is very simple to program. Generic versions of the randomised algorithms are nevertheless available in Matlab (Genetic Algorithm and Direct Search toolbox[3], and from user-provided packages[4]) and in Weka (Class GeneticSearch[5], class SimulatedAnnealing[6]). A genetic algorithm is furthermore available in R (Package genalg[7]).

6.4.2 Random thoughts

"Relevant" ?

What do we mean precisely by "relevant" ? See [Kohavi & John, 1997; Guyon & Elisseeff, 2003; Dash & Liu, 1997; Blum & Langley, 1997] for a total of nearly ten different definitions for the *relevance* of a feature subset with respect to a response value or class label. The definitions vary depending on whether a particular feature is relevant but not unique, etc. A feature can be useful for prediction and at the same time irrelevant for the application, think of a bias term for instance. Conversely, a feature relevant for the application can be useless for prediction if the prediction model is not able to exploit it.

Redundancy

Is redundancy always evil for prediction ? Surprisingly, the answer is no. Redundancy can be useful in different ways. First, redundant features can be averaged to filter out noise to a certain extent. Instead of choosing one out of two highly correlated features, it may be more interesting to build a feature that is the average of both, giving a chance for the measurement noises to cancel each other out. Second, two correlated features may carry information about the response value in their differences. It is especially interesting to keep this in mind in the case of functional data; focussing on the difference between two consecutive features can be seen as working with the derivative of the function. Finally, if several highly correlated features are selected together by a distance-based estimator (either for mutual information or noise variance), it can be seen as an sort of

[3] `http://www.mathworks.com/products/gads/`
[4] `http://www.mathtools.net/MATLAB/Genetic_algorithms/`
[5] `http://weka.sourceforge.net/doc/weka/classifiers/bayes/net/search/local/GeneticSearch.html`
[6] `http://weka.sourceforge.net/doc/weka/classifiers/bayes/net/search/local/SimulatedAnnealing.html`
[7] `http://cran.r-project.org/src/contrib/Descriptions/genalg.html`

"work around" found by he algorithm to actually weight the features in the definition of the distance, to "mimic" a metric of the form of (4.1).

Features useful with others

Can a feature be non relevant individually and still useful in conjunction with others ? The typical example is the XOR problem. Consider two features X_1 and X_2 normally distributed and a response variable such that $\mathrm{Y} = 1$ if X_1 and X_2 have the same sign and zero otherwise. The union of X_1 and X_2 allows a perfect prediction of Y, however, neither feature is able to bring the slightest piece of information about the value of Y alone. Another example is $\mathrm{Y} = \sin(2\pi(\mathrm{X}_1 + \mathrm{X}_2))$, with X_1 and X_2 drawn from a uniform distribution over $[0, 1]$. Both are needed to predict Y, but each of them is useless alone; whatever the value of X_1 is, it is impossible to know to which angle value $2\pi(\mathrm{X}_1 + \mathrm{X}_2)$ corresponds.

Discarded features recycled

What to do with features that are discarded ? Caruana has suggested to used them as extra outputs to increase the performances of the prediction model [Rich Caruana, 2003]. These extra outputs force the model to learn the mapping from the selected features to the discarded ones. Caruana has found that the model can benefit from this extra knowledge and reach better performances. Another interesting idea is to use the discarded features, especially those discarded because of redundancy, along with some other relevant features to build one or several other models and group them into an *ensemble* model [Günter & Bunke, 2002]. This ensemble model combes the results of several different models (built on different feature subsets) to provide a more robust prediction.

6.4.3 A proposed methodology

Given all possible combinations of relevance criteria and subset search procedures, ranging from the simplest to the most time-consuming, it clearly appears that the best feature selection methodology is a matter of trade-off between optimality of the result and computational time; it is furthermore highly dependent on the data.

Some recommendations

Although there exists no ideal procedure that would work in all situations, the following practical recommendations can be used.

1. Exploit any domain knowledge you have access to, to eliminate obviously useless features, especially if this renders the heuristic search tractable. If you are not an expert in the domain, ask one, either before selecting features to avoid processing obviously useless features. Removing two or three features a priori can make a huge difference for an exponential algorithm, but also for a cubic algorithm! You can also ask the expert after the selection process, to check whether the selected features make sense.

2. Perform simple ranking with the correlation coefficient (6.2). If there is a strong linear link between features and the response value, it is important to know it because nonlinear models are rarely good at modelling linear mappings, and will certainly not outperform a linear model in such a case.

3. Perform simple ranking first using mutual information to get baseline results. It will allow you to spot the most individually relevant features, and maybe to identify features that are completely independent from the response value. Excluding them might be considered only if it leads to a significant reduction in computation time. Building a prediction model with the first three up to the first ten, for instance, relevant features gives cheap baseline prediction results and provides with an rough idea of the computation time needed for estimating relevance. This is an important information that can help decide which options to consider.

4. If tractable, consider the exhaustive search of all pairs or triplets of features with a Gamma test and/or mutual information. See Section 7.4 for some insights about which to choose. This way, you avoid risks of ignoring important features that are useful together only. If it is not tractable, consider only pairs or triplets involving at least one individually relevant feature, or even only relevant features. The ideas is always to trade extension of the results vs. computational burden.

5. Initiate a forward strategy with the best subset from previous step. Starting from three features instead of one will make the procedure less sensitive to the initialisation.

6. End with a backward search using the actual prediction model, to remove any redundant features and further reduce the dimensionality. It is wise at this stage to use the prediction model because (i) probably few iterations will be performed and (ii) this focusses the selection procedure towards the capabilities/flaws of the prediction model.

7. If the forward-backward procedure gives results that are not much better than those obtained with the ranking approach, resort to a randomised algorithm like the genetic algorithm first with mutual information or gamma test, to spare time, then with the actual model if necessary. Those algorithms will take much more time than the incremental procedure but they might lead to better solutions as they are less restrictive than the greedy algorithms.

Chapter summary

Feature selection methods combine a feature relevance estimator with a subset search procedure. The relevance of the features can be estimated using the performances of a prediction model or by statistical measures. The latter are not able to take into account the characteristics of the prediction model that will subsequently be used, but they are fare less demanding in computation time. Searching for the optimal feature subset can be done by exhaustive search, but this of course becomes quickly intractable when the initial number of features is important. Incremental (greedy) methods are a good alternative; they have a quadratic complexity and often reach good, though sub-optimal, results.

7

Practical Feature Selection for High-Dimensional Data

Selecting features from high-dimensional data is a difficult task for two reasons: the needed computation time grows rapidly with the number of initial features, and the procedure becomes very sensitive to the values of its parameters. This chapter proposes to enhance the mutual information-based forward feature selection with two methods to cope with those difficulties. The first one is to group features together to reduce the number of them. The second one is to use resampling methods to help in the choice of the parameters of the method, namely the number of neighbours in the relevance estimator and the stopping criterion for the forward search. Finally, this chapter proposes some further enlightenments about the use of the mutual information vs. the gamma test for relevance estimation.

When computation time is a sparse resource, the combined use of a nearest neighbours-based mutual information estimator or a noise variance estimator, and a forward feature search is an affordable solution. Nevertheless, even with large computation power available, selecting features from high-dimensional data often reveals impractical. Furthermore, high-dimensional data often have the property to have highly correlated attributes, which, in addition to the high dimensionality, renders the selection procedure unstable, very sensitive to the choice of its parameters. Section 7.1 explains these difficulties in more detail, while Section 7.2 and 7.3 provide solutions, namely a reduction of the initial number of features through grouping and more control over the parameters of the method (relevance criterion and stopping criterion) using resampling techniques. It should be noted that most of the results in this chapter are described for the mutual information, but the methods presented could be used with few modifications with the gamma test. Finally, Section 7.4 briefly compares the properties of Mutual Information and the Gamma Test to score features.

7.1 Problems specific to feature selection from high-dimensional data

Two main obstacles are encountered when selecting features from high-dimensional data. Firstly, given the large number of initial features, even greedy approaches can become costly. Secondly, the results of the procedure are often unstable: the selected feature subset sometimes highly depends on the values of the parameters of the feature selection method.

7.1.1 Large computation times even for incremental procedures

With $d = 100$ or 1000 features, exhaustive search is of course impossible. But considering that, for each feature subset that has to be scored, a non-trivial relevance estimation has to be performed, the incremental methods also become computationally very expensive. Most of the time, even the backward feature elimination procedure cannot be considered because the large number of initial features will prevent many model fitting algorithms from converging and many nonparametric relevance estimations from providing any accurate estimation.

The forward feature search is cubic in the dimensionality

The forward search is also demanding when the data dimensionality d is large. It requires $O\left(d^2\right)$ subset evaluations. Since each subset evaluation requires at most as many operations as there are features in the subset, i.e. $O\left(d\right)$ operations, the total cost of the forward procedure is $O\left(d^3\right)$ operations. With $d = 1000$, this can lead to more than 10^8 operations in worst cases. Even if only the first 10 features are selected, more than 10^6 relevance estimations have to be performed. Depending on the complexity of the estimation, it can take from days to weeks to complete on a standard computer.

Even simple ranking takes time; for instance, estimating the individual mutual information between 10 000 features and the response variable with 1000 observations takes more than three minutes on a standard PC using the simplest method (histograms.) Using a simple nearest neighbour-base estimator, it takes more than 10 minutes.

7.1.2 Unstable results

Another characteristic high-dimensional data often have, beside the large number of features, is the fact that many features are highly-correlated to several others, a property called *collinearity*. Collinearity is present especially when the data come from the discretisation of a signal like a sound or a spectrum for instance. For such data, referred to as *functional data* [Ramsay & Silverman, 1997], the feature labelling has a meaningful order: for sound samples it corresponds to time, while for spectra it correspond to wavelengths. Two features that have close labels (e.g. X_{56} and X_{57}) are often highly correlated.

Collinearity brings instability

Consider two consecutive features X_{56} and X_{57} that are highly correlated. The respective relevance measurements between each of these features and the response variable are thus very similar. Given the variance of those relevance measurements, the order of relevance for those two features (i.e. which one is more relevant than the other) is more likely to be influenced by the parameter of the relevance measure, or by the noise, than by the real difference in relevance between both features.

The feature subset obtained by the forward procedure is then often highly influenced by the parameter of the relevance estimator and by the stopping criterion for the forward search. Those two parameters must thus be carefully chosen.

The collinearity property furthermore renders the interpretation of the selected features very difficult. What do we conclude if two runs of the forward selection bring two completely different results? Are the subsets $\{X_{34}, X_{62}, X_{24}\}$ and $\{X_{32}, X_{68}, X_{23}\}$ to be considered as different ? Is there any reason why feature X_{68} has been selected and not (say) X_{67}? What if features X_{68}, X_{69} and X_{70} are selected together even though they are obviously redundant ?

7.2 Grouping correlated features

Grouping correlated features is one way to reduce their number and thus to accelerate computations. It furthermore renders the procedure less sensitive because the resulting groups are less collinear than the original features. Finally, when features

are grouped according to their correlations to each others, for instance by grouping only consecutive features in spectral data, the result is interpretable in terms of ranges of wavelengths instead of single values of the wavelength, which were actually artificially introduced by the sampling procedure.

7.2.1 Spline modelling of spectra

An elegant way of grouping features for functional data is by first modelling these data with B-splines, and then working with the coefficients of the spline representation of the spectra instead of the original features [Alsberg, 1993][1]. Each coefficient is indeed associated to a range of features (i.e. wavelengths or time samples.)

Modelling functional data with splines consists in building a spline basis in which functional data are then described. Order b Splines are piecewise order b polynomial functions, with continuous derivatives up to order $b-2$, defined on the union of the intervals $[u_0, u_1[$, $[u_1, u_2[$, ..., $[u_{t-1}, u_t]$, where $[u_0, u_t]$ is the range of indices of the features. Each "piece" (there are $t-1+b$ of them) is a *B-spline*: $B_1^b, \ldots, B_{t-1+b}^b$. Each B-spline is a function with a localized domain: it is non-zero on only at most b consecutive intervals. For $b = 1$, there are t B-splines, $B_1^1, \ldots, B_t^1$, given by

$$B_l^1(i) = \begin{cases} 1 \text{ if } i \in [u_{i-l}, u_l[, \\ 0 \text{ otherwise.} \end{cases} \tag{7.1}$$

This corresponds to approximating the spectra with a piecewise constant function. For $b = 2$, it corresponds to a piecewise linear approximation. Each BSpline is a "triangle" function defined on two intervals. It grows linearly on $[u_{i-l}, u_l[$ then decreases linearly on $[u_l, u_{i+l}[$.

Data are reconstructed with fewer features

The data elements are modelled as linear combinations of B-splines, therefore by the approximation

$$\mathbf{x}^{(j)} = \text{discretization of } \sum_{l=1}^{t-1+b} a_{j,l} B_l^b. \tag{7.2}$$

[1] This section is based on an original idea of F. Rossi

The $\mathbf{x}^{(j)}$ are thus seen as vectors sampling the spline reconstruction of the spectra.

The coefficients $a_{j,l}$ are obtained by minimizing the square reconstruction error for each spectrum. The parameter t can be chosen by leave-one-out on the spectra [Rossi *et al.* , 2006]: each original feature, one at a time, is estimated from a spline reconstruction of the spectra built without that feature. The leave-one-out error corresponding to a value of t is the squared estimation error averaged over all features and all spectra.

Each spline coefficient is built as a linear combination of all original features (wavelengths.) However, due to the locality property of splines, each spline coefficient corresponds to a small effective wavelength range [Rossi *et al.* , 2006]. The coefficients of the linear combination corresponding to a spline coefficient are indeed significantly nonzero only in the region of the spectra covered by the corresponding spline.

The BSpline-based method is illustrated on the NITROGEN dataset[2]. This dataset consists of scans and chemistry gathered from fescue grass. The data set contains 141 spectra discretised to 1050 different wavelengths, from 400 to 2 498 nm. The goal is to determine the nitrogen content of the grass samples (ranging from 0.8 to 1.7 approximately).

Selecting the spline coefficient instead of the original features

The spectra are modelled in a basis comprising 149 B-splines of order 5. Choosing the appropriate BSpline basis takes less than 17 minutes when the optimal number of B-splines is chosen between 50 and 500. Then, a forward feature selection procedure with the mutual information is performed on the coefficients of the spline representation.

The forward feature selection takes approximately 12 minutes on a standard personal computer. By contrast, running the same procedure with the original (1050) variables takes about 2 hours. In both cases, the forward search is stopped after 105 features, because the dataset contains only 105 elements. The time needed to build the actual prediction model (`RBFN`) is less

[2] Available from `http://kerouac.pharm.uky.edu/asrg/cnirs/shoot_out_1998/`

than 2 minutes.

Table 7.1 shows the results of prediction (in Normalised Mean Squared Error–NMSE) on the test set. The 10 features selected by maximization of the mutual information cannot be used to construct a linear model with performances comparable to the ones of the optimal linear models (PCR: Principal Component Regression, PLSR: Partial Least Squares Regression and LR: Linear Regression.) Moreover, the nonlinear model constructed on those variables has the best performances.

Method	Variables	NMSE (test)
PCR	10	$1.57\,10^{-1}$
PLSR	9	$1.51\,10^{-1}$
MI + RBFN	10	$1.21\,10^{-1}$
MI + LR	10	$2.59\,10^{-1}$

Table 7.1: Normalised mean square error on the test set for the nitrogen content prediction problem (shootout database)

While the mutual information maximization leads to the selection of 10 variables, they are calculated using only three intervals of the original wavelength range: $[400, 816]$, $[874, 1118]$ and $[2002, 2478]$. Figure 7.1 represents the normalised (between zero and one) coefficients used to compute the new variables. It appears clearly that only some of the original wavelengths are used.

7.2.2 Other methods

This approach of modelling the data with splines brings *de facto* a grouping of the features. However, other methods could be used to cluster features. In [Dijck & Hulle, 2006], a hierarchical clustering using full linkage and correlation was used to cluster features. Although this method gives clusters containing consecutive features, it should be possible to adapt the usual hierarchical or flat tools to group consecutive features together only. Finally, it could be interesting to perform an Independent Component Analysis of the functional data and subsequently

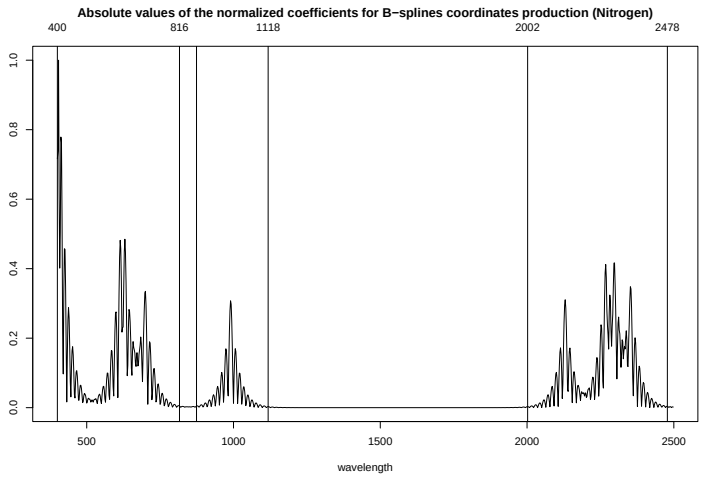


Figure 7.1: Normalised absolute value of the coefficients used to compute the selected variables from the original spectral variables. Only the original variables for which the coefficient is non-zero participate in the construction of the new features, i.e. the coefficients of the spline that were selected by the forward procedure.

select the sources that appear.

Independent Component Analysis

For example, let us consider the TECATOR dataset (described in [Borggaard & Thodberg, 1992]). An Independent Component Analysis algorithm was applied on the dataset. The aim of the `ICA` algorithm is to extract "sources" from the data, that is independent signals from which the data can be reconstructed by linear combinations, see for instance [Hyvärinen *et al.* , 2001] for a basic introduction. Figure 7.2 displays the independent sources that were extracted from the whole dataset, with six sources. Each spectrum can be expressed as a weighted sum of those sources, which are statistically independent. All the sources are rather flat signals with one localized "bump". Each source can thus be associated to a wavelength range. For instance, source S4 spans the range $[925, 945]$, which corresponds to the fat content in the meat sample [Borggaard & Thodberg, 1992]. That source is indeed highly correlated (more than 0.8)

with the response value. This approach was successfully applied to the prediction of the fat content [Krier *et al.* , 2006b].

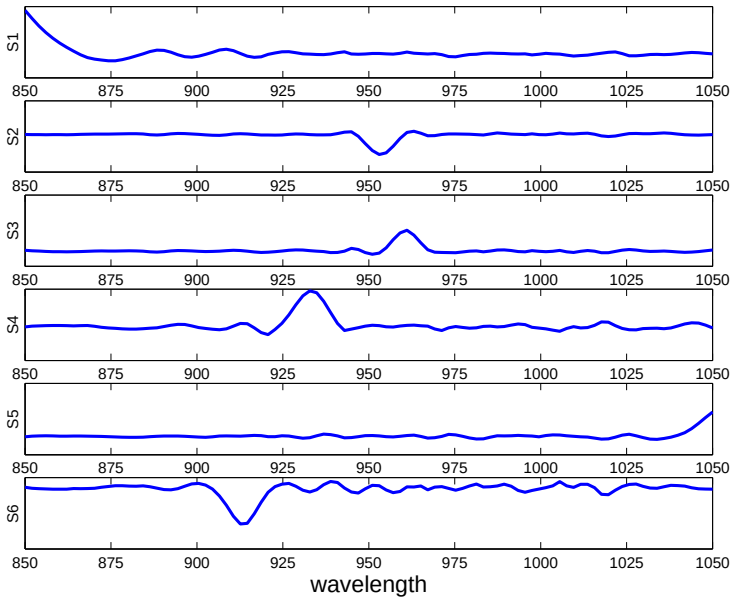


Figure 7.2: Sources extracted from the Tecator dataset with FastICA. Each source spans a compact range of the features (wavelengths.)

7.3 Using resampling methods

While grouping the features reduces both dimensionality and collinearity, and hence improves the procedure's robustness, the robustness can still be improved by carefully choosing the parameters of the method. Resampling methods help addressing this issue.

Resampling methods have heavy computational requirements that increase the time needed to perform the forward selection procedure. However, the running time of the scheme pro-

posed in this section remains acceptable compared to the computational burden of alternate solutions that could be used to choose the number of features and the parameter of the estimator (e.g. optimising those elements based on the performances of a prediction model.) The next subsection presents the resampling tools that will be used in the methods described in Sections 7.3.2 and 7.3.3. Section 7.3.4 then illustrates the methods on an example, Section 7.3.5 considers the computational burden.

7.3.1 K-fold resampling and permutation (randomised) test

The K-fold resampling is very similar to the K-fold cross-validation scheme used for validating prediction models, except that it is used in an unsupervised manner. Given $\mathbf{x}_i$ a vector comprising the values of the ith feature for all observations, and $\mathbf{y}$ the corresponding response values, and some statistic θ, it consists in estimating the K values of $\hat{\theta}$ where one (or several) data element(s) has (have) been removed from the analysis. Typically, the sample is partitioned into K clusters of roughly equal size, and the statistic is estimated K times successively on the sample from which the Kth cluster was excluded. The average of those estimations is often found to be a better estimator of θ, while the variance of the estimations gives an idea of the sensitivity of the estimator to the particular sample.

The permutation test

The permutation test [Good, 1994] is a nonparametric hypothesis test over some estimated statistics $\hat{\theta}$ involving $\mathbf{x}_i$ and $\mathbf{y}^{(j)}$ ($1 \leq j \leq N$), respectively realizations of X_i and Y. Let $\hat{\theta}_i$ be the value of the statistics for the given $\mathbf{x}_i^{(j)}$ and $\mathbf{y}^{(j)}$, both samples of size N drawn from X_i and Y_i respectively. The aim of the test is to answer the following question: how likely is the value $\hat{\theta}_i$ given the samples $\mathbf{x}_i^{(j)}$ and $\mathbf{y}^{(j)}$ ($1 \leq j \leq N$), if we suppose that X_i and Y are independent and thus that the statistic θ_i should be zero?

The permutation test considers the empirical distributions $\mathbf{x}_i^{(j)}$ and $\mathbf{y}^{(j)}$ to be fixed, as well as the sample size N. The random variable of interest is the value of the statistic θ. In such a framework, the distribution of $\hat{\theta}$ is the set of all values of $\hat{\theta}_l$ for all $N!$ possible permutations of the elements of the vector $\mathbf{x}_i^{(j)}$, or,

equivalently, all permutations of the elements of the vector $\mathbf{y}^{(j)}$. The p-value α associated to the test is the proportion of $\hat{\theta}_l$ that are larger than $\hat{\theta}_i$.

Randomised permutation test

In practice, it is not necessary to perform all $N!$ permutations. Several tens or hundreds of them are randomly performed. In this case, the exact p-value cannot be known but a 95% confidence interval around the observed p-value can be estimated [Opdyke, 2003].

The next two sections describe how to take advantage of the resampling methods explained above in a nearest-neighbour-based estimation of the mutual information used in a forward feature search procedure. The resampling methods can be used to (i) choose the number of neighbours, and (ii) to halt the forward procedure.

7.3.2 The optimal number of neighbours

In order to determine the optimal number of neighbours in the estimation of the mutual information, the notion of optimality must be explicitly defined since there is no obvious criterion that we could maximize or minimize. We do not want to optimise the number of neighbours with respect to the performances of a prediction model built with the variables chosen by the procedure, because this would render the search procedure too time-consuming.

Choose k that maximises separability

The goal is to discriminate between features that are relevant for the problem and features that are useless. We therefore consider the optimal value of k to be the value for which the separation between the relevant features and an independent feature is maximum. Since the estimator of the mutual information has some variance, it is important to take this variance into account in measuring the separability. If we had access to the distribution of the mutual information estimate over the data, we could calculate a separation between $MI(\mathrm{X};\mathrm{Y})$ and $MI(\mathrm{U};\mathrm{Y})$ (considered as random variables) for an important feature X and an useless feature U (as illustrated in Figure 7.3).

In practice, we rely on a combined K-fold/permutation test to estimate the bias and the variance of the estimator for relevant features and for independent ones. The idea is the following.

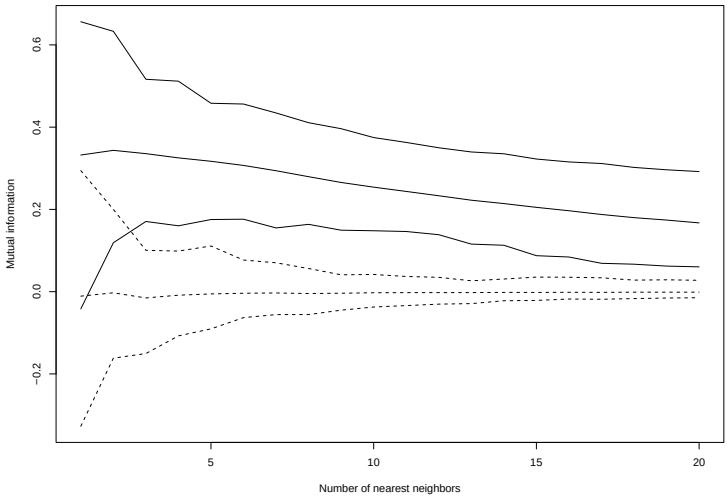


Figure 7.3: Mutual information estimator distribution for an artificial example. Solid lines correspond to a useful feature and dashed lines to an irrelevant feature. The outer lines correspond to the standard deviation (estimated using a K-fold resampling technique) of the estimation of the mutual information, while the innermost line corresponds to the mean of the estimation. The "optimal" number of neighbours is the one that best separates the respective distributions of mutual information of the useful and useless feature.

Consider X_i a feature that is supposed to be relevant to predict Y. Two resampling distributions are built for both $MI\left(\mathrm{X}_i;\mathrm{Y}\right)$ and $MI\left(\mathrm{X}_i^{\pi};\mathrm{Y}\right)$ where X_i^{π} denotes a randomised version of X_i that is made independent from Y through permutations. This is done by performing several estimations of (i) the mutual information between X_i and Y using subsets of the original sample and (ii) the mutual information between a randomised version of X_i and Y, using several non-overlapping subsets of the original sample, in a K-fold cross validation-like scheme. A good value for K is around 20 or 30. A value of K less than 20 renders the estimation of mean and variance hazardous, while the estimations with more than 30 are often very close to those with K=30 in practice. The procedure results in two samples of

estimates of $MI\,(\mathrm{X}_i;\mathrm{Y})$ and $MI\,(\mathrm{X}_i^{\pi};\mathrm{Y})$.

The optimal value of k is the one that best separates those two distributions, for instance according to a Student-like measure:

$$t_{i,k} = \frac{\mu - \mu_\pi}{\sqrt{\sigma^2 + \sigma_\pi^2}}, \tag{7.3}$$

where μ and σ^2 represent the mean and variance of the cross-validated distribution of $MI\,(\mathrm{X}_i;\mathrm{Y})$, and μ_π and σ_π^2 are those of the cross-validated distribution of $MI\,(\mathrm{X}_i^{\pi};\mathrm{Y})$ (illustrated on Figure 7.4).

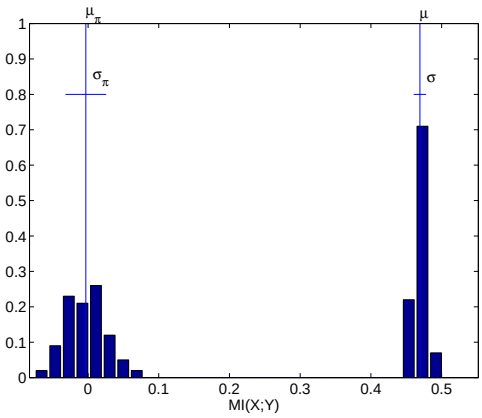


Figure 7.4: Distribution of mutual information for a relevant feature. On the left, the distribution of the mutual information of the features with permuted values, on the right, the distribution of the mutual information of the relevant feature, as given by the K-fold method. The value of k is chosen so as to best separate those two distributions.

The optimal value kept for the whole procedure

The optimal k is chosen commonly for all features as the one corresponding to the largest value of $t_{i,k}$ over all values of k over all features. This way, features that are useless do not participate in the choice of the optimal value. Using useless features to choose the value that best separates the resampling of

the mutual information from the permuted sample would indeed make no sense if they were independent from the output value. It should be noted that other solutions could be thought of like, for instance, to optimise the mean value of $t_{i,k}$ over features for which $t_{i,k}$ is above a pre-specified significance threshold, but at the cost of an additional parameter.

It is also straightforward to extend this criterion to the case of several variables and therefore to optimise k at each step of the forward procedure. The main drawback of this option is the increased computational burden. The next section presents a halting criterion for the forward search, based on the permutation test.

7.3.3 A stopping criterion

When nested subsets of features are considered, as in the forward search, the mutual information is a non-decreasing function of the subset size; it can only increase or remain constant as more features are added. The chain rule theorem for information indeed states that [Cover & Thomas, 1991]

$$I(X_1, X_2, \ldots, X_i; Y) = I(X_1; Y) + I(X_2; Y|X_1) + \cdots + I(X_i; Y|X_1, \ldots, X_{i-1}),$$

which can be rewritten using a short-hand notation:

$$I(X_1, X_2, \ldots, X_i; Y) = \sum_{l=1}^{i} I\left(X_l; Y|X_{l-1}, X_{l-2}, \ldots, X_1\right).$$

By applying the chain rule theorem twice, we show:

$$\begin{aligned} I(X_1, \ldots, X_i; Y) &= \sum_{l=1}^{i} I\left(X_l; Y|X_{l-1}, \ldots, X_1\right) \\ &= \sum_{l=1}^{i-1} I\left(X_l; Y|X_{l-1}, \ldots, X_1\right) + I\left(X_i; Y|X_{i-1}, \ldots, X_1\right) \\ &= I(X_1, \ldots, X_{i-1}; Y) + I\left(X_i; Y|X_{i-1}, \ldots, X_1\right), \end{aligned}$$

which means that the mutual information between a set of features and the variable to predict can be decomposed into the sum of the mutual information between the first $i-1$ features and the response value and the mutual information between the last feature and the response value, conditioned on the set

of $i - 1$ previous features. As both are non-negative [Cover & Thomas, 1991], the mutual information can only increase as more features are added.

Maximizing the mutual information therefore does not make sense: the whole feature subset will always, in theory, have the largest mutual information with the value to predict.

Friedman's example

Let us consider here a synthetic prediction problem, derived from Friedman's [Friedman, 1991]. We consider 10 input variables X_i and one output variable Y given by

$$Y = 10 \sin(X_1 \cdot X_2) + 20 (X_3 - 0.5)^2 + 10X_4 + 5X_5 + \epsilon. \quad (7.4)$$

All $X_i, 1 \leq i \leq 10$, are uniformly distributed over $[0, 1]$, and ϵ is a centred Gaussian noise with unit variance. Variables X_6 to X_{10} are just noise and have no predictive power. The sample size is 100.

In practice, as illustrated by Figure 7.5, the evaluation of the mutual information tends to decrease when useless variables are added, especially with an estimator based on the distances between observations. It is therefore tempting to look for the maximum value of the mutual information. But again, as shown by Figure 7.5, this will frequently lead to sub-optimal feature subsets. On this example, stopping the forward procedure at the first peak selects a wrong number of features in almost all cases. Moreover, searching for the global maximum does not improve a lot the situation: the optimal set of features is selected only in four cases (for k equal to 1, 3, 5 and 6). While the optimal choice of k should in theory prevent estimator problems to lead to bad estimations of the mutual information, we can guarantee neither that the proposed criterion will always choose an optimal k, nor that, even if we had the optimal k, the optimal feature subset would correspond to the highest value of the estimated mutual information. For instance, for the dataset used in Figure 7.5, the parameter selection criterion chooses $k = 19$; maximizing the mutual information then keeps only features X_4 and X_2.

A more promising approach consists in trying to avoid adding useless features to the current subset by comparing the value of the mutual information with the added feature to the one without that feature in a way that incorporates the variability

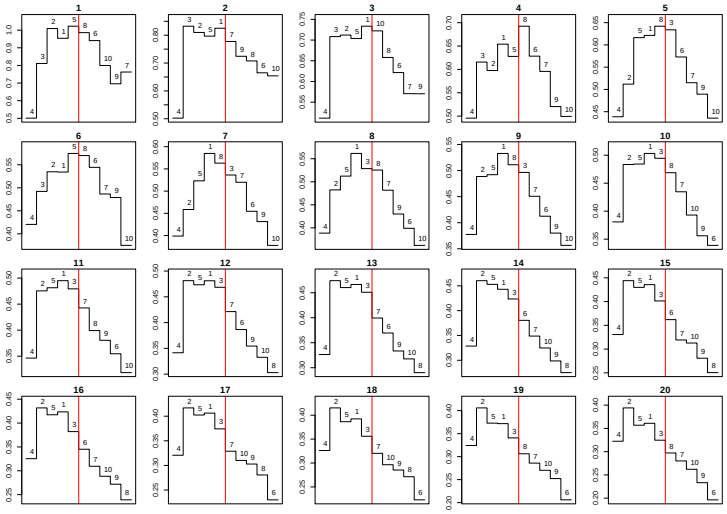


Figure 7.5: Result of the forward procedure on the artificial example with different values of the number of neighbours (shown above each graph.) On each graph, the value of the mutual information of the "winner subset" at each iteration is shown, below the index of the feature that is selected at that iteration. Only for well-chosen values of the number of neighbours the correct features are selected (X_1 to X_5.)

of the estimator.

Estimation of the optimal criterion by permutation test

Let us consider S, the subset of already selected features, and X^*, the best candidate among all remaining features. Take the distribution of $MI\,(S \cup \{\mathrm{X}^*\}; \mathrm{Y})$ under the hypothesis that X^* is independent from Y and S, that is all values of

$$MI\,(S \cup \{\mathrm{X}^{*\pi}\}; \mathrm{Y})$$

where $\mathrm{X}^{*\pi}$ is a randomised permutation of X^*. If the p-value of $MI\,(S \cup \{\mathrm{X}^*\}; \mathrm{Y})$ is small and the hypothesis that the mutual information is the same with and without the new feature is rejected, it means that X^* brings sufficient new information about Y to be added to the current feature subset.

Note that the increment (or lack thereof) in mutual information between $MI\left(S \cup \{\mathrm{X}^*\}; \mathrm{Y}\right)$ and $MI\left(S; \mathrm{Y}\right)$ is estimated without comparing estimations of mutual information on subsets of different sizes. In theory this should not be an issue, in practice however, it is important as illustrated by Figure 7.5: while, in theory, adding an informative variable should strictly increase the mutual information, the contrary is frequently observed.

7.3.4 Application to real data

To further validate the interest of the proposed approach, the forward procedure is applied to 100 datasets randomly generated from Equation 7.4. For each dataset, the optimal value of k is selected between 1 and 20, then the forward procedure is conducted. The feature set that maximizes the mutual information and the best feature set according to the stopping criterion presented in the previous section are retained for comparison. Results are summarized in Tables 7.2, 7.3 and 7.4.

Number of features	1	2	3	4	5	6
Maximal mutual information	7	45	33	14	1	0
Stopping criterion	0	1	12	52	29	6

Table 7.2: Number of feature subsets of a given size obtained by both criteria

It appears clearly from Table 7.2 that maximizing the mutual information does not provide good results: this leads to the selection of an optimal set of features (5 variables) only in one case out of one hundred. The stopping criterion defined in the previous section tends to select more features: in fact, the feature sets obtained by this methods have strictly more features that the ones selected by maximizing the mutual information in 84 % of the cases (and equal sizes in other situations).

Moreover, the additional features are generally informative ones, as illustrated by Table 7.3. The positive aspect of maximizing the mutual information is that it leads, on those experiments, only to the selection of relevant features. The proposed stopping criterion selects sometimes irrelevant features

Number of informative features	1	2	3	4	5
Maximal mutual information	7	45	33	14	1
Stopping criterion	0	1	16	66	17

Table 7.3: Number of feature subsets that contain the specified number of relevant features obtained by both criteria

(see Table 7.4), but it also selects always at least as much relevant features as the former method. Moreover, in 79% of the experiments, it selects strictly more relevant features than the maximizing strategy. In 5 % of the experiments, the feature set selected by the significance stopping criterion consists in the set that maximizes the mutual information with an additional uninformative variable: this corresponds to the error level expected as the forward procedure was controlled by using the 95% percentile of the permutation distribution.

Number of uninformative features	0	1	2
Stopping criterion	75	22	3

Table 7.4: Number of feature subsets that contain the specified number of irrelevant features obtained by the proposed stopping criterion.

This simulation study shows that while the proposed stopping criterion is not perfect, it provides significant improvements over the standard practice of maximizing the mutual information. Moreover, it does not lead to the selection of too large feature sets that would reduce its practical benefit.

The utility of the method is further illustrated below on a well-known dataset from the UCI Machine Learning Repository (Housing), on a high-dimensional nitrogen spectra data and on a high-dimensional data set from the Delve repository.

The HOUSING dataset

The optimal value (on the learning set) of k, searched between 1 and 20, is found to be 18. The forward search procedure described in the previous section is run with 50 permutations on the learning examples. The threshold p-value is set to 0.05. When the mutual information is below the 95% percentile of

the permutation distribution, the procedure is halted.

Figure 7.6 displays the mutual information as a function of the forward search iterations. The horizontal lines correspond to the critical values (i.e. the 95% percentile of the permutation distribution) while the circles represent the mutual information between the selected subset and the value to predict. Four features are selected (X_6, X_{13}, X_1 and X_4). Interestingly, the procedure does not stop when the peak in mutual information is observed.

A RBFN model was built using the selected features and optimized by 5-fold cross-validation on the learning set, according to the method described in [Benoudjit & Verleysen, 2003]. The root mean squared error (RMSE) on the test set is 9.48. By comparison, the RMSE on the test set with the all set of features is 18.97, while the RMSE with the first two features, corresponding to the peak in mutual information, is 19.39.

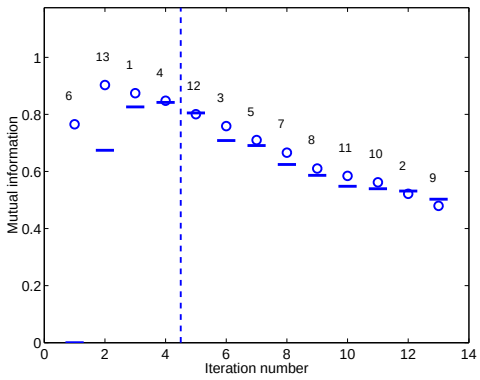


Figure 7.6: The evolution of the mutual information in a forward feature subset search on the Boston Housing dataset. Thresholds (horizontal lines) are computed as the 95% percentile of the permutation distribution; the actual values of the mutual information are represented with circles. The procedure stops after four features have been selected (dashed line).

The NITROGEN dataset originates from a software contest organized at the International Diffuse Reflectance Conference[3] held in 1998 in Chambersburg, Pennsylvania, USA. It consists of scans and chemistry gathered from fescue grass (*Festuca elatior*). The data set contains 141 spectra discretised to 1050 different wavelengths, from 400 to 2 498 nm. The goal is to determine the nitrogen content of the grass samples (ranging from 0.8 to 1.7 approximately). The data can be obtained from the Analytical Spectroscopy Research Group of the University of Kentucky[4].

The NITROGEN dataset

The dataset is split into a test set containing 36 spectra and a training set with the remaining 105 spectra. The spectra are described in a BSpline basis, as described in section 7.2; the optimal B-spline basis consists in 166 B-splines of order four.

Figure 7.7 illustrates the behaviour of the forward feature selection with resampling on this dataset. The optimal number of neighbours is 12. It leads to the selection of 25 variables (among the 166 B-spline coordinates). The RMSE on the test set, using a `RBFN` model built on those features, is 0.6649. Maximizing the mutual information leads to a smaller feature set with 6 features. The RMSE on the test, using a `RBFN` model built on those features, is 0.7753. As a reference, the RMSE on the test set when all features are used is 3.1197.

For the sake of this analysis, we used only 104 of the 139 original features. We indeed discarded the features that are too much correlated with the value to predict like the average price, the first and third percentiles, etc. In the dataset, 52 regions were found to have a median house price of zero; they were considered to be erroneous and removed from the analysis[5].

The DELVE dataest

Of the 22,732 remaining observations, 14,540 are used for the test set. The 8192 remaining observations are split into 8 subsets and used for training. This corresponds to the classical splitting for this dataset; it allows one to study the variability of the fea-

[3] `http://www.idrc-chambersburg.org/index.htm`

[4] `http://kerouac.pharm.uky.edu/asrg/cnirs/shoot_out_1998/`

[5] The preprocessed data can be downloaded from the UCL Machine Learning Group website: `http://www.ucl.ac.be/mlg`

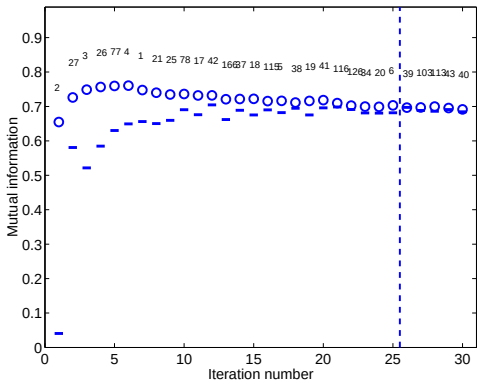


Figure 7.7: The evolution of the mutual information in a forward feature subset search on the nitrogen dataset. Thresholds (horizontal lines) are computed as the 95% percentile of the permutation distribution; the actual values of the mutual information are represented with circles. Twenty-five features are selected.

ture selection procedure while retaining enough data both for learning and testing. For each observation subset, the optimal feature subset is determined using the proposed approach and a RBFN model is built using a 3-fold cross validation procedure. The RBFN model is then applied on the test set and the results are compared with those obtained using the peak in mutual information and using all features.

Figure 7.8 displays the evolution of the mutual information and of the thresholds found by permutation over each iteration of the forward search procedure. Figure 7.8 shows the results of the first of the eight learning sets. The number of selected features is eight, while the maximum of mutual information is observed for six features.

Table 7.5 shows the Root Mean Square Error of the model on the test set, for each learning subset. The permutation approach always selects either 8 or 9 features, while stopping the forward procedure at the peak of mutual information gives from 2 to 6 features. Except for Subset number 2, the results obtained with the permutation are either equivalent either far better than

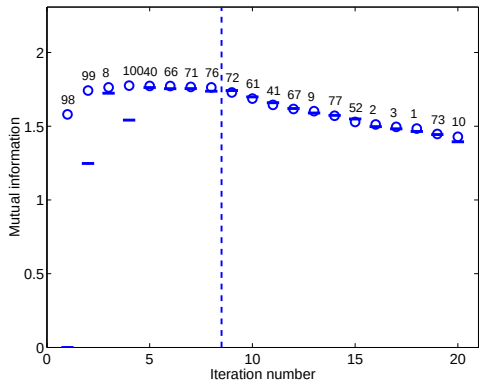


Figure 7.8: The evolution of the mutual information in a forward feature subset search on the first subset of the Delve dataset. Thresholds (horizontal lines) are computed as the 95% percentile of the permutation distribution; the actual values of the mutual information are represented with circles. Eight features are selected.

those obtained with features selected by taking the peak of mutual information.

Subset number	Using permutations # features	test RMSE ($\times 10^4$)	Using the peak # features	test RMSE ($\times 10^4$)	All features test RMSE ($\times 10^4$)
1	8	1.3286	6	1.3223	1.4304
2	9	1.0748	6	0.9472	1.5393
3	8	1.2883	3	2.5643	1.4338
4	8	1.2214	2	2.3125	1.419
5	9	1.2575	3	1.1799	1.4628
6	8	0.9504	5	2.363	1.4146
7	8	1.1987	2	2.2381	2.1855
8	9	1.1929	3	1.19	1.5314

Table 7.5: Root mean square error on the test set obtained by the RBFN built on each learning subset.

7.3.5 Computational burden

In most traditional resampling schemes, the overall computation time is simply multiplied by the number of resamplings performed. In this case however, a more detailed analysis is needed to grasp the overhead cost brought by the proposed method.

Cost of the stopping criterion

The number of mutual information estimations to perform at iteration t in the forward search, is equal to the number $d-t+1$ of features that are candidate for entering the optimal feature subset plus the number P of permutations performed to evaluate the threshold of the stopping criterion. The cost of each iteration, in terms of mutual information estimation, thus amounts to $d-t+1+P$. As the number of permutations is often limited to 20 or 30, the additional cost at each iteration needed to estimate the threshold is small compared with the cost needed to find the feature that should be added to the optimal feature subset. For instance, on a 100-dimensional dataset (like the TECATOR dataset), 955 estimations of the mutual information are needed to find the optimal subset of size 10 while 200 estimations, that is a bit more than 15% were used to set the threshold. Of course, when the number of original features is small, permutations tend to represent a more important part of the total computational burden.

Cost of optimising the number of neighbours

The cost of the choice of the optimal number of neighbours is roughly equal to the cost of the first step of the forward search multiplied by K, the number of folds in the cross-validation scheme used in the proposed method. In practice, K is chosen between 20 and 30. If the expected number of optimal features has the same order of magnitude, the total cost of the forward procedure will also be of the same order of magnitude than the cost of the cross-validation, which means that the overall cost is roughly doubled. However, this is much less than if the number of neighbours was optimized using the performances of the prediction model, as this would imply performing as many forward searches as the number of values that are tested.

The total cost of the automatic determination of the parameters, in the case of high-dimensional data, is thus a bit more than the double of the cost when the number of neighbours is chosen arbitrarily and the mutual information is maximized.

This additional cost brings in better and more stable results, as shown in the next section.

7.4 Mutual information or Gamma test?

How to choose between the mutual information and the gamma test is a very difficult question. Both have the same computational complexity. Interestingly, the mutual information measure independence while the gamma test measures dependence. A value of mutual information of zero between two variables means that they are independent, while a value of the gamma test of zero denotes a perfect, noiseless, mapping existing between them. Mutual information seems thus more suited at discarding useless features while the gamma test is more akin to spot relevant ones. Both furthermore have limitations.

Drawback of the mutual information

The mutual information is symmetric. It measures not only how much information a feature brings about the response value, but also how much information the response value can bring to predict the features. If the relationship between the feature and the response value is bijective, a symmetric relevance measure is suitable. However, if the function to approximate is not bijective, predicting the response value from the feature may be easy while at the same time predicting the feature from the response value can be impossible. In such case, the mutual information will give low scores to the features even if the response value is perfectly predictable from them, simply because the converse, i.e. predicting the features from the response value is difficult. Think of a feature X being uniformly distributed over $[0, 2\pi]$ and the output being a sinusoidal function of the feature. The mutual information between X and the sine waveform decreases with the frequency of the sine even though the response value is always perfectly predictable from the input (Figure 7.9).

Drawback of the gama test

The Gamma test also has limitations. It measures the variance of the noise, which, if the noise is Gaussian white, brings all information about the noise intensity. However, if the noise is strongly non-Gaussian, or if the noise intensity depends in some way on the value of the input, the measure given by the Gamma test may not be relevant to score a feature. Figure 7.10

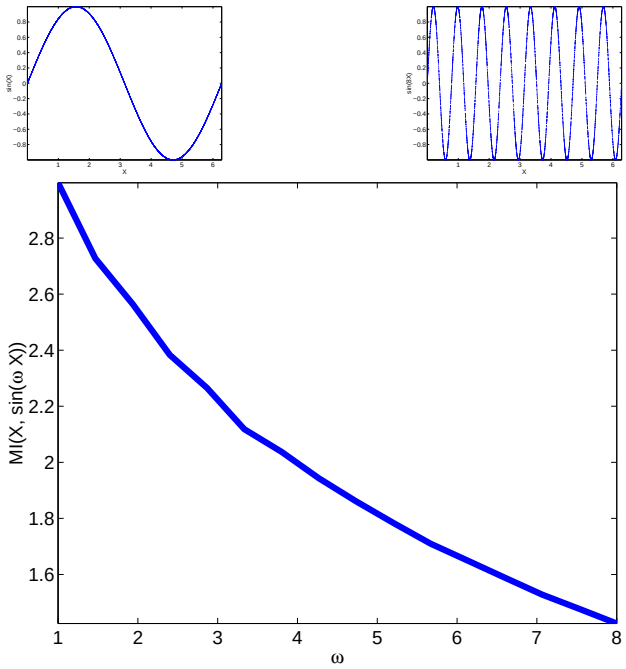


Figure 7.9: Mutual information between X and $\sin \omega X$ for $\omega \in [1, 8]$. The graphs of $\sin X$ and $\sin 8X$ are shown in the top figures. The mutual information is estimated with histograms computed over 10000 draws from a uniform distribution over $[0, 2\pi]$. The mutual information decreases as frequency ω increases.

shows the scatter (feature/response value) plots of two features. One is a pure random variable and the other contains information but also many "outliers". According to the Gamma test, the random variable is more relevant than the one actually carrying some information about the response value!

Chapter summary

Feature selection in the case of a large number of highly correlated features is not easy; classical methods have to be adapted to cope with the large number of features and collinearity. Re-

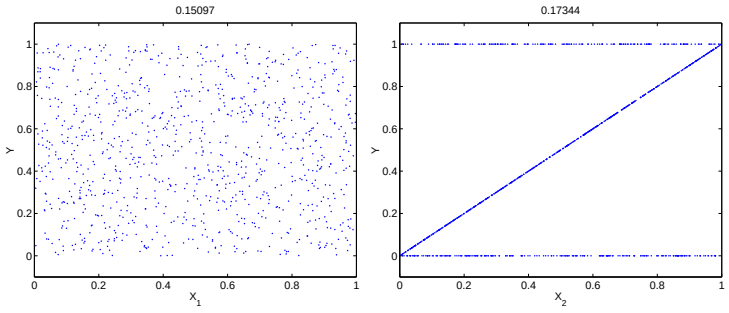


Figure 7.10: Two features X_1 and X_2 are scored with the Gamma test to estimate their prediction power with respect to Y. Feature X_1 is independent from Y and thus has no predictive power. Feature X_2 is equal to Y, except that it is altered with a strongly non-Gaussian noise scheme (burst noise): each observation is either left unaltered (with probability 0.6) or is replaced with an extremum value (with probability 0.4). This feature brings much information about Y, even if it is noisy. Unfortunately, the gamma test is not able to identify this relationship since the variance of the noise is 0.1509 for X_1 and 0.1734 for X_2.

sampling methods bring a lot of information about the relevance of the estimator. It allows choosing the parameter of the estimator (the number of neighbours for instance) and automatically determining when to halt the selection procedure. Feature clustering (e.g. by using spline representation of functional data) can be used to reduce the number of features by taking advantage of their correlations, hence improving the computation time and making the selected features more interpretable.

8

Conclusions

"In mathematics you don't understand things. You just get used to them. – J. von Neumann

Data analysis has become a research field, distinct from statistics, artificial neural networks, machine learning or evolutionary computing. Of course, the field of data analysis benefits from the theory, the tools and methods, the experience and the expertise coming from these domains, but it is not limited to them; it has challenges of its own.

Many data analysis tools borrowed from statistics or machine learning were designed for low-dimensional, large sample data, based on a scheme that the human mind can apprehend. Nowadays, however, data tend to become highly dimensional; they are described by a lot (sometimes over a thousand or more) of attributes. And high-dimensional spaces, where high-dimensional data live, have strange and counter-intuitive properties.

Goal of the thesis

Moreover, it has become common knowledge that many data analysis tools perform better when used on data with few attributes. Their performances degrade when dimensionality increases, either because of irrelevant attributes, or increased noise, or simply because the models are not able to deal with complex data.

The goal of this Thesis was to study the possible link between the counter-intuitive properties of high-dimensional spaces and the observed loss of performances of the data analysis tools on high-dimensional data. More specifically, we focused on how distance-based data analysis tools are affected by the property of distances to concentrate in high-dimensional spaces, and on how feature selection methods perform with high-dimensional

data.

The Euclidean norm and the Gaussian kernels have always been used intensively without much questioning despite the concentration of distances phenomenon. Recently, it was proposed to use fractional distances, a generalization of the Minkowski family of distances comprising among others the Euclidean distance, to reduce the concentration. Concerns rose however about those results as they seem to depend on strong assumptions, namely that the data are uniformly distributed and the sample size is small.

What we found out

The results of this thesis show that the concentration phenomenon does not depend on the sample size and is actually an intrinsic property of the norm, relating to the Laws of Large Numbers. The concentration does not depend either on the distribution of the data; but if the data are not uniformly distributed, fractional norms are not always less concentrated than higher-order norms. The concentration also depends on the intrinsic dimension of the data rather than on the dimension of the space in which they are described.

This study furthermore showed that the tools that only depend on comparing distances in order to rank elements in the process of building the response value or the class label do not suffer from the concentration as far as the Euclidean norm is concerned. Kernel-based methods, by contrast, loose their locality property when built on the Gaussian kernel; it is actually very difficult, if not impossible, to tune the parameter of the Gaussian kernel (its width) to ensure that it covers a well-defined and non-empty portion of the dataspace. However, this locality property is important for the stability, and the interpretation, of those models. Generalised Gaussian kernels, that have an extra slope parameter, are a good alternative in the sense that it is possible to choose the values of both parameters (slope and width) to ensure that they are localised.

New similarity measures

The results of this study show that fractional norms should be considered to compare high-dimensional data, not because they are (in some cases) less concentrated, but because they are able to cope with non-standard noise schemes that may appear with high-dimensional data. The value of the parameter of the norm should be optimised either to minimize (an estimation

of) the generalization error of the model using it, or to avoid one more cross-validation loop, according to a nonparametric measure of noise variance like the gamma test.

Generalised Gaussian kernels should be used with high-dimensional data. The extra degree of freedom brought by the second parameter of the Generalized Gaussian kernel (its slope) actually allows building models faster. These two parameters indeed relate to some characteristics of the distribution of the distances around the centre of the kernel. Those characteristics can be easily measured, and a simple formula gives the corresponding parameter values. Using this method, results can be obtained with Radial-Basis Function Networks and Least-Squares Support Vector Machines, of the same quality than those obtained with Gaussian kernels whose width is optimised by cross-validation.

Improving feature selection

Beside those results, it appears clearly that distance-based tools are badly influenced by non-informative attributes. For high-dimensional data, the least time-consuming feature selection method is, besides simple ranking, to perform a forward search with a nearest neighbours-based mutual information criterion. This method is however still time consuming and is furthermore very sensitive to the values of its parameters: the number of neighbours and the stopping criterion.

Reducing the initial number of features can be done, for functional data, by describing them in a B-spline basis. The coefficients of the reconstruction of the data in that basis are the new (fewer) features to select. Each coefficient actually corresponds to a range of consecutive features. This approach reduces computation times and furthermore brings more interpretability about the results. Instead of selecting features that were artificially introduced by the sampling of the data, ranges of features, corresponding to ranges of wavelengths for instance, are selected. Since the number of coefficients is often much lower than the number of initial features, this accelerates the process of selection. But it also makes it more stable because the coefficients are less collinear than the original features are.

Another way to make the feature selection more robust is to choose adequately the parameters of the method. A procedure was developed that uses randomisation and resampling

to choose the number of neighbours in the estimation of the mutual information so as to best discriminate between relevant features and useless ones. The procedure further uses randomised tests to decide when to halt the forward search procedure.

Chemometry

The methods that were developed, i.e. Radial-Basis Function Networks with Generalised Gaussian Kernels, the fractional distance-based nearest neighbours classifier and the feature selection method based on randomisation and resampling were tested successfully on spectroscopic data.

For each test, the computation times were lowered drastically compared with traditional methods, while at the same time preserving the same prediction performances, sometimes even better ones.

Open issues

While this work proposes methods to make sound choices for the norm and the kernel to use, work still has to be carried on to test how norms and kernels interact. Is the optimal norm for nearest-neighbour search the optimal one also for kernel-based methods? It is obvious that the method for choosing the parameters of the Generalised Gaussian kernel depends on the choice of the norm. Can this one be optimised first and then the kernel be chosen? Is it possible to perform both optimizations at the same time? In a more general framework, could we apply the principles of model selection and/or feature selection for metric selection?

Nevertheless, in our quest for the truth about high-dimensional data, we learned one thing: there is no reason why the exponent should always be equal to two...

References

AGGARWAL, C. C., & YU, P. S. 2001. Outlier detection for high dimensional data. *Pages 37–46 of: SIGMOD '01: Proceedings of the 2001 acm sigmod international conference on management of data*. Santa Barbara, California, United States: ACM Press.

AGGARWAL, C. C., HINNEBURG, A., & KEIM, D. A. 2001. On the surprising behavior of distance metrics in high dimensional spaces. *Pages 420–434 of:* DEN BUSSCHE, J. VAN, & VIANU, V. (eds), *Eight international conference on database theory*. Lecture Notes in Computer Science, vol. 1973. London, United Kingdom: Springer.

AHA, D.W. 1997. lazy learning. *Editorial. artificial intelligence review*, **11(1-5)**, 1–6.

ALSBERG, BJORN K. 1993. Representation of spectra by continuous functions. *Journal of chemometrics*, **7**, 177–193.

ATKESON, C. G., MOORE, A. W., & SCHAAL, S. 1997. Locally weighted learning. *Intelligence review*, **11**, 11–73.

BATTITI, R. 1994. Using the mutual information for selecting features in supervised neural net learning. *IEEE transactions on neural networks*, **5**, 537–550.

BECKENBACH, E. F., & BELLMAN, RICHARD E., THIRD PRINTING. 1971. *Inequalities*. third printing edn. Ergebnisse der Mathematik und ihrer Grenzgebiete Series. Berlin: Springer-Verlag.

BELLMAN, R. 1961. *Adaptive control processes: A guided tour*. Princeton University Press.

BENOUDJIT, N., & VERLEYSEN, M. 2003. On the kernel widths in radial-basis function networks. *Neural processing letters*, **18**, 139–154.

BERCHTOLD, S., BOHM, C., KEIM, D. A., & KRIEGEL, H.-P. 1997. A cost model for nearest neighbor search in high-dimensional data space. *Pages 78–86 of: 16th ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems (PODS) may 12-14, 1997, Tucson, Arizona, USA*. New York, NY, USA: ACM Press.

BEYER, K. S., GOLDSTEIN, J., RAMAKRISHNAN, R., & SHAFT, U. 1999. When is "nearest neighbor" meaningful? *Pages 217–235 of: Seventh international conference on database theory*. Lecture Notes in Computer Science, vol. 1540. Jerusalem, Israel: Springer-Verlag.

BIAU, G., BUNEA, F., & WEGKAMP, M.H. 2005. Functional classification in hilbert spaces. *IEEE transactions on information theory*, **51**(6), 2163– 2172.

BLUM, A., & LANGLEY, P. 1997. Selection of relevant features and examples in machine learning. *Artificial intelligence*, **97**(1-2), 245–271.

BONNLANDER, B. V., & WEIGEND, A. S. 1994. Selecting input variables using mutual information and nonparametric density estimation. *Pages 42–50 of: Proc. of the 1994 int. symp. on artificial neural networks (ISANN 94)*.

BORGGAARD, C., & THODBERG, H.H. 1992. Optimal minimal neural interpretation of spectra. *Analytical chemistry*, **64**, 545–551.

BOSER, BERNHARD E., GUYON, ISABELLE M., & VAPNIK, VLADIMIR N. 1992. A training algorithm for optimal margin classifiers. *Pages 144–152 of: COLT '92: Proceedings of the fifth annual workshop on computational learning theory*. New York, NY, USA: ACM Press.

BURGES, C. J. C. 1998. A tutorial on support vector machines for pattern recognition. *Data min. knowl. discov.*, **2**(2), 121–167.

CAROTHERS, N. L. 2004. *A short course on banach space theory (london mathematical society student texts)*. Cambridge University Press.

CHAPELLE, O., SCHÖLKOPF, B., & ZIEN, A. (eds). 2006. *Semi-supervised learning*. Cambridge, MA: MIT Press.

CHAVEZ, E., NAVARRO, G., BAEZA-YATES, R. A., & MARROQUIN, J. L. 2001. Searching in metric spaces. *ACM computing surveys*, **33**(3), 273–321.

CORTES, C., & VAPNIK, V. 1995. Support-vector networks. *Machine learning*, **20**(3), 273–297.

COVER, T. M., & HART, P. E. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory*, **13**(1), 21–27.

COVER, T.M., & THOMAS, J.A. 1991. *Elements of information theory*. New York: Wiley.

DASH, M., & LIU, H. 1997. Feature selection for classification. *Intelligent data analysis*, **1**(3), 191–156.

DEMARTINES, P. 1994. *Analyse de données par réseaux de neurones auto-organisés*. Ph.D. thesis, Institut National Polytechnique de Grenoble, Grenoble, France.

DENG, K., & MOORE, A. 1998. On the greediness of feature selection algorithms. *Pages 62–77 of: Proc. international conference on machine learning (ICML), june 1998*.

DIJCK, G. VAN, & HULLE, M. M. VAN. 2006. Speeding up the wrapper feature subset selection in regression by mutual information relevance and redundancy analysis. *Pages 31–40 of: Submitted to ICANN 2006: International Conference in Aritificial Neural Networks, held in Athens, Greece, in september 2006*.

D.J. NEWMAN, S. HETTICH, C.L. BLAKE, & MERZ, C.J. 1998. *UCI repository of machine learning databases*.

DONOHO, D. 2000 (August). *High-dimensional data analysis: The curses and blessings of dimensionality*. Lecture delivered at the conference "Math Challenges of the 21st Century" held by the American Math. Society organised in Los Angeles, August 6-11.

EFRON, B., & TIBSHIRANI, R.J. 1993. *An introduction to the bootstrap*. New York: Chapman & Hall.

EVANGELISTA, P.F., EMBRECHTS, M.J., & SZYMANSKI, B.K. 2006. Taming the curse of dimensionality in kernels and novelty detection. *In:* ABRAHAM, A., BAETS, B.D., KÖPPEN, M., & NICKOLAY, B. (eds), *Applied soft computing technologies: The challenge of complexity*. Advances in Soft Computing, vol. 14. Berlin: Springer Verlag.

EVANS, M., HASTINGS, N., & PEACOCK, B. 2000. *Statistical distributions 3rd ed.* New York: Wiley.

FAUSETT, L. V. 1994. *Fundamentals of neural networks*. Prentice Hall.

FLEURET, F. 2004. Fast binary feature selection with conditional mutual information. *Journal of machine learning research*, **5**(November), 1531–1555.

FRANCOIS, D., WERTZ, V., & VERLEYSEN, M. tba. The concentration of fractional distances. *IEEE transactions on knowledge and data engineering*, **submitted**.

FRANÇOIS, D., WERTZ, V., & VERLEYSEN, M. 2005. Non-euclidean metrics for similarity search in noisy datasets. *Pages 339–344 of: ESANN 2005, european symposium on artificial neural networks*.

FRIEDMAN, J. 1991. Multivariate adaptive regression splines (with discussion). *Annals of statistics*, **9**(1), 1–141.

FRIEDMAN, J. H. 1997. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, **1**(1), 55–77.

FUKUNAGA, K., & HAYES, R.R. 1989. Effects of sample size in classifier design. *IEEE transactions on pattern analysis and machine intelligence*, **11**(8), 873–885.

GOLLI, A. EL, CONAN-GUEZ, B., & ROSSI., F. 2004. Self organizing map and symbolic data. *Journal of symbolic data analysis*, **2**(1), 19–25.

GOLUB, G. H., & VAN LOAN, C. F. 1996. *Matrix computations*. 3rd edn. Johns Hopkins University Press.

GOOD, P. 1994. *Permutation tests*. NewYork: Springer.

GÜNTER, S., & BUNKE, H. 2002. Creation of classifier ensembles for handwritten word recognition using feature selection algorithms. *Page 183 of: Iwfhr '02: Proceedings of the eighth international workshop on frontiers in handwriting recognition (iwfhr'02)*. Washington, DC, USA: IEEE Computer Society.

GUYON, I., & ELISSEEFF, A. 2003. An introduction to variable and feature selection. *Journal of machine learning research*, **3**(March), 1157–1182.

GUYON, ISABELLE, GUNN, STEVE, NIKRAVESH, MASOUD, & ZADEH, LOTFI A. 2006. *Feature extraction: Foundations and applications (studies in fuzziness and soft computing)*. Springer.

HAASDONK, B. 2005. Feature space interpretation of svms with indefinite kernels. *Ieee trans. pattern anal. mach. intell.*, **27**(4), 482–492.

HAMAMOTO, Y., UCHIMURA, S., MATSUURA, Y., KANAOKA, T., & TOMITA, S. 1990. Evaluation of the branch and bound algorithm for feature selection. *Pattern recognition letters*, **11**(7), 453–456.

HARTIGAN, J. A., & WONG, M. A. 1979. Algorithm AS 136: A k-means clustering algorithm. *Applied statistics*, **28**(1), 100–108.

HASTIE, T., & TIBSHIRANI, R. 1996. Discriminant adaptive nearest neighbor classification. *IEEE transactions on pattern analisys and machine intelligence*, **18**(6), 607–616.

HECHT-NIELSEN, R. 1991. *Neurocomputing*. Reading, MA,: Addison-Wesley Publishing Company.

HINNEBURG, A., AGGARWAL, C.C., & KEIM, D.A. 2000. What is the nearest neighbor in high dimensional spaces? *Pages 506–515 of:* ABBADI, AMR EL, BRODIE, MICHAEL L., CHAKRAVARTHY, SHARMA, DAYAL, UMESHWAR, KAMEL, NABIL, SCHLAGETER, GUNTER, & WHANG, KYU-YOUNG (eds), *26th international conference on very large data bases*. Cairo, Egypt: Morgan Kaufmann Publishers Inc.

HUGHES, G. 1968. On the mean accuracy of statistical pattern recognizers. *IEEE transactions on information theory*, **14**(1), 5 – 63.

HYVÄRINEN, A., KARHUNEN, J., & OJA, E. 2001. *Independent component analysis*. John Wiley & Sons.

JOHN, G. H., KOHAVI, R., & PFLEGER, K. 1994. Irrelevant features and the subset selection problem. *Pages 121–129 of:* HIRSH, W. W. COHENAND H. (ed), *Proceedings of the eleventh international conference on machine learning, 1994, in new brunswick, NJ, USA, july 10-13*. Morgan Kaufmann.

KATAYAMA, N., & SATOH, S. 1997 (May). The sr-tree: An index structure for high-dimensional nearest neighbor queries. *Pages 369–380 of:* PECKHAM, J. (ed), *ACM SIGMOD international conference on management of data, may 13-15, 1997, tucson, AZ, USA*.

KAUFMANN, L., & ROUSSEEUW, P. J. 1987. Clustering by means of medoids. *Pages 405–416 of:* DODGE, Y. (ed), *Statistical data analysis based on the L1 norm*. Amsterdam: Elsevier/North Holland.

KENDALL, M.G. 2004. *A course in the geometry of n dimensions*. Dover Publications.

KOHAVI, R., & JOHN, G. H. 1997. Wrappers for feature subset selection. *Artificial intelligence*, **97**(1-2), 273–324.

KOHONEN, T. 1995. *Self-organizing maps*. Springer Series in Information Sciences, vol. 30. Berlin, Germany: Springer.

KORN, F., PAGEL, B.-U., & FALOUTSOS, C. 2001. On the "dimensionality curse" and the "self-similarity blessing". *IEEE transactions on knowledge and data engineering*, **13**(1), 96–111.

KRASKOV, A., STÖGBAUER, HARALD, & GRASSBERGER, P. 2004. Estimating mutual information. *Physical review e*, **69**, 066138.

KRIER, C., FRANCOIS, D., WERTZ, V., & VERLEYSEN, M. 2006a (August). Feature scoring by mutual information for classification of mass spectra. *In: FLINS 2006, 7th international flins*

conference on applied artificial intelligence, august 29-31, 2006, genova (italy).

KRIER, C., FRANÇOIS, D., & VERLEYSEN, M. 2006b. Une approche orientée données pour la projection de variables spectrales en spectrométrie. *Pages 53–59 of: Chimiométrie 2006*.

KWAK, N, & CHOI, C.-H. 2002a. Input feature selection by mutual information based on parzen window. *IEEE transactions on pattern analysis and machine intelligence*, **24**(12), 1667–1671.

KWAK, N., & CHOI, C.-H. 2002b. Input feature selection for classification problems. *IEEE transactions on neural networks*, **13**(1), 143–159.

LANCKRIET, G., CRISTIANINI, N., BARTLETT, P., GHAOUI, L. EL, & JORDAN, M. 2004. Learning the kernel matrix with semidefinite programming. *Journal of machine learning research*, **5**, 27–72.

LAND, A. H., & DOIG, A. G. 1960. n automatic method for solving discrete programmingproblems. *Econometrica*, **28**, 497–520.

LANDERS, D., & ROGGE, L. 1987. Laws of large numbers for pairwise independent uniformely integrable random variables. *Mathematische nachrichten*, **130**, 189–192.

LEE, J.A., LENDASSE, A., & VERLEYSEN, M. 2003. Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing*, **57**, 49–76.

LENDASSE, A., SIMON, G., WERTZ, V., & VERLEYSEN, M. 2005. Fast bootstrap methodology for model selection. *Neurocomputing*, **64**, 161–181.

LIN, H.-T., & LIN, C.-J. 2003 (Mar). *A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods*. Tech. rept. National Taiwan University.

LIN, K.-I., JAGADISH, H. V., & FALOUTSOS, C. 1994. The TV-tree: An index structure for high-dimensional data. *VLDB journal: Very large data bases*, **3**(4), 517–542.

LOWE, D. 1995. Similarity metric learning for a variable-kernel classifier. *Neural computation*, **7**(1), 72–85.

MARTINETZ, T., & SCHULTEN, K. 1991. A "neural gas" network learns topologies. *Pages 397–402 of:* KOHONEN, T., MÄKISARA, K., SIMULA, O., & KANGAS, J. (eds), *Artificial neural networks*. Amsterdam: Elsevier.

MESSER, K., MATAS, J., KITTLER, J., LUETTIN, J., & MAITRE, G. 1999 (March). XM2VTSDB: The extended M2VTS database. *Pages 72 – 77 of: Proc. 2nd int. conf. on audio- and video-based person authentication (AVBPA'99),.*

MOODY, J. 1992. The effective number of parameters: an analysis of generalization and regularization in nonlinear learning systems. *Pages 847–854 of:* MOODY, J., HANSON, J., & LIPPMANN, R. (eds), *Advances in neural information processing systems 4*. Palo Alto: Morgan Kaufmann.

MOODY, J. E., & DARKEN, C. 1989. Fast learning in networks of locally-tuned processing units. *Neural computation*, **1**, 281–294.

ONG, C.S., MARY, X., CANU, S., & SMOLA, A. J. 2004. Learning with non-positive kernels. *Page 81 of: Icml '04: Proceedings of the twenty-first international conference on machine learning*. New York, NY, USA: ACM Press.

OPDYKE, J.D. 2003. Fast permutation tests that maximize power under conventional monte carlo sampling for pairwise and multiple comparisons. *Journal of modern applied statistical methods*, **2**(1), 27–49.

ORR, M.J.L. 1995. Regularisation in the selection of radial basis function centres. *Neural computation*, **7**(3), 606–623.

PARDALOS, P., & J.H. GLICK, AND J.B. ROSEN. 1987. Global minimization of indefinite quadratic problems. *Computing*, **39**, 281–291.

PARK, J., & SANDBERG, I. W. 1991. Universal approximation using radial-basis-function networks. *Neural computation*, **3**(2), 246–257.

PEKALSKA, E., PACLIK, P., & DUIN, R. P. W. 2002. A generalized kernel approach to dissimilarity-based classification. *Journal of machine learning research*, **2**, 175–211.

PELCKMANS, K, BRABANTER, J DE, SUYKENS, J. A.K., & MOOR, B. DE. 2005. The differogram: Non-parametric noise variance estimation and its use for model selection. *Neurocomputing*, **69**(1), 100–122.

POINCARÉ, H. 1963. *Mathematics and science: Last essays*. New York: Dover Publications, Inc.

POWELL, M. 1987. Radial basis functions for multivariable interpolation: a review. *Pages 143–167 of:* J.C. MASON, M.G. COX (EDS.) (ed), *Algorithms for approximation*. New York, NY, USA: Clarendon Press.

QUINLAN, J. ROSS. 1993. *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

RAMSAY, J. O., & SILVERMAN, B. W. 1997. *Functional data analysis*. Springer.

RAO, C. 1948. The utilisation of multiple measurements in problems of biological classification. *Journal of the royal statistical society b*, **10**, 159–203.

RAUDYS, S.J., & JAIN, A.K. 1991. Small sample size effects in statistical pattern recognition: recommendations for practitioners. *IEEE transactions on pattern analysis and machine intelligence*, **13**, 252–262.

RICH CARUANA, VIRGINIA R. DE SA. 2003. Benefitting from the variables that variable selection discards. *Journal of machine learning research*, **3**(March), 1245–1264.

ROSSI, F., LENDASSE, A., FRANÇOIS, D., WERTZ, V., & VERLEYSEN, M. 2005. Mutual information for the selection of relevant variables in spectrometric nonlinear modelling. *Chemometrics and intelligent laboratory systems*, **2**, 215–226.

ROSSI, F., FRANÇOIS, D., WERTZ, V., MEURENS, M., & VERLEYSEN., M. 2006. Fast selection of spectral variables with b-spline compression. *Chemometrics and intelligent laboratory systems*, **to appear**(80).

RUCKER, R., & POVILAITIS, D. 1984. *The fourth dimension: A guided tour of the higher universes*. Boston: Houghton Mifflin Company,.

RUI, Y., HUANG, T. S., ORTEGA, M., & MEHROTRA, S. 1998. Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE transactions on circuits and systems for video technology, special issue on segmentation, description, and retrieval of video content*, **8**(5), 644–655.

RUSSELL, S. J., & NORVIG., P. 2003. *Artifical intelligence: A modern approach*. 2nd edn. Prentice Hall.

SALTON, G., & BUCKLEY, C. 1990. Improving retrieval performance by relevance feedback. *Journal of the american society for information science*, **41**(June), 288–297.

SCHÖLKOPF, B., & SMOLA, A. 2002. *Learning with kernels.* Cambridge, MA: MIT Press.

SCHOENBERG, I. J. 1938. Metric spaces and completely monotone functions. *The annals of mathematics*, **39**(4), 811–841.

SCOTT, D., & THOMPSON, J. 1983. Probability density estimation in higher dimensions. *Pages 173–179 of:* DOUGLAS, S.R. (ed), *Computer science and statistics. proceedings of the fifteenth symposium on the interface*. Amsterdam, The Nederlands: North Holland-Elsevier.

SCOTT, D.W. 1992. *Multivariable density estimation: Theory, practice, and visualization*. New-York: Wiley.

SHANNON, C.E., & WEAVER, W. 1949. *The mathematical theory of communication*. Urbana, IL: University of Illinois Press.

SILVERMAN, B.W. 1986. *Density estimation for statistics and data analysis*. Chapman and Hall.

SKALA, M. 2005. Measuring the difficulty of distance-based indexing. *Pages 103–114 of:* CONSENS, MARIANO P., & NAVARRO, GONZALO (eds), *String processing and information retrieval, 12th international conference, SPIRE 2005, buenos aires, argentina, november 2-4, 2005, proceedings*. Lecture Notes in Computer Science, vol. 3772. Springer.

SMOLA, A., MURATA, N., SCHÖLKOPF, B., & MÜLLER, K.-R. 1998. Asymptotically optimal choice of ε-loss for support vector machines. *Pages 105–110 of:* NIKLASSON, L., BODÉN, M., & ZIEMKE, T. (eds), *Proceedings of the international conference on artificial neural networks*. Perspectives in Neural Computing. Berlin: Springer.

STEFÁNSSON, A., KONCAR, N., & JONES, A. J. 1997. A note on the gamma test. *Neural computing & applications*, **5**(3), 131–133.

STEINBACH, M, ERTÖS, L., & KUMAR, V. 2004. The challenges of clustering high dimensional data. *Pages 273–307 of:* WILLE, L. T. (ed), *New directions in statistical physics : Econophysics, bioinformatics, and pattern recognition*. Berlin: Springer.

STEPHEN P. BROOKS, NIAL FRIEL, & KING, RUTH. 2003. Classical model selection via simulated annealing. *Journal of the royal statistical society: Series b (statistical methodology)*, **65**(2), 503–520.

STOPPIGLIA, H., DREYFUS, G., DUBOIS, R., & OUSSAR, Y. 2003. Ranking a random feature for variable and feature selection. *Journal of machine learning research*, **3**, 1399–1414.

SUYKENS, J., GESTEL, T. VAN, BRABANTER, J. DE, MOOR, B. DE, & VANDEWALLE, J. 2002. *Least squares support vector machines*. World Scientific.

TAKAGI, T., & SUGENO, M. 1985. Fuzzy identification of systems and its applications to modeling and control. *IEEE transactions on systems, man and cybernetics*, **SMC-15**(1), 116–132.

TENENBAUM, J. B., DE SILVA, V., & LANGFORD, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, **290**(5500), 2319–2313.

VAPNIK, V. 2000. *The nature of statistical learning*. 2nd edition edn. Springer.

VAPNIK, VLADIMIR N. 1998. *Statistical learning theory*. Wiley-Interscience.

VERLEYSEN, M. 2003. Learning high-dimensional data. *Pages 141–162 of:* ABLAMEYKOA, S., GORAS, L., GORI, M., & PIURI, V. (eds), *Limitations and future trends in neural computation*. Amsterdam (The Netherlands): OS PressIOS Press.

WARDJR, J.H. 1963. Hierarchical grouping to optimize an objective function. *Journal of the american statistical association*, **58**(301), 236–244.

YANG, J., & HONAVAR, V. G. 1998. Feature subset selection using a genetic algorithm. *IEEE intelligent systems*, **13**(2), 44–49.

YIANILOS, P. 1999 (January). *Excluded middle vantage point forests for nearest neighbor search*. Tech. rept. NEC Research Institute. Presented at the Sixth DIMACS Implementation Challenge: Near Neighbor Searches workshop, January 15, 1999.

YU, K., JI, L., & ZHANG, X. 2002. Kernel nearest-neighbor algorithm. *Neural processing letters*, **15**(2), 147–156.

Index

References

Aggarwal & Yu [2001] . . . 18, 151
Aggarwal *et al.* [2001] 35, 36, 39, 40, 42, 65, 77, 82, 91, 151
Aha [1997] 21, 27, 45, 58, 151
Alsberg [1993] 124, 151
Atkeson *et al.* [1997] . 45, 53, 151
Battiti [1994] 106, 151
Beckenbach & Bellman [1971] . . 31, 151
Bellman [1961] 14, 79, 151
Benoudjit & Verleysen [2003] 41, 48, 60, 114, 138, 151
Berchtold *et al.* [1997] . . . 37, 151
Beyer *et al.* [1999] . 16, 34, 35, 37, 39, 75, 152
Biau *et al.* [2005] 96, 152
Blum & Langley [1997] . 116, 152
Bonnlander & Weigend [1994] . . 106, 152
Borggaard & Thodberg [1992] 9, 127, 152
Boser *et al.* [1992] 27, 59, 152
Burges [1998] 57, 152
Carothers [2004] 67, 152
Chapelle *et al.* [2006] 59, 152
Chavez *et al.* [2001] . . 43, 83, 152
Cortes & Vapnik [1995] . . 44, 45, 152
Cover & Hart [1967] 27, 152
Cover & Thomas [1991] 36, 133, 134, 153
D.J. Newman & Merz [1998] 91, 153
Dash & Liu [1997] 116, 153
Demartines [1994] . . . 36, 74, 153
Deng & Moore [1998] . . 114, 153
Dijck & Hulle [2006] . . . 126, 153
Donoho [2000] 7, 153
Efron & Tibshirani [1993] . . 111, 153
Evangelista *et al.* [2006] . . 18, 21, 153
Evans *et al.* [2000] 33, 153
Fausett [1994] 102, 153
Fleuret [2004] 106, 153
François *et al.* [2005] . 42, 65, 153
Francois *et al.* [tba] . . 35, 36, 153
Friedman [1991] 134, 154
Friedman [1997] 17, 21, 154
Fukunaga & Hayes [1989] . . . 19, 154
Günter & Bunke [2002] . 117, 154
Golli *et al.* [2004] 29, 154
Golub & van Loan [1996] 46, 154
Good [1994] 129, 154
Guyon & Elisseeff [2003] . 7, 116, 154
Guyon *et al.* [2006] 104, 154
Haasdonk [2005] 58, 154
Hamamoto *et al.* [1990] 114, 154

Hartigan & Wong [1979] 27, 154
Hastie & Tibshirani [1996] ... 53, 154
Hecht-Nielsen [1991] 11, 154
Hinneburg *et al.* [2000] .. 34, 39, 76, 155
Hughes [1968] 18, 155
Hyvärinen *et al.* [2001] . 127, 155
John *et al.* [1994] 110, 155
Katayama & Satoh [1997] 37, 155
Kaufmann & Rousseeuw [1987] 29, 155
Kendall [2004] 11, 155
Kohavi & John [1997] .. 116, 155
Kohonen [1995] 27, 155
Korn *et al.* [2001] 37, 97, 155
Kraskov *et al.* [2004] ... 107, 155
Krier *et al.* [2006a] 112, 155
Krier *et al.* [2006b] 128, 156
Kwak & Choi [2002a] .. 106, 156
Kwak & Choi [2002b] .. 106, 156
Lanckriet *et al.* [2004] ... 54, 156
Land & Doig [1960] 114, 156
Landers & Rogge [1987] . 93, 156
Lee *et al.* [2003] 83, 156
Lendasse *et al.* [2005] .. 111, 156
Lin & Lin [2003] 57, 156
Lin *et al.* [1994] 37, 156
Lowe [1995] 54, 156
Martinetz & Schulten [1991] . 27, 156
Messer *et al.* [1999] 9, 156
Moody & Darken [1989] 45, 157
Moody [1992] 20, 156
Ong *et al.* [2004] 58, 157
Opdyke [2003] 130, 157
Orr [1995] 21, 157
Pardalos & J.H. Glick [1987] . 58, 157
Park & Sandberg [1991] .. 29, 44, 157
Pekalska *et al.* [2002] 58, 157
Pelckmans *et al.* [2005] .. 68, 157
Poincaré [1963] 9, 157
Powell [1987] 27, 58, 157
Quinlan [1993] 102, 157
Ramsay & Silverman [1997] . 14, 123, 157
Rao [1948] 7, 157
Raudys & Jain [1991] 19, 158
Rich Caruana [2003] ... 117, 158
Rossi *et al.* [2005] 108, 158
Rossi *et al.* [2006] 125, 158
Rucker & Povilaitis [1984] 9, 158
Rui *et al.* [1998] 69, 158
Russell & Norvig. [2003] ... 111, 158
Salton & Buckley [1990] . 69, 158
Schölkopf & Smola [2002] ... 29, 158
Schoenberg [1938] 57, 158
Scott & Thompson [1983] 16, 158
Scott [1992] 106, 158
Shannon & Weaver [1949] .. 105, 158
Silverman [1986] 15, 159
Skala [2005] 78, 159
Smola *et al.* [1998] 60, 159
Stefánsson *et al.* [1997] . 68, 108, 110, 159
Steinbach *et al.* [2004] ... 16, 159
Stephen P. Brooks & King [2003] 115, 159
Stoppiglia *et al.* [2003] . 112, 159
Suykens *et al.* [2002] 59, 159
Takagi & Sugeno [1985] . 27, 159
Tenenbaum *et al.* [2000] . 83, 159
Vapnik [1998] 57, 159
Vapnik [2000] 20, 159
Verleysen [2003] 7, 160
WardJr [1963] 27, 160
Yang & Honavar [1998] 115, 160
Yianilos [1999] 77, 160
Yu *et al.* [2002] 58, 160

A

absolute 39
absolute contrast 34
almost surely 87
attribute 2

B

b-spline 124

backtracking 114
blessing 19
burst noise 69

C

centre 45
centroid 45
centroids 29
class 2
classification 2
clustering 1
clusters 1
collinearity 123
coloured 64
concentration of distances
phenomenon 3
conditional entropy 105
conjecture
independence 84, 94
contrast 76
correlation 105
cover 46
covering 46
curse of capacity 20
curse of dimensionality 3, 14, 20, 21

D

dataset
APPLE 41, 59, 60, 65, 69, 96, 97
BOATGOAT 96, 97
DELVE 139
DELVE-CENSUS 8
DELVE 59, 60
HOUSING 137
HOUSING 59, 60, 69
IONOSPHERE 97
LL 61
LSSVM 61
MUSK 97
NITROGEN 139
NITROGEN 125
RBFN 61
SVM 61
TECATOR . 8, 59, 61, 69, 96, 97, 127, 142
WINE 59, 60, 69
XM2VTS 9, 34
YESNO 96, 97
degrees of freedom 74
dimension 2
dimensionality
curse 14
dissimilarity matrices 28

E

effective number of parameters . 21
embedding 83
empty 16
empty space phenomenon ... 79
empty space phenomenon 3
empty space shenomenon ... 16
ensemble 117

F

feature 2
feature selection 22, 102
feature space 2
feature vector 29
feature vectors 28
festuca elatior 139
final prediction error 21
forward feature selection ... 114
fractal 97
functional data 14, 123

G

generalised gaussian kernel . 54
genetic algorithm 114
gram matrix 57
greedy methods 114

H

high-dimensional 3
hypercube 10
hypersphere 11

I

i.i.d. 35, 75, 77, 78, 82, 92, 97
identically distributed 83, 92
independent 83

intrinsic dimension 83

K

kernel
  covering 46
  definition 29, 44
  Gaussian 45
  interpretation 44
  locality 46
  methods 29
  width 45
kernel 28, 44

L

lemma
  expectation of the norm 80, 84
  variance of the norm 80
local 45
locality 46

M

manifold 46
mercer conditions 44
mercer kernels 29
metric 2
metric 2
metric learning 53
mutual information 105

N

neighbourhood 28, 45
norm
  fractional 31
  Minkowski 30
numerical features 2

O

optimism 111

P

parsimonious 22
phenomenon
  concentration of distances . 30
  empty space 16
positive semi-definite 57
projection 22
proposition
  identical distributions .. 83, 92
  Relative variance with uniform data 82

R

radial-basis 29
ranking 28
regression 1
relative contrast
  definition 33
  theorem 35
relative contrast 34, 76
relative variance
  definition 33
  theorem 35
relative variance 34, 79
relevance 116
relevance criterion 103
robustness 40

S

Schoenberg 57
search procedure 103
sigmoid 57
simulated annealing 115
smoothness prior 17
stability 37
stable 38
submanifold 21
support vector 45
support vectors 59

T

temperature 115
tesseract 10
theorem
  Aggarwal 77
  Beyer 75
  Concentration and norm .. 81
  Demartines 74
  Hinneburg 76
  limit ratio 90
  Relative contrast (Beyer) .. 35

Relative variance ... 35, 80, 84
Schoenberg 57
tool
1-NN 68, 69, 72, 108
C4.5 102
ICA 127
K-MEANS 27, 29, 37, 65
K-MEDOIDS 29
k-NN .. 2, 3, 17, 21, 27–29, 37, 52, 58
KkNN 57, 58
Lazy Learning 44
LL 58–62
LR 126
LSSVM 59, 61, 62
LVQ 27–29
MLP 20, 102
PCR 126
PLSR 126
RBFN ... 21, 22, 27, 29, 44, 55, 58–62, 69, 102, 103, 125, 126, 138–141
SOM 27, 29, 37, 65
SVM 18–21, 27, 29, 44, 45, 55–57, 59, 61, 62, 102, 103
tools
distance-based 27
feature vector-based 29
kernel-based 28
matrix distance-based 29
rank-based 28
transductive learning 54

V

vapnik-chervonenkis dimension 20
variable 2
vc-dimension 20

W

width 28, 44