NON-METRIC PAIRWISE PROXIMITY DATA

VORGELEGT VON
DIPL. ING. PHYS. JULIAN LAUB
AUS BERLIN

VON DER FAKULTÄT IV – ELEKTROTECHNIK UND INFORMATIK
DER TECHNISCHEN UNIVERSITÄT BERLIN
ZUR ERLANGUNG DES AKADEMISCHEN GRADES EINES
DOKTORS DER NATURWISSENSCHAFTEN
GENEMIGTE DISSERTATION

Promotionsausschuss

Vorsitzender: Prof. Dr. K. Obermayer
Gutachter: Prof. Dr. S. Jähnichen
Gutachter: Prof. Dr. K.-R. Müller (Uni Potsdam)
Gutachter: Prof. Dr. J. M. Buhmann (ETH Zürich)

Tag der wissenschaftlichen Aussprache

10. Dezember 2004

BERLIN, OKTOBER 2004
D 83

SUMMARY

There are two common data representations in intelligent data analysis, namely the vectorial representation and the pairwise representation. The translation of latter into the former is called embedding. This is a non-trivial issue of ongoing scientific interest. While the pairwise representation imposes less restriction on the data and is thus potentially able to capture richer structure, the vectorial representation has the advantage to offer many powerful data analytical tools, in particular as a consequence of the existence of probabilistic data models in such spaces.

Pairwise data satisfying restrictive conditions can be faithfully translated into a vectorial representation. Pairwise data, for which this is not possible are called *non-metric* pairwise data.

This thesis is about non-metric pairwise data. It is an investigative and explorative study of non-metric pairwise data, based on theoretical and conceptual as well as empirical considerations. The reader is first made familiar with the two data representations. Pairwise data are illustrated and first issues raised. Common embedding strategies are developed. It is then shown that these two data representations coincide for a certain class of learning algorithms, even when the pairwise data is non-metric and traditional techniques only obtain approximate vector representations. The new embedding developed is *exact* with respect to structure.

The major focus lies on apprehending the nature and consequences of metric violations. While the scientific community seems aware of such an issue, it has never been clearly formulated to the best of the authors knowledge. Metric violations have commonly been considered an accidental byproduct of noise and have received corresponding mathematical treatment. It is shown by simple modeling of metric violations that this assumption is wrong. A particular embedding method is used to visualize and interpret the information coded by metric violations.

Finally the structure coded by metric violations is shown to be efficiently extracted by a simple algorithm which evaluates structure based on a stability index.

KEYWORDS. Pairwise data, exploratory data analysis, machine learning, clustering, embedding, visualization, metric violations, multidimensional scaling, feature discovery, structure learning.

ZUSAMMENFASSUNG

In intelligenter Datenanalyse gibt es zwei gängige Datenrepräsentationen, nämlich die vektorielle Repräsentation und die paarweise Repräsentation. Die Übersetzung der letzteren in die erstgenannte nennt man Einbettung, eine nicht triviale Problematik von stetem, wissenschaftlichen Interesse. Während die paarweise Repräsentation den Daten weniger Einschränkungen auferlegt und so potentiell fähig ist, reichere Struktur festzuhalten, wartet die vektorielle Repräsentation mit vielen mächtigen datenanalytische Werkzeugen auf, da man in solchen Räumen über probabilistische Modelle für die Daten verfügt.

Paarweise Daten, die restriktive Bedingungen erfüllen, können getreu in eine vektorielle Repräsentation abgebildet werden. Paarweise Daten, für die dies nicht möglich ist, werden *nicht metrisch* genannt.

Diese Doktorarbeit betrifft nicht-metrische, paarweise Daten. Es ist eine investigative und explorative Studie nicht metrischer, paarweiser Daten, gestützt auf theoretische und konzeptuelle, sowie auf empirische Betrachtungen. Zuerst wird der Leser mit den beiden Datenrepräsentationen vertraut gemacht. Paarweise Daten werden illustriert und die ersten Problematiken angesprochen. Gängige Einbettungsmethoden werden dargestellt. Dann wird gezeigt, dass diese beiden Datenrepräsentationen für eine gewisse Klasse von Lernalgorithmen übereinstimmen, sogar wenn die paarweisen Daten nicht metrisch sind, und traditionelle Techniken nur zu approximativen Vektorrepräsentation führen. Die neuentwickelte Einbettung ist *exakt* in Bezug auf Struktur.

Das Hauptgewicht liegt im Erfassen der Natur und der Folgen von metrischen Verletzungen. Obwohl die wissenschaftliche Gemeinschaft die Problematik wahrzuhaben scheint, wurde diese nach des Autors bestem Wissen nie klar formuliert. Metrische Verletzungen wurden gemeinhin als zufälliges Nebenprodukt von Rauschen betrachtet und wurden mathematisch dementsprechend behandelt. Eine einfache Modellierung metrischer Verletzungen zeigt, dass diese Annahme falsch ist. Eine spezielle Einbettung wird benutzt um den Informationsgehalt metrischer Störungen zu visualisieren und interpretieren.

Schliesslich wird gezeigt, dass ein einfacher Algorithmus, der die Struktur über einen Stabilitätsindex auswertet, effizient die Struktur, die von metrischen Verletzungen kodiert wird, extrahieren kann.

SCHLÜSSELWÖRTER. Paarweise Daten, explorative Datenanalyse, maschinelles Lernen, Clustering, Einbettung, Visualisierung, metrische Verletzungen, multidimensional scaling, Merkmalentdeckung, Lernen von Struktur.

ACKNOWLEDGMENTS

First of all I would like to thank my supervisors Prof. Dr. Klaus-Robert Müller, Prof. Dr. Joachim Buhmann and Prof. Dr. Stefan Jähnichen. Without their help, sometimes silent, nevertheless fundamental, this thesis could not have been written. Of course, I owe the utmost gratitude to Prof. Dr. Klaus-Robert Müller who through his constant encouragements and participation, his deep scientific knowledge and curiosity and above all his wit and legendary optimism made this work only possible. He helped me go through the dark side of research, when everything seems vain, and look forward. I also thank him for leaving me a great freedom in research, an invaluable good in today's world.

I owe a particular recognition to Dr. Volker Roth with whom I had the pleasure to often work in common and who made major contributions to this work. Without our neverending discussions, both of general and very specific nature, this thesis would not be what it is. I really do admire how he allies scientific rigor with everyday pragmatism.

A very technical and mathematical thank goes to Dr. Motoaki Kawanabe with whom I shared the room for many years and who always had an open ear for my questions and time to discuss them. He often continued to think about them long after the discussions ended and came up with proofs or counter examples correcting many of my false intuitions.

The major part of this work has been done in Professor Müller's Group Intelligent Data Analysis at the Fraunhofer FIRST. I would like to thank all people of this institute and in particular the members of our group for the continued pleasure to arrive in my office every morning.

Other parts were done in Professor Buhmann's group, back at the university of Bonn, now at the ETH Zürich. There too, I profited from the welcoming atmosphere and scientific excellence.

Financialwise, I gratefully acknowledge the grants # MU 987/1-1, # BU 914/4-1 and # JA 379/13-2 from the Deutsche Forschungsgemeinschaft, as well as PASCAL Network of Excellence (EU # 506778).

Julian Laub, August 2004.

CONTENTS

1. INTRODUCTION

Data analysis

The subject of this thesis is data analysis. Tautologically yet usefully defined, data analysis is the process of *exploring*, *analyzing* and *understanding* a *set of objects*, typically measurements from natural sciences or engineering. Data analysis encompasses everything from the meticulous and slow item by item examination of Tycho Brahe's astronomical observations to the large scale automated gene finding with intelligent algorithms. It is the prerequisite to every inductive step from a *sample* to a *rule*, from the particular to the general, and thus it is at least as old as modern science starting with Galilei.

Computer science

The faster and richer data acquisition due to the multiplication of scientific interests and invention of more sophisticated experimental techniques called for new tools to process the data. While statistic gives a mathematically rigorous approach to those problems, they were of little use in practice. The tools capable of treating thousands or even millions of objects were only developed with the boom of the *computer*. To such an extend that today, data analysis is largely understood as a particular field of computer science.

Artificial intelligence

Alas, computers happen to be quite dumb when left to themselves and even though one was now able to quickly compute statistical descriptors like the mean or the variance, as quickly it became clear that data would rather reveal their mystery and hidden message to an expert in the field armed with, even modest, a priori knowledge than to a machine. The fact that we dispose of large data sets in a short time and that they could be fed to powerful computers turned out to be often insufficient. The computers had to be made intelligent, at best replacing the human expert, at worst helping him. Unlike believed at the dawn of the computer, the former would turn out to be a long way into the future, even at the beginning of the new century and millennium.

Machine learning

Machine learning is the field of artificial intelligence which studies how machines could be led to apprehend and interpret their environment. The idea is to implement generic algorithms which can learn from, adapt to, and model a given environment given by a set of objects. It is the attempt to implement an inductive procedure in a machine. The computer now no longer acts as a mere tool of an human expert who has all the knowledge but as a *participant* of the process of analysis and understanding.

This participation is still very basic and one should not expect miracles from a machine. Let us be fair: a few hundreds or thousands lines of code and a couple of minutes or hours of autistic learning versus a few billions of neurons and more than twenty years of apprenticeship aided by myriads of people, it

should come to no surprise that machine learning is in its very infancy.

Intelligent data analysis. A possible definition.

In science we do expect every analysis to be intelligent. But as we have seen, the expert knowledge may be finite or not able to handle large data sets. Intelligent data analysis combines the best of both worlds. It develops and uses machines that learn and interact with the experimenter. There is a constant interplay of these two very unequal actors who both need each other. Also, even in the field of machine learning, we should not be afraid of speaking about *human* learning.

This is the spirit of this thesis. Its goal is to understand the phenomenon of non-metric pairwise proximity data, its origin, its signification and its repercussions. This subject will be treated from the perspective from both the machine and the human.

Similarity

The mother of data analysis is *similarity*. Exploring data is looking for equal, similar, dissimilar or differing patterns in order to classify, group, discriminate. The objects composing the data set come from numerous fields of empirical sciences ranging from astronomy and high energy physics, genomics and proteonomics, cognitive psychology and social sciences to web mining and financial stock market analysis. Many of these data sets can be differently analyzed according to a special focus. The abstract data objects themselves do not predetermine the way to go. Similarity does. The similarity encodes meaning and only given a similarity between objects will the data analysis start.

Two data types

There are two main data types, called *vectorial data* and *pairwise proximity data*. The former are *feature* based and the latter are *relation* based. These two data types fostered two different approaches in intelligent data analysis, the geometric approach and the syntactic, or structural approach.

The geometric approach for vectors enjoys the presence of numerous and powerful tools, famous ones being *Support Vector machines* and *Fisher Discriminant Analysis*, but it makes a rather strong assumption on the data, namely that it fits into the quite restrictive structure of a Hilbert space. The structural approach is less developed but is able to treat more generic data.

The problem

It is an ongoing issue of how these two data types and associated approaches translate into each other. The attempts to unify them had a rather marginal existence. The choice of a distance measure allows to pass from vectorial data to pairwise proximity data. However, this choice is not intrinsic to the data and largely determines the outcome of the analysis.

Conversely, it can be shown that when pairwise proximity data satisfy a certain number of requirements, then there is a set of vectors and an appropriate distance measure such that the mutual distance between the vectors is the same as the set or pairwise proximities. *However, this is not the case in general*. This thesis studies such pairwise data, called *non-metric*. It shows that there is a simple and elegant unification of the pairwise and the vectorial representa-

tion in the context of a certain class of algorithms exemplified by the k-means algorithm. It further develops several models on how such pairwise data occur and what different interpretations they admit. It unravels properties related to non-metricity which have so far gone unnoticed.

The study addresses both the issues of machine learning and our understanding of a certain data type. It is organized as follows:

Organization of the thesis

CHAPTER 2. The second chapter introduces pairwise data. It gives a few examples and shows different possible representations of pairwise data. It then recalls a few necessary definitions to the understanding of the difference between vectorial data and pairwise data. Further, the traditional translation of pairwise data to vectors by way of *embedding* is presented both in its exact and its approximated version. A discussion at the end of the chapter elaborates on the issues raised and introduces the main contributions on this thesis.

CHAPTER 3. The third chapter presents a new embedding strategy called *Constant Shift Embedding*. It shows that non-metric pairwise data can still be embedded *loss-free* into a Euclidean space when considering a certain class of learning algorithms. This is a step forward both theoretically and practically. It shows, on one hand, that a unification of vectorial and pairwise data can be found not with respect to metricity but with respect to the outcome of some learning algorithm. On the other hand, it makes a large class of pairwise data available to many powerful tools in a vector space.

CHAPTER 4. The fourth chapter delves into the significance of metric violations. It is mainly conceptual in nature. We distinguish between accidental and inherent non-metricity. Several models are presented that explain the occurrence of inherent non-metricity. We show furthermore how the information hidden in the systematic metric violations can be recovered and illustrate that indeed it must be considered for a deeper understanding of the data.

CHAPTER 5. In the fifth and last chapter, we make a modest but determined step towards automated structure discovery in non-metric pairwise data. The idea is to let the machines profit from the insight we gained in the previous chapters. A small yet efficient algorithm is presented that detects structure by computing a simple stability index. It is shown that it detects the structure coded by metric violations.

A brief discussion concludes this thesis.

NOTATION. This table summarizes the symbols and their explanation. Symbols may have a different signification in a different context but the meaning of a symbol is recalled when judged necessary. In general matrices are denoted by capital letters and their elements as indiced lower case letters.

Symbol	Explanation
A, B	Usually some generic matrices
D	A dissimilarity matrix
S	A similarity matrix
C	A (pseudo-)covariance matrix
T	The counting matrix
P	The amplitude matrix
Q	The projection matrix $I_n - ee^t$
I	The identity matrix, $i_{ij} = 1$ if $i = j$, 0 else
X	The matrix containing vectors x_i
V	The row matrix consisting of eigenvectors
Λ	The diagonal matrix consisting of eigenvalues
x_i	A vector indexed by i.
	This is usually *not* the i^{th} coordinate of some vector x!
v_i	The i^{th} eigenvector
e	The vector $(1, 1, \dots 1)^t$
d_{ij}	The dissimilarity between object i and j
s_{ij}	The similarity between object i and j
d_o	A real constant
λ_i	The i^{th} eigenvalue
α_i	The i^{th} coefficient of an expansion
n	Usually the number of samples in a data set
p	Usually the dimension of a some vector x_i
i, j, k, l	Index variables
$\lvert E \rvert$	Cardinality of the set E
a, b, c, d	Counting variables in binary image matching
ω_{ij}	Some weight
$d(\cdot, \cdot)$	A metric
$\lVert \cdot \rVert$	A norm
$\langle \cdot, \cdot \rangle$	An inner product
$.^t$	The transposed of its argument
$.^c$	The centralized of its argument
$M(\cdot)$	A partition of its input data set
L	The subspace of projections in classical scaling or PCA
$\mathbb{R}$	The set of real number
$\mathcal{C}_D$	The equivalence class of C's yielding a given D

2. PAIRWISE DATA

In this chapter we discuss a specific type of data which arise in a variety of fields in machine learning : We discuss *pairwise data*. After an introduction on two fundamentally different data types we will give several examples of such data and illustrate representation of pairwise data. We then move on to a mathematical formulation and give a definition for the relevant spaces. We have a first glimpse on metric violations, the main topic of this thesis, as well as the issue of embedding pairwise data into a Euclidean space. Common embedding strategies are presented.

§. 2.1. INTRODUCTION.

Two main data types

We will distinguish two data types in this work, namely *vectorial* and *pairwise* data. A *data point* refers to either of these two concepts.

Vectorial representation

Vectorial data, or simply vectors, are data represented in a vector space. A vector space is very general in nature and there is no intrinsic notion of distance in a vector space. At this point, however, our interest lies elsewhere. As a consequence of the axiom of choice, *every* vector space has a basis (Weisstein, 2004), that is, any vector x can be expressed as a linear combination of some minimal set of vectors $e_1, e_2, \ldots$ which span the whole space.

$$x = \sum_i \alpha_i e_i.$$

A (data) point in a vector space can thus be represented as a collection of coefficients α_i, usually real numbers.

$$x = (\alpha_1, \alpha_2, \ldots).$$

(In Section 2.3 the definition of a vector space will be given.)

Features

The basis vectors represent and summarize the whole vector space. In terms of data analysis, they represent *features*, measured quantities which determine a data point.

The intuition on vector spaces is mainly fostered by physics and the notion of measurement. Every object is uniquely defined by some quantitative "variable", the coordinates. In classical mechanics, e.g. the phase space is defined as the space spanned by q and p, where q measures the location of a particle and p its linear momentum. The vector space (q, p) entirely determines the physical state of a system. In data analysis we call q and p features, that is, characteristics we are interested in and which describe our system, the data to be analyzed.

Pairwise representation

Data points from a pairwise data set have *no* features. They do not exists irrespectively of the other points like in a vector space where every point exists as an entity independent of all others. An object in this set is only defined by its relationships to all other objects. For pairwise data, the homologue of features is *relationship*. We do not have access to a set of variables determining a point (such a set might not exist) but only to the *pairwise* relationships among the points, hence their name.

Proximity data

These pairwise relationships are most often given as real numbers representing the degree either of similarity or dissimilarity of the respective pairs of points. Pairwise data is then called pairwise *proximity* data.

$$s_{ij} \in \mathbb{R} \text{ for all } i = 1, 2, \ldots \text{ and } j = 1, 2, \ldots$$

From now on, we will omit "proximity", assuming that our pairwise data is given as similarities or dissimilarities. Note that the distinction between vectorial data and pairwise data exists independently of the notion of similarity. However, the notion of similarity (or dissimilarity) is the basis of data analysis. Without the notion of similarity, there is no data analysis and no machine learning.

For vectorial data, one needs first to define a similarity. While this is quite natural as we shall see in Section 2.3, it still is subject to a choice which strongly influences the outcome of the subsequent data analysis. In pairwise data, the similarities are given beforehand, they *are* the data. These similarity usually are of more general nature than the ones obtained by an a posteriori choice of e.g. an inner product in a vector space. *Pairwise data can capture structure which can inherently not be captured by vectorial data.*

Vectorial vs. syntactic approach

These two main data types call for different analytical approaches. Following Goldfarb (1985) we will distinguish between "geometric" approaches to handle vectorial data and "syntactic" (or "structural") approaches designed for pairwise data. It is important to stress that these approaches differ in nature and do not naturally carry over one into the other. As we have mentioned in the introduction, the bulk of analytical tools are only available for vectorial data. This raises the issue of transforming pairwise data into vectorial data, a procedure known as *embedding*. This problem is all but trivial. Embedding will be the topic of Section 2.4 and Section 2.5.

§. 2.2.

EXAMPLES OF PAIRWISE DATA.

In this section we will give some of examples of pairwise data. In particular we will speak about their different possible representations.

Occurrence of pairwise data

Whereas vectors in a vector space are the result of measurements with respect to certain chosen characteristics, the pairwise data arise as direct comparison between different data points. These comparisons most naturally express a similarity of a dissimilarity of the respective two objects.

There are many possibilities for pairwise data to occur. A few fields are:

- Bioinformatics. Pairwise data occur e.g. in genomics, as alignment scores between two DNA or protein sequences obtained by an alignment algorithm, see for instance Altschul et al. (1997) or Pearson and Lipman (1988). These pairwise data are the starting point for large scale structure of function prediction of proteins.

- Text or web mining. Pairwise data occur as similarities between different texts. The similarity measure can be of simple nature, counting e.g. co-occurrence of certain words, or more complex, measuring topical closeness. Subsequent data analysis permits to classify text documents based on these pairwise comparisons. See e.g. Hofmann et al. (1998), Jacobs et al. (2000).

- Cognitive psychology and social sciences. Pairwise data occur as human similarity judgments (Gati and Tversky, 1982, Goldstone et al., 1991). Human test subjects rate the similarity of a pair of objects on a predefined scale. Psychologist gain insight into mental processes by analysis these pairwise data. It can also occur as result of pairwise comparisons in social sciences, called preference data, or as output of some social comparison of e.g. countries.

Table 2.2 gives a simple instance of pairwise data obtained from human similarity judgments of the auditory morse code (Everitt and Rabe-Hesketh, 1997). The entries correspond to the percentage of a large number of observers who responded “same” to the row signal followed by the column signal. We note that this proximity matrix is *asymmetric*. Furthermore we note that there is no one unique upper similarity indicating that two signals are identical (given by the diagonal of the matrix). For dissimilarities it is natural to request that the diagonal be zeros, i.e. that the dissimilarity of an object to itself be zero.

Asymmetry

It is not uncommon for pairwise data to be asymmetric, i.e. for two object to have different similarity according to the order in which they are presented.

	1	2	3	4	5	6	7	8	9	10
1 (.----)	84	63	13	8	10	8	19	32	57	55
2 (..---)	62	89	54	20	5	14	20	21	16	11
3 (...--)	18	64	86	31	23	41	16	17	8	10
4 (....-)	5	26	44	89	42	44	32	10	3	3
5 (.....)	14	10	30	69	90	42	24	10	6	5
6 (-....)	15	14	26	24	17	86	69	14	5	14
7 (--...)	22	29	18	15	12	61	85	70	20	13
8 (---..)	42	29	16	16	9	30	60	89	61	26
9 (----.)	57	39	9	12	4	11	42	56	91	78
10(-----)	50	26	9	11	5	22	17	52	81	94

Table 2.1. Test subjects are asked to judge the pairwise similarity of auditory morse code (long and short tones). The entries correspond to the percentage of a large number of observers who responded "same" to the row signal followed by the column signal.

In cognitive psychology, asymmetry comes as a consequence of the mere complexity of the though process. Other examples of asymmetric data involve e.g. the well known salesman problem, where the journey time from town A to town B may very well vary from the journey time from town B to town A, particularly in hilly regions. A further example is the similarity between people: a child is often seen similar to one or both of its parents, whereas a parent is rarely considered similar to his child (Borg and Groenen, 1997).

Representation as a graph...

There are several standard ways to represent pairwise data. Let us consider a toy data set given by 25×25 dissimilarities of five points A through E. Figure 2.1 gives the pairwise data represented by a weighted graph. A weighted graph is a pair $(V, \mathcal{E})$ of vertices V and weighted edges $\mathcal{E}$. In our example, the vertices are given by the points A to E and the weighted edges by the dissimilarities. If there are no missing values, i.e. all dissimilarities are known, the graph is fully connected.

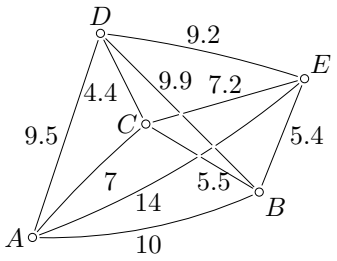


Figure 2.1. Representation of pairwise dissimilarities as a graph. The data points are given by the vertices A to E, the pairwise relation between them by the weighted edges. Symmetric similarities correspond to undirected graphs. Note that the vertices must not be confounded with points in a vector space!

The representation of pairwise data as a graph mainly serves to illustrate theoretic issues and concept. For real data sets it quickly becomes cumbersome and untracktable. It must be stressed here that the vertices do *not* correspond to points in a two dimensional space: only in rare cases may pairwise data be represented in two dimensions.

... a table...

A more common way to represent pairwise data—and the most natural one—is to simply list the values of the similarity or dissimilarity in a table. Table 2.2 gives the same toy data set as seen from this representation. A table naturally

	A	B	C	D	E
A	0	10	7	9.5	14
B	10	0	5.5	9.9	5.4
C	7	5.5	0	4.4	7.2
D	9.5	9.9	4.4	0	9.2
E	14	5.4	7.2	9.2	0

Table 2.2. Representation of pairwise dissimilarities as a table or as a matrix. This is a natural representation in the sense that similarities or dissimilarities are given as numbers. However, for large data sets it becomes awkward to print and quite illegible.

... or a checkerboard pattern

carries over to a matrix, thus making the dissimilarities available to the myriad of algebraic matrix operations. *All subsequent treatments of pairwise data use the matrix notations*. While the table or matrix notation is the only mathematical usable one, it quickly becomes awkward to display large sets of pairwise data. To this effect, one replaces the matrix of values by a matrix of e.g. gray values representing these values: see Figure 2.2. One easily squeezes thou-

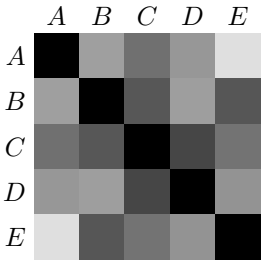


Figure 2.2. Representation of pairwise dissimilarities as a checkerboard pattern. The values of the matrix are represented as colors or gray values. This representation is of great advantage for large data sets, consisting of hundreds or thousands of elements. Note that the values are taken from Figure 2.1: the gray squares are symmetric around the diagonal. Dark values represent small dissimilarities, light values large dissimilarities.

sands of data points into such a representation. While we can not single out individual similarity values, we have a good overview of the global structure. Figure 3.8 on page 55 shows three dissimilarity matrices for pairwise aligned protein sequences. We see different structures which we would probably miss when looking at the numbers.

A further advantage is that the structure is not swamped by scale when comparing different pairwise data sets. The gray values only reflect structure. See

e.g. Figure A.3 on page 110 for three different matrices that have a similar structure while their individual values are on totally different scales.

ISSUES.

RELATIONSHIP BETWEEN SIMILARITY AND DISSIMILARITY. Up to now, we invariably spoke of similarities or dissimilarities. In a certain way we will continue to do so. There is no natural relationship between them, even though it will often prove necessary to move between them, namely pass from a similarity measure to a dissimilarity measure. This point will be discussed in further detail in Section 2.4.

ASYMMETRY. We saw that pairwise data may be asymmetric. Asymmetry calls for non standard procedures. We will not discuss asymmetric pairwise data. Part of our results will be valid for asymmetric pairwise data, other will not. Mention of the particular requirements will be made in due course.

MISSING VALUES. Sometimes not all pairwise relationships are known. The data set is said to have missing values. We do not discuss these cases which require special treatment.

§. 2.3.

MATHEMATICAL STATEMENT.

In this section we will formalize pairwise data. We first will introduce the definition of the spaces necessary for a mathematical treatment. We will recall the definition of a distance function and the implication for pairwise proximity data. Violations of the requirements on distance functions are discussed in a first approach. The problem of embedding is reviewed.

DEFINITIONS.

We start by recalling the definitions of four important spaces, namely a vector space, a metric space, a Hilbert space and a Euclidean space. Minimum algebraic and analytic knowledge is taken as a prerequisite.

DEFINITION 2.3.1. Let K be a field. A *vector space* over K is an abelian group $(E, +)$ and an application $(\lambda, x) \to \lambda x$ of $K \times E$ to E such that

Vector space

$$\begin{aligned}
&\text{For all } \lambda, \mu \in K \text{ and for all } x \in E, \lambda(\mu x) = (\lambda\mu)x,\\
&\text{For all } \lambda \in K \text{ and for all } x, y \in E, \lambda(x + y) = \lambda x + \lambda y,\\
&\text{For all } \lambda, \mu \in K \text{ and for all } x \in E, (\lambda + \mu)x = \lambda x + \mu x,\\
&\text{For all } x \in E, 1_K x = x.
\end{aligned}$$

Common vector spaces in data are the set of p-tuples K^p over a certain field K. With $K = \mathbb{C}$ or $K = \mathbb{R}$ we recover the well known complex or real vector space. For the remainder of this work, we invariably will take $K = \mathbb{R}$ and omit the mention to more general fields.

If a set $\{e_1, e_2, \ldots\}$ is minimal (i.e. no two vectors are a linear combination of each other) and spans the vector space it is called a *basis*. The number of basis vectors is called the *dimension*. It may be finite of infinite.

A vector space is a very general space and as abstract as one wishes. It has no notion of similarity and is, as such, of little importance for us. We need the concept of *measurement* so that objects can be described as similar or dissimilar.

DEFINITION 2.3.2. The pair (E, d) is called a *metric space* if E is a non-empty set and the function $d : E \times E \to \mathbb{R}$ satisfies the following conditions:

Metric space

$$\begin{aligned}
&d(x, y) \geqslant 0 \text{ for all } x, y \in E,\\
&d(x, y) = 0 \text{ if and only if } x = y,\\
&d(x, y) = d(y, x) \text{ for all } x, y \in E,\\
&d(x, y) \leq d(x, z) + d(z, y) \text{ for all } x, y, z \in E.
\end{aligned}$$

A metric space (E, d) is said to be *complete* if every Cauchy sequence converges in E. Note that E need *not* be a vector space. Any set of object is a metric space when endowed with a function that satisfies the proper requirements, given above. We will come back to these requirements and therefore postpone a closer look at them.

A vector space and a metric space are two independent notions of great generality and not restrictive enough for many situations to be useful. We therefore define the following:

DEFINITION 2.3.3. E is called a real *Hilbert space* if E is vector space and if it has an complete inner product, that is, a function $\langle \cdot, \cdot \rangle : E \times E \to \mathbb{R}$,

Hilbert space

$(x, y) \mapsto \langle x, y\rangle$, such that

$$\begin{aligned}
&\langle x, x\rangle \geqslant 0 \text{ for all } x,\\
&\langle x, x\rangle = 0 \text{ if and only if } x = 0,\\
&\langle x, y\rangle = \langle y, x\rangle \text{ for all } x \text{ and } y,\\
&\langle x, \lambda_1 y_1 + \lambda_2 y_2\rangle = \lambda_1\langle x, y_1\rangle + \lambda_2\langle x, y_2\rangle \text{ for all } x, y_1, y_2, \lambda_1, \lambda_2.
\end{aligned}$$

An inner product induces a natural norm via $\|x\| = \sqrt{\langle x, x\rangle}$, which in its turn induces a natural distance via $d(x, y) = \|x - y\|$. We thus have a vector space in which we have a measure for the similarity of points. *The large majority of data analytical tools is formulated for points lying in a Hilbert space*. This is in particular due to the fact that in such a space we have a probabilistic formulation of the data, i.e. we can assume that it be drawn from some random source with a certain distribution $x_i \sim P(x)$. Sometimes, by abuse of language, we say vector space instead of Hilbert space, assuming the preexistence of some inner product.

Hilbert spaces are ubiquitous. In physics they play a major role, not only in quantum mechanics for the space of wave functions but also in special relativity and in classical mechanics, in which a particularly simple Hilbert space is used, namely the Euclidean space:

Euclidean space

DEFINITION 2.3.4. E is called a (real) Euclidean space if $E = \mathbb{R}^p$ and if the inner product and the norm are given by:

$$\langle x, y\rangle = \sum_{i=1}^{p} x_i y_i \text{ and } \|x\| = \sqrt{\langle x, x\rangle}. \tag{2.1}$$

The norm $\|x\| = (\sum_{i=1}^{p} x_i^2)^{1/2}$ is called the *Euclidean norm* and is usually denoted by $\|\cdot\|_2$. The Euclidean space is the mathematical replica of our intuitions notion of space. Its the mother of all spaces. In three dimensions it is the physical world we perceive. In machine learning low dimensional Euclidean spaces are used e.g. for visualization purposes.

In data analysis and machine learning, points lying in a Hilbert space E are called vectorial data, typically denoted

$$\{x_1, x_2, \ldots\}, \tag{2.2}$$

where $x_i \in E$. If the set of points is finite, n shall denote is cardinality. We henceforth suppose that there be no repetition, i.e. that there be no two identical points with different subscripts. If the dimension of E is finite, it shall be denoted by p. If $n < \infty$ and $p < \infty$ the set of vectors x_i is often written as a matrix $X \in \mathbb{R}^{p\times n}$ where the i^{th} column represent x_i.

Other norms

Albeit its popularity and immediateness, Euclidean spaces are not the only ones used in data analysis. Often the space $E = \mathbb{R}^p$ is endowed with other norms than the Euclidean norm. A family of norms is given by the Minkowski norm:

$$\|x\|_l = \Big(\sum_i^p \|x_i\|^l \Big)^{\frac{1}{l}}. \tag{2.3}$$

For $l = 2$ we recover the Euclidean norm. For $l = 1$ we obtain the Manhattan norm. The choice of the norm relates to the problem of model selection. The Euclidean norm is only natural in appearance.

Feature space

Hilbert spaces are very restrictive structures and are the prerequisite for many results in machine learning to hold. In general, they are called *feature space*. As we have noted, most machine learning algorithms have been formulated for feature space representation. In discriminant analysis, for example, scalar functions are found that separate labeled points as well as possible (Fisher, 1936, Fukunaga, 1990). Support vector machines maximize the margin between points of different classes (Vapnik, 1998, Wahba, 1999, Müller et al., 2001). k-means clustering finds prototype vectors of k groups in the feature space (Duda et al., 2001), and Principal Component Analysis finds direction of high variance (Jolliffe, 1986).

For data which is naturally represented as vectors these methods are powerful analytical tools. However, there are many situations, where there exists no obvious vectorial representation. This brings us to pairwise data.

PAIRWISE DATA.

S and D

Our starting point will be the (dis-)similarity matrix obtained from some data. Let us first fix the notation.

$$\text{Similarity: } S \in \mathbb{R}^{n \times n},$$
$$\text{Dissimilarity: } D \in \mathbb{R}^{n \times n}.$$

n is the number of objects. s_{ij} is an increasing function of similarity, whereas d_{ij} is an decreasing function of similarity.

Note that any similarity matrix can be converted to a dissimilarity matrix and conversely by some decreasing function. Therefore we may invariably speak about similarity or dissimilarity matrices.

In the most general case, no further assumptions are made on S or D. It is only by adopting the semantics of distance to dissimilarities that one could be tempted to introduce the minimal requirements that $d_{ij} \geqslant 0$ for all $i, j = 1, 2, \ldots n$ and that $d_{ii} = 0$ for all $i = 1, 2, \ldots n$. In the following we will focus on dissimilarity matrices.

Let be recalled the following:

Metric function

DEFINITION 2.3.5. Let $E = \{x_1, x_2, \ldots\}$ be a set of points—not necessarily vectors—with no repetition (i.e. no two identical points with different subscripts), and let $d : E \times E \to \mathbb{R}$. The function d is called a *metric* if:

$$d(x_i, x_j) \geqslant 0 \text{ for all } x_i, x_j \in E \tag{2.4}$$
$$d(x_i, x_j) = 0 \text{ if and only if } x_i = x_j \tag{2.5}$$
$$d(x_i, x_j) = d(x_j, x_i) \text{ for all } x_i, x_j \in E \tag{2.6}$$
$$d(x_i, x_k) + d(x_k, x_j) \geqslant d(x_i, x_j) \text{ for all } x_i, x_j, x_k \in E \tag{2.7}$$

We recognize the requirements of a metric space. The conditions 2.4 to 2.7 are respectively called *positivity*, *reflectivity*, *symmetry* and *triangle inequality*.

Metric dissimilarity matrix

DEFINITION 2.3.6. A dissimilarity matrix $D = (d_{ij})$ is called *metric* if there exists a metric function d such that $d_{ij} = d(\cdot, \cdot)$.

This is equivalent to the statement that if D is metric, then its element d_{ij} satisfy the four conditions (2.4) to (2.7).

A generic dissimilarity matrix usually satisfies (2.5), often (2.4), sometimes (2.6) and rarely (2.7).

Euclidean dissimilarity matrix

DEFINITION 2.3.7. A distance matrix $D = (d_{ij})$ is called *Euclidean* if and only if there exist vectors $x_1, x_2, \ldots x_n \in \mathbb{R}^p$ such that $d_{ij} = \|x_i - x_j\|_2$, where $\|\cdot\|_2$ denotes the Euclidean norm.

PAIRWISE VS. VECTORIAL DATA. As we have seen, the definition of a similarity is a definition on top of an existing vectorial data. In pairwise proximity data, the dis-/similarity is the data itself. There are no features, i.e. there is no basis from which to span the space. Whereas in vectorial data, each single data point comes as a vector of p entries, the size of a pairwise representation is never less than n^2. In pairwise data, there is no clear notion of inter point distance, since one point is only defined with respect to all others. Therefore, if

Generative process

one looks at the data as being generated by some physical process, it is unclear how to formalize a generative model for pairwise data. It is unclear how the semantics of e.g. source and noise carry over to a pairwise setting.

The lack of probabilistic model for pairwise data is the reason why most machine learning algorithms formulated for Hilbert spaces fail to carry over to pairwise data.

Power of pairwise data...

On the other hand, constraining data to fit the restrictive structure of a Hilbert space seems to limit possible understanding. An inner product and the derived norm and distance can only account for a limited class of similarity measurements. Pairwise data need not satisfy the metric conditions and are thus potentially able to capture much richer structure.

METRIC VIOLATIONS.

There are no formalized assumptions on the similarity and dissimilarity matrices except that they "somehow" represent similarity or dissimilarity and that they be square matrices.

The freedom on similarities seems even larger than the freedom on dissimilarities, since the latter are semantically bound to the definition of a metric and one wishes them to conform at least to the requirement of positivity. Negative dissimilarities hurt the intuition like negative distances, while for similarities the same does not hold.

... and its drawback

Very general dissimilarities usually satisfy none of the requirements imposed upon a function to be metric. Let us have a brief glimpse on the respective violations:

POSITIVITY. Positivity can be accidentally violated, usually when obtaining a dissimilarity matrix from a similarity matrix via some decreasing function. Typical choices like $D = 1 - S$ or $D = -\log(S)$ only yield positive D's with prior assumptions on S which are not always natural. Note that in many cases positivity can be enforced by some trick without changing the problem.

REFLECTIVITY. Reflectivity can be violated in two ways, the usual one being that a zero dissimilarity does not imply identity of the points. That is: we often find zero elements on the off-diagonal of the dissimilarity matrix. Conversely the case of non-zero elements on the diagonal does occur, again, usually as an improper transformation from S to D. But this violation might also be an intrinsic feature of the data from cognitive psychology: somebody might find himself less attractive than someone else, so the "distance" to himself is larger than the "distance" to the other person. The violation of reflectivity in the former case does not pose a particular problem in terms of data analysis. In the latter, the data may be considered so exotic that it calls for very individual treatment anyway. Functions which do not satisfy reflectivity are also called *partial metrics*.

SYMMETRY. Symmetry is violated rarely, yet in a natural way as is clear from Section 2.2. Asymmetric dissimilarities usually pose problems in as much they may yield complex eigenvalues in subsequent calculations. They are symmetrized via $\frac{1}{2}(D + D^t)$. For a specific treatment of asymmetric data, see e.g. Everitt and Rabe-Hesketh (1997).

TRIANGLE INEQUALITY. Triangle inequality is often violated. This is the interesting violation. While violations of positivity and reflectivity seems above

all accidental and violation of symmetry are quite scarce, the violation of triangle inequality calls for particular devotion. Let us have a closer look of the violation of (2.7). Several cases should be distinguished.

– The triangle inequality may not be satisfied as a result of fallible estimates or a noisy data set. The comparison algorithm may yield noisy scores. Many of these algorithms rely upon some initialization or some elastic matching which may lead to violations of triangle inequality or even of symmetry (see Figure 2.3 to Figure 2.5 for a *schematic* illustration).

Figure 2.3. Distance as measured by some tool. The oscillation about the straight line from O_i to O_j represents the noise.

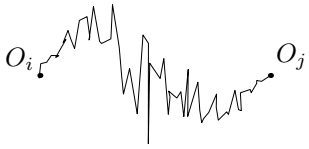


Figure 2.4. Distance as measured by some tool with more noise.

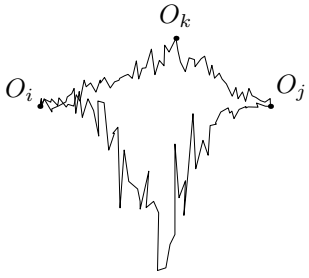


Figure 2.5. The triangle inequality is violated.

– The violation may be an intrinsic feature of the distance measure. The Minkowsky distance given by $\|x_i - x_j\|_l$ for some Minkowsky norm $\|\cdot\|_l$ given by Equation 2.3 is non-metric for $0 < l < 1$. Other non-metric distance measures are noise robust pseudo-metrics like $d(x_i, x_j) = \text{median}_k(|x_{i,1} - x_{j,1}|, |x_{i,2} - x_{j,2}|, \ldots |x_{i,n} - x_{j,n}|)$ where median_k is the k-th value of the ordered difference vector. Other examples involve e.g. the Kullback-Leibler measure of cross-entropy. It is asymmetric and violates the triangle inequality (Kapur and Kesavan, 1992). In Jacobs et al. (2000) the problem specific advantages of such non-metric distance measures are discussed.

– Many data sets are inherently non-metric. This particularly applies on data sets based upon some human judgment, for which many relationships are not transitive, e.g. "X likes Y, Y likes $Z \nRightarrow X$ likes Z".

DEFINITION 2.3.8. Pairwise data will be called *non-metric* if the dissimilarity matrix representing the pairwise data violates one or several of the conditions (2.4) to (2.7). We will speak about *metric violations*.

REMARK. We do not refer to non-metric data as a more general data type like data qualitative or ordinal in nature. Non-metric data stands for data whose mutual distances do not satisfy the requirements of a metric function. Also note that the adjective "metric" in "metric violations" does not refer to the nature of the violation but to the property violated.

Embedding

We have already mentioned the problem of embedding which tries to find points in a feature space such that their mutual distance is given by the dissimilarity matrix. If this were always possible there would be no need for structural approaches, i.e. data analytical methods developed for pairwise data. A Hilbert space is, a fortiori, a metric space, thus making it intrinsically impossible to represent non-metric data in such a space.

Non-metricity in itself is no impediment for algorithms of data analysis which directly relay on pairwise input. The need for such algorithms arose

exactly because of the impossibility to embed pairwise data. Two popular algorithms for pairwise data are: k-means and k nearest neighbors.

However, from a data analytical perspective, pairwise data still suffers from a lack of analytical tools. This is mainly due to the lack of probabilistic model for pairwise data. The data only exists "as such". Furthermore, structural approaches are often expensive in computations since the information is contained in $n \times n$ relationships. So we again raise the question of embedding pairwise data in a feature space. In particular, we will be interested in the *embedding of non-metric pairwise data into a Euclidean space*.

ISSUE OF EMBEDDING.

Concept

Embedding tries to find a set of n points in some space, usually a feature space, such that their pairwise distances $d(x_i, x_j)$, $i, j = 1, 2, \ldots n$, given by a metric function $d(\cdot, \cdot)$, is "as close as possible" to the given dissimilarity matrix D with respect to some cost function, the mutual distances being ideally identical to the original pairwise distances.

Embedding is a very general concept. It applies to finding vectorial representatives of pairwise data, possibly non-metric, as well as to finding low dimensional representations of high dimensional data, in general to fit a data set in a given space.

While it is possible to embed data in some general metric space, one usually is interested in *Euclidean embeddings*, i.e. embeddings into a Euclidean space. We look for vectors in a Euclidean space to represent the data closest to human intuition and interpretability. This is particularly the case for low dimensional embeddings, typically of two or three dimensions, which allow to visualize the data in a familiar space. Figure 2.6 shows a schematic representation of the

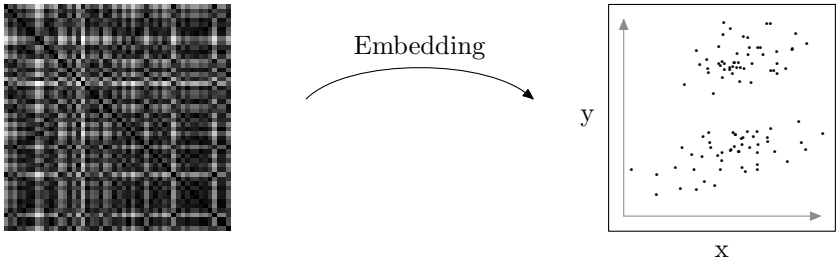


Figure 2.6. Schematic representation of the embedding problem. On the left hand side we have pairwise dissimilarity data represented as a checkerboard pattern. On the right hand side we have vectorial data: points in a proper Hilbert space such that their mutual (pairwise) distance is as close as possible to the original pairwise dissimilarities, as measured by some cost function.

embedding of pairwise data into a two dimensional Euclidean space.

The crucial point of the embedding is the conservation of the dissimilarities. Pairwise representation naturally come about when the gain of structural representations outweight the featural presentation. Embedding in such restrictive structures as Hilbert spaces will not come without a price. This price is what the present thesis is about.

Geometric loss

A Hilbert space has an inner product and thus a natural metric. It is therefore intrinsically impossible to find points such that their mutual distance will be identical to the dissimilarities we started from. The embedding will incur a loss in terms of metric. We will call this loss *geometric loss*. This will be treated in Section 2.5.

An Euclidean embedding is an even harder problem since we loose the freedom on the metric. Even metric pairwise data may not have a loss-free representation in a Euclidean space. Only if the pairwise dissimilarities are Euclidean this is possible, as will be seen in the following section.

§. 2.4.

EMBEDDING INTO A EUCLIDEAN SPACE.

In this section we will discuss the special case of embedding Euclidean dissimilarities into a Euclidean space. We have seen that the reason for Euclidean embeddings are not only their usefulness for existing data analytical tools but also their tangibility for the human experimenter. The reason for choosing Euclidean dissimilarities is that there is a powerful way to enforce Euclideanness[1] for (quite) general dissimilarity matrices.

Let $D = (d_{ij})$ be a Euclidean dissimilarity matrix. From Definition 2.3.7 we recall that there exist vectors $x_1, x_2, \dots x_n \in \mathbb{R}^p$ such that $d_{ij} = \|x_i - x_j\|_2$. The goal of this section is to find them.

We first need to introduce a few definitions and establish a few results.

Centralized matrix

DEFINITION 2.4.1. Let $A = (a_{ij})$ be any matrix and let $Q = I - \frac{1}{n}ee^t$ be the projection matrix on the orthogonal complement of $e = (1, 1, \dots 1)^t$.

The *centralized* matrix A^c is the matrix

$$A^c = QAQ.$$

A centralized matrix has row and column sum equal to zero. Looking at the

[1] This is not a neologism. In very rare cases one also encounters the term "Euclideanity".

components of A^c

$$a^c_{ij} = a_{ij} - \frac{1}{n}\sum_{k=1}^{n} a_{ik} - \frac{1}{n}\sum_{k=1}^{n} a_{kj} + \frac{1}{n^2}\sum_{k,l=1}^{n} a_{kl}, \tag{2.8}$$

one easily verifies that $\sum_{i=1}^{n} a^n_{ij} = 0$ and $\sum_{j=1}^{n} a^n_{ij} = 0$.

To understand the semantic of the centralizing operation, consider the following setting. Let $\{x_1, x_2, \dots x_n\}$ be some set of n vectors of some vector space in a given basis and gathered in the row matrix X, i.e. the i^{th} row of X contains x_i. Let C_X be the corresponding covariance matrix:

$$C_X = \frac{1}{n} X X^t. \tag{2.9}$$

If $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ denotes the arithmetic mean of $\{x_1, x_2, \dots x_n\}$ then $\{x_1 - \bar{x}, x_2 - \bar{x} \dots x_n - \bar{x}\}$ corresponds to the set of points shifted about the origin.

THEOREM 2.4.1. *Let $\{x_1, x_2, \dots x_n\}$ be a set of points gathered in a matrix X and let $\bar{X}$ denote the matrix of points $x_i - \bar{x}$, where $\bar{x}$ denotes the arithmetic mean. C denotes the covariance matrix as defined in Equation 2.9 and C^c is the centralized covariance matrix. Then we have $C^c_X = C_{\bar{X}}$, in other words, the diagram*

$$\begin{array}{ccc} X & \xrightarrow{\textit{centralizing}} & \bar{X} \\ {\scriptstyle \textit{cov}}\downarrow & & \downarrow{\scriptstyle \textit{cov}} \\ C_X & \xrightarrow{\textit{centralizing}} & C_{\bar{X}} \end{array}$$

commutes.

Proof. Let $e = (1, 1, \dots 1)^t$ as before. By simple algebra, one verifies that $\bar{x} = \frac{1}{n} X^t e$ and $\bar{X} = X - e\bar{x}^t$. Then:

$$\begin{aligned} C^c_X &= \big(I - \frac{1}{n} e e^t\big) \frac{1}{n} X X^t \big(I - \frac{1}{n} e e^t\big)^t \\ &= \frac{1}{n} X X^t - \frac{1}{n^2} X X^t e e^t - \frac{1}{n^2} e e^t X X^t + \frac{1}{n^3} e e^t X X^t e e^t \\ &= \frac{1}{n} X X^t - \frac{1}{n} X \bar{x} e^t - \frac{1}{n} e \bar{x}^t X + \frac{1}{n} e \bar{x}^t \bar{x} e^t \\ &= \frac{1}{n} \big(X - e\bar{x}^t\big)\big(X - e\bar{x}^t\big)^t \\ &= C_{\bar{X}}. \end{aligned}$$

□

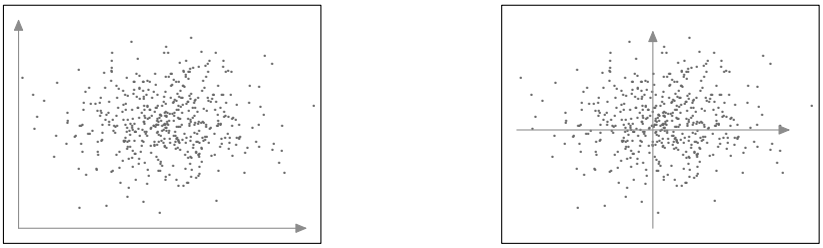


Figure 2.7. Centralizing a covariance matrix: on the left hand side, we have some vectorial data in a two dimensional Euclidean space. The covariance matrix depends on the position of the points with respect to the origin of the space. Centralizing the data or the covariance matrix corresponds to moving the origin to the center of gravity of the data, i.e. the point given by its arithmetic mean.

Centralizing a covariance matrix thus corresponds to center the corresponding points around the origin, which in view of embedding, does not change the problem (see Figure 2.7).

LEMMA 2.4.1. *A^c is unique.*

Proof. Let A^c and $A^{c'}$ be two centralized matrices of A. Simple algebra yields $a^c_{ij} - a^{c'}_{ij} \equiv 0$ for all $i, j = 1, 2, \dots n$. □

Squared Euclidean dissimilarity matrix

DEFINITION 2.4.2. A dissimilarity matrix $D = (d_{ij})$ is called *squared Euclidean* if and only if there exist vectors $x_1, x_2, \dots x_n \in \mathbb{R}^p$ such that $d_{ij} = \|x_i - x_j\|_2^2$, where $\|\cdot\|_2$ denotes Euclidean norm.

REMARK. It will turn out to be more useful to speak about squared Euclidean dissimilarities. Therefore, unless stated otherwise, a dissimilarity matrix D will be taken squared.

Let $D = (d_{ij})$ be squared Euclidean and fixed. D can be decomposed as follows:

Decomposition

$$d_{ij} = c_{ii} + c_{jj} - 2c_{ij}. \tag{2.10}$$

$C = (c_{ij})$ is not fixed by the choice of D, since we always may change its diagonal elements, yet recover the same D. Let $\mathcal{C}_D$ denote the equivalence class of all C yielding the same D by Equation 2.10. In particular we note, by simple algebra, that $C^c \in \mathcal{C}_D$.

We have the following two important results:

Relation between C and D

LEMMA 2.4.2. *Let C and D be two matrices related to each other by $d_{ij} = c_{ii} + c_{jj} - 2c_{ij}$. Then*

$$C^c = -\frac{1}{2}D^c. \tag{2.11}$$

Proof. Substituting Equation 2.10 into Definition 2.8 of the centralized C^c yields

$$\begin{aligned}
c_{ij}^c &= c_{ij} - \frac{1}{n}\sum_{k=1}^{n} c_{ik} - \frac{1}{n}\sum_{k=1}^{n} c_{kj} + \frac{1}{n^2}\sum_{k,l=1}^{n} c_{kl} \\
&= -\frac{1}{2}\Big((d_{ij} - c_{ii} - c_{jj}) - \frac{1}{n}\sum_{k=1}^{n}(d_{ik} - c_{ii} - c_{kk}) \\
&\quad - \frac{1}{n}\sum_{k=1}^{n}(d_{kj} - c_{kk} - c_{jj}) + \frac{1}{n^2}\sum_{k,l=1}^{n}(d_{kl} - c_{kk} - c_{ll})\Big) \\
&= -\frac{1}{2}\Big(d_{ij} - \frac{1}{n}\sum_{k=1}^{n} d_{ik} - \frac{1}{n}\sum_{k=1}^{n} d_{kj} + \frac{1}{n^2}\sum_{k,l=1}^{n} d_{kl}\Big) \\
&= -\frac{1}{2} d_{ij}^c.
\end{aligned}$$

□

The next result is of paramount importance, since it establishes a strong link between the squared Euclideanness of a dissimilarity and the spectrum of the associated centralized covariance matrix C^c.

Main theorem

THEOREM 2.4.2. *D is squared Euclidean if and only if C^c is positive semi-definite.*

Proof. Torgerson (1958) referring to Young and Householder (1938), or the following simple algebra: ($\Rightarrow$) Because D is squared Euclidean, we can take vectors $x_1, x_2, \ldots x_n \in \mathbb{R}^p$ ($p \leqslant n-1$) which satisfy $d_{ij} = \|x_i - x_j\|^2$. Then,

$$\begin{aligned}
d_{ij}^c &= d_{ij} - \frac{1}{n}\sum_{k=1}^{n} d_{ik} - \frac{1}{n}\sum_{k=1}^{n} d_{kj} + \frac{1}{n^2}\sum_{k,l=1}^{n} d_{kl} \\
&= \|x_i\|^2 + \|x_j\|^2 - 2x_i x_j - \Big(\|x_i\|^2 + \frac{1}{n}\sum_{k=1}^{n}\|x_k\|^2 - \frac{2}{n}\sum_{k=1}^{n}(x_i x_k)\Big) \\
&\quad - \Big(\frac{1}{n}\sum_{l=1}^{n}\|x_l\|^2 + \|x_j\|^2 - \frac{2}{n}\sum_{k=1}^{n}(x_l x_j)\Big) \\
&\quad + \Big(\frac{1}{n}\sum_{l=1}^{n}\|x_l\|^2 + \frac{1}{n}\sum_{k=1}^{n}\|x_k\|^2 - \frac{2}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n}(x_l x_k)\Big) \\
&= -2(x_i x_j - x_i\bar{x} - \bar{x}x_j + \bar{x}\bar{x}) \\
&= -2(x_i - \bar{x})(x_j - \bar{x}),
\end{aligned}$$

where $\bar{x} = \frac{1}{n}\sum_{k=1}^{n} x_k$. That is,

$$D^c = -2\bar{X}\bar{X}^t \quad \text{and} \quad C^c = \bar{X}\bar{X}^t, \tag{2.12}$$

where $\bar{X} = (x_1 - \bar{x}, x_2 - \bar{x}, \dots x_n - \bar{x})^t$. From this equation, it is obvious that C^c is positive semi-definite.

($\Leftarrow$) When C^c is positive semi-definite, there exists a $n \times p$ $(p \leqslant n-1)$ matrix X which satisfies

$$C^c = XX^t. \tag{2.13}$$

Let $X = (x_1, x_2, \dots x_n)^t$, that is, x_i's are the row vectors of X. Then, from the relationships between D and C^c,

$$\begin{aligned} d_{ij} &= c^c_{ii} + c^c_{jj} - 2c^c_{ij} \\ &= x_i x_i + x_j x_j - 2x_i x_j \\ &= \|x_i - x_j\|^2. \end{aligned}$$

This shows the matrix D is squared Euclidean. □

This theorems gives us a necessary and sufficient condition on the spectrum of C^c for D to be loss-free embeddable in a Euclidean space.

REMARK. The condition "D metric" is not strong enough. One can construct examples of metric D's such that the associated C^c is indefinite.

If C^c is positive semi-definite it is a Mercer kernel (i.e. a dot product, for example by the existence of X such that $C^c = \frac{1}{n}XX^t$).

CONSEQUENCES. $C^c = -\frac{1}{2}D^c$ relates the distance matrix to the centralized covariance matrix. The spectrum of C^c tells us whether D is a squared Euclidean or not. Thus we really are only interested in C^c, a covariance matrix in case it is positive semi-definite, a *pseudo-covariance* matrix else. In the latter case, C is also called *generalized covariance*.

Dissimilarity and metric

REMARK. A dissimilarity matrix may be called a metric if it satisfies the necessary conditions. On the other hand any distance can be readily interpreted as a dissimilarity.

Similarity and covariance

A similarity matrix S may be interpreted as covariance if it is positive semi-definite, or as a pseudo-covariance else. However, conversely, it is not always straight forward to interpret a covariance matrix as a similarity matrix since the covariance takes into account the length of vectors, which is hard to justify passing to similarities—implying that to same points are more similar the further away they are from the origin. C depends on the origin!

Relation between S and D

There is no *a priori* relationship between S and D. Since embedding yields a representation in terms of distances, the quantity which a solution will be

judged by, a direct embedding of S does not seem appropriate, the semantics of S being scalar products and not distances.

When starting from S we first must choose an associated distance matrix D. There are several standard ways to achieve this. One often encounters the choices $D = 1 - S$, $d_{ij} = -\log(s_{ij})$, $d_{ij} = \sqrt{-\log(s_{ij})}$, $d_{ij} = \frac{1}{s_{ij}} - 1$ or $d_{ij} = s_{ii} + s_{jj} - 2s_{ij}$. This last choice will be assumed implicitly—unless stated otherwise—for the remainder of this work, when we take S as starting point. This choice corresponds to interpreting the similarity matrix S as a covariance.

Non-euclidean $\not\Rightarrow$ non-metric

METRIC SPACE VS. EUCLIDEAN SPACE. It must be stressed here, that C^c non positive semi-definite *does not* mean that the corresponding D is not metric. It means that D is not squared Euclidean and hence may not be embedded loss-free with respect to the Euclidean metric into a Euclidean space. D may be metric and therefore there may be an embedding into a metric space. However, a metric space is still a very abstract structure and may not help in understanding the data.

Non-metric $\Rightarrow$ non-Euclidean

If D is non-metric, then *a fortiori* it is not Euclidean and by the above theorem C^c is not positive semi-definite. *It is important to keep in mind that the converse is not true.*

EXAMPLE. Because the metric property in Definition 2.3.7 assumes Euclidean metric, it is stronger than the condition that all triangle inequalities hold, i.e.

$$\sqrt{d_{ij}} + \sqrt{d_{jk}} \geqslant \sqrt{d_{ik}}, \quad \text{for all } i \neq j \neq k \tag{2.14}$$

Let us consider the following distance matrix

$$\sqrt{D} = \begin{pmatrix} 0 & 3 & 4 & 1 \\ 3 & 0 & 5 & 2 \\ 4 & 5 & 0 & 3 \\ 1 & 2 & 3 & 0 \end{pmatrix}. \tag{2.15}$$

It is easy to check Condition 2.7. The squared distance matrix and its centralized version become

$$D = \begin{pmatrix} 0 & 9 & 16 & 1 \\ 9 & 0 & 25 & 4 \\ 16 & 25 & 0 & 9 \\ 1 & 4 & 9 & 0 \end{pmatrix},$$

$$D^c = \begin{pmatrix} -5 & 1 & 5 & -1 \\ 1 & -11 & 11 & -1 \\ 5 & 11 & -17 & 1 \\ -1 & -1 & 1 & 1 \end{pmatrix}.$$

Because the eigenvalues of D^c are $\{-26.05,\ -7.44,\ 0,\ 1.49\}$, D^c is not negative semi-definite. This means $C^c = \frac{1}{2}D^c$ is not positive semi-definite, or equivalently D is not metric in the sense of Definition 2.3.7. The intuitive explanation of the example is as follows. The samples 1, 2, and 3 form a triangle (with edges length 3, 4, and 5). From the relationships of the distances, we have

$$\sqrt{d_{14}} + \sqrt{d_{42}} = 1 + 2 = 3 = \sqrt{d_{12}}$$
$$\Leftrightarrow \text{ Point } 4 \text{ should be on the edges connecting the points } 1 \text{ and } 2,$$
$$\sqrt{d_{14}} + \sqrt{d_{43}} = 1 + 3 = 4 = \sqrt{d_{13}}$$
$$\Leftrightarrow \text{ Point } 4 \text{ should be on the edges connecting the points } 1 \text{ and } 3,$$
$$\sqrt{d_{24}} + \sqrt{d_{43}} = 2 + 3 = 5 = \sqrt{d_{23}}$$
$$\Leftrightarrow \text{ Point } 4 \text{ should be on the edges connecting the points } 2 \text{ and } 3.$$

However, of course, it is not possible to find a point (4) in Euclidean space. In other word, we can not place four points in Euclidean space so that they have the distance $\sqrt{D}$.

Recovering vectors from squared Euclidean D's.

We are now ready to solve the problem proposed at the beginning of this section, namely finding the vectors x_i such that their mutual distance is given by the initial dissimilarity matrix supposed to be squared Euclidean.

Algorithm

Since D is squared Euclidean, C^c is positive semi-definite by Theorem 2.4.2 and we have the following algorithm to recover the data points (Cox and Cox, 2001):

1. Calculate centralized kernel matrix $C^c = -\frac{1}{2}QDQ$ from the distance matrix D.

2. Get the eigenvalue decomposition of C^c,

$$C^c = V\Lambda V^t,$$

   where $V = (v_1, v_2, \dots\ v_n)$ is the row matrix composed of the eigenvectors v_i and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \dots\ \lambda_n)$ the diagonal matrix with eigenvalues $\lambda_1 \geqslant \cdots \geqslant \lambda_n \geqslant 0$. Notice that due to the centralization, which introduces a linear dependency between all vectors, at least one eigenvalue equals zero.

3. Calculate the $n - 1 \times n$ map matrix

$$X = \Lambda^{\frac{1}{2}} V^t, \tag{2.16}$$

where V is the column-matrix of the eigenvectors and Λ the diagonal matrix of the corresponding eigenvectors.

The columns of X contain the vectors x_i, $i = 1, 2, \ldots n$, in the $n-1$-dimensional subspace. *The mutual distances coincide with D, i.e.* $d_{ij} = \|x_i - x_j\|^2$. In other words: there is a direct algebraic transformation between D and the set of x_i's.

Principal component analysis

One may choose not to retain the full set of $n-1$ eigenvectors, but use only a subset L of the leading eigendirections $L = \{v_1, v_2, \ldots v_t\}$ with $t \leqslant n$. In this case the vectors represent the least squares approximates of a set of vectors whose mutual squared distance is given by D. This algorithm then effectively amounts to PCA (Jolliffe, 1986).

The assumption that D be squared Euclidean assured positive semi-definiteness of C^c so that $\Lambda^{\frac{1}{2}}$ is well-defined for all eigendirections, which yields identical mutual distances to the pairwise dissimilarities we started from. For non-metric pairwise data, this assumption does not hold and the above algorithm will fail because of complex eigenvalues. The embedding of non-metric pairwise data will be discussed in the next section.

§. 2.5.

EMBEDDING OF NON-METRIC PAIRWISE DATA.

Consequence of non-metricity

Non-metricity of pairwise data, by implying non squared Euclideanness, readily translates into the spectrum of C^c having negative eigenvalues. Therefore C^c cannot be looked at as a covariance matrix of some set of vectors. (For an investigation of non-metricity not based upon the eigenvalue spectrum see Appendix A.) A distortionless embedding into a vector space is not possible, even in high dimensions. It is a common misconception that higher dimensions could straighten out a faulty metric and represent even the most general dissimilarity matrix D. It must be stressed here, once again, that for non-metric D's, there is no loss-free embedding (in the sense of geometric loss) in a Hilbert space, be it of price of many supplementary dimensions. We therefore will always be confronted to particular workarounds.

In this section we will briefly pass in review common workarounds in these cases, namely Multidimensional Scaling and embedding into a pseudo-Euclidean space.

MULTIDIMENSIONAL SCALING.

Presumingly the most popular embedding method for non-metric data is *Multidimensional Scaling* (MDS). See e.g. Cox and Cox (2001), Borg and Groenen (1997), Everitt and Rabe-Hesketh (1997) and Buja et al. (2001) for recent overviews. MDS was invented for the analysis of proximity data and for dimension reduction. In MDS one seeks a vectorial representation of data—typically in low dimension—such that the distortion of the pairwise dissimilarities d_{ij} is minimal. The oldest form of MDS is due to Torgerson (1952, 1958) and Gower (1952) and is called *classical scaling*. It is of particular importance to us and will be discussed in detail below. Today, the leading MDS methods are based upon works by Kruskal (1964a,b). The goodness of fit between the dissimilarities D and their vectorial representatives is measured by a cost function called "stress" which has the form:

Classical scaling

Stress

$$\text{stress}(x_1, x_2, \dots x_n) = \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \omega_{ij} \big(\|x_i - x_j\| - d_{ij} \big)^2, \qquad (2.17)$$

where ω_{ij} are weights. Typically these weights read:

$$\omega_{ij} = \frac{1}{n(n-1)d_{ij}^2}, \; \omega_{ij} = \frac{1}{\sum_{k,l} d_{kl}^2} \;\text{ or }\; \omega_{ij} = \frac{1}{d_{ij} \sum_{k,l} d_{kl}}. \qquad (2.18)$$

The choice in Equation 2.18 relates to the minimization of relative, absolute or intermediate error (Duda et al., 2001).

A simple case of cost function is given by the residual sum of squares:

$$\text{stress}(x_1, x_2, \dots x_n) = \Bigg(\sum_{i \neq j = 1}^{n} \big(\|x_i - x_j\| - d_{ij} \big)^2 \Bigg)^{\frac{1}{2}}.$$

Another widely used cost function is the squared stress criterion, denoted by $\text{sstress}(x_1, x_2, \dots x_n)$, where the difference between the squared norm $\|x_i - x_j\|^2$ and the squared dissimilarities d_{ij}^2 is optimized (Takane et al., 1977).

Sstress

MDS always finds a vectorial representation, whether or not the pairwise data be metric. However, the *a priori* chosen cost function, possibly of high complexity and with many parameters which can be tuned, often makes it hard to understand how the low-dimensional representation is found.

Kruskal-Shepard distance scaling

The above MDS version via the optimization of $\text{stress}(x_1, x_2, \dots x_n)$ is called the *Kruskal-Shepard distance scaling*. It is based upon direct fitting of the vectorial distances to the original dissimilarities. *Classical Torgerson-Gower inner-product scaling* a.k.a classical scaling is based on converting the dissimilarities into a form naturally fitted by inner products.

Algorithm

CLASSICAL SCALING. Classical scaling is based upon Theorem 2.4.2 by Young and Householder (1938). The idea underlying classical scaling is to suppose that the dissimilarities are Euclidean distances and then to find coordinates for exploring them.

Classical scaling proceeds similarly to the algorithm given on page 24. However, since C^c needs not to be positive semi-definite, the projection is not defined for the eigendirections associated to negative eigenvalues.

1. Calculate centralized matrix $C^c = -\frac{1}{2}QDQ$ from the distance matrix D.

2. Get the eigenvalue decomposition of C^c

$$C^c = V\Lambda V^t,$$

   where $V = (v_1, v_2, \dots v_n)$ is the row matrix composed of the eigenvectors v_i and the diagonal matrix $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \dots \lambda_n)$ with eigenvalues $\lambda_1 \geqslant \cdots \geqslant \lambda_p > \lambda_{p+1} = \cdots = \lambda_{p+k} = 0 > \lambda_{p+k+1} \geqslant \lambda_n$.

3. Select the t-dimensional subspace L out of the p eigenvectors v_i associated to the positive eigenvalues and calculate the $t \times n$ map matrix

$$X_L = \Lambda_L^{\frac{1}{2}} V_L^t, \tag{2.19}$$

   where V_L is the column-matrix of the selected eigenvectors and Λ_L the diagonal matrix of the corresponding eigenvectors.

The columns of X_L contain the vectors x_i, $i = 1, 2, \dots n$, in the chosen t-dimensional subspace.

Classical scaling vs. PCA

There is *no* direct algebraic transformation between D and the set of x_i's. Furthermore, unlike in PCA, the x_i's do *not* represent the least squares approximate of a set of vectors whose mutual squared distance is given by D. However, if D happens to be squared Euclidean, all eigenvalues are positive and we recover the algorithm from page 24.

Classical scaling can also be formulated as an optimization problem. The corresponding cost function is called *strain*. One possible form of strain is a residual sum of squares:

$$\text{strain}(x_1, x_2, \dots x_n) = \left(\frac{\sum_{i,j=1}^n (c_{ij}^c - \langle x_i, x_j \rangle)^2}{\sum_{i,j=1}^n \langle x_i, x_j \rangle^2} \right)^{\frac{1}{2}}.$$

In MDS literature, many further MDS variants can be found. Note that one of it is named *non-metric* MDS. In this case, the term non-metric does not refer to the conditions imposed on pairwise data to be metric, but refers to ordinal data, i.e. data for which only the rank order is taken into account.

PSEUDO-EUCLIDEAN SPACE.

Unify vectorial and pairwise representation

The pseudo-Euclidean approach to embedding is the fruit of the attempt to unify vectorial and structural data into one global type, thereby unifying the vectorial and the structural approach to data analysis, so as to profit of both their respective advantages. It is based upon the works by Goldfarb (1984, 1985).

The pseudo-Euclidean space is a generalization of the well known Euclidean space to indefinite inner products. It effectively amounts to two Euclidean spaces one of which has a positive semi-definite inner product and the other a negative semi-definite inner product.

For squared Euclidean distances, the embedding procedure as discussed relies on the centralized covariance matrix $C^c = -\frac{1}{2}D^c$ with a subsequent spectral decomposition $C^c = \frac{1}{n}XX^t$. Dividing the embedding space into two Euclidean spaces with positive and negative semi-definite inner products with respective dimensions p and q amounts to posing

$$C^c = \frac{1}{n}X \begin{pmatrix} M & \\ & 0_{k\times k} \end{pmatrix} X^t,$$

where

$$M = \begin{pmatrix} I_{p\times p} & \\ & -I_{q\times q} \end{pmatrix},$$

and $0_{k\times k}$ is the $k \times k$ matrix consisting of zeros. Note that $p + q + k = n$, so that

$$XMX^t = V\Lambda V^t = V|\Lambda|^{\frac{1}{2}}M|\Lambda|^{\frac{1}{2}}V^t,$$

where V is the column-matrix of the eigenvectors and Λ the diagonal matrix of the corresponding eigenvectors.

The vectors can be recovered as follows:

$$X_L = |\Lambda_L|^{\frac{1}{2}}V_L^t,$$

where V_L is the column-matrix of the *selected* eigenvectors and Λ_L the diagonal matrix of the corresponding eigenvectors. X_L contains the vectors in the pseudo-Euclidean space.

For L full index set, we recover the pseudo-covariance matrix

$$\text{cov}(X) = \frac{1}{n}X^tXM = \frac{1}{n}|\Lambda|M = \frac{1}{n}\Lambda.$$

X is a result of a mapping the sense of a PCA projection and the embedding procedure can thus be interpreted as kernel-PCA, where C is the reproducing kernel of the pseudo-Euclidean feature space (Greub, 1975).

Distances in a pseudo-Euclidean space

An interesting interpretation of the distances in a pseudo-Euclidean space is that they can be looked at as a difference of squared Euclidean distances

from the "positive" and the "negative" space, by the decomposition $\mathbb{R}^{(p,q)} = \mathbb{R}^p + i\mathbb{R}^q$.

Thus

$$d_{ij} = d_{ij}^{(\mathbb{R}^p)} - d_{ij}^{(\mathbb{R}^q)}. \tag{2.20}$$

DETAILS. In this somewhat more technical exposé we follow Appendix A1 from Pękalska et al. (2001). A pseudo-Euclidean space E is a real linear vector space equiped with a non-degenerate, indefinite, symmetric bilinear function $\langle \cdot, \cdot \rangle$, called inner product (Greub, 1975). A pseudo-Euclidean space an be interpreted as composed from two Euclidean subspaces, i.e. E_+ of dimensionality p and E_- of dimensionality q such that $E = E_+ \oplus E_-$. The inner product is positive definite on E_+ and negative definite on E_-. E is characterized by the signature (p, q) (Goldfarb, 1984). A basis $\{e_1, e_2, \ldots\ e_{p+q}\}$ is called orthonormal in a pseudo-Euclidean space if

$$\langle e_i, e_i \rangle = \begin{cases} +1, i = 1, 2 \ldots\ p \\ -1, i = p+1, \ldots\ p+q \end{cases}$$

and $\langle x_i, x_j \rangle = 0$ for $i \neq j$.

The inner product between two vectors x and y reads:

$$\langle x, y \rangle = \sum_{i=1}^{p} x_i y_i - \sum_{i=p+1}^{p+q} x_i y_i = x^t M y.$$

Thus, a sphere of equidistant points in a Euclidean space becomes a hyperboloid in a pseudo-Euclidean space.

The norm if a non-zero vector x in a pseudo-Euclidian space is defined as:

$$\|x\|^2 = \langle x, x \rangle = x^t M x.$$

It can be positive, negative or zero (even for non-zero vectors called *isotropic* vectors). The definition of a squared distances follows naturally:

$$D(x, y) = \|x - y\|^2 = \langle x - y, x - y \rangle = (x - y)^t M (x - y).$$

Consider the decomposition

$$M = M_+ + M_- = \begin{pmatrix} I_{p \times p} & \\ & 0 \end{pmatrix} + \begin{pmatrix} 0 & \\ & -I_{q \times q} \end{pmatrix},$$

then

$$\begin{aligned} D(x, y) &= (x - y)^t M (x - y) \\ &= (x - y)^t (M_+ + M_-)(x - y) \\ &= (x - y)^t M_+ (x - y) - (x - y)^t (-M_-)(x - y) \\ &= D_+ - D_-, \end{aligned}$$

where $D_+ = (x-y)^t M_+ (x-y)$ and $D_- = (x-y)^t (-M_-)(x-y)$. D_+ and D_- belong to a Euclidian space of dimension p resp. q. This decomposition yields the formula 2.20.

Pseudo-Euclidean space in physics

REMARK. Pseudo-Euclidean spaces seem to be a quite "artificial" construction and of mere mathematical interest. However they play an important role in fields as prominent as special relativity, where the line element of the Minkowski space reads $ds^2 = dx^2 + dy^2 + dz^2 - dt^2$, thus combining "space-like" vectors and "time-like" vectors.

§. 2.6.

DISCUSSION.

Glimpses of possible issues and research directions have been given along this presentation of pairwise data. It seems at hand that *non-metricity* and *embedding* are an ongoing issue.

On one hand, one allows pairwise data to be very general and thus capture rich structure, possibly non-metric, on the other one seems quite unable to profit from the "rich" structure thus obtained: embedding is used to recover vectors, preferably in a Euclidean space, so as to make the data available to the zoo of data analytical tools developed for vectors. However, rarely is embedding not an enforcement where the initial freedom of structural representation is sacrificed to vectorial tractability. The pseudo-Euclidean approach, however elegant, has rather marginal an existence. It seems that the pretended unification has failed in practice. This is not so surprising since many analytical tools require the specific property of positive semi-definiteness of the inner product of a Hilbert space (called Mercer kernels in the corresponding literature). This is not the case in a pseudo-Euclidean space, hence its limited interest. We still believe in a possible unification, but rather on a "local" level, i.e. on the level of the data analytical tools and not the representation. This idea will be explored in the next chapter. It will be shown that the equivalence of the representation should not only be considered from geometrical point of view. The unification will not come by unifying the data representation, but by unifying the data with the subsequently used analytical tools and showing that identical results can be obtained whatever the representation.

Chapter 3

Chapter 4

Furthermore, the structural approaches claim to profit from the less restrictive structure of pairwise data. However, if we look in particular at metric violations, it is yet to be proved that the captured structure is richer in terms of

information rather than simply noisier. At this point in the thesis, it is still unclear what metric violations mean and whether at all we can come along with a sensible interpretation, or even model explaining metric violations. This will be the topic of the fourth chapter.

§. 2.7.

CONCLUSION.

In modern data analysis data arises in a variety of forms which require appropriate treatment. For major fields, data is often not available as feature vectors in a vector space, thus precluding the use of well established data analytical tools. For instance, genomics typically produce data represented as strings from some alphabet, psychology yields sets of similarity judgments, yet other fields like social sciences measure so called preference data.

Non-vectorial data sets as such are difficult to handle, and for data mining purposes we need to relate them to some mathematical concept. A common approach is to replace the initial data by a collection of real numbers representing some "comparison" among the elements of the data set. This can be straight forward, as for similarity judgments, or highly non trivial as for string data, where the similarity score may be derived by a complex alignment algorithm. This procedure yields a matrix gathering the pairwise proximity relations between the original objects. We have to stress here that such a matrix is by no means naturally related to the common viewpoint of objects being embedded in some "well behaved" space with a vector space. In particular, for pairwise data, there is no probabilistic model.

There are two data analytical approaches, namely the vectorial approach and the structural approach. The advantage of the vectorial approach is the myriad of techniques which can be deployed in a vector space to analyze the data. However, a normed vector space is a restrictive structure and this is where the advantage of the structural approach lies relying upon pairwise input which has the potential to capture much richer structure.

Embedding pairwise data into a vector space is the attempt to combine the best of both worlds. Several embedding procedures have been presented. The discussion of their insufficiencies opened the main research axis of the next chapters.

3. OPTIMAL EMBEDDING

In this chapter we study properties of embedding strategies in the context of clustering. We will proceed as follows: we begin with a short overview of proximity based data grouping, and then focus on reformulating such problems with vectorial data representations. For the class of pairwise clustering methods that are related to minimizing a shift-invariant cost function, our main contribution is a new embedding strategy, which we call *Constant Shift Embedding* as proposed in Roth et al. (2003a,b). A surprising property of this embedding is *the complete preservation of group structure*. The original non-metric pairwise clustering problem can be restated as a grouping problem over points in a vector space, yielding identical assignments of objects to clusters. Using the constant shift embedding principle, we then demonstrate the equivalence between the *pairwise clustering* cost function and the classical k-means grouping criterion in the embedding space.

§. 3.1.

INTRODUCTION.

Clustering

Unsupervised grouping or *clustering* aims at extracting hidden structure from data (Duda et al., 2001). The term data refers to both a set of objects and a set of corresponding object representations resulting e.g. from some physical measurement process. As we have seen in the previous chapter, different types of object representations are possible, the two most common of which are *vectorial data* and *pairwise proximity data*. In the first case, a set of n objects is represented as n points in a d-dimensional vector space, whereas in the second case we are given a $n \times n$ pairwise proximity matrix.

The problem of grouping vectorial data has been widely studied in the literature, and many clustering algorithms have been proposed (Duda et al., 2001, Jain et al., 1999). One of the most popular method is k-means clustering. It derives a set of k prototype vectors which quantize the data set with minimal quantization error. Figure 3.1 shows a simple example of two dimensional data and two possible clustering solutions. Other popular clustering algorithms are *hierarchical clustering* where the solution is obtained either by iteratively

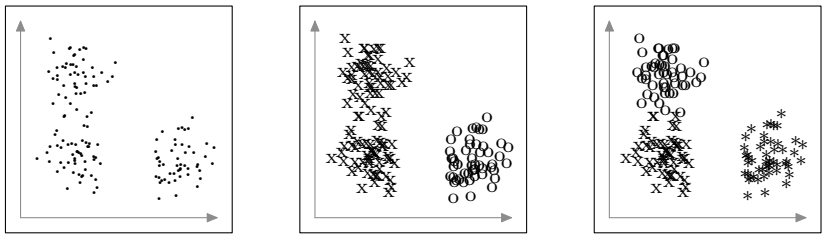


Figure 3.1. Clustering of some data in a Euclidean space (left), solution with two clusters (middle) and solution with three clusters (right). The different labels indicate the different clusters obtained. There is no definite true *solution. The choice of the number of clusters is related to the problem of model selection.*

splitting up (divisive) or putting together (agglomerative) dissimilar resp. similar points, typically via some neighborhood proximity, or *spectral* algorithms which are based upon spectral graph theory (Chung, 1997). Recently, new algorithms have appeared, like *superparamagnetic clustering* (Blatt et al., 1996) which is based upon an analogy with physics, namely the Pott spin model.

Ill-definedness

Clustering is an ill-defined problem, in as much a ground truth does not exist. This poses the problem of validation. In supervised learning the validation is performed based upon the information given by the labels which accompany every data point. In clustering we must rely on other criteria. These can be *extrinsic* like the validation by an expert with a priori knowledge, or *intrinsic* like the validation by e.g. stability analysis. Often it is only a combination of both which leads us towards new insights, as it has been explored e.g. in Schäfer and Laub (2005). The same problem arises in model selection, typically concerning the number of clusters. The interaction of machine driven automation, optimization of some optimality criterion and a subsequent expert interpretation, with possibility to change the preceding criterion, leads to intelligent data analysis.

Extrinsic vs. intrinsic validation

Stability as intrinsic criterion for model selection and validation

In particular, we have to give an answer to the question of how many clusters should be chosen. In Roth et al. (2002), cluster stability has been shown to be a suitable model selection criterion for unsupervised grouping problems. The term stability here refers to structural similarity of partitionings for different problem instances drawn from the same data source. This quantity can be empirically estimated by iteratively splitting the data into two disjoint sets, and measuring the distance between the grouping solutions. However, the stability concept is more than a pure heuristic approach, since it has a clear theoretical interpretation. In terms of statistical learning theory, the principle of favoring solutions with a high stability can be viewed as selecting the most self-consistent labeling of the data. For details see Roth et al. (2002).

§. 3.2.

PROXIMITY BASED CLUSTERING.

Non-metric pairwise data

Partitioning pairwise proximity data is considered a much harder problem, since the inherent structure is hidden in n^2 pairwise relations. Figure 3.2 illustrates clustering pairwise data. It is the pairwise *pendant* to Figure 3.1. Note that in this representation the cluster membership is only given by the ordering; the middle and right figures cannot be distinguished without the labels.

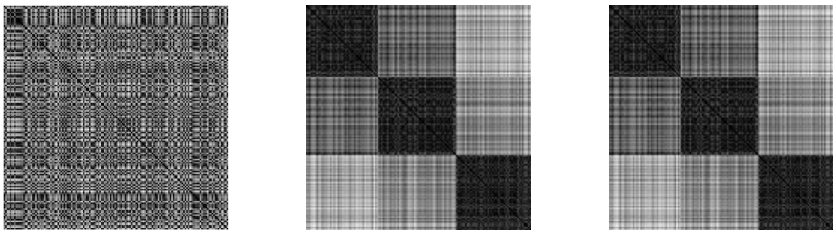


Figure 3.2. Unordered pairwise data (left), ordered according to the solution with two clusters (middle) and, identically, according to the solution with three clusters (right). This figure is the pairwise pendant *to Figure 3.1. Note the usefulness of the checkerboard representation which allows to visualize the cluster structure of the result.*

As we have seen, the proximities can violate the requirements of a distance measure, i.e. they may be non-symmetric and negative, and the triangle inequality does not necessarily hold. A loss-free embedding into a vector space is therefore not possible, so that grouping problems of this kind cannot directly be transformed into geometrically *equivalent* vectorial problems by means of classical scaling.

Drawback of MDS

When one forcefully embeds non-metric pairwise data by embedding strategies like MDS, the problem is that clustering the embedded data vectors in general yields partitionings *different* from those obtained by directly solving the pairwise problem. Even worse, by guaranteeing low (but nonzero) distortions of the proximities, it is still unclear how the object assignments are affected by the embedding.

Cost based clustering

Among several methods for clustering proximity based data, in the following we will focus on those techniques that explicitly minimize a certain cost function. This subset of clustering methods includes e.g. graph-theoretic approaches like several variations of *Cut* criteria (Shi and Malik, 2000), and several methods derived from an axiomatization of pairwise cost functions in Puzicha et al. (1999). From a theoretical viewpoint, cost based clustering meth-

ods are interesting insofar, as many properties of the grouping solutions can be derived by analyzing invariance properties of the cost function.

Shift invariant cost functions

Among the class of cost based criteria, the main focus of this work concerns those cost functions which are invariant under constant additive shifts of the pairwise dissimilarities. For this subset of clustering criteria we show that there always exists a set of vectorial data representations such that the grouping problem can be equivalently restated in terms of Euclidean distances between these vectors. A special cost function of this kind is the *pairwise clustering cost function*. It is of particular interest, since it combines the properties of additivity, scale and shift invariance, and statistical robustness (Puzicha et al., 1999). In Hofmann and Buhmann (1997) this grouping problem is stated as a combinatorial optimization problem, which is optimized in a *deterministic annealing* framework after applying a mean-field approximation.

Pairwise clustering cost function

Main result

According to the Theorem 3.4.1 given on page 45, we can always find a vectorial data representation such that the optimal partitioning with respect to the pairwise cost function is *identical* to k-means partitioning in the embedding space. This property demonstrates that the embedding method is optimal with respect to to distortions of the *data partition*. This distortion preserving embedding has to be contrasted with alternative, in our view not consistent, approaches that are optimal with respect to some *a priori* chosen MDS distortion measure.

Consequences of the theorem

Formulating pairwise clustering as a k-means problem yields several advantages, both of theoretical and technical nature:

1. The availability of prototype vectors defines a generic rule for using the learned partitioning in a predictive way.
2. We can apply standard noise and dimensionality reduction methods in order to separate the “signal” part of the data from underlying “noise”.
3. Fast and efficient local search heuristics for optimizing the clustering cost functional often work much better in low dimensional embedding spaces.

THE PAIRWISE CLUSTERING COST FUNCTION.

Compactness criterion

The modeling idea behind the pairwise clustering cost function is to minimize the sum of *pairwise* intra-cluster distances, emphasizing *compact* clusters. Optimizing a compactness criterion is certainly a very intuitive meta-principle for exploratory data analysis. It should be noticed, however, that other meta-principles have been proposed, such as *separation* measures, mixed *compactness/separation* measures or *connectivity* measures. We will discuss the relation of pairwise clustering to some of these methods in Section 3.5.

Binary assignment matrix

In order to formalize pairwise clustering, we define for each object a binary assignment variable that indicates its cluster membership. Let these variables be summarized in the $(n \times k)$ binary stochastic assignment matrix $M = (m_{ij}) \in \{0,1\}^{n \times k}$ such that $\sum_{\nu=1}^{k} m_{i\nu} = 1$. Given a $(n \times n)$ dissimilarity matrix D, the pairwise clustering cost function reads:

$$H^{\text{pc}} = \frac{1}{2} \sum_{\nu=1}^{k} \frac{\sum_{i,j=1}^{n} m_{i\nu} m_{j\nu} d_{ij}}{\sum_{l=1}^{n} m_{l\nu}}.$$

The optimal assignments $\hat{M}$ are obtained by minimizing H^{pc}. The minimization itself is a NP hard problem (Brucker, 1978), and some approximation heuristics have been proposed: in Hofmann and Buhmann (1997) a *mean field annealing* framework has been presented (see the discussion in Section 3.2 of this work for some comments and new results on annealing). In Puzicha et al. (1999) it has been shown that the time-honored *Ward's method* can be viewed as a hierarchical approximation of H^{pc}.

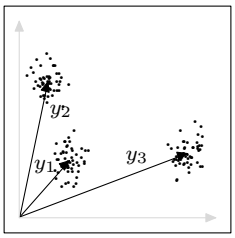


Figure 3.3. k-means prototype vectors for three clusters.

A SPECIAL CASE: k-MEANS CLUSTERING.

For the special case of squared Euclidean distances between vectors $x_1, x_2, \ldots x_n$, $x_i \in \mathbb{R}^p$, it is well known that H^{pc} is identical to the classical k-means cost function, see Duda et al. (2001). We now briefly review this relationship. The k-means cost function is defined as

$$H^{\text{km}} = \sum_{\nu=1}^{k} \sum_{i=1}^{n} m_{i\nu} \|x_i - y_\nu\|^2. \tag{3.1}$$

It measures the sum of squared intra-cluster distances to the prototype vectors

$$y_\nu = \frac{\sum_{i=1}^{n} m_{i\nu} x_i}{n_\nu}, \tag{3.2}$$

where $n_\nu = \sum_{l=1}^{n} m_{l\nu}$ denotes the number of objects in cluster ν (Figure 3.3). H^{km} can be written in a pairwise fashion by exploiting a simple algebraic identity for squared Euclidean distances:

$$\|x_i - y_\nu\|^2 = \frac{1}{n_\nu} \sum_{j=1}^{n} m_{j\nu} \|x_i - x_j\|^2 - \frac{1}{2n_\nu^2} \sum_{j,l=1}^{n} m_{j\nu} m_{l\nu} \|x_j - x_l\|^2,$$

$$\sum_{i=1}^{n} m_{i\nu} \|x_i - y_\nu\|^2 = \frac{1}{2n_\nu} \sum_{j,l=1}^{n} m_{j\nu} m_{l\nu} \|x_j - x_l\|^2.$$

Substituting the latter into Equation 3.1, we obtain

$$H^{\text{km}} = \frac{1}{2}\sum_{\nu=1}^{k} \frac{\sum_{i,j=1}^{n} m_{i\nu} m_{j\nu} \|x_i - x_j\|^2}{\sum_{l=1}^{n} m_{l\nu}} = H^{\text{pc}}.$$

From this viewpoint, k-means clustering can be interpreted as a method for minimizing the sum of squared *pairwise* intra-cluster distances $d_{ij} = \|x_i - x_j\|^2$. The reader should notice, however, that in the general case of arbitrary dissimilarities d_{ij} a direct algebraic re-transformation of H^{pc} into H^{km} is *not* possible since there is no algebraic relationship between the d_{ij}'s and $\|x_i - x_j\|$ as we have seen in the previous chapter. Despite this fact, we will show that it is still possible to obtain the optimal assignment variables $\hat{M}$ with respect to $H^{\text{pc}}(M)$ by minimizing a suitably transformed k-means problem. The key ingredient will be the *shift invariance property* of the pairwise clustering cost function described in the following subsection.

INVARIANCE PROPERTIES OF THE PAIRWISE CLUSTERING COST FUNCTION.

Important invariance properties

The pairwise clustering cost function has two important invariance properties:

1. H^{pc} is invariant under symmetrizing transformations

$$\tilde{d}_{ij} = \frac{1}{2}(d_{ij} + d_{ji}) \;\Rightarrow\; \tilde{H} = H. \tag{3.3}$$

2. H^{pc} is invariant (up to a constant) under additive shifts of the *off-diagonal* elements of the dissimilarity matrix:

$$\tilde{d}_{ij} = d_{ij} + d_0(1 - \delta_{ij}) \;\Rightarrow\; \tilde{H} = H + \frac{1}{2}(n-k)d_0 = H + \text{const.} \tag{3.4}$$

Note that the optimal assignments of objects to clusters are not influenced by adding a constant to the cost function, i.e. $\hat{M}(\tilde{D}) = \hat{M}(D)$.

§. 3.3.

CONSTANT SHIFT EMBEDDING.

Consequence of shift-invariance

In Section 3.2 we have introduced the cost function H^{pc} as a special instance of pairwise clustering problems. Due to the shift-invariance property (Equation 3.4), the partitioning of the data set (i.e. the assignments of a set of n

objects to k clusters) is not affected by a constant additive shift on the off-diagonal elements of the pairwise dissimilarity matrix $D = (d_{ij}) \in \mathbb{R}^{n \times n}$. We will consider general dissimilarity matrices D, restricted only by the constraint that all self-dissimilarities be zero, i.e. that D has zero diagonal elements. We show that by exploiting the above shift invariance we can always embed such data into a Euclidean space without influencing the cluster structure. An off-diagonal shifted dissimilarity matrix reads

Off-diagonal shift

$$\tilde{D} = D + d_o(e_n e_n^t - I_n), \tag{3.5}$$

where $e_n = (1, 1, \dots 1)^t$ is a vector $\in \mathbb{R}^n$ of ones and I_n the $n \times n$ identity matrix. In other words, Equation 3.5 describes a constant additive shift $\tilde{d}_{ij} = d_{ij} + d_o$ for all $i \neq j$.

Let us now consider only *symmetric* dissimilarity matrices. Note that for the clustering criterion H^{pc} this requirement imposes no restrictions on the general applicability, since H^{pc} is invariant under symmetrizing transformations (Equation 3.3). Given such a symmetric and zero-diagonal matrix D, let us decompose it as in Equation 2.10 in the following way by introducing a new matrix $C = (c_{ij})$:

$$d_{ij} = c_{ii} + c_{jj} - 2c_{ij}.$$

Decomposition

For general dissimilarities, C^c will be indefinite. By shifting its diagonal elements, however, we can transform it into a positive semi-definite matrix: the following lemma states that for any matrix A, a positive semi-definite matrix $\tilde{A}$ can be derived by subtracting the smallest eigenvalue from all of its diagonal elements:

LEMMA 3.3.1. *Let $\tilde{A} = A - \lambda_n(A) I_n$, where $\lambda_n(\cdot)$ is the minimal eigenvalue of its argument. Then $\tilde{A}$ is positive semi-definite.*

Proof. The spectrum of $\tilde{A}$ is given by the roots λ of the characteristic polynomial defined by $\det(\tilde{A} - \lambda I_n)$. $\det(\tilde{A} - \lambda I_n) = \det\big(A - (\lambda_n(A) + \lambda) I_n\big)$, so that $\lambda_i(\tilde{A}) = \lambda_i(A) - \lambda_n(A)$. The smallest eigenvalue of $\tilde{A}$ is given by $\lambda_n(\tilde{A}) = \lambda_n(A) - \lambda_n(A) = 0$. Therefore $\tilde{A}$ is positive semi-definite. □

Given a matrix D, there exists a unique matrix C^c by Lemma 2.4.2. If C^c is not positive semi-definite, Lemma 3.3.1 states that by subtracting $\lambda_n(C^c)$ from its diagonal elements, we obtain a positive semi-definite $\tilde{C}$. Returning to Equation 2.10 with our fixed matrix C^c, such a diagonal shift of C^c corresponds to an *off-diagonal* shift of the dissimilarities

$$\tilde{d}_{ij} = \tilde{c}_{ii} + \tilde{c}_{jj} - 2\tilde{c}_{ij} \;\Leftrightarrow\; \tilde{D} = D - 2\lambda_n(C^c)(e_n e_n^t - I_n), \tag{3.6}$$

since $\tilde{d}_{ij} = \tilde{c}_{ii} + \tilde{c}_{jj} - 2\tilde{c}_{ij} = \tilde{c}^c_{ii} + \tilde{c}^c_{jj} - 2\tilde{c}^c_{ij} - \lambda_n(C^c)\big((I_n)_{ii} + (I_n)_{jj} - 2(I_n)_{ij}\big) = D - \lambda_n(C^c)\big((I_n)_{ii} + (I_n)_{jj} - 2(I_n)_{ij}\big)$ and $\big((I_n)_{ii} + (I_n)_{jj} - 2(I_n)_{ij}\big)$ equals 0 for $i = j$ and 2 for $i \neq j$.

In other words, if we were given $\tilde{D}$ instead of our original D, then $\tilde{C}$ would be a positive semi-definite member of the equivalence class $\mathcal{C}_{\tilde{D}}$ of matrices fulfilling the decomposition $\tilde{d}_{ij} = \tilde{c}_{ii} + \tilde{c}_{jj} - 2\tilde{c}_{ij}$. Theorem 2.4.2 then tells us that this off-diagonally shifted matrix $\tilde{D}$ derives from a squared Euclidean distance. Since every positive semi-definite matrix is a dot product matrix in some vector space, there exists a matrix X of vectors such that $\tilde{C} = XX^t$. The matrix $\tilde{D}$ then contains squared Euclidean distances between these vectors.

Preservation of cluster assignments

We can now insert $\tilde{D}$ into our clustering procedure (which is assumed shift-invariant), and we will obtain the same partition of the objects as if we had clustered the original matrix D. Contrary to directly using D, however, the matrix $\tilde{D}$ now contains squared Euclidean distances between a set of vectors $\{x_1, x_2, \ldots x_n\}$ which can be recovered according to the PCA algorithm presented in the previous chapter (see page 24).

Summary

The above procedure can be summarized as follows:

$$
\begin{array}{c}
D \\
\downarrow \text{ decomposition via } d_{ij} = c_{ii} + c_{jj} - 2c_{ij} \\
C \in \mathcal{C}_D \\
\downarrow \text{ centralization via } C^c = QCQ \\
C^c = -\frac{1}{2}D^c \\
\downarrow \text{ diagonal shift via } \tilde{C} = C^c - \lambda_n(C^c)I_nV \\
\tilde{C} = XX^t \\
\downarrow \text{ off-diag. shifted dissimilarities} \\
\tilde{d}_{ij} = \tilde{c}_{ii} + \tilde{c}_{jj} - 2\tilde{c}_{ij} \\
\downarrow \text{ clustering assignments} \\
M(\tilde{D}) = M(D).
\end{array}
$$

Figure 3.4 illustrates this additive shift, based upon the triangle in Figure 2.5 which violated triangle inequality because of its noisy distances. The triangle in Figure 3.4 now satisfies the triangle inequality.

In principle, the above derivation holds true not only for the centralized matrix C^c, but for any member C of the of the equivalence class $\mathcal{C}_D$. Some of these members, however, will eventually have very large negative eigenvalues, which means that we would have to add a very large constant to all off-diagonal entries of D. For numerical reasons we want to avoid these problems, which leads us to the question of the *minimal* necessary shift. The next theorem states that our above choice of using C^c is optimal in this sense:

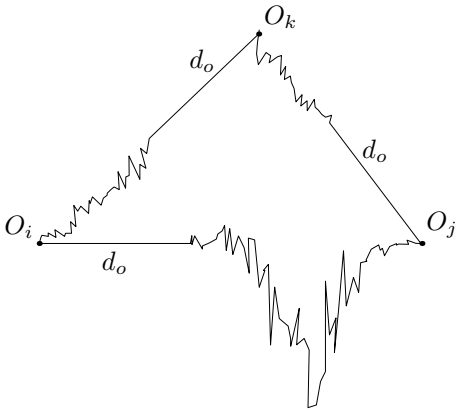


Figure 3.4. Distance as measured by some tool. See Figure 2.5 for the original triangle violating the triangle inequality. Thanks to the minimal shift, the transformed triangle now satisfies the triangle inequality. The clustering solution for the dissimilarities thus rendered metric does not change.

THEOREM 3.3.1. *(Cox and Cox, 2001). $d_o = -2\lambda_n(C^c)$ is the minimal constant such that $\tilde{D} = D + d_o\big(e_n e_n^t - I_n\big)$ derives from squared Euclidean distance.*

Minimal shift

Proof. A proof is given in Cox and Cox (2001). It also follows from Theorem 2.4.2 and Lemma 3.3.1, or the following simple argument:

Suppose that D is non-metric. In order to get a metric distance, we add a constant $d_o > 0$, i.e.

$$\tilde{d}_{ij} = d_{ij} + d_o(ee^t - I_n).$$

The centralized kernel matrix becomes

$$\tilde{C}^c = -\frac{1}{2}\tilde{D}^c = -\frac{1}{2}D^c + \frac{1}{2}d_oQ = C^c + \frac{1}{2}d_oQ.$$

Let $\lambda_1 \geqslant \cdots \geqslant \lambda_p > \lambda_{p+1} = 0 \geqslant \lambda_{p+2} \geqslant \cdots \geqslant \lambda_n$ be the eigenvalues. Then,

$$\lambda_1 + \frac{d_o}{2} \geqslant \cdots \geqslant \lambda_p + \frac{d_o}{2} > \lambda_{p+2} + \frac{d_o}{2} \geqslant \cdots \geqslant \lambda_n + \frac{d_o}{2},\ \lambda_{p+1} = 0,$$

therefore, $\tilde{C}^c$ is positive semi-definite, if $d_o \geqslant -2\lambda_n$.

The distortion caused by this change is (Cox and Cox, 2001)

$$\text{tr}(\tilde{D} - D)^2 = \sum_{i,j} d_o(1 - \delta_{ij})d_o(1 - \delta_{ji}) = n(n-1)d_o^2.$$

where δ_{ij} is the Kronecker symbol, $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ for $i \neq j$. Therefore, we must choose d_o as small as possible. This means that $d_o = -2\lambda_n$ gives the optimal constant shift to metric distance $\tilde{D}$. □

EXAMPLE. Let D be a squared matrix such that

$$\sqrt{D} = \begin{pmatrix} 0 & 1 & 3 \\ 1 & 0 & \sqrt{2} \\ 3 & \sqrt{2} & 0 \end{pmatrix},$$

which does not satisfy triangle inequality since $3 \nleq 1 + \sqrt{2}$

C^c is given by

$$\begin{pmatrix} 2 & \frac{1}{3} & -\frac{7}{3} \\ \frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{7}{3} & 0 & \frac{7}{3} \end{pmatrix},$$

with eigenvalues $\{-0.5166, -0.0000, 4.5166\}$.

Now, $\tilde{D} = D + d_o(ee^t - I_n) = D - 2\lambda_n(C^c)(ee^t - I_n)$ so that

$$\sqrt{\tilde{D}} = \begin{pmatrix} 0 & 1.4259 & 3.1675 \\ 1.4259 & 0 & 1.7416 \\ 3.1675 & 1.7416 & 0 \end{pmatrix},$$

which "just" satisfies triangle inequality. The corresponding $\tilde{C^c}$ is positive semi-definite.

RECONSTRUCTING THE EMBEDDED VECTORS.

Given a general dissimilarity matrix D, in the last section we have shown how to obtain a shifted matrix $\tilde{D}$ which derives from squared Euclidean distances between points $x_1, x_2, \ldots x_n$ in some vector space. This property of $\tilde{D}$ implies that the corresponding matrix $\tilde{C^c}$ is positive semi-definite, and thus a dot product matrix $\tilde{C^c} = XX^t$. According to Lemma 2.4.2, $\tilde{C^c}$ can be calculated as $\tilde{C^c} = -1/2\tilde{D^c}$. The vectors $x_1, x_2, \ldots x_n$ can be recovered by an eigenvalue decomposition of $\tilde{C^c}$ as in the algorithm given in the previous chapter (see page 24).

Preprocessing

So far we have discussed an exact reconstruction of the structure preserving vectors in the embedding space. While this has both important theoretical and practical consequences (see Section 3.2), in many applications we would like to insert some preprocessing step in our clustering procedure. A typical example of this kind would be the suppression of noise. When focusing on noise reduction, we are interested in some sort of approximative reconstructions of the exact vectors. The reader should notice that given the vectorial representations $x_1, x_2, \ldots x_n$ in a Euclidean space, the issue of separating the "noisy" part of the data from the "signal" part can be handled within a well-defined framework. On the contrary, in the original pairwise setting without a common vector space structure, to our knowledge there exist no general purpose denoising methods. For instance, it is not clear how to define a global noise model

Denoising

that specifies the amount of noise by which each single object is corrupted. The semantics of a generative model which is responsible for the "signal" part is also unclear.

PCA

In Principal Component Analysis (PCA), one usually assumes that the directions corresponding to small eigenvalues contain the noise (Mika et al., 1999). We can thus obtain a representation in a space of reduced dimension (with the well-defined error of PCA reconstruction) when choosing $t < n - 1$ dimensions in the PCA algorithm of page 24: $X_t = V_t \Lambda_t^{1/2}$, where V_t consists of the first t column vectors of V and Λ_t is the top $t \times t$ submatrix of Λ. The vectors in $\mathbb{R}^t$ then differ the least from the vectors in $\mathbb{R}^p$ in the sense of quadratic approximation error. This means that the embedded vectors are the best least squares error approximation to the optimal vectors which preserve the group structure. The mathematical tractability of error constitutes the main difference to directly decomposing C^c (i.e. without shifting) and projecting onto a subset of eigenvectors with positive eigenvalue, as in classical scaling. In the latter case, there exist no optimal vectors (in the sense of structure preservation), since only the positive eigenvalues can be used for deriving a vector representation. For classical scaling, it is thus unclear, what "objects" are approximated and with what error.

Summary

The processing pipeline of both the loss-free vector reconstruction and the PCA approximation is summarized in the following algorithm:

$$
\begin{array}{c}
D \\
\downarrow \text{ constant shift embedding} \\
\tilde{D} \\
\downarrow \text{ decomposition } \tilde{d}_{ij} = \tilde{c}_{ii} + \tilde{c}_{jj} - 2\tilde{c}_{ij} \\
\tilde{C} \\
\downarrow \text{ centering} \\
\tilde{C}^c = -\frac{1}{2}\tilde{D}^c \\
\downarrow \text{ loss-free reconstruction} \\
X = V\Lambda^{\frac{1}{2}} \\
\downarrow \text{ approximation and denoising} \\
X_t = V_t\Lambda_t^{\frac{1}{2}},\ t < n-1.
\end{array}
$$

Dimension reduction

It should be noticed, however, that given the exactly reconstructed vectors in $\mathbb{R}^p$, we can also apply any other standard method for dimensionality reduction or visualization, such as *projection pursuit* (Huber, 1985), *locally linear em-*

bedding (LLE) (Roweis and Saul, 2000), *Isomap* (Tenenbaum et al., 2000) or *Selforganizing maps* (Kohonen, 1995). These methods can also be viewed as approximations of the optimal structure preserving vectors, employing, however, an approximation criterion different from the squared error as in the case of the above PCA framework.

PREDICTING THE CLUSTER MEMBERSHIP OF NEW DATA.

First notice that due to the eigenvalue equation $\tilde{C}^c V = V\Lambda$, we can rewrite Equation 2.19 in the form:

$$X = \tilde{C}^c V \Lambda^{-\frac{1}{2}}.$$

Consider now the situation where we are given m new objects and the corresponding $m \times n$ matrix of pairwise dissimilarities d_{ij}^{new} between these new objects and all n original objects. In order to predict the cluster membership of the new objects, we first have to project them into the Euclidean space spanned by the eigenvectors V of the centered dot product matrix $\tilde{C}^c$. Then, we assign each new object to the cluster with the closest centroid. For the projection itself, two steps are required. First compute the matrix C_{new} defined by

$$d_{ij}^{\text{new}} = c_{ii}^{\text{new}} + \tilde{c}_{jj}^c - 2c_{ij}^{\text{new}}. \tag{3.7}$$

Similar to the situation in Equation 2.10, we still have the problem of ambiguities due to the freedom of choosing c_{ii}^{new}. This problem, however, is automatically overcome by re-expressing the matrix C^{new} in the centered coordinate system:

$$(c^{\text{new}})_{ij}^c = c_{ij}^{\text{new}} - \frac{1}{n}\sum_{k=1}^{n} c_{ik}^{\text{new}} - \frac{1}{n}\sum_{k=1}^{n} \tilde{c}_{kj}^c + \frac{1}{n^2}\sum_{k,l=1}^{n} \tilde{c}_{kl}^c.$$

Substituting Equation 3.7 into the above equation and noticing that $\tilde{D}$ and $\tilde{C}^c$ are connected by $\tilde{d}_{ij} = \tilde{c}_{ii}^c + \tilde{c}_{jj}^c - 2\tilde{c}_{ij}^c$, we can restate $(C^{\text{new}})^c$ solely in terms of D^{new} and $\tilde{D}$:

$$(c^{\text{new}})_{ij}^c = -\frac{1}{2}\Big(d_{ij}^{\text{new}} - \frac{1}{n}\sum_{k=1}^{n} d_{ik}^{\text{new}} - \frac{1}{n}\sum_{k=1}^{n} \tilde{d}_{kj} + \frac{1}{n^2}\sum_{k,l=1}^{n} \tilde{d}_{kl}\Big),$$

that is,

$$(C^{\text{new}})^c = -\frac{1}{2}\Big(D^{\text{new}}(I_n - \frac{1}{n}e_n e_n^t) - \frac{1}{n}e_m e_n^t \tilde{D}(I_n - \frac{1}{n}e_n e_n^t)\Big).$$

Second, project the objects represented by $(C^{\text{new}})^c$ into the coordinate system spanned by the eigenvectors V of the matrix $\tilde{C}^c$:

$$X^{\text{new}} = (C^{\text{new}})^c \, V \Lambda^{-\frac{1}{2}}. \tag{3.8}$$

Summary

The whole process flow for predicting the cluster membership of new objects is summarized as follows:

$$
\begin{aligned}
& D^{\text{new}} \\
& \Big\downarrow \text{ decomposition } d_{ij}^{\text{new}} = c_{ii}^{\text{new}} + \tilde{c}_{jj}^{c} - 2c_{ij}^{\text{new}} \\
& C^{\text{new}} \\
& \Big\downarrow \text{ Centering} \\
& (C^{\text{new}})^c = -\tfrac{1}{2}\Big(D^{\text{new}}(I_n - \tfrac{1}{n}e_n e_n^t) - \tfrac{1}{n}e_m e_n^t \tilde{D}(I_n - \tfrac{1}{n}e_n e_n^t)\Big) \\
& X^{\text{new}} = (C^{\text{new}})^c V \Lambda^{-\frac{1}{2}} \\
& \Big\downarrow \text{ assignment to closest centroid } \{y_1, y_2, \dots\ y_k\} \\
& \hat{\nu}(x^{\text{new}})_i = \arg\min_\nu \|(x^{\text{new}})_i - y_\nu\|^2.
\end{aligned}
$$

Prediction (schematic): from the preceding clustering step we are given the squared Euclidean distances $\tilde{D}$, the centered dot-product matrix $\tilde{C}^c = -\frac{1}{2}\tilde{D}^c$, its eigenvectors and its eigenvalues $V,\ \Lambda$, and the cluster centroids $\{y_\nu\}_{\nu=1}^k$. Prediction step 1: decomposing D^{new} and re-expressing the matrix C^{new} in the centered coordinate system of $\tilde{C}^c$. Step 2: projecting the new objects on the eigenvectors V of $\tilde{C}^c$. Step 3: assigning objects to the cluster with the closest centroid vector y_ν.

§. 3.4.

SUMMARY.

Relationship between k-means cost function and pairwise cost function

For the special case of squared Euclidean distances, the pairwise cost function and the k-means cost function can be transformed into each other by using a simple algebraic identity, cf. Section 3.2. With the results of the last section, we are now able to prove that a similar relationship between both cost functions holds in the general setting:

THEOREM 3.4.1. *Given an arbitrary $(n \times n)$ dissimilarity matrix D with zero self-dissimilarities, there exists a transformed matrix $\tilde{D}$ such that*

1. *the matrix $\tilde{D}$ can be interpreted as a matrix of squared Euclidean distances between a set of vectors $\{x_1, x_2, \dots\ x_n\}$ with dimensionality dim$(x_i) \leqslant n-1$,*

2. *the original pairwise clustering problem defined by the cost function $H^{pc}(D)$ is equivalent to the k-means problem with cost function H^{km} in this vector space, i.e. the optimal cluster assignment variables $\hat{m}_{i\nu}$ are identical in both problems: $\hat{M}^{pc}(D) = \hat{M}^{km}(\tilde{D})$.*

Proof. 1. Let $\tilde{D}$ be the symmetrized and off-diagonal shifted version of D:

$$D_{\text{sym}} = \frac{1}{2}(D + D^t) \tag{3.9}$$

$$C^c = -\frac{1}{2} Q D_{\text{sym}} Q = -\frac{1}{2} D^c_{\text{sym}} \tag{3.10}$$

$$\tilde{D} = D_{\text{sym}} - 2\lambda_n(C^c)(e_n e_n^t - I_n). \tag{3.11}$$

According to Section 3.3 and the theorems mentioned therein, there exists a set of vectors $\{x_1, x_2, \ldots x_n\}$ with dimensionality $\dim(x_i) \leqslant n-1$ such that $\tilde{D}$ contains squared Euclidean distances between these vectors. 2. Since $\tilde{D}$ represents squared Euclidean distances, Equation 3.2 implies that the pairwise clustering cost function is identical to the k-means function: $H^{\text{pc}}(\tilde{D}) = H^{\text{km}}(\tilde{D})$. According to the invariance properties given by Equation 3.3 and Equation 3.4, the optimal assignments $\{\hat{m}_{i\nu}\}$ of objects to clusters are not influenced by the transformations given by Equation 3.9 and Equation 3.11 of D into $\tilde{D}$, i.e. $\hat{M}(D) = \hat{M}(\tilde{D})$. □

Consequences

The above theorem has several important consequences.

INTERPRETATION AND REPRESENTATION. Rewriting pairwise clustering as a k-means problem naturally introduces the notion of cluster centroids or cluster representants.

PREDICTION. The cluster prototypes define a generic prediction rule for new objects.

DATA PREPROCESSING AND DENOISING. The vectorial representation of the objects allows us to apply standard preprocessing and denoising methods. Note that the usual semantics of "signal" and "noise" is closely related to some sort of generative model in a vector space.

OPTIMIZATION. Minimizing the pairwise clustering cost function is an NP-hard problem. The associated k-means problem with loss-free reconstructed vectors has the same complexity, since the dimensionality of the vectors grows with n, see Drineas et al. (1999). Thus, for handling real-word problems, in both cases efficient approximation algorithms or schemes are necessary. In Hofmann and Buhmann (1997) it has been proposed to optimize H^{pc} by way of

deterministic annealing. Since annealing methods are not our main focus, we only mention that deterministic annealing is feasible only for *factorial* Gibbs distributions Puzicha et al. (1999). For $H^{\mathrm{pc}}(D)$, this constraint requires the use of a *mean-field approximation*. Applying Theorem 3.4.1, however, we are able to anneal the shifted k-means cost function $H^{\mathrm{km}}(\tilde{D})$, for which the mean-field approximation becomes *exact*. For details on annealing and mean-field approximations, the interested reader is referred to Hofmann and Buhmann (1997), Rose et al. (1990).

Preprocessing and optimization

If one decides to insert a denoising and dimensionality reduction step into the clustering procedure, this will usually not only speed up the computations, but it will also "robustify" optimization heuristics for the k-means problem. For instance, applying PCA approximations according to Section 3.3, the energy landscape typically will be smoothed out, which makes local search heuristics (such as the classical iterative k-means algorithm) less sensitive to being trapped in local minima.

SUMMARIZING DIAGRAM.

Let D be a dissimilarity matrix possibly violating symmetry and triangle inequality. Let $\tilde{D}$ be its symmetrized and shifted version. $M(\cdot)$ denotes a partition (assignment matrix) of the data contained in its argument. The first line of the diagram represents the data on the level of pairwise data. The second on the level of a loss-free embedding with respect to cluster assignment. Finally the last on the level of a low dimensional approximation. The left half of the diagram shows what can be achieved with a general dissimilarity matrix, the right half with its symmetrized and shifted version.

$$
\begin{array}{ccccccc}
D & \xrightarrow{k\text{-means}} & M(D) & = & M(\tilde{D}) & \xleftarrow{k\text{-means}} & \tilde{D} \\
\downarrow & & \downarrow & & \| & & \downarrow \\
\emptyset & & \emptyset & & M(x_1, x_2, \dots x_n) & \xleftarrow{k\text{-means}} & (x_1, x_2, \dots x_n) \in \mathbb{R}^{n-1} \\
{\scriptstyle \mathrm{MDS}}\downarrow & & & & & & \downarrow{\scriptstyle \mathrm{PCA}} \\
(x_1, x_2, \dots x_n) \in \mathbb{R}^t & \xrightarrow{k\text{-means}} & M(x_1, x_2, \dots x_n) & & M(x_1, x_2, \dots x_n) & \xleftarrow{k\text{-means}} & (x_1, x_2, \dots x_n) \in \mathbb{R}^t
\end{array}
$$

§. 3.5.

RELATIONS TO GRAPH-THEORETIC CLUSTERING METHODS.

Equivalence of several cost functions to k-means

In this section we discuss the relations between graph-theoretic grouping principles and the constant shift embedding method for pairwise clustering. As main result, we show that both the *Averaged Association* and the *Averaged Cut* cost function are shift-invariant. With this invariance property, the *Averaged Association* principle turns out to be equivalent to the k-means clustering algorithm in the embedding space. Using the same strategy, we show that *Averaged Cut* is equivalent to the *pairwise separation* cost function. The latter can also be stated in terms of Euclidean distances between embedded vectors. For the *Normalized Cut* method, on the other hand, the constant shift embedding method is not applicable. In the case of balanced partitions with similar structure among all clusters, however, the differences between *Averaged Association*, *Averaged Cut* and *Normalized Cut* become vanishingly small. In such situations, all three methods can be reasonably well approximated by k-means.

A graph $G = (V, E)$ can be partitioned into disjoint sets $A^\nu,\ \nu = 1, \ldots, k$ by removing edges: $\bigcup_{\nu=1}^{k} A^\nu = V,\ A^\nu \cap A^\mu = \emptyset$ for $\nu \neq \mu$. Following Shi and Malik (2000), we define the similarity between the sets A^ν and $V - A^\nu$ by the total weight of the edges that have been removed

$$\text{cut}(A^\nu, V - A^\nu) = \sum_{\substack{u \in A^\nu \\ v \in (V - A^\nu)}} w(u, v),$$

where the weight on each edge, $w(u, v)$, is a function of the similarity between nodes u and v. We further introduce a measure of association between two sets, $\text{assoc}(A, B)$, as the total connection from nodes in set A to the nodes in set B. It follows immediately that both measures are connected by the formula

$$\text{cut}(A^\nu, V - A^\nu) = \text{assoc}(A^\nu, V) - \text{assoc}(A^\nu, A^\nu).$$

We further denote by W the similarity (weight) matrix with unit self-similarities: $w_{ii} = 1,$ for all $i = 1, \ldots, n$. Based on this similarity matrix, we define a dissimilarity matrix by $D = e_n e_n^t - W$, with $e_n = (1, 1, \ldots 1)^t$ as before. Together with the notation of the binary assignment variables $m_{i\nu}$ and the def-

inition $n_\nu = |A^\nu|$, we can write the association measure in the form

$$\begin{aligned}\text{assoc}(A^\nu, A^\nu) &= \sum_{i,j=1}^{n} m_{i\nu} m_{j\nu} w_{ij} = \sum_{i,j=1}^{n} m_{i\nu} m_{j\nu}(1 - d_{ij}) \\ &= n_\nu^2 - \sum_{i,j=1}^{n} m_{i\nu} m_{j\nu} d_{ij}.\end{aligned} \tag{3.12}$$

For two sets, $A \cup B = V$, $A \cap B = \emptyset$, in Shi and Malik (2000) the *Averaged Association* cost function has been defined as

Averaged Association...

$$\text{AvAssoc} = \frac{\text{assoc}(A, A)}{|A|} + \frac{\text{assoc}(B, B)}{|B|}.$$

It can be easily extended for a k-partitioning problem:

$$\text{AvAssoc}_k = \sum_{\nu=1}^{k} \frac{\text{assoc}(A^\nu, A^\nu)}{n_\nu}.$$

Inserting $D = e_n e_n^t - W$ and Equation 3.12, we see that maximizing the averaged association is equivalent to minimizing the *pairwise clustering* cost function H^{pc}:

$$\text{AvAssoc}_k(W) = \sum_{\nu=1}^{k} \frac{\text{assoc}(A^\nu, A^\nu)}{n_\nu} = n - 2H^{\text{pc}}(e_n e_n^t - W). \tag{3.13}$$

According to Theorem 3.4.1, it is always guaranteed that the (possibly shifted) matrix $C^c = -\frac{1}{2} D^c$ is a positive semi-definite dot-product matrix which can be used to embed the data into a Euclidean space. In this space the problem of minimizing the pairwise clustering function reduces to a standard k-means problem.

... is identical to k-means

The *Averaged Cut* cost function, cf. Shi and Malik (2000), is defined as

Averaged Cut...

$$\text{AvCut}_k = \sum_{\nu=1}^{k} \frac{\text{cut}(A^\nu, V - A^\nu)}{n_\nu} = \sum_{\nu=1}^{k} \frac{\text{assoc}(A^\nu, V) - \text{assoc}(A^\nu, A^\nu)}{n_\nu}.$$

In the following we will show that AvCut is equivalent to the *Pairwise Separation* cost function H^{ps} (in Puzicha et al. (1999) this cost function is denoted by H^{ps1a}):

$$\begin{aligned}H^{\text{ps}} &= -\sum_{\nu=1}^{k} \sum_{i=1}^{n} m_{i\nu} \frac{1}{k-1} \sum_{\mu \neq \nu} \frac{\sum_{j=1}^{n} m_{j\mu} d_{ij}}{\sum_{j=1}^{n} m_{j\mu}} \\ &= -\frac{1}{k-1} \Bigg(\sum_{\nu=1}^{k} \frac{1}{n_\nu} \sum_{i,j=1}^{n} m_{i\nu} d_{ij} - 2H^{\text{pc}} \Bigg).\end{aligned}$$

With Equation 3.13 and the identity

$$\text{assoc}(A^\nu, V) = \sum_{i,j=1}^{n} m_{i\nu} m_{ij} = n n_\nu - \sum_{i,j=1}^{n} m_{i\nu} d_{ij},$$

AvCut can be reformulated in terms of H^{ps}:

$$\begin{aligned} \text{AvCut}_k &= \sum_{\nu=1}^{k} \frac{\text{assoc}(A^\nu, V)}{n_\nu} - n + 2H^{\text{pc}} \\ &= kn - \sum_{\nu=1}^{k} \frac{1}{n_\nu} \sum_{i,j=1}^{n} m_{i\nu} d_{ij} - n + 2H^{\text{pc}} \\ &= (k-1)n + (k-1)H^{\text{ps}}. \end{aligned} \tag{3.14}$$

Minimizing the averaged cut cost function based on the affinity matrix W is thus equivalent to minimizing H^{ps} with distances $D = e_n e_n^t - W$. Note that the separation measure H^{ps} has the same shift-invariance property as its compactness counterpart H^{pc}:

$$H^{\text{ps}}\big(D + d_0(1 - \delta_{ij})\big) = H^{\text{ps}} + \text{const}.$$

... is shift invariant

We can thus directly apply the constant shift embedding framework of Section 3.3.

Normalized Cut...

The *Normalized Cut* cost function, cf. Shi and Malik (2000), is an intermediate grouping criterion that combines both the compactness and separation principle. The k-cluster version is defined as

$$\text{Ncut}_k = \sum_{\nu=1}^{k} \frac{\text{cut}(A^\nu, V - A^\nu)}{\text{assoc}(A^\nu, V)} = k - \sum_{\nu=1}^{k} \frac{\text{assoc}(A^\nu, A^\nu)}{\text{assoc}(A^\nu, V)}.$$

Rewriting this in terms of distances $D = e_n e_n^t - W$, we arrive at

$$\text{Ncut}_k = k - \sum_{\nu=1}^{k} \Big(\frac{n_\nu - n_\nu^{-1} \sum_{i,j=1}^{n} m_{i\nu} m_{j\nu} d_{ij}}{n - n_\nu^{-1} \sum_{i,j=1}^{n} m_{i\nu} d_{ij}} \Big). \tag{3.15}$$

... is almost shift invariant

Contrary to AvAssoc and AvCut, the Ncut cost function is not shift invariant. For non-metric (dis)similarities, it is thus not possible to apply the constant shift embedding trick to obtain a grouping problem in a vector space. However, for the special case of balanced partitionings, $n_\nu = \frac{n}{k}$ for all ν, and similar distribution of intra-cluster distances among all groups, all the row-sums of the distance matrix tend to be similar. Assuming $\sum_{j=1}^{n} d_{ij} = \text{const}$ and substituting this into Equation 3.14, or Equation 3.15 respectively, we see that in this

special case both the AvCut_k and the Ncut_k criteria become equivalent to the AvAssoc_k criterion, and hence equivalent to the H^{pc} cost function. This equivalence means that for clustering problems with similar group structure and balanced partitions large differences between the models will become vanishingly small. The somewhat surprising results of a large-scale comparison study of graph partitioning algorithms for image segmentation tasks in Soundararajan and Sarkar (2001) could be explained in the light of this analysis.

§. 3.6.

APPLICATIONS.

Three applications

We will illustrate the constant shift embedding by three applications from protenomics. The first example shows that CSE can be successfully applied to denoise pairwise data in a mathematical rigorous fashion, which cannot be achieved for non-metric pairwise data with traditional techniques. The second application is a worked through example of combining CSE and low-dimensionsal approximations, model selection and clustering to globin protein sequences. In the third example, we apply CSE to cluster protein sequences of the ProDom database with respect to structural similarity.

BACTERIAL *GyrB* AMINO ACID SEQUENCES.

The GyrB data set

Our first illustration involves the gyrase subunit B. The data set consists of 84 amino acid sequences from five genera in *Actinobacteria*: 1: *Corynebacterium*, 2: *Mycobacterium*, 3: *Gordonia*, 4: *Nocardia* and 5: *Rhodococcus*. A detailed description can be found in Kasai et al. (1998). This data set was used in Tsuda et al. (2002) for illustration of marginalized kernels. The authors hinted at the possibility of computing the distance matrix by using BLAST scores (Altschul et al., 1990), noting, however, that these scores could not be converted into positive semi-definite kernels.

Computation of the similarity matrix & CSE

In our experiment, the sequences have been aligned by the Smith-Waterman algorithm (Pearson and Lipman, 1988) which yields pairwise alignment scores. The associated pseudo-covariance matrix exhibits a few strongly negative eigenvalues as seen in Figure 3.5. Using constant shift embedding a *positive semi-definite* kernel is obtained, leaving the cluster assignment unchanged for shift invariant cost functions.

Denoising

The important step is the denoising. Several projections to lower dimensions have been tested and $t = 5$ turned out to be a good choice, eliminating the bulk of noise while retaining the essential cluster structure.

Figure 3.5. The spectrum of the centralized covariance matrix. We see that it exhibits strongly negative eigenvalues pointing to severe metric violations. It cannot be embedded loss-free into a Euclidean space with respect to a metric. Denoising is not properly defined on such data.

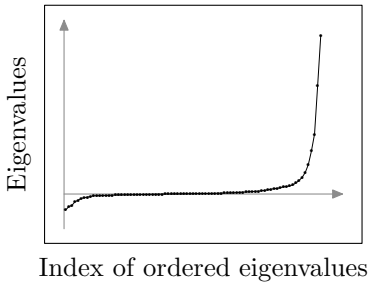


Figure 3.6 shows the striking improvement of the distance matrix after denoising. On the left hand side the ideal distance matrix is depicted, consisting solely of 0's (black) and 1's (white), reflecting the true cluster membership. In the middle and on the right the original and the denoised distance matrix are shown, respectively. Denoising visibly accentuates the cluster structure in the pairwise data.

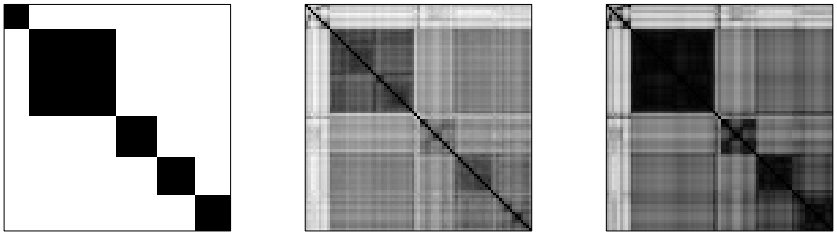


Figure 3.6. Dissimilarity matrix: On the left the ideal distance matrix reflects the true cluster structure. In the middle and on the right: distance matrix before and after denoising. Dark values represent small dissimilarities, light values large dissimilarities.

Improvement obtained by denoising

Since we dispose of the true labels, we can quantitatively assess the improvement by denoising. We performed usual k-means clustering, followed by a majority voting to match cluster labeling. For the denoised data we obtained 3 misclassifications ($3.61\,\%$) whereas we got 17 ($20.48\,\%$) for the original data. This simple experiment corroborates the usefulness of our embedding and denoising strategy for pairwise data.

In order to fulfill the spirit of the theory of constant shift embedding, the cost function of the data-mining algorithm subsequent to the embedding needs to be shift invariant. We may, however, go a step further and apply algorithms for which this condition does not hold. In doing so, however, we give up the mathematical traceability of the error.

Comparison to previous results

To illustrate that denoised pairwise data can act as standalone quality data

independent of the framework of algorithms based on shift invariant cost functions, and in order to compare to the results obtained in (Tsuda et al., 2002), a linear SVM is trained on 25 % of the total data to mutually classify the genera-pairs: 3 – 4, 3 – 5 and 4 – 5. Genera 1 and 2 separate errorless and have therefore been omitted. Model selection over the regularization parameter C has been performed by choosing the optimal value out of 10 equally spaced values from $[10^{-4}, 10^{2}]$. The results have been averaged by a 1000-fold sampling (cf. Table 3.1). The best values are emphazised.

Genera	FK	MCK2	Undenoised	Denoised
3 – 4	10.4	8.48	*5.06*	5.43
3 – 5	10.9	5.71	5.72	*3.83*
4 – 5	23.1	11.6	7.55	*3.17*

Table 3.1. Comparison of mean test-error of supervised classification by linear SVM of genera with training sample 25 % of the total sample. The results for MCK2 (Marginalized Count Kernel) and FK (Fisher Kernel) is obtained by kernel Fisher discriminant analysis which compares favorably to the SVM in several benchmarks (Tsuda et al., 2002).

Discussion

For the classification of genera 3 – 5 and 4 – 5 we obtain a substantial improvement by denoising. Interestingly this is not the case for genera 3 – 4 which may be due to the elimination of discriminative features by the denoising procedure. The error still is significantly smaller than the error obtained by MCK2 and FK, which is in agreement with the superiority of a structure preserving embedding of Smith-Waterman scores even when left undenoised: FK and MCK are kernels derived from a generative model, whereas the alignment scores are obtained from a matching algorithm specifically tuned for protein sequences, reflecting much better the underlying structure of protein data.

CLUSTERING OF PROTEIN SEQUENCES.

The globin data set

In this experiment with globin sequences, we present a worked-through example of combining constant shift embedding, low-dimensional approximations, model selection and clustering in the embedding space. From the SWISS-PROT and TrEMBL databases (Boeckmann et al., 2003) we extracted all approximative 1200 sequences annotated as “globins” or as “globin-like”. The heuristic FASTA scoring method (Pearson and Lipman, 1988) was used for computing pairwise alignment scores, which in turn were length-corrected, a Bayesian approach for correcting local alignments, following Durbin et al. (1998), and normalized to the length of the alignment. From the pair-scores s_{ij}, we derived dissimilarities by setting $d_{ij} = s_{ii} + s_{jj} - 2s_{ij}$. Note that other transformations (e.g. of the form $d_{ij} = \exp(-s_{ij})$) may be applied as

Metric violations

well. Our experimental results, however, favor the first choice. The eigenvalue spectrum of the centered matrix C^c shows some highly negative entries, indicating that the dissimilarities do not derive from squared Euclidean distances. By way of the constant shift embedding procedure, however, the sequences are represented as points in a vector space without distorting the grouping solution.

Given these vectors, we are left with two problems:

1. choosing an appropriate denoising mechanism and
2. minimizing the k-means cost function for different values of k and selecting the "optimal" number of clusters k. In the following we present details for both the model selection procedure and the final clustering results.

DENOISING. The left panel in Figure 3.7 shows the 25 leading eigenvalues of the centered matrix C^c. The eigenvalue curve suggests that there are only very few dominating directions in the embedding space. We thus decided to discard all but the first ten leading eigenvectors. Since in this control experiment we have access to the ground-truth labels, we are able to test this hypothesis about "signal" and "noise". The plotted denoised and original distance matrices in Figure 3.8 indicate that the space spanned by the first ten eigenvectors indeed accentuates the main structure of the protein (sub-)families.

OPTIMIZATION AND MODEL SELECTION. For minimizing the k-means functional in the embedding space a deterministic annealing method was applied. Concerning the selection of the "correct" number of clusters, we used the concept of *cluster stability* which has been introduced in Dudoit and Fridlyand (2002) and refined in Lange et al. (2003). The main idea is to draw resamples from the data set and then to compare the inferred data-partitions across these resamples. The variations of the partitions are transformed into an instability index, which is normalized such that a *random* procedure yields instability 1, and a perfect correspondence between solutions yields instability 0. The right panel in Figure 3.7 depicts the estimated instability for different numbers of clusters. The bars show the standard deviations estimated in the resampling procedure. The most stable solution partitions the data into three clusters, and two another distinct local minima occur for $k = 5$ and $k = 9$.

CLUSTERING RESULTS. For the solutions with $k = 3$ and $k = 9$, we have plotted the corresponding distance matrices in Figure 3.8. On the left panels we have also depicted the "true" group membership of the proteins, as annotated in the SWISS-PROT database. The groups are: *Plant* (plant globins), *HB-*α (hemoglobin-α), *MYG* (myoglobin), *HB-*β (hemoglobin-β) and *GLB*

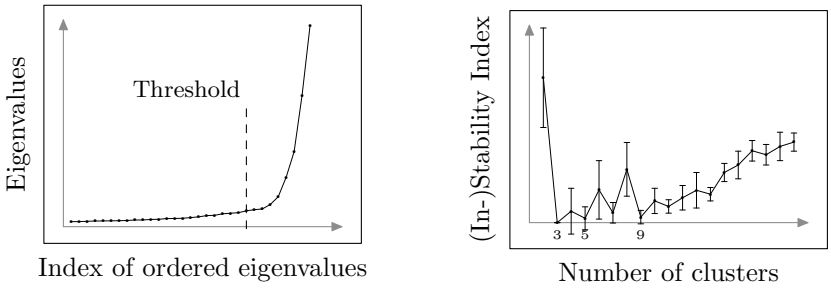


Figure 3.7. Clustering of globin proteins. Left: leading eigenvalues of the centered matrix C^c. Right: instability of the partition vs. number of clusters k.

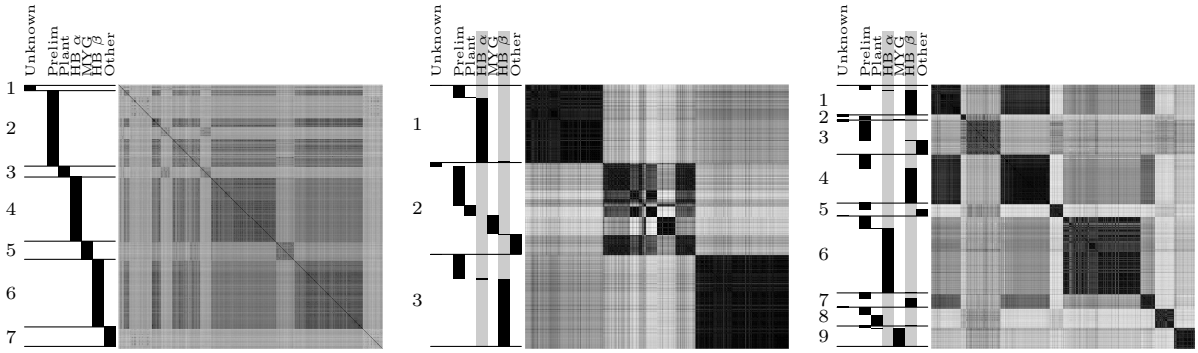


Figure 3.8. Dissimilarity matrices for the embedded clustering problems, permuted with respect to cluster labels. Left: original dissimilarities (without denoising, plotted in the permutation of the true labels). Middle: $k = 3$ and left: $k = 9$. Dark values represent small dissimilarities, light values large dissimilarities.

(other globins, e.g. globin I-IV or insect globins). The column marked *Prelim* indicates "preliminary" sequences from the TrEMBL database with missing or uncertain annotations. The automatically found solutions divide the sequences into biologically meaningful groups: the 3-cluster solution separates both hemoglobin-α and hemoglobin-β from the rest. The 9-cluster solution defines a refinement of these groups, in the following sense: the β-hemoglobins are split into two subgroups (cluster no. 1 and no. 4), both the myoglobins and the plant globins are now contained in individual clusters, and the other globins are also separated into two sub-clusters (the first of which now mainly contains insect globins). It is interesting to notice that successively increasing the number of clusters automatically leads to a natural hierarchical representation of the group structure, which has *not* been introduced by the algorithm as

a modeling bias.

COMPARISON WITH MDS. From a theoretical viewpoint, the constant shift embedding principle has one major advantage over classical MDS embedding: for shift-invariant clustering cost functions, CSE yields cluster preserving embeddings in $n-1$ dimensional vector spaces, while for MDS no such guarantees are available. Taking a practical perspective, however, one might be interested in differences between CSE and MDS in *low dimensional* embedding spaces. Designing experiments which allow "fair" comparisons of this kind, however, is difficult, since both the CSE method (different reduction methods like PCA, LLE, etc.) and MDS (different cost functions, choice of weights, etc.) can be varied in several ways. Nevertheless, we conclude this section with a comparison of k-means clustering results in two dimensions, once directly embedded using MDS (stress cost function, relative weights, see Equation 2.17), and the second time embedded with CSE and PCA. In the upper left panel of Figure 3.9 and Figure 3.10 the two-dimensional MDS embedding of the above data set is depicted. The different point symbols refer to the SWISS-PROT labels. Given these two dimensional data set, we then minimized the k-means clustering cost function with $k = 3$, leading to the labels shown in the lower left panel. It is interesting to note that the typical "ring artifacts" of MDS embedding produce elongated structures which cannot be recovered by the compactness based k-means clustering criterion. In the case of CSE with succeeding PCA embedding, the situation looks very different: the embedded data clearly show three relatively compact groups (upper right panel): one corresponds to hemoglobin-α proteins, another to hemoglobin-β proteins, the third one is a mixture of the other protein families. These three compact groups are perfectly recovered in the 3-means solution (lower right panel).

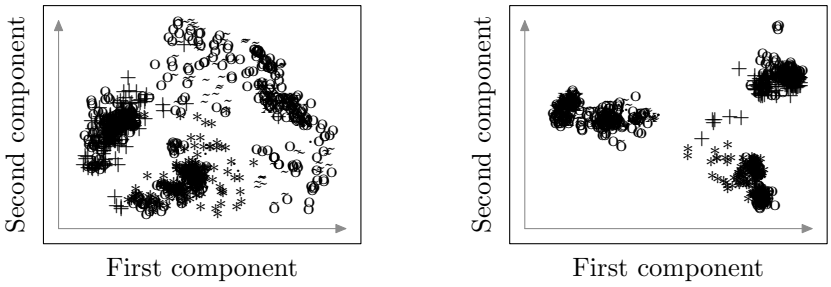


Figure 3.9. Embedded proteins with original SWISS-PROT labels. Left: MDS (Stress, local weights), right: CSE with PCA embedding.

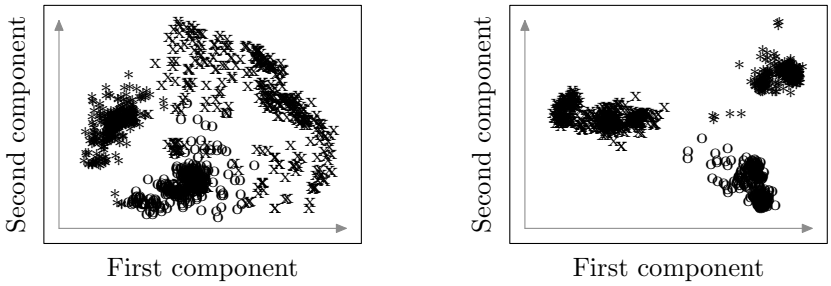


Figure 3.10. Embedded proteins with inferred k-means labels, $k = 3$. Left: MDS (Stress, local weights), right: CSE with PCA embedding.

CLUSTERING OF PRODOM SEQUENCES.

The ProDom data set

The analysis described in this section aims at finding a partition of domain sequences from the ProDom database (Corpet et al., 2000) that is meaningful with respect to *structural similarity*. In order to measure the quality of the grouping solution, we use the computed solution in a predictive way to assign group labels to SCOP sequences, which have been labeled by experts according to their structure (Murzin et al., 1995). The predicted labels are then compared with the "true" SCOP labels.

For demonstration purposes, we select the following subset of sequences from `prodom2001.2.srs`: among all sequences we choose those which are highly similar to at least one sequence contained in the first four folds of the SCOP database.[1] Between these sequences, we compute pairwise (length-corrected and standardized) Smith-Waterman alignment scores, summarized in the similarity matrix $S = (s_{ij})$. These similarities are transformed into dissimilarities by setting $d_{ij} = s_{ii} + s_{jj} - 2s_{ij}$. The centralized covariance matrix $C^c = -\frac{1}{2}D^c$ possesses some highly negative eigenvalues, indicating that metric properties are violated. Applying the constant shift embedding method, a valid positive semi-definite kernel is derived, with an eigenvalue spectrum that shows only a few dominating components over a broad "noise"-spectrum (see Figure 3.11). Extracting the first 16 leading principal components[2] leads to a vector representation of the sequences as points in $\mathbb{R}^{16}$. These points are then clustered by minimizing the k-means cost function within a deterministic annealing framework. The model order was selected by applying a re-sampling based *stability* analysis, which has been demonstrated to be a suitable

Similarity matrix

CSE

[1] "Highly similar" here means that the highest alignment score exceeds a predefined threshold. The result is a subset of roughly 2700 ProDom domain sequences.

[2] Subsampling techniques or deflation can be used to reduce computational load for large-scale problems. We only used a subset of 800 randomly chosen proteins for estimating the 16 leading eigenvectors.

model order selection criterion for unsupervised grouping problems in Roth et al. (2002).

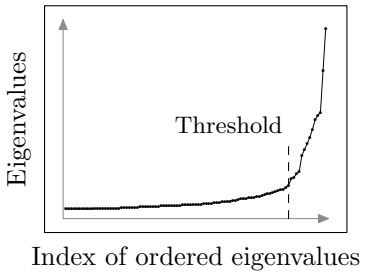


Figure 3.11. (Partial) eigenvalue spectrum of the shifted score matrix. Only the 16 leading eigenvalues have been retained, thus conserving the main structure while eliminating the bulk of noise. This corresponds to a denoising of the pairwise data.

Results

In order to measure the quality of the grouping solution, all 1158 SCOP sequences from the first four folds are embedded into the 16-dimensional space. The predicted group structure on this test set is then compared with the true SCOP fold-labels. Figure 3.12 shows both the predicted group membership of these sequences and their true SCOP fold-label in the form of a bar diagram: the sequences are ordered by increasing group label (the lower horizontal bar), and compared with the true fold classification (upper bar). In order to quantify the results, the inferred clusters are re-labeled ("re-colored") according to the maximum number of correctly identifiable fold-labels. This procedure allows us to correctly identify the fold label of roughly 94 % of the SCOP sequences.

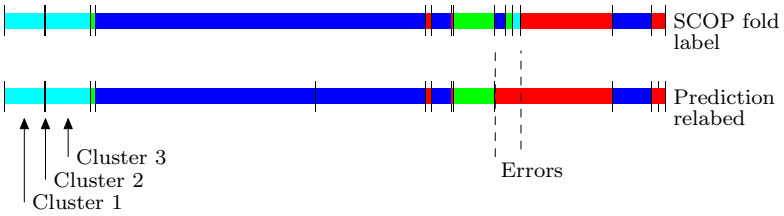


Figure 3.12. Visualization of cluster membership of the chosen 1158 SCOP sequences contained in folds 1 – 4.

Despite this surprisingly high percentage, it is necessary to deeper analyze the biological relevance of the inferred grouping solution. In order to check to what extend the above "over-all" result is influenced by artifacts due to highly related (or even almost identical) SCOP sequences, we repeated the analysis based on the subset of 128 SCOP sequences with less than 50 % sequence identity (PDB-50). Predicting the group membership of these 128 sequences and using the same re-labeling approach, we can correctly identify 86 % of the fold-labels (Figure 3.13). This result demonstrates that we have not only found trivial groups of almost identical proteins, but that we have indeed extracted

relevant structural information.

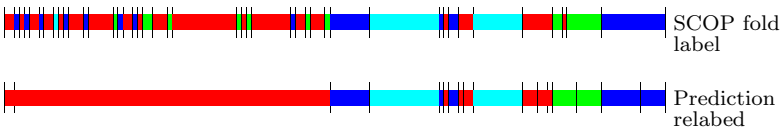


Figure 3.13. Visualization of cluster membership of the 128 PDB-50 sequences.

§. 3.7. DISCUSSION.

Summary

We have introduced an optimal embedding procedure for pairwise clustering by means of constant shift embedding (CSE). For the class of shift-invariant clustering methods, it optimizes a fundamentally different criterion compared to classical embedding approaches based on MDS. The most prominent property of CSE is the complete preservation of the group structure in the embedding space. For MDS methods, on the other hand, such a preservation can only be guaranteed in the special (and rather uninteresting) case of zero distortions ("stress") of the pairwise dissimilarities. For non-zero distortions, to our knowledge no bounds on *structural* distortions are known.

Outstanding property of CSE

For shift-invariant cost functions we can always embed non-metric pairwise data in a Euclidean space and obtain a statistically equivalent problem formulation. This represents a unification of the vectorial and pairwise data representation, not on the level of geometry, which incurs distortions by the embedding procedure, but on the level of structure itself, which is preserved.

Unification of vectorial and pairwise representation

The possibility of restating a pairwise grouping problem in a vector space has important theoretical consequences. For instance, we are able to statistically describe the clusters by defining cluster prototypes in the embedding space, and by measuring the variance in each of the clusters. These Prototypes, in turn, define a generic rule for extending the grouping solution to a predictive discrimination rule for estimating the cluster membership of new objects. Concerning the problem of finding efficient optimization algorithms for minimizing clustering cost functions, the shown equivalence of pairwise clustering and k-means shed light on the probabilistic structure of the solution space: the problem of minimizing H^{pc} belongs to the class of combinatorial optimization problems for which the classical *mean-field approximation* becomes *exact*.

Theoretical consequences

There are also a couple of practical consequences of CSE: a common vector

Practical consequences

space representation renders the data accessible to standard dimensionality and noise reduction methods which lack a clear meaning for pairwise data. Such preprocessing methods, however, have to be chosen carefully, depending on the requirements and/or the prior knowledge available for each special application. For the task of clustering the globin proteins, it turned out that a classical PCA denoising worked surprisingly well. A comparison with the known family structure of these proteins revealed that the low-dimensional PCA embedding space accentuated the relevant structure while suppressing the alignment noise. It should be noticed, however, that in general unsupervised situations, such high-level domain knowledge may be hardly available. In these situations, one should rely on general statistical descriptors, such as the form of the eigenvalue spectrum of the covariance matrix.

Despite the fact that "wrong" preprocessing methods clearly have the potential to distort the cluster structure (which we naturally want to preserve), the CSE framework at least tells us that these distortions are not caused by the general restrictions of a vector space. We know that there always exists a Euclidean space which contains the optimal structure preserving vectors, which means that there might be hope to find more suitable low-dimensional approximations.

§. 3.8.

CONCLUSION.

For several major applications of data analysis, objects are often not represented as feature vectors in a vector space, but rather by a matrix gathering pairwise proximities. Such pairwise data often violates metricity and, therefore, cannot be naturally embedded in a vector space. Concerning the problem of unsupervised structure detection or *clustering*, in this chapter a new embedding method for pairwise data into Euclidean vector spaces was introduced. We have shown that all clustering methods, which are invariant under additive shifts of the pairwise proximities, can be reformulated as grouping problems in Euclidian spaces. The most prominent property of this *constant shift embedding* framework is the complete *preservation of the cluster structure* in the embedding space. Restating pairwise clustering problems in vector spaces has several important consequences, such as the statistical description of the clusters by way of *cluster prototypes*, the generic extension of the grouping procedures to a discriminative *prediction rule,* and the applicability of standard *preprocessing methods* like denoising or dimensionality reduction.

4. FEATURE DISCOVERY

In this chapter we will study the issue of the signification and interpretation of metric violations. In literature, metric violations are usually discarded as mathematical artifact of noise, and solutions to elude the mathematical annoyance of negative eigenvalues are ready at hand. Only a few authors hint at the possibility of inherent non-metricity and the danger of a forceful metrization of the data. The central and so far unanswered question is therefore: *Does the negative part of the spectrum of a similarity matrix code anything useful other than noise?* The answer to this question was given in Laub and Müller (2004) and will be presented here. We will systematically study the occurence of negative spectra. Models are developed to explain these spectra and simple projection techniques are presented to visualize the information coded by the metric violations. Several applications will illustrate the theory.

§. 4.1.

INTRODUCTION.

Distortion of embeddings

From a geometric point of view, non-metric pairwise data can not be embedded distortionless into a Euclidean space. So, in general, embedding into a Euclidean space (and often subsequent dimension reduction) amounts to distorting pairwise data to enforce Euclideanness. This procedure is exemplified by MDS.

Possible loss of information

Little is known about the information loss incurred by enforcing metricity, when non-metric data is forcefully embedded into a vector space on the assumption that non-metricity be a mere artifact of noise. This assumption certainly holds for many cases, especially when the pairwise comparison is the output of some algorithm tuned to be metric but relying on some random initialization. It does not hold for pairwise data which is inherently non-metric, e.g. for human similarity judgments, where geometrical (metric) and categorial thinking (possibly non-metric) is superposed.

Traditional approaches for pseudo-covariance matrices

Technically, non-metricity translates into indefinite covariance matrices (Theorem 2.4.2), a fact, which imposes severe constraints on the data analysis procedures. Typical approaches to tackle these problems involve omitting alto-

gether the negative eigenvalues like in classical scaling or shifting the dissimilarities so as to enforce squared Euclideanness as in CSE. An important point is to notice that these issues will crucially depend on the magnitude of the negative eigenvalues. If the negative eigenvalues are small in magnitude, they are commonly associated to noise and leaving them away will at best improve the result, at worse leave it unchanged. If they are large, some argue that classical scaling is still an appropriate dimension reduction technique (Cox and Cox, 2001).

Possible loss for CSE?

In the previous chapter we have seen that non-metric pairwise data may be embedded without loss for subsequent clustering if the cost function is shift invariant. However, in practical applications, the need for dimension reduction to speed up optimization and robustify solutions, effectively results in retaining only the leading eigendirections and cutting off large parts of the spectrum. For other cases than noise corrupted non-metric pairwise data it is an open question whether the removal of negative eigenvalues leads to an information loss.

Inherent non-metricity

Several authors (Jacobs et al., 2000, Torgerson, 1958) notice that it may not always be of advantage to embed the data, especially if it comes at the price of high distortion. Violation of triangle equality or symmetry as property of the distance measure should not be regarded at as noise but as intrinsic feature of the data set. Some problems (e.g. where transitivity is violated) might get an erroneous treatment when forcibly embedded in an Euclidean space.

Non-metricity in classification

In Pękalska et al. (2001) we read: *There is still an open question about the consequences on classification tasks of transforming the problem into a Euclidean space, either by neglecting the negative eigenvalues or by directly enlarging D by a constant*. They show that the retention of the negative eigendirection can be beneficial to the classification result and is thus a sensible choice in machine learning (for a similar finding, see Graepel et al. (1999)). However, their positive results seems due to a particular instance of denoising and the question about the signification of metric violations and the thus induced "negative variance" remains unsolved. Improvement of a classification rate suggests that it is other than noise. It is still utterly unclear, whether we should look at non-metricity as a mere mathematical artifact of no further importance except for algorithmic reasons or whether it reveals us *new* insight into the structure of the data.

Non-metricity in intelligent data analysis

We will not be interested in a classification task but merely in the explanation of variance. In a sense, this means that we want to know whether *at all* there is "something interesting" in the negative eigenvalues else than random noise.

We adopt the point of view that

1. embedding pairwise data in a Euclidian vector space of low dimension for visualization it is a good idea in a first approach to understand the data and that
2. variance can capture problem specific information.

Visualization

The fact that we are interested in Euclidean embeddings allows us, via Theorem 2.4.2, to consider the eigenspectra of pseudo-covariance matrices to measure the metric violations of the underlying dissimilarity matrix. For more generic measures of metric violations and more general embeddings, see Appendix A.

It is important that we again stress our interest in *visualization*. This chapter is conceptual in nature and relies upon visualization as the simplest way to gaining insight into complex pairwise data (Everitt and Rabe-Hesketh, 1997).

This study comprises a general look on different spectra, on several models to explain them, illustrated by simple and intuitive examples. We will limit our illustrations to embeddings in two dimensions, which allows for visual appreciation, an unquestionable advantage in unsupervised learning. We will show that the negative eigenvalues can indeed correspond to variance non negligible to the problem, in the sense that the latter can be related to "relevant" features.

§. 4.2.
UNDERSTANDING NEGATIVE EIGENVALUES.

We will start this study by some general considerations on the nature of the spectrum of pseudo-covariance matrices associated to dissimilarity matrices violating metric requirements. We will discuss variance, information and the loss thereof.

SHAPE OF THE SPECTRUM.

Sorting the spectrum

The spectrum of a matrix is the set of its eigenvalues. For real symmetric matrices it can be shown to be real (Lüthkepohl, 1996). To give sense to notions like shape we need to introduce an ordering. Assuming that the numbers are real, we may simply order them in increasing values.

Trivial and non-trivial spectra

The spectrum can be of any shape, however, we will mainly be interested in the following two (common) cases (see Figure 4.1):

1. Flat negative spectrum: the negative eigenvalues do not differ much in magnitude from the bulk of eigenvalues.
2. Strongly decreasing negative spectrum in the last few eigenvalues.

Distribution ⇒ spectrum

The data distribution always gives us algorithmically the spectrum. The converse is not true unfortunately.

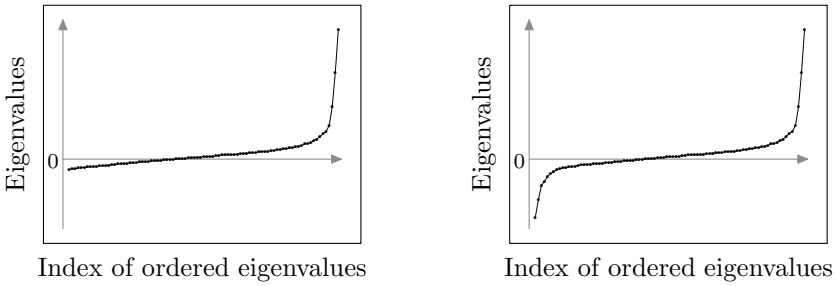


Figure 4.1. Spectrum with trivial (left) and non-trivial (right) negative eigenvalues. The trivial negative spectrum is characterized by a slowly falling negative tail while the non-trivial negative spectrum exhibits a strongly falling negative tail.

Spectrum $\not\Rightarrow$ distribution

A rule of thumb is that whatever typical data distribution yields a given spectrum, there is always a "pathological" distribution which yields a similar spectrum. "Pathological" in this context just means that the spectrum gives us no valuable hint at the distribution expected by the typical case, this being a Gaussian distribution or some regular distribution with finite support.

This fact is exemplified for simple statistics as means and variance. Duda et al. (2001) gives an example of four different distributions with identical mean and variance (second order statistics).

One might ask what then justifies the use of "pathological". As a matter of fact, second order statistics—and in the same vein the distribution of the eigenvalues—do give relevant information when one assumes some Gaussian process produced the data, which in natural processes certainly is reasonable.

Typcial distributions

Typically, for a flat spectrum, we will expect the data to be isotropically distributed in space. (Note that notions like isotropic distribution only make sense once an ordering fixed for the eigenvalues and hence for the corresponding eigenvectors.) On the other hand, the directions spanned by the eigenvectors associated to large negative eigenvalues in magnitude somehow defy this interpretation and we expect to find there non negligible variance.

Pathological cases

Pathological cases in this context may be, e.g. spectra whose shape is an artifact of small sample size, missing values, outlier or, in general, very exotic distributions.

VARIANCE EXPLAINED.

PCA

Recovering vectors from a squared Euclidean D according to the algorithm (Cox and Cox, 2001) given on page 24 corresponds to a principal component analysis (PCA). PCA has a nice interpretation as variance explained: choosing

the embedding subspace associated to the eigenvalues $\lambda_1, \lambda_2, \dots \lambda_k$, where $1 \leqslant i < k \leqslant n$, amounts to retaining the fraction

$$\frac{\lambda_i + \cdots \lambda_k}{\sum_{i=1}^{n} \lambda_i} \tag{4.1}$$

of the total variance.

When the rigorous mathematical framework of PCA is not given—as it is the case in classical scaling—, we can still measure the adequacy of a representation by measures like Equation 4.1. In the case of non-metric D's, PCA is not well defined because of negative eigenvalues. Instead of the measure given in Equation 4.1, one typically uses (Everitt and Rabe-Hesketh, 1997)

Measures when there are negative eigenvalues

$$\frac{|\lambda_i| + \cdots |\lambda_k|}{\sum_{i=1}^{n} |\lambda_i|}, \tag{4.2}$$

or

$$\frac{\lambda_i^2 + \cdots \lambda_k^2}{\sum_{i=1}^{n} \lambda_i^2}.$$

Another choice involves counting only the positive eigenvalues in the denominator (Cox and Cox, 2001). Note that Equation 4.2 corresponds to a formulation of the variance in a pseudo-Euclidean space.

On information, relevance and loss.

Ill-definedness

Data analysis aims at extracting information from data. Given a certain task, a certain scientific question, one attempts to extract information *relevant* to the specific problem. This is an intrinsically ill-defined problem, since there is no rigorous definition of information, let alone of relevance.

As stated in the introduction, we adopt here a very modest and down to earth approach: we look at variance and try to understand it.

We *look* at variance. So to tackle the problem of relevance, we would rather address the issue of low dimensional visualization. We define information simply by the explained variance and leave notions like relevance to the intuition of the reader by the question: is a given explained variance interesting for this problem?

This evasiveness on the definition of relevance does not mean that unsupervised learning aims at recovering features known beforehand. It rather stresses the data explorative aspect of this study, not concealing its epistemological limits.

In the realm of unsupervised learning there is a trade off to find between pure explorative research and automation. Disposing intrinsically of no ground truth, one must be aware that no algorithm will produce such.

Trivial spectra

For spectra with a flat negative part, the cut off on the eigenvalues required for embedding may not result in a loss of relevant information, since a flat spectrum translates isotropic distributions commonly associated with noise. This is not the case for spectra with a non-trivial negative part which could contain essential information (variance).

Non-trivial spectra

On the other hand, for spectra with a steeply falling tail, this interpretation does no longer hold and usual dimensionality reduction procedures and subspace methods retaining only leading eigenvalues are prone to a loss of information. In general, all procedures relying on distance based cost functions may suffer the same drawbacks since they will only take into account large variance. In the following section we will discuss non-trivial spectra in more detail.

§. 4.3.

CODING INFORMATION IN THE NEGATIVE PART OF SPECTRUM.

We will first illustrate the coding ability of negative eigenvalues by a from-scratch construction of a similarity matrix. We will speak in these constructions about clusters, as prominent representants explaining variance by a clear separation.

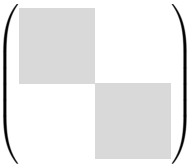
SIMPLE MODEL I.

Consider the following abstract setting: n objects, labeled $1, 2, \dots n$, presenting two salient features. Suppose that they cluster into $\{1, \dots \frac{n}{2}\}$ and $\{\frac{n}{2} + 1, \dots n\}$ according to the first feature, and into $\{1, 3, 5, \dots n-1\}$ and $\{2, 4, 6, \dots n\}$ according to the second.

S_1 and S_2

Let S_1 and S_2 be the similarity matrices corresponding to feature 1 and 2 respectively. S_1 has a block structure, S_2 a line structure (gray areas correspond to high similarities):

$$S_1 = \left(\right) \quad \text{and} \quad S_2 = \left(\right) .$$



They both will obviously have a clear pronounced structure in the positive eigenvalues corresponding to the two clusters defined.

Combining similarity matrices is in no way trivial. Tsuda *et al.* have studied different mix kernels ("similarity matrices") in a supervised learning task (Tsuda et al., 2004). In an unsupervised setting we generally have no a priori idea of how to mix different similarity matrices.

A starting idea could be to pose:

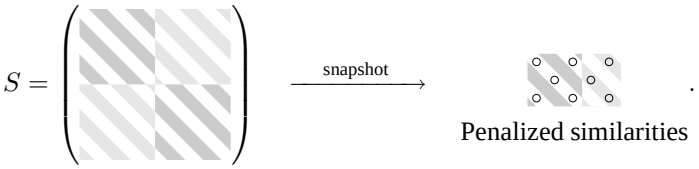
$$S = aS_1 + bS_2.$$

The most straight forward way would certainly be $a = 1$ and $b = 1$ in which case we expect to recover four clusters when projecting onto the first few leading eigenvalues, namely: $\{1, 3, \dots \frac{n}{2}-1\}$, $\{2, 4, \dots \frac{n}{2}\}$, $\{\frac{n}{2}+1, \frac{n}{2}+3, \dots n-1\}$ and $\{\frac{n}{2}+2, \frac{n}{2}+4, \dots n\}$.

The information extracted from this four-cluster solution however is not satisfactory given the initial setting of the problem, since one may not be able to relate the four clusters to the two coded features, in particular if there is no clear hierarchical structure in the solution.

Penalization

Interestingly, a recurrent mixing seems to be given by the case $a = 1$ and $b = -1$ yielding—save exception—a non-trivial negative spectrum (Figure 4.1, right). This corresponds to a "penalization" by S_2 of S_1.



The penalized similarities of S are the s_{ij} for which $(S_2)_{ij}$ is large. If $(S_2)_{ij}$ is small or even zero, $s_{ij} \sim (S_1)_{ij}$, and the similarities remain unpenalized.

From S_1 and S_2 we obtain dissimilarities D_1 and D_2 via some decreasing function, from which the corresponding covariances C_1 and C_2 are computed.

Since C_1 is positive semi-definite and $-C_2$ is negative semi-definite, C is indefinite by the following theorem:

THEOREM 4.3.1 (Weyl). *Let $A, B \in M_n$ be Hermitian and let the eigenvalues $\lambda_i(A)$, $\lambda_i(B)$ and $\lambda(A+B)$ be arranged in decreasing order. For each $k = 1, 2, \dots n$ we have*

$$\lambda_k(A) + \lambda_n(B) \leqslant \lambda_k(A+B) \leqslant \lambda_k(A) + \lambda_1(B).$$

Proof. Horn and Johnson (1995) □

We pose $A = C_1$ and $B = -C_2$ and make the reasonable assumption that $\lambda_n(C_1) = \lambda_1(-C_2) = 0$. From the above follows that $\lambda_n(A+B) = \lambda_n(C) \leqslant$

0. Furthermore, excluding the unlikely case $\lambda_i = 0$ for all $i = 1, 2, \ldots n$, there exists k such that $\lambda_k(C) < 0$.

Note that this does not prove that the spectrum actually has a non-trivial negative spectrum. We can only assess that it has negative eigenvalues.

Penalized similarities may form a structure on their own which by this construction is encoded by the metric violations in the negative eigenvalues of C.

Discussion

The construction of $S = S_1 - S_2$ may seem somewhat arbitrary, even defying intuition. However, our concern in this section is to give an idea how negative spectra come about, regardless of interpretation. Note that the construction of S corresponds to the difference of squared Euclidean distances as presented in the Section 2.5 on pseudo-Euclidean spaces.

In order to foster intuition on negative spectra, the symbolic model of a difference of two similarities may be understood as a sum of a similarity and a dissimilarity. The information contained in the similarity will be encoded by the positive eigenvalues whereas the information contained in the dissimilarity will be encoded in the negative ones.

Conversely, the decomposition of the distances in a pseudo-Euclidean space (or some generic, non-metric dissimilarity) into a difference of squared Euclidean distances may be looked at as a sum of a dissimilarity and a similarity, the roles of the positive and negative eigenvalues now being flipped. The information contained in the dissimilarity is encoded in the positive eigenvalues and the one contained in the similarity in the negative.

The question on whether the sum of similarities and dissimilarities makes any sense for defining a similarity (or a dissimilarity) is not well-defined, as the notion of similarity and dissimilarity both lack a clear cut definition. One rather has to start from the fact, that non-metric dissimilarities *do* exist. The penalization model as presented above is one first explanation.

EXAMPLE I.

Let $n = 8$ and the object grouped according to the scheme described, i.e. the 8 object cluster like $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$ in the one feature, and like $\{1, 3, 5, 7\}$ and $\{2, 4, 6, 8\}$ in a second.

The similarities might look like follows (the matrices were obtained by an artificial from scratch construction):

$$S_1 = \begin{pmatrix} 2.30 & 1.24 & 1.28 & 1.58 & 0 & 0 & 0 & 0 \\ 1.24 & 2.54 & 1.50 & 1.79 & 0 & 0 & 0 & 0 \\ 1.28 & 1.50 & 2.85 & 1.70 & 0 & 0 & 0 & 0 \\ 1.58 & 1.79 & 1.70 & 2.64 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.44 & 1.88 & 1.65 & 1.62 \\ 0 & 0 & 0 & 0 & 1.88 & 2.52 & 1.11 & 1.58 \\ 0 & 0 & 0 & 0 & 1.65 & 1.11 & 2.68 & 1.54 \\ 0 & 0 & 0 & 0 & 1.62 & 1.58 & 1.54 & 2.42 \end{pmatrix},$$

$$S_2 = \begin{pmatrix} 2.04 & 0 & 1.30 & 0 & 1.52 & 0 & 1.74 & 0 \\ 0 & 2.11 & 0 & 1.82 & 0 & 1.29 & 0 & 1.45 \\ 1.30 & 0 & 2.59 & 0 & 1.50 & 0 & 1.52 & 0 \\ 0 & 1.82 & 0 & 2.10 & 0 & 1.28 & 0 & 1.12 \\ 1.52 & 0 & 1.50 & 0 & 2.26 & 0 & 1.83 & 0 \\ 0 & 1.29 & 0 & 1.28 & 0 & 2.29 & 0 & 1.48 \\ 1.74 & 0 & 1.52 & 0 & 1.83 & 0 & 2.13 & 0 \\ 0 & 1.45 & 0 & 1.12 & 0 & 1.48 & 0 & 2.81 \end{pmatrix},$$

and

$$S = \begin{pmatrix} 0.26 & 1.24 & -0.02 & 1.58 & -1.52 & 0 & -1.74 & 0 \\ 1.24 & 0.43 & 1.50 & -0.03 & 0 & -1.29 & 0 & -1.45 \\ -0.02 & 1.50 & 0.26 & 1.70 & -1.50 & 0 & -1.52 & 0 \\ 1.58 & -0.03 & 1.70 & 0.50 & 0 & -1.28 & 0 & -1.12 \\ -1.52 & 0 & -1.50 & 0 & 0.18 & 1.88 & -0.18 & 1.62 \\ 0 & -1.29 & 0 & -1.28 & 1.88 & 0.23 & 1.11 & 0.10 \\ -1.74 & 0 & -1.52 & 0 & -0.18 & 1.11 & 0.55 & 1.54 \\ 0 & -1.45 & 0 & -1.12 & 1.62 & 0.10 & 1.54 & -0.39 \end{pmatrix}.$$

From this symmetric S we compute D via $d_{ij} = s_{ii} + s_{jj} - 2s_{ij}$, and C via $C = -\frac{1}{2}D^c$. The respective spectra are given in Figure 4.2.

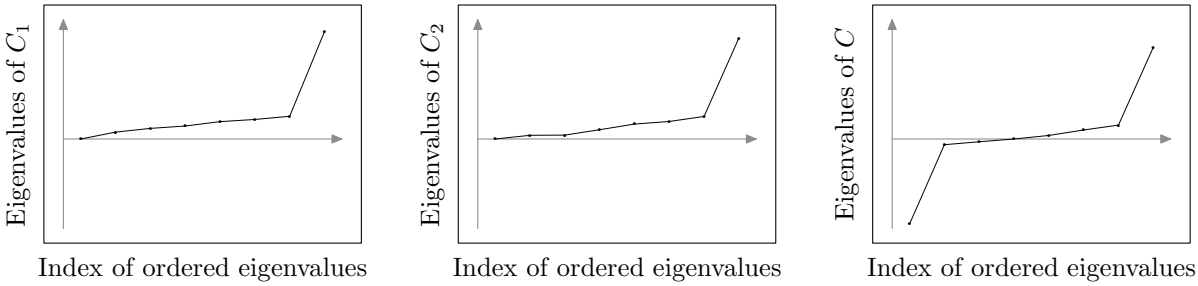


Figure 4.2. Spectrum of C_1 (left), C_2 (middle) and C (right). The spectrum of C is non-trivial.

SIMPLE MODEL II.

The second model presented below treats a construction of similarity encountered in many fields of data analysis. A simple approach is given by posing:

$$s_{ij} = \frac{(S_1)_{ij}}{(S_2)_{ij}},$$

with the assumption that $(S_2)_{ij} \neq 0$ for all $i, j = 1, 2, \ldots n$. Such similarity scores occur in various image matching algorithms or in text mining via the *min-max* similarity, see e.g. Banerjee and Ghosh (2002) or Dagan et al. (1995).

This second model acts similarly as the previous one, i.e. by penalization. The same discussion as the one following Model I holds for Model II. The quotient of similarities usually is understood as some normalization procedure. However, considering that the inverse of a similarity may be looked at as a dissimilarity we now face the interpretation of a similarity as a product of similarity and dissimilarity. As before, we claim that the question about its semantic is ill-defined.

EXAMPLE II.

The same setting as in example I is taken. Consider the following similarity matrices:

$$S_1 = \begin{pmatrix} 2.45 & 1.38 & 1.48 & 1.70 & 0.18 & 0.18 & 0.13 & 0.15 \\ 1.43 & 2.66 & 1.60 & 1.99 & 0.11 & 0.14 & 0.14 & 0.15 \\ 1.42 & 1.62 & 2.96 & 1.86 & 0.17 & 0.15 & 0.12 & 0.10 \\ 1.74 & 1.94 & 1.86 & 2.76 & 0.18 & 0.12 & 0.13 & 0.14 \\ 0.16 & 0.15 & 0.14 & 0.13 & 2.54 & 2.07 & 1.76 & 1.75 \\ 0.18 & 0.14 & 0.17 & 0.19 & 2.06 & 2.68 & 1.24 & 1.74 \\ 0.15 & 0.12 & 0.12 & 0.19 & 1.78 & 1.23 & 2.87 & 1.68 \\ 0.13 & 0.19 & 0.12 & 0.12 & 1.80 & 1.71 & 1.74 & 2.55 \end{pmatrix},$$

$$S_2 = \begin{pmatrix} 2.17 & 0.13 & 1.43 & 0.14 & 1.67 & 0.20 & 1.93 & 0.13 \\ 0.10 & 2.24 & 0.18 & 1.96 & 0.19 & 1.48 & 0.17 & 1.61 \\ 1.42 & 0.11 & 2.76 & 0.12 & 1.60 & 0.11 & 1.70 & 0.13 \\ 0.16 & 2.01 & 0.14 & 2.24 & 0.10 & 1.46 & 0.16 & 1.26 \\ 1.71 & 0.18 & 1.60 & 0.14 & 2.38 & 0.16 & 1.95 & 0.14 \\ 0.15 & 1.44 & 0.13 & 1.42 & 0.13 & 2.46 & 0.13 & 1.68 \\ 1.89 & 0.18 & 1.72 & 0.19 & 1.95 & 0.19 & 2.28 & 0.15 \\ 0.16 & 1.64 & 0.19 & 1.26 & 0.13 & 1.58 & 0.18 & 2.96 \end{pmatrix},$$

such that

$$S = \begin{pmatrix} 1.13 & 10.93 & 1.03 & 12.16 & 0.10 & 0.89 & 0.07 & 1.17 \\ 13.79 & 1.19 & 9.10 & 1.02 & 0.56 & 0.09 & 0.83 & 0.09 \\ 1.00 & 14.95 & 1.07 & 15.86 & 0.10 & 1.35 & 0.07 & 0.76 \\ 10.70 & 0.97 & 13.65 & 1.23 & 1.76 & 0.08 & 0.83 & 0.11 \\ 0.09 & 0.81 & 0.09 & 0.92 & 1.07 & 13.18 & 0.90 & 12.67 \\ 1.25 & 0.10 & 1.36 & 0.13 & 16.24 & 1.09 & 9.81 & 1.04 \\ 0.08 & 0.68 & 0.07 & 1.00 & 0.91 & 6.43 & 1.26 & 10.95 \\ 0.82 & 0.11 & 0.64 & 0.10 & 13.65 & 1.08 & 9.70 & 0.86 \end{pmatrix}.$$

The S's are symmetrized via $S+S^t$ and the C's are computed in the usual way. The respective spectra are are shown in Figure 4.3. (See also the application

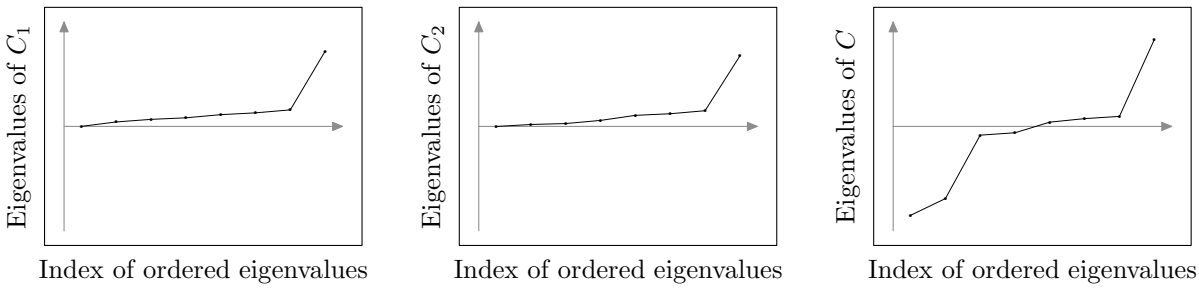


Figure 4.3. Spectrum of C_1 (left), C_2 (right) and $s_{ij} = \frac{(S_1)_{ij}}{(S_2)_{ij}}$ (right). The spectrum of C is non-trivial.

on USPS handwritten digits for a negative spectrum explained by this model.)

SIMPLE MODEL III.

Cognitive psychology

The last simple model to explain negative spectra is inspired by approaches in cognitive psychology to explain human similarity judgments which typically yield non-metric dissimilarities (Thomas and Mareschal, 1997). We will generalize them to explain the spectra often encountered in this particular field.

Let $\{f_1, f_2, \dots f_n\}$ be a set of feature vectors. A given data point x_i can be decomposed as follows:

$$x_i = \sum_{k=1}^{n} \alpha_k^{(i)} f_k.$$

The squared Euclidean distance between x_i and x_j therefore reads:

$$d_{ij} = \|x_i - x_j\|^2 = \Big\| \sum_{k=1}^{n} \big(\alpha_k^{(i)} - \alpha_k^{(j)}\big) f_k \Big\|^2.$$

However this assumes constant feature-perception, i.e. a constant mental image with respect to different tasks. In the realm of human perception this is often not the case, as illustrated by the following well known visual "traps" (Figure 4.4).

Feature perception

Figure 4.4. Left: What do you see? A small cube in the corner of a room or a large cube with a cubic hole or a small cube sticking with one corner on a large one? Right: What do you see? A young lady or an old woman? If you were to compare this picture to a large set of images of young ladies or old women, the (unwilling) perception switch could induce large individual weights on the similarity.

Our perception has several ways to interpret the figures which can give rise to large deviations of the perceived dissimilarities. It is important to notice here that in the realm of human similarity judgments, one may not speak of artifact or erroneous judgments with respect to a Euclidean norm. The latter seems rather exceptional in these cases.

A possible way to model different interpretation of a give geometric object is to introduce weights $\{\omega^{(1)}, \omega^{(2)} \dots \omega^{(d)}\}$, $\omega^{(l)} \in \mathbb{R}^n$ for $l = 1, 2, \dots d$, affecting the features.

The similarity judgment between objects then depends on the perceptual state (weight) the observer is in. Assuming that he be in state $\omega^{(l)}$ the distance

becomes:

$$d_{ij} = \|x_i - x_j\|^2 = \Big\| \sum_{k=1}^{n} \big(\alpha_k^{(i)} - \alpha_k^{(j)}\big)\omega_k^{(l)} f_k \Big\|^2. \tag{4.3}$$

With no further restriction this model yields non-metric distance matrices. See Example III for a simple illustration.

Perception switches and non-metricity

In the worst case l is random, but usually perception-switches can be modeled and l becomes some function of (i, j). For l random, non-metricity is an artifact of sample size, since when averaging the d's over p observers the mean dissimilarity is asymptotically metric in p ($\langle d\rangle \to$ metric as $p \to \infty$): the mean weight becomes constant for all i, j equal to the expectation of its distribution.

On the other hand, if we suppose that the function l of (i, j) does not vary much between observers, then the averaging does not flatten out the non-metric structure induced by the perception-switch.

EXAMPLE III.

Consider a weight $\omega^{(l)}$ constant for all feature-vectors, taken to be the unit vectors e_k in this example. Then Equation 4.3 becomes

$$d_{ij} = \big(\omega^{l_{ij}}\big)^2 \Big\| \sum_{k=1}^{n} \big(\alpha_k^{(i)} - \alpha_k^{(j)}\big) e_k \Big\|^2 = \big(\omega^{l_{ij}}\big)^2 \|x_i - x_j\|_2^2,$$

where $\|\cdot\|_2$ is the usual unweighted Euclidean norm.

For a simple illustration we take 16 points distributed in two gaussian blobs (Figure 4.5, left) with squared Euclidean distance given by d_2 to represent the objects to compare. Suppose a test person is to pairwise compare these objects (which are not points!) to give it a dissimilarity score and that his perception is strongly affected for the pairs $(2, 3)$, $(7, 2)$ and $(6, 5)$ translating in a strong weighting of these dissimilarities. For the sake of the example, we chose the weights to be 150, 70 and 220 respectively.

The weights then acts as follows:

$$\begin{aligned} d(2,3) &= d_2(2,3) \cdot 150, \\ d(7,2) &= d_2(7,2) \cdot 70, \\ d(6,5) &= d_2(6,5) \cdot 220. \end{aligned}$$

The spectrum of the associated centralized pseudo-covariance matrix is given in Figure 4.5, right, and exhibits a clear negative spectrum.

REMARK. In applications we only dispose of S and *no* handy decomposition. By the preceding explicite construction, it is clear, that the negative eigenvalues potentially code important information, even when there is no obvious process, which is responsible for the negative part of these spectra.

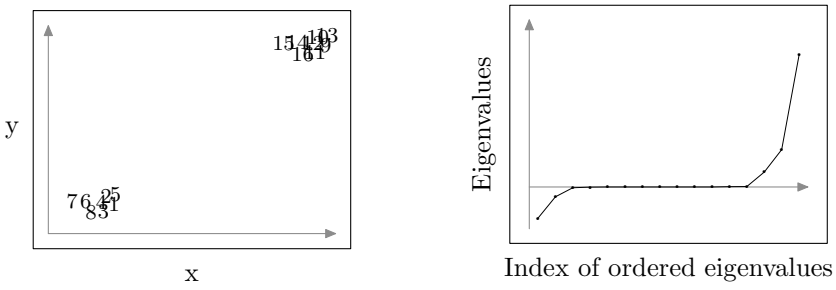


Figure 4.5. Simple data distribution (left) and spectrum associated to the weighted distance matrix.

§. 4.4.

RECOVERING THE INFORMATION CODED BY THE NEGATIVE PART OF THE SPECTRUM.

There are two simple algorithms to recover the information coded by the negative part of the spectrum.

Algorithm

For the first one, we essentially follow the idea of the constant shift embedding by metricizing C through a simple shift, except that we replace the minimal shift by some offset $> c_o$ (unidimensional constant). Projection follows like for the leading eigendirections. Let D be a non-squared Euclidean dissimilarity matrix.

$$
\begin{array}{c}
D \\
\downarrow C^c = -\frac{1}{2} D^c \\
C^c \text{ with negative eigenvalues} \\
\downarrow \text{shift} \\
C^c_+ = C^c + c_o I_n \\
\downarrow \text{spectral decomposition} \\
V \Lambda V^t \\
X_L = \Lambda_L^{\frac{1}{2}} V_L^t,
\end{array}
$$

with $c_o > |\lambda_n(C^c)|$ to have a positive semi-definite C^c_+ and avoid singularities around the origin. L is the chosen subspace, given by the retained set of eigendirections v_i.

Visualization

Retaining only the first two coordinates ($L = \{v_1, v_2\}$) of the obtained vectors corresponds to a projection onto the first two leading eigendirections. Retaining the last two ($L = \{v_{n-1}, v_n\}$) is a projection onto the last two eigendirections: *This corresponds—up to a scaling factor of the order of $\sqrt{c_o}$—to a projection onto directions which corresponds to the non-metric part of C.*

Non-metric part of C

DEFINITION 4.4.1. We define the *non-metric* part of C—or the spectrum thereof—to be the eigedirections resp. negative eigenvalues induced by the metric violations of the associated D.

Drawback of this algorithm

The shifting procedures by the scaling factors tends to even out the differences between the eigenvalues. In the majority of cases this effect is negligible, especially if the difference of the eigenvalues associated to the direction we project is small. However if $\lambda_1 \gg \lambda_2$ (and likewise $\lambda_{n-1} \gg \lambda_n$) then the shift might affect the interpretation of the embedded data, usually in a stronger way than by simply projecting onto the leading positive eigendirections (Mardia, 1978).

pseudo-Euclidean approach

To elude this potential drawback, we consider the pseudo-Euclidean approach, which comes down to taking the absolute value of the negative eigenvalues and projecting onto the corresponding eigenvectors (see Section 2.5).

Algorithm

The algorithm then reads:

$$
\begin{array}{c}
D \\
\Big\downarrow {\scriptstyle C=-\frac{1}{2}QDQ} \\
C \text{ with } p \text{ positive and } q \text{ negative eigenvalues} \\
\Big\downarrow \text{spectral decomposition} \\
V\Lambda V^t = V|\Lambda|^{\frac{1}{2}} M |\Lambda|^{\frac{1}{2}} V^t \\
X_L = |\Lambda_L|^{\frac{1}{2}} V_L^t,
\end{array}
$$

where M is the block matrix consisting of the blocks $I_{p\times p}$, $-I_{q\times q}$ and $0_{k\times k}$ (with $k = n - p - q$).

The columns of X_L contain the vectors x_i in the l-dimensional subspace L. At this point L can be very general. However, as for PCA, we will find it sensible to choose a few leading eigendirections *which can also include eigendirections associated to the negative part of the spectrum.*

Visualization

Retaining only the first two coordinates ($L = \{v_1, v_2\}$) of the obtained vectors corresponds to a projection onto the first two leading eigendirections. Retaining the last two ($L = \{v_{n-1}, v_n\}$) is a projection onto the last two eigendirections: *This corresponds to a projection onto directions related to the metric violations of D.*

Equivalence of these two algorithms for two dimensional embeddings

When considering two dimensional embeddings, it is easily shown that there always exists a shift, namely $c_o = |\lambda_n| + |\lambda_{n-1}|$, such that the embedding be identical to the one obtained in the pseudo-Euclidean space, up to inversion of the last and second last component (rotation by $\frac{\pi}{2}$). This result is based on the simple identities:

$$\lambda_n + c_o = \lambda_n + |\lambda_n| + |\lambda_{n-1}| = |\lambda_{n-1}|$$
$$\lambda_{n-1} + c_o = \lambda_n + |\lambda_{n-1}| + |\lambda_{n-1}| = |\lambda_n|,$$

which hold for $\lambda_n < 0$ and $\lambda_{n-1} < 0$ (see Figure 4.6).

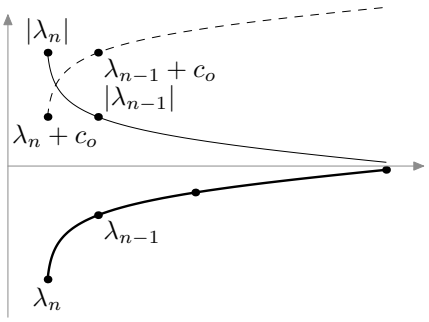


Figure 4.6. Schematic representation of the equivalence of the shift procedure and the embedding into pseudo-Euclidean space. This equivalence only holds in two dimensions. From the figure it is evident that the projection are identical only up to a rotation of $\frac{\pi}{2}$.

Consequences

The above algorithms allow to extract the information coded by the negative eigenvalues induced by metric violations. On may discover features accounted for in the negative eigenvalues which are "cut away"—or otherwise neglected—by usual embedding procedures. This is usually the way to go, since we only dispose of an S for which a priori no obvious decomposition exists.

INTERPRETING NEGATIVE EIGENSPACES. For a positive semi-definite C the projections along the leading eigendirections can readily be interpreted as projections along the axis of high variances of the data. For pseudo-covariance matrices this still holds up to a scaling factor when shifting the spectrum so as to assure positive semi-definiteness.

Pseudo-Euclidean space revisited

For projections onto the negative eigendirections the interpretation is not so straightforward since there is no clear intuition on what "negative variance" represents. However, the second above presented algorithm relies on a pseudo-Euclidean-style decomposition of the embedding space. As we have seen in the second chapter, the pseudo-Euclidean space effectively amounts to two Euclidean spaces one of which has a positive semi-definite inner product and the other a negative semi-definite inner product. As we have seen in the

second chapter, an interesting interpretation of the distances in a pseudo-Euclidean space is that they can be looked at as a difference of squared Euclidean distances from the "positive" and the "negative" space, by the decomposition $\mathbb{R}^{(p,q)} = \mathbb{R}^p + i\mathbb{R}^q$, so that $d_{ij} = d_{ij}^{(\mathbb{R}^p)} - d_{ij}^{(\mathbb{R}^q)}$, where the d_{ij} are squared-Euclidean. This is the rationale behind the first construction of a non-metric D via $d_{ij} = (D_1 - D_2)_{ij}$.

The power of this decomposition resides in the fact that the negative eigenvalues now admit the natural interpretation of variances of the data projected onto directions in $\mathbb{R}^q$. Thus the variance along v_n is $\sqrt{|\lambda_n|}$, the variance along v_{n-1} is $\sqrt{|\lambda_{n-1}|}$, etc.

EXAMPLE I (CONT.).

We project the data according to the above algorithm. As expected, we re-

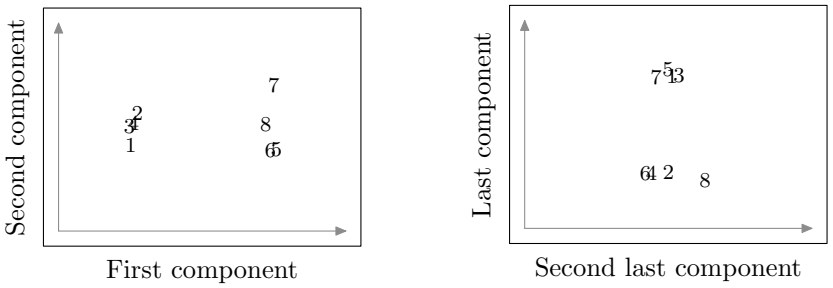


Figure 4.7. Projection onto the two leading positive eigendirection (left), projection onto the two leading negative eigendirections (right).

cover the variance due to the cluster structure $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$ in the positives, the cluster structure $\{1, 3, 5, 7\}$ and $\{2, 4, 6, 8\}$ in the negatives (See Figure 4.7).

Neglecting the non-metric part would have resulted in the loss of the second cluster structure. Figure 4.8 shows all possible projections onto the directions given by the components of the eigendecomposition in the pseudo-Euclidean space. The cluster structures $\{1, 2, 3, 4\}$, $\{5, 6, 7, 8\}$ and $\{1, 3, 5, 7\}$, $\{2, 4, 6, 8\}$ are unidimensional and are only recovered by projections involving the first or last index. Of course, other projections are possible, but we claim that as the current methods are based on large variance, they will inherently not be able to capture the cluster structure $\{1, 3, 5, 7\}$, $\{2, 4, 6, 8\}$.

EXAMPLE II (CONT.).

We project the data according to the above algorithm. As expected, we re-

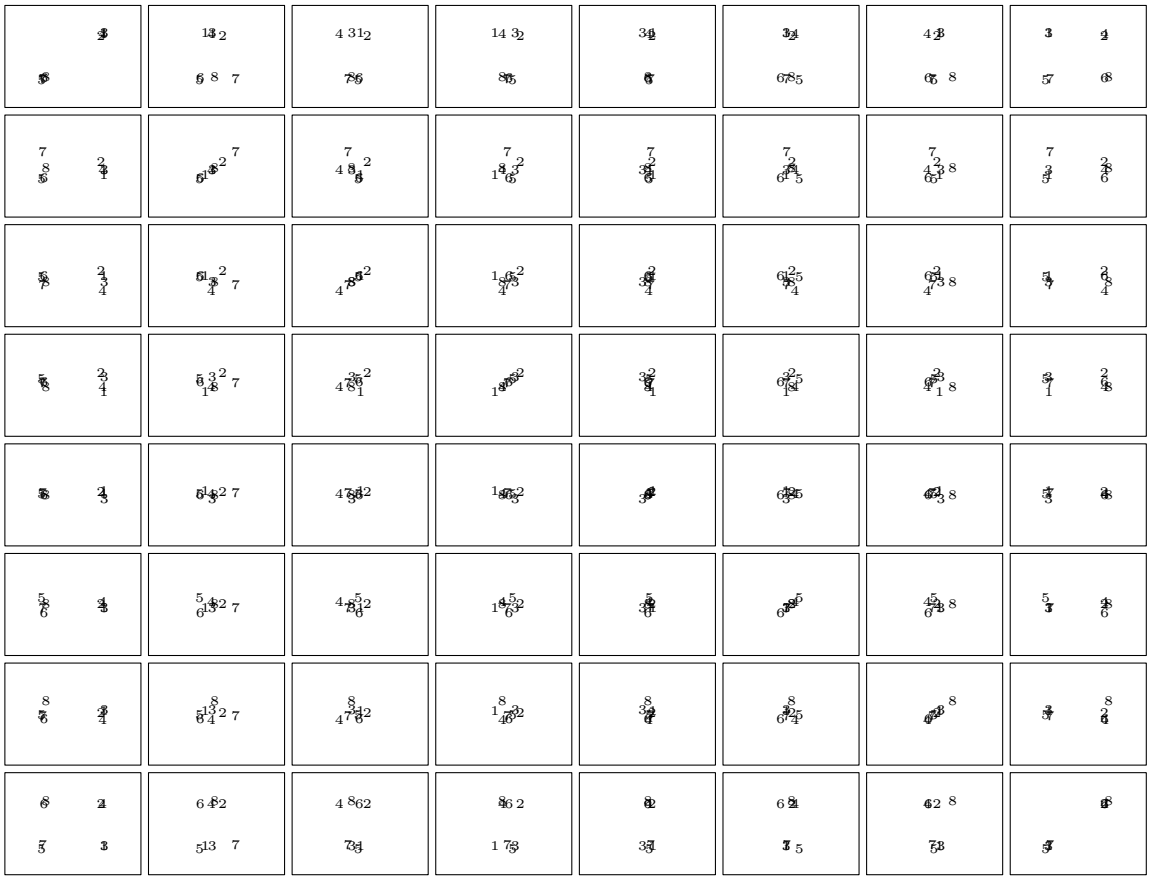


Figure 4.8. Exhaustive projections in two dimensions onto the 8×8 *possible subspaces. The rows are a loop over* i *and give the abscissa, the columns are a loop over* j *and give the ordinate. The first row and first column systematically separate* $\{1, 2, 3, 4\}$ *from* $\{5, 6, 7, 8\}$ *while the last row and last column systematically separate* $\{1, 3, 5, 7\}$ *from* $\{2, 4, 6, 8\}$*. These two cluster structures are unidimensional. They are not recovered by projections involving other that the first or last index.*

cover the variance due to the cluster structure $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$ in the positives, the cluster structure $\{1, 3, 5, 7\}$ and $\{2, 4, 6, 8\}$ in the negatives (See Figure 4.9).

Neglecting the non-metric part would have resulted in the loss of the second cluster structure.

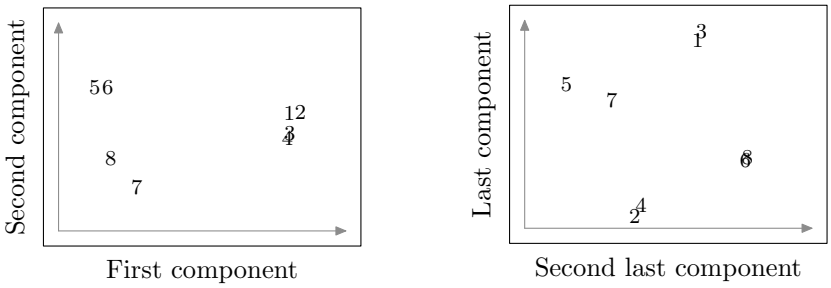


Figure 4.9. Projection onto the two leading positive eigendirection (left), projection onto the two leading negative eigendirections (right).

EXAMPLE III (CONT.).

Figure 4.10 shows the recovery of the points from the weighted distance matrix yields the same cluster solution in the positive part (left) and no definite structure in the negative (right). However, we see that the variance in the negative corresponds to the points whose mutual distance has been (strongly) weighted.

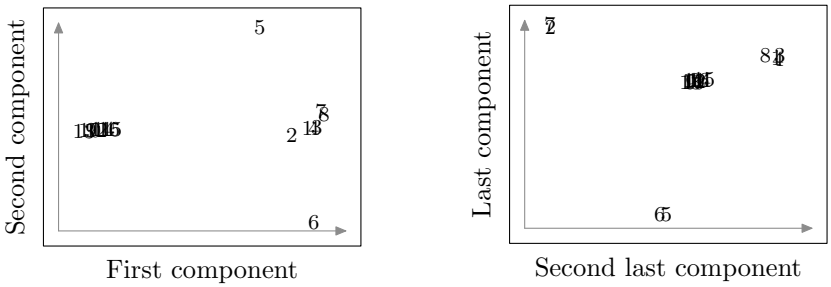


Figure 4.10. Recovery from weighted distance matrix. Projection onto the two leading positive eigendirection (left), projection onto the two leading negative eigendirections (right).

The information contained in the negative part here codes for the individual weighting of the (dis)similarity. This also is encountered, e.g. in pairwise alignments of proteins, where the length itself of the compared protein largely contributes to the score and must be corrected so that the score translates the genuine, evolutionary distance between the strings.

Note that the projection on the last two components admits a simple explanation with models I and II as well. The individual weighting of D can be modeled by the addition to D of a sparse matrix with entries roughly given by the weighted element of D times its weight factor. This addition translates into

high similarity in the projection onto the last two components.

§. 4.5.

SUMMARY.

We summarize the procedure and the rationale behind it (see *schematic* diagram Figure 4.12).

Apples!

Consider the following illustrative setting: we have apples of different sizes and two colors (Figure 4.11). There are two salient features: size (geometric) and color (categorial).

Figure 4.11. Initial data-objects: apples of different size and two colors. (The images were kindly furnished by http://www.marzipanworld.com/)

Obtention of D or S

These apples are pairwise compared, either by a computer algorithm, a human test subject or any other mechanism. This comparison yields a dissimilarity matrix D or a similarity matrix S.

In the later case a problem specific dissimilarity matrix is obtained from S. Typical choices involve $D = 1 - S$, $d_{ij} = s_{ii} + s_{jj} - 2s_{ij}$, $d_{ij} = -\log(s_{ij})$, $d_{ij} = \sqrt{-\log(s_{ij})}$ or $d_{ij} = \frac{1}{s_{ij}} - 1$.

Computation of C

The embedding procedure: from D we compute the centralized pseudo-covariance matrix C^c and we compute its spectrum. C^c is positive semi-definite if and only if D is squared Euclidean.

Projection onto the leading positive *and* negative eigendirections

We project the data onto the first two leading eigenvectors explaining the variance associated to the first feature (size). Second we project the data onto the last two eigenvectors accounting for the variance of the second feature (color). This last step is done either by shifting the spectrum, thus enforcing the distances to be squared Euclidean, or by going into the pseudo-Euclidean space.

Feature discovery

The second feature is lost by methods relying exclusively on high variance.

Conversely we propose the exploration of the negative eigenspectrum for *feature discovery*.

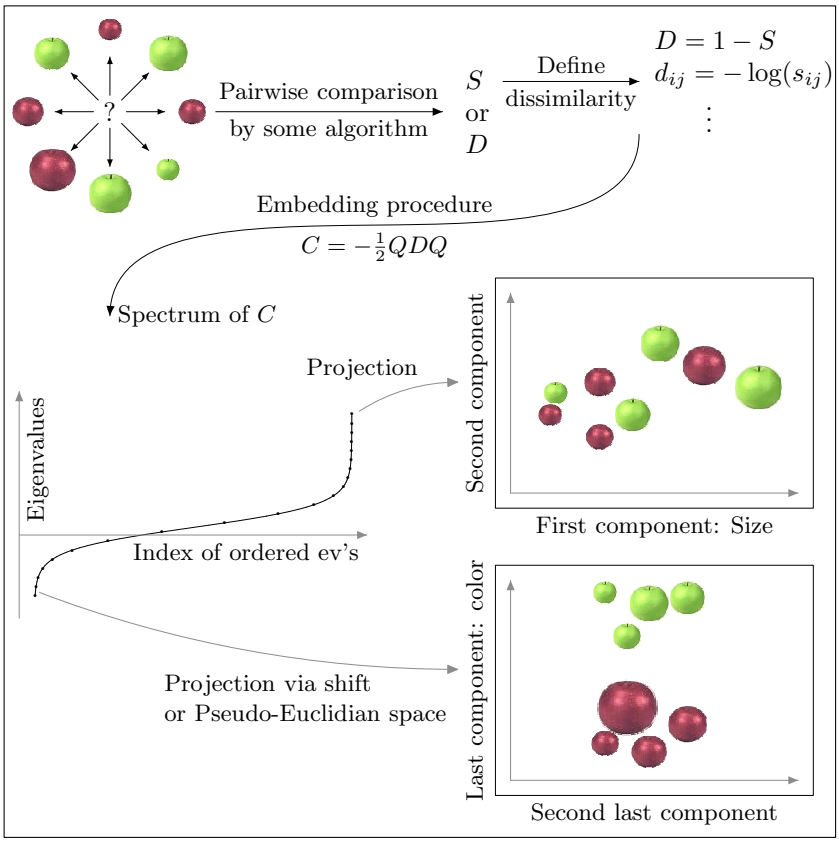


Figure 4.12. Summarizing diagram. The variance coded by the negative eigenvalues can code for features different in nature than the one coded by positive eigenvalues.

§. 4.6.

APPLICATIONS.

We will illustrate with three real world problems the importance of investigating the negative spectrum for a deeper understanding of the data.

USPS HANDWRITTEN DIGITS.

The data set

The similarity matrix is obtained from binary image matching on the digits 0 and 7 of the USPS data set. Digits 0 and 7 have been chosen since they exhibit clear geometric difference. All images have been sorted according to decreasing sum of pixel value (1 to 256) thus separating the bold digits from the light ones. Shown in Figure 4.13 are the 25 boldest and lightest for the 0's and the 7's. A total of 1844 samples have been retained. The images have been normalized and discretized to have binary pixel value 0 and 1.

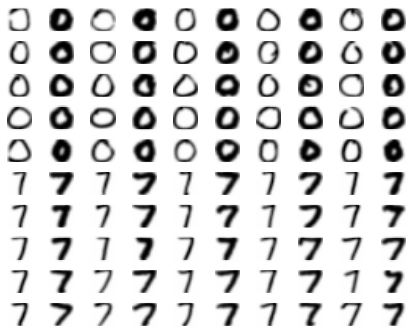


Figure 4.13. 100 handwritten digits from the USPS database. To illustrate how different features can be coded by penalization we chose a data set consisting of two geometrical shapes, namely 0 and 7. The digits with boldest and lightest stroke weight were chosen, thereby obtaining categorical distinction.

BINARY IMAGE MATCHING. Let r and s denote the label of two images and s_{rs} the score rating mutual similarity.

Score matrix

In the case of binary images, s_{rs} is a function of a, b, c and d, where a counts the number of variables, where both objects s and r score 1, b the number of variables, where r scores 1 and s scores 0, etc. (see Table 4.1). The counting

		Object s	
		1	0
Object r	1	a	b
	0	c	d

Table 4.1. Construction of similarity scores for binary data. a to d are counting variables that stand for different possibles binary pixel-matiching. Thanks to these counting variables, a myriad of similarity scores can be defined.

variables a, b, c and d allow to define a variety of similarity scores s_{rs}. See e.g. Cox and Cox (2001), Everitt and Rabe-Hesketh (1997). Note that the same constructions also appears in other fields of taxonomy (Gower, 1971).

Simpson score

We will be interested in the *Simpson* score, defined by:

$$s_{rs} = \frac{a}{\min(a+b, a+c)}. \tag{4.4}$$

The Simpson score for every pair of images yields a similarity matrix which is converted to a dissimilarity matrix via $d_{ij} = s_{ii} + s_{jj} - 2s_{ij}$. The associated pseudo-covariance matrix C exhibits a strongly falling negative spectrum, corresponding to highly non-metric data for the chosen subset of USPS digits (see Figure 4.14).

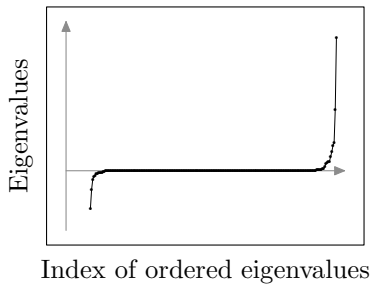


Figure 4.14. Spectrum of C^c. As expected there are a couple of leading eigenvalues indicating large concentration of variance. However, on the other side of the spectrum, a non-trivial tail of negative eigenvalues of large magnitude indicate severe metric violations.

Results

Projection onto the eigenvectors associated to the first leading eigenvalues and projection onto the eigenvectors associated to the last eigenvalues yield results of different nature: see Figure 4.15 and Figure 4.16.

In each case there is a clear interpretation of the variance according to salient features. The variance in the "positive" eigenvectors corresponds to the geometrical distinction between the shapes of the 0's and the 7's. In the "negative" eigenvectors, however, the variance is associated to the feature of stroke weight.

This interesting feature would have been lost if we had embedded the data by conventional methods thereby cutting away the negative part of the spectrum.

Simpson decomposed

The Simpson score allows for a nice interpretation in terms of the second simple model presented. If we pose $(S_1)_{rs} = a$ and $(S_2)_{rs} = \min(a+b, a+c)$, s_{rs} simply reads:

$$s_{rs} = \frac{(S_1)_{rs}}{(S_2)_{rs}}.$$

Figure 4.17 to Figure 4.19 show the corresponding projections. A subset of only 100 digits has been used to stress the separation (see Figure 4.13). The figures depict the spectra and recovered points of S_1, S_2 and S respectively.

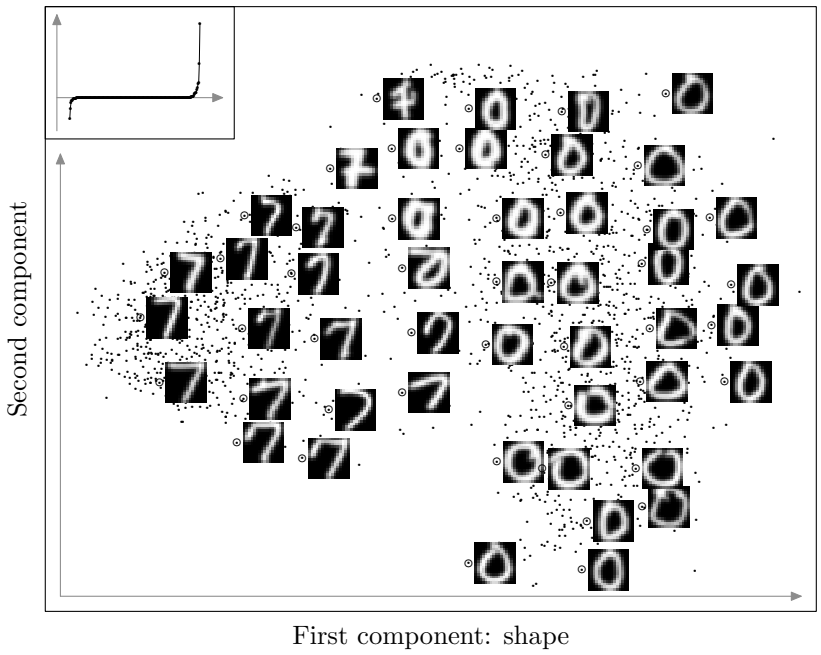


Figure 4.15. Projection onto the first two positive eigendirections. The first component separates the geometrical shapes 0 *from* 7.

The variance along the first principal component separates the 0 from the 7. For S_2 it separates bold from light. This latter variance is recovered in the last eigendirection of S.

Note that the variance corresponding to the second leading eigenvalue of C_1 (covariance matrix associated to S_1) also corresponds to a separation of bold vs. light. We have four nicely separated clusters in the first two leading eigenvalues, as obtained by other binary image matching scores.

However, we have to recall that in the generic case, we *do not* dispose of a decomposition of S so that this information, even though it might exist, is not available to us. Since we dispose only of S, finding the features associated to the stroke weight of the digits requires to look at the negative eigendirections.

COMPARISON WITH MDS. Different projections obtained by MDS have been confronted to our results. The experiments have been carried out with the program `XGvis` (Buja et al., 2001) which allows for a variety of MDS cost functions. It allows to users to chose between four MDS variants, namely between the Torgerson-Gower inner-product scaling and Kruskal-Shepard distance scal-

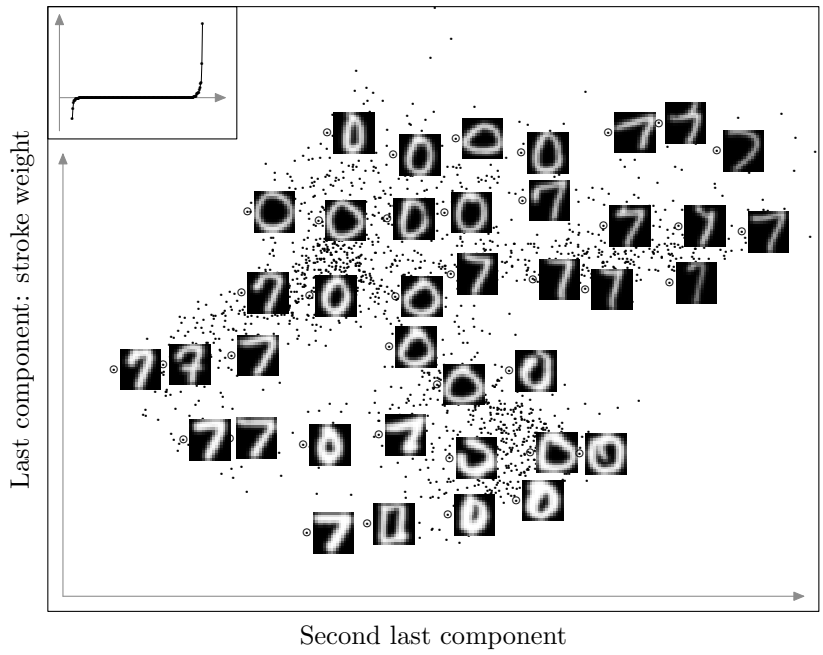


Figure 4.16. Projection onto the onto the last two negative eigendirections. The last component separates the stroke weight into light and bold.

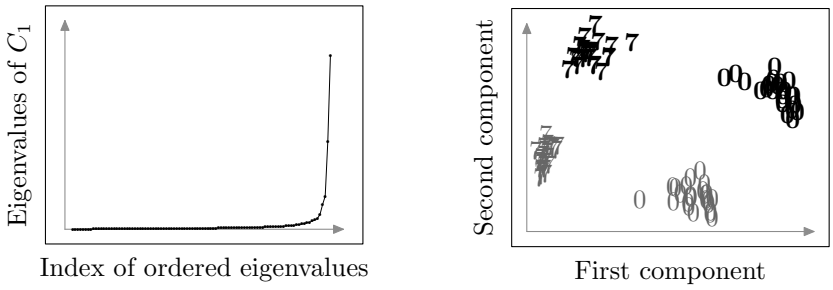


Figure 4.17. Spectrum of the covariance matrix associated to the numerator S_1 (left) and corresponding projection onto the leading two eigendirections.

ing. For each variant, one can choose between metric and non-metric scaling. Remember that "non-metric" in this context refers to proximity data for which only the rank order is taken into the account. To avoid confusion we will instead call it "rank-only".

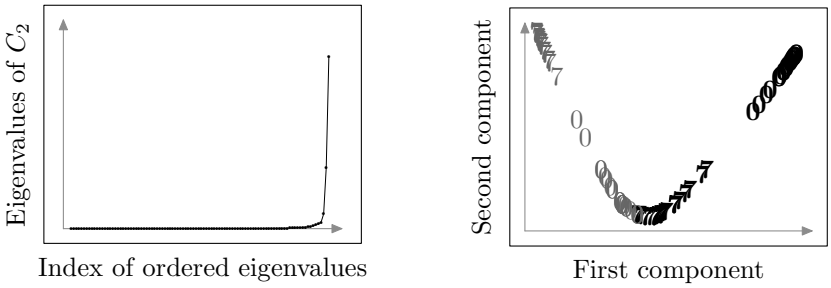


Figure 4.18. Spectrum of the covariance matrix associated to the denominator S_2 (left) and corresponding projection onto the leading two eigendirections.

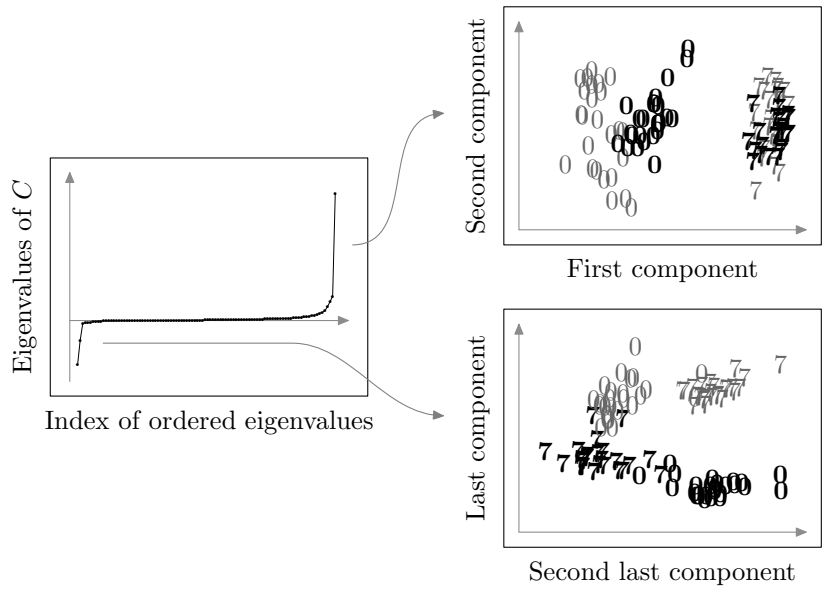


Figure 4.19. Spectrum of S and the points recovered for the positive and negative eigenvalues.

The general Kruskal-Shepard distance scaling for metric scaling optimizes:

$$\text{stress}(x_1, x_2, \dots x_n) = \left(1 - \frac{\left(\sum_{i,j}^n \omega_{ij} d_{ij} \|x_i - x_j\|\right)^2}{\left(\sum_{i,j}^n \omega_{ij} d_{ij}^2\right)\left(\sum_{i,j}^n \omega_{ij} \|x_i - x_j\|^2\right)}\right)^{\frac{1}{2}},$$

with $\omega_{ij} = d_{ij}^r$, with $-4 \leqslant r \leqslant 4$ is a weight factor. Notice that if $d_{ij} = \|x_i - x_j\|$, $\text{stress}(x_1, x_2, \dots x_n) = 0$.

For the default parameter $r = 1$ one obtains a projection separating the 0's from the 7's, corresponding to the projection along the first leading eigendirections. This was to be expected since MDS is a distance based algorithm.

Figure 4.20 shows the result for classical scaling (left) and for Kruskal-Shepard (Krsk/Sh) distance scaling (right). They both seem to separate quite well the 0's from the 7's, except in the central region for Krsk/Sh distance scaling which also heavily suffers the drawback of initialization as seen on Figure 4.21.

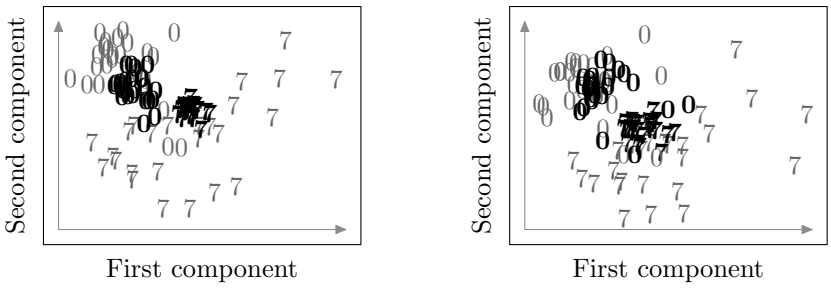


Figure 4.20. Metric Classic (left), Metric Krsk/Sh distance scaling (right).

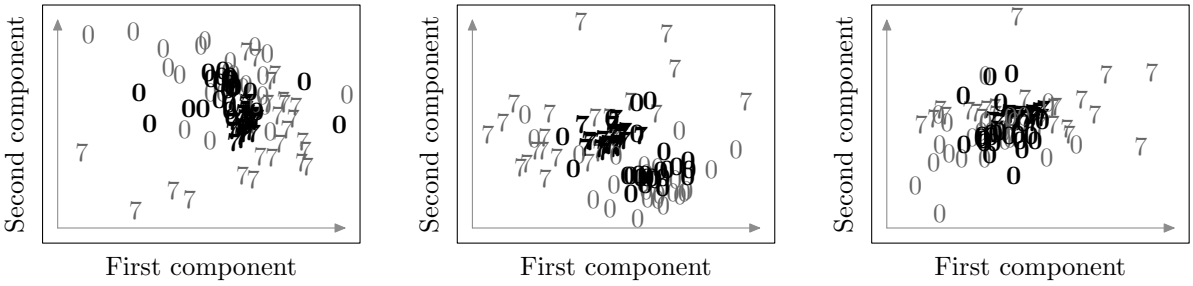


Figure 4.21. Metric Krsk/Sh with three new initializations. While the 0's are globally separated from the 7's except for the central region (see also Figure 4.20, right) the obtained projections vary a lot from one initialization to the other.

Figure 4.22 also shows two instances of some exotic variants of MDS, both sensible (left) and very hard to interpret (right). Different parameters like the data power and the weight factor have been tried, both for metric and rank-only variants. `XGvis` even implements a rank-only version of classical scaling which seems to be a contradiction in terms since classical scaling is geometric in nature.

Drawbacks of MDS

The main problem of MDS is that it does not tell the experimenter what to do. Varying the parameters yields a myriad of projections with little handy in-

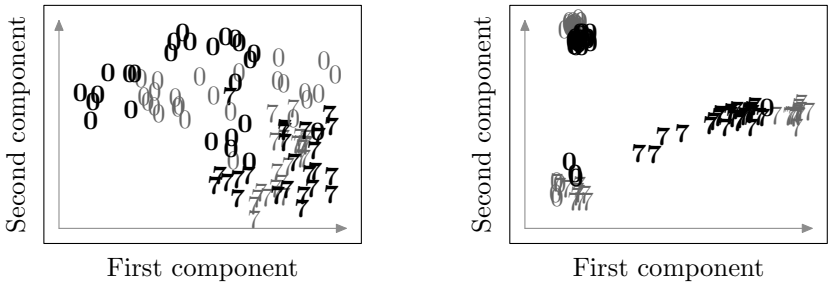


Figure 4.22. Exotic MDS variants: rank-only classical scaling (left), Metric Krsk/Sh with D^5. Data Power and Minkowski Metric parameters turn out to be of little importance (right).

terpretation, notwithstanding the problem of local minima leading to different projections for different initializations (Figure 4.21). The intuition is impaired by the complexity of the cost function so that finding good projections seems to be left to chance. MDS still leaves open the issue about model selection, whereas our solution is obtained without choosing parameters.

But the main point to retain here, is that whatever choice of parameters we have explored *MDS proves unable to separate the bold digits from the light ones like in Figure 4.16*. As before, we claim that this is inherently impossible with MDS which, as all current distance based methods, takes into account large variance.

TEXT-MINING.

The data set

We are interested in the semantic structure of nouns and adjectives from different text sources. In this application we chose two topically unrelated sources: on one hand, Grimm's Fairy Tales[1], on the other popular science articles about space exploration[2]. Both sources contributed 60 documents containing roughly between 500 and 1500 words each.

A subset of 120 nouns and adjectives has been selected, containing both very specific and very general terms out of both data sources.

SIMILARITY MEASURE FOR WORDS. We are not interested in the absolute recurrence of a word, i.e. how many times it occurs within a given document. We only consider whether a word appears or not in a document.

[1]Project Gutenberg `http://promo.net/pg/`

[2]Science at Nasa articles `http://science.nasa.gov/headlines/news_-archive.htm`

Contingency table

From a set of p documents and a choice of n keywords we can construct a contingency table, by simply indicating whether word i ($i = 1, 2, \dots n$) appears in document k ($k = 1, 2, \dots p$) or not. This yields a $p \times n$ boolean matrix X, with $x_{ki} = 1$ if word i appears in document k and 0 else (see Table 4.6).

	Word 1	Word 2	$\cdots$	Word n
Doc 1	1	0	$\cdots$	1
Doc 2	1	1	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Doc p	1	0	$\cdots$	1

Table 4.2. The contingency table X indicating whether word i appears in document k or not. This table does not take into account the frequency with which a given word appears.

Let X_i denote the ith column of X (associated to word i).

Keyword Semantic Proximity

We will take the *Keyword Semantic Proximity* as similarity measure (Rocha (2001) or Rocha and Bollen (2001) and references therein), which expresses that two words are similar if they often appear together in a document. This similarity is penalized if they individually spread over a large number of documents:

$$\begin{aligned} s_{ij} &= \frac{\#\{\text{documents where word } i \textit{ and} \text{ word } j \text{ appear}\}}{\#\{\text{documents where word } i \textit{ or} \text{ word } j \text{ appear}\}} \\ &= \frac{\sum_{X_i+X_j=2} 1}{\sum_{X_i=1} 1 + \sum_{X_j=1} 1 - \sum_{X_i+X_j=2} 1}. \end{aligned} \tag{4.5}$$

From this similarity measure, we obtain a dissimilarity matrix via, e.g. $d_{ij} = -\log(s_{ij})$. In Rocha (2001) the author uses $d_{ij} = 1/s_{ij} - 1$ which is another possible choice. In either case, the resulting dissimilarity matrix d is not squared Euclidean such that the associated (pseudo-)covariance matrix exhibits strong negative eigenvalues (see inset in Figure 4.23).

Results

The data is projected on the first two leading eigenvectors. The result is given in Figure 4.23.

On the far left we find the words stemming from the popular science articles whereas on the far right (e.g. “nuclear”, “computer”, “physics” etc.), we have those from Grimm’s Fairy Tales (e.g. “castle”, “queen”, “ravens” etc.). Towards the center they mix with words spreading over both text sources. The variance corresponds to the semantic context of the words.

Projection onto the last two eigendirections yields a distribution over a new interesting feature. The result is given in Figure 4.24.

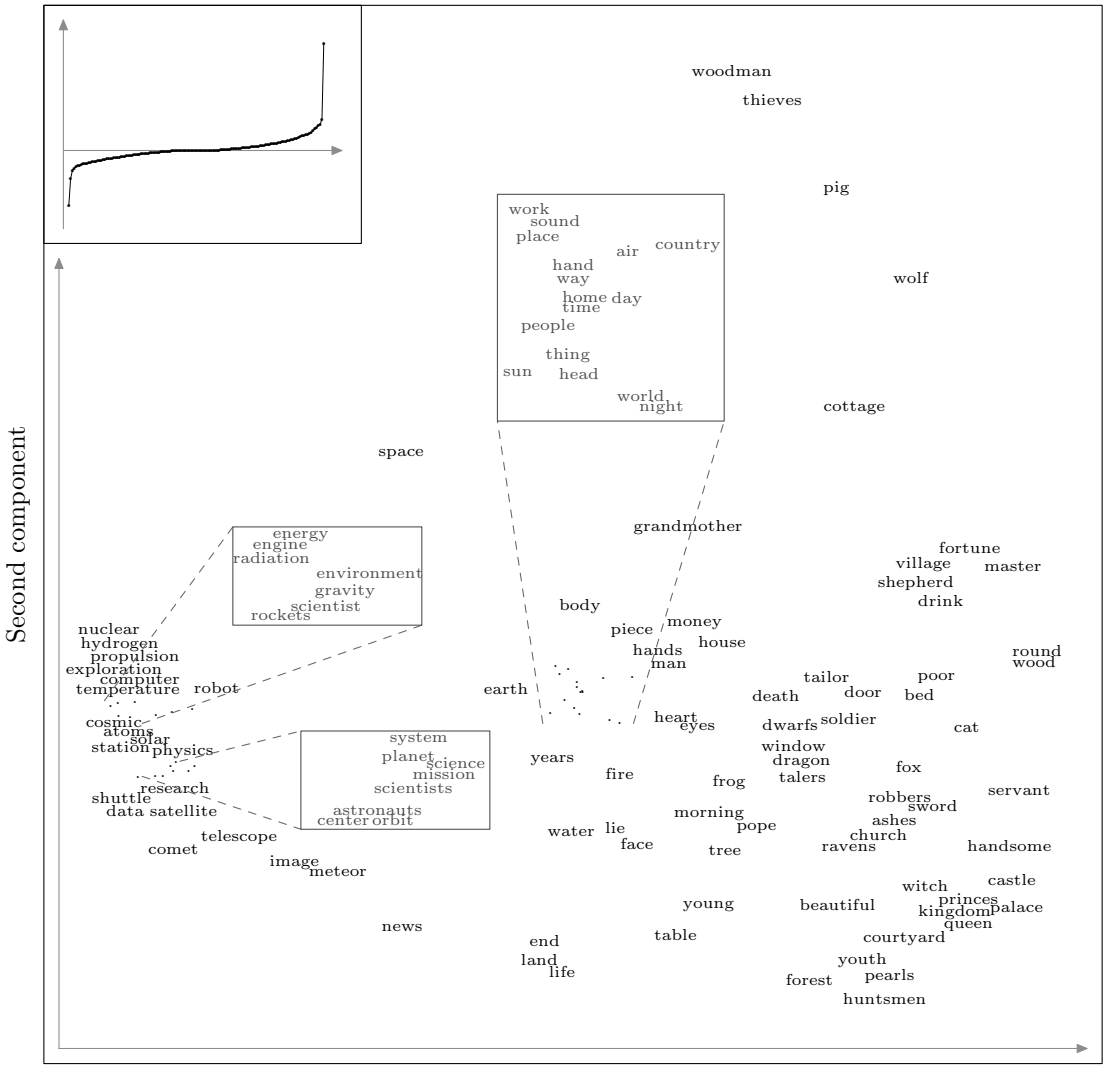


Figure 4.23. Projection onto the onto the first two eigendirections. The first eigendirection separates the semantic context.

We notice that in the upper half we find words of high specificity of either of the sources (e.g. “astronauts”, “wolf”, ‘witch‘” etc.). In the lower half we see an accumulation of words with general, unspecific, meaning, expected to be

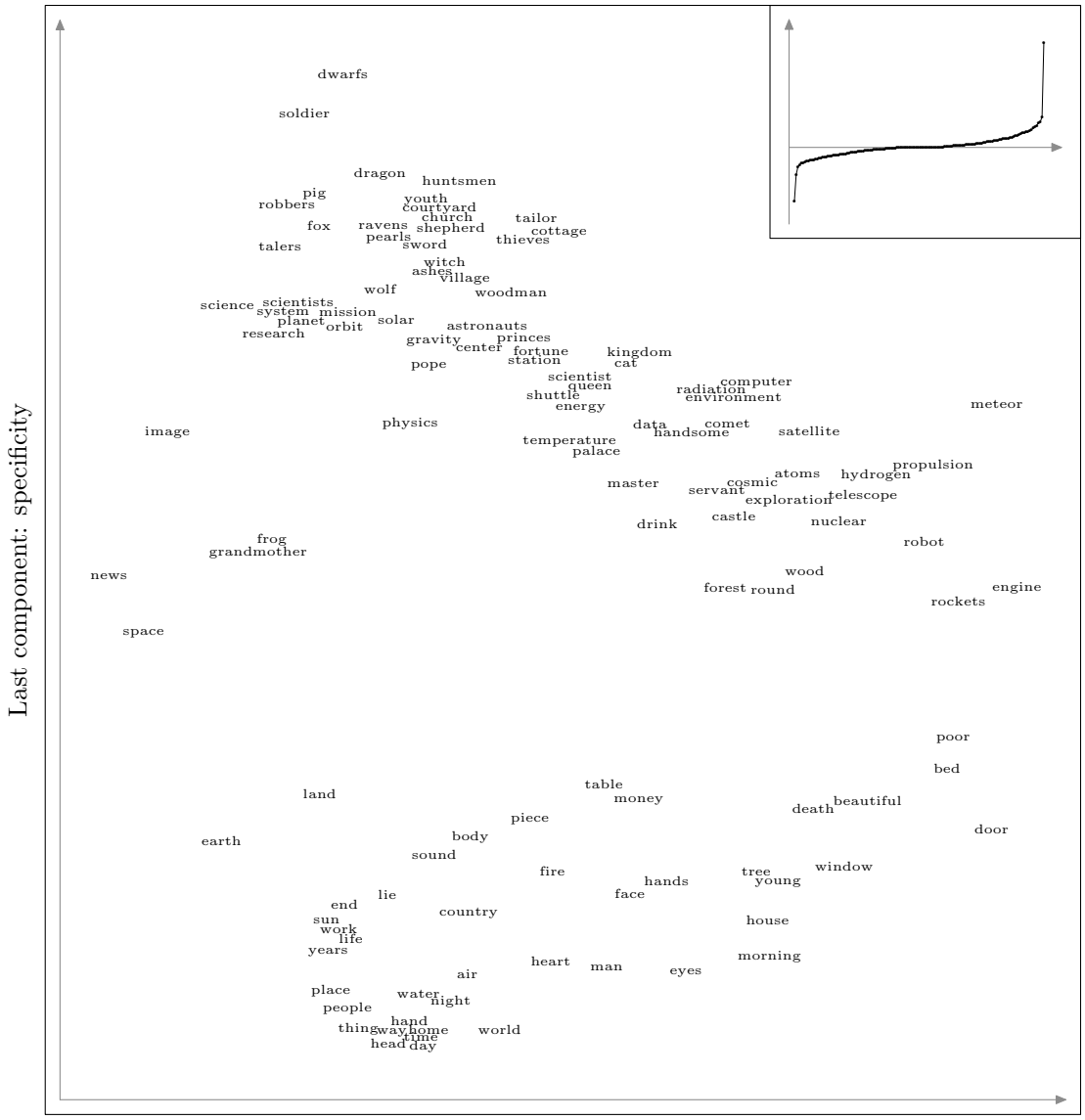


Figure 4.24. Projection onto the onto the last two negative eigendirections. The last eigendirection separates the specificity of the words.

found in a large variety of documents (e.g. “day”, “world”, “thing” etc.). Thus the variance associated to the last eigendirection corresponds to the specificity of the words.

This feature would have been lost by algorithms not specifically taking into account the negative part of the spectrum.

Of course the notion of specificity respectively generality is not absolute but depends on the underlying data sources. “Day”, “world”, “thing” etc. are general with respect to the Grimm’s Fairy Tales and the NASA articles.

HUMAN SIMILARITY JUDGMENTS FROM COGNITIVE PSYCHOLOGY.

We finally present an example from human similarity judgments in cognitive psychology. This will also allow us to illustrate Model III (page 70).

The data set

The pairwise dissimilarity data (Table 4.6) is obtained from Gati and Tversky (1982). The stimuli tested consist of 16 images of flowers having leafs of varying elongation and stems of increasing size (Figure 4.25). These two stimuli were presented to a group of thirty undergraduate students from Hebrew University who, individually, evaluated their mutual dissimilarity on a 20-point scale. (See Gati and Tversky (1982) for details and Table 4.6 for the averaged results.)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	7.9	9.5	10.2	3.8	10.7	11.2	11.9	6.7	15.4	15.4	16.4	9	17.7	17.2	18.7
2		0	5.1	7.3	12.6	4.5	7.3	9.8	15.4	7.1	11.9	15	17.7	9.1	13.9	15.8
3			0	5	12.3	9	4.3	7.6	16.3	12.8	7.5	11.1	18.5	14.1	9.3	11.6
4				0	14.9	10.5	7.7	4.2	17.6	15.8	11	6.5	19.1	17.1	12.9	8.8
5					0	10.6	10.6	13	4.3	12.1	13.2	14.9	5.8	15.2	16.4	16.9
6						0	5.7	9.1	13.6	4.9	9.8	13.4	15.9	6.9	12.7	15.1
7							0	5.9	14.4	10.6	4.8	8.2	16.8	12.7	6.8	10.3
8								0	15.7	12.5	8.5	5.1	18.2	15.5	9.8	6
9									0	10	12	13.8	4.4	12.1	13.8	15.2
10										0	7.3	10.6	13.8	4.3	8.4	13
11											0	6.6	14.7	9.3	4.3	8.2
12												0	16.5	12.9	8.1	3.5
13													0	11.1	11.5	13.7
14														0	6.8	11.1
15															0	5.8
16																0

Table 4.3. Average ratings for dissimilarity between plants. The table is taken to be symmetric.

We have processed the data according to the presented algorithm.

Results

In the positive eigendirections we obtain a very neat reconstruction of the two geometric features, namely the elongation of the leafs and the size of the stem. There seems to be no tendency to favor one over the other. The first component explains the variance in leaf elongation (horizontal axis), the second the variance of the stem size (vertical axis).

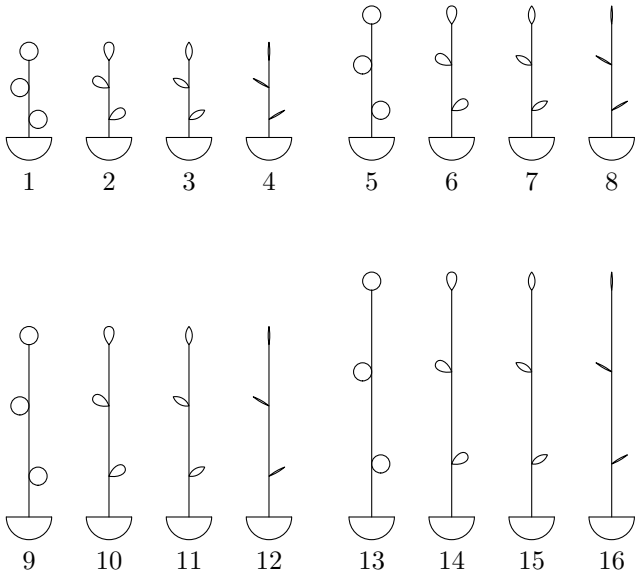


Figure 4.25. Images of the flowerpots presented to the test person. On one hand we have flowerpots with plants of increasing stem size, on the other we have plants with varying leaf elongation.

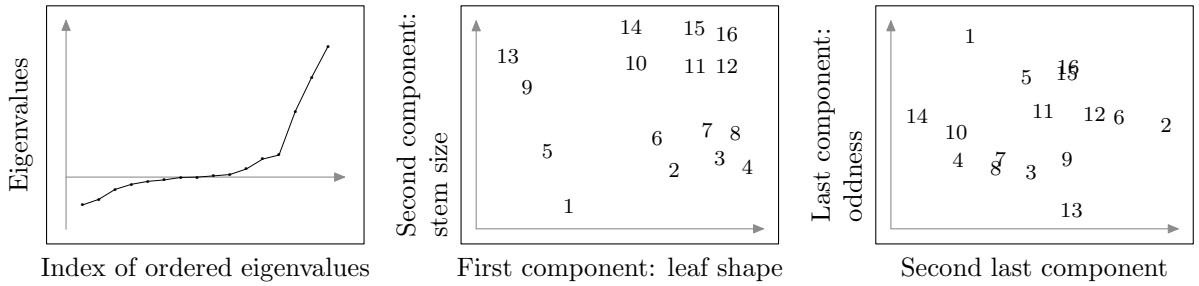


Figure 4.26. Left: Spectrum of the similarity matrix. Middle: Projection onto the leading two positive eigenvalues. Right: projection onto the last two eigendirections.

The projection onto the last two negative eigendirections exhibits further information, as shown by Figure 4.26, right. The interpretation, however, is not so straight forward as previously. This is, where feature discovery begins. Two cluster loosely form, separated by the last eigendirection (vertical axis). They are $\{1, 2, 5, 6, 11, 12, 15, 16\}$ and $\{3, 4, 7, 8, 9, 10, 13, 14\}$. A possible feature could be the oddness of a plant, such that the first clusters contains

the odd plants, and the second the "normal"-ones. Indeed, we tend to expect plants with small leafs to be of small size and plants with large leafs to be of greater size. The odds here are the small plants with large leafs and the large plants with small leafs. This would correspond to a categorial perception while judging similarity.

Features like the concept of normality, or expectation, are not uncommon in cognitive psychology, e.g. in Navarro and Lee (2002) features like the normality or familiarity of faces are discussed in the context of the Modified Contrast Model, along with certainly not easily graspable features like relationships in parenthood. While the authors focus on common and distinctive features and distinguish between conceptual and perceptual features, the interpretation of the discovered features remains—as in our three applications—as a second independent step in data analysis.

MODELING THE FLOWERPOT EXPERIMENT. We model the flowerpot experiment according to Model III (page 70) by starting from a uniform distribution of 16 points in three dimensions. The feature vectors f_k, $k = 1, 2, 3$ were chosen to be the unit vectors $e_1 = (1, 0, 0)$ etc.

The weight vectors are obtained by fitting the d heuristically to the experimental dissimilarity by minimization of the mean over the difference of all matrix elements.

We obtain a good model fit for six weight vectors $\{(8.3, 0, 0), (0, 3.5, 0), (4.7, 4.7, 4.7), (6.4, 6.4, 0), (0, 3.4, 3.4), (3.1, 0, 3.1)\}$. See Figure 4.27.

In other words, following the semantics of the third model presented, one can explain the results of the obtained dissimilarities by six perceptual states of the observer. These seem to outnumber the actually observed features (in the two dimensional representations) which are three in number (the two geometric features in the positives and the categorial one in the negatives). However, we must keep in mind that one may reduce the number of required weights to approximate d by a deeper knowledge of the initial feature presentation, including its dimensionality. We have taken a uniform distribution in three dimensions for lack of this precise knowledge.

§. 4.7.

DISCUSSION.

Summary

This chapter studies the potential of relevant information being coded specifically by the non-metric part of the spectrum of a pseudo-covariance matrix.

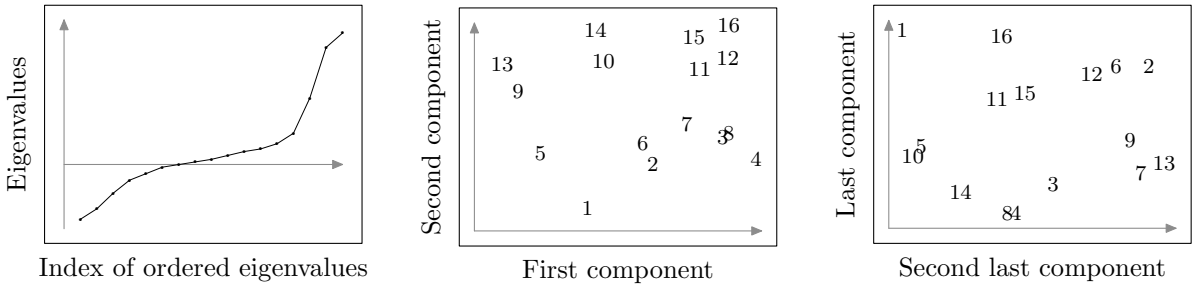


Figure 4.27. Left: Spectrum associated to the model. Middle and right: Prediction of flowerpot experiment.

To the best of the authors knowledge this issue has never been addressed, maybe partly because of the underlying ill-definedness of its unsupervised aspect and the difficulty of problems related to the eigenvalue spectrum.

Nature of this study

We have first chosen a conceptual approach to metric violations and then done an explorative research to show that the negative part of the spectrum *can* code for interesting variance. The stress is on "can" and "interesting". As a matter of fact, every direction associated to some eigenvalues codes for something. The lesson here is that whatever relevant information we look for about a data set, it should not only be sought for in the few leading eigendirections.

We can not assess that for all spectra like Figure 4.1 (right), there is relevant structure coded by the negative eigenvalues, since one might be in a situation of e.g. some fancy noise. This study rather is an incentive to further systematically study non-trivial spectra of pairwise data.

Simple penalization models

Beside this explorative research which heavily relies on examples, we tried to gain some insight on how these spectra come about: penalization by subtraction or division, individual scaling of dissimilarities by perception-switches or algorithmic artifacts.

These models explain simple situations where one specific feature is coded in the non-metric part of the spectrum.

Complex weighting models

In more complex settings, like they arise in human similarity judgments it becomes quite hazardous to speak of a definite number of features. If we compare for instance the images of different faces in order to estimate their similarity, we face a virtually infinite number of features. This really is, where feature discovery begins.

Limits

In order not to deceive expectations, it must be stressed here, that this is a highly non-trivial task in the majority of problems. Thus we often encounter situations where we are utterly incapable of giving any sensible meaning to the distribution of the points along the directions associated to the negative eigenvalues. It is here where the difficult second step of *interpretation* has to set in.

§. 4.8.

CONCLUSION.

Pairwise data in empirical sciences typically violates metricity, either due to noise, fallible estimates, or due to intrinsic non-metric features, such as they arise for human judgments. Non-metricity translates to indefinite pseudo-covariance matrices which precludes the usual processing.

So far the problem of non-metric pairwise data has been tackled by cutting away the negative eigenvalues or shifting the spectrum for a subsequent (Kernel-) PCA analysis. However, little to none attention has been paid to the negative part of the spectrum itself. In particular no answer was given to whether the directions associated to the negative eigenvalues can at all code variance other than noise related.

We have shown that the negative eigenvalues *can* code for relevant structure in the data, thus leading to the discovery of new features, which were lost by common techniques. Three models explain the occurence of non-trivial negative spectra and show that relevant information can be coded by metric violations. The significance of the negative eigenvalues was illustrated on several real world applications, namely USPS handwritten digits, text-mining and human similarity judgments.

5. TOWARDS STRUCTURE LEARNING

In this chapter we will go a first step towards automated structure discovery in non-metric pairwise data. A simple algorithm called *Stability Component Analysis* is developed to detect stable and potentially interesting structure. It can be applied to non-metric pairwise data and successfully extracts the structure coded by non-metricity which can hide, as we have seen in the previous chapter, further information about the data.

§. 5.1. INTRODUCTION.

Visualization

Visualization is part of "human learning". This is the real rationale behind projection onto subspaces of dimension 1, 2 or 3. In the previous chapter, visualization allowed us to understand how metric violations can code for useful information. In visualization, one often learns by local inspection of the correlations. Structure is often recognized by our intelligence on the specific field, by a priori knowledge on the data set rather than by abstract concepts. A biologist, for example, can learn much from a two dimensional data cloud which may hardly be distinguished from a gaussian blob. He will look at local correlations and relate unknown data points to their neighbor in an expert fashion that machines have yet to equal. Far reaching data exploration thus seems rather hindered by strong model assumptions. However, visualization requires the choice of a subspace which critically determines the interpretation. For general problems, there are many candidate projections whether the subspace be obtained by PCA or MDS (see for example Figure 4.8). We are therefore looking for a way to *automatically* select interesting directions.

Let the machines learn!

We claim: let the machines learn! Only with their help are we able to e.g. quantify results and rigorously asses their quality. This comes at the price of model assumptions, the first of which being a definition of structure.

Ill-definedness of the problem

Structure is an ill-defined concept, intuitive on the first glimpse, all but self-explanatory on the second. We therefore are confronted to similar problems as in unsupervised learning and will never obtain sensible results unless we

sacrifice generality to achievability, which usually is a lesson of modesty.

Stability as structure determining

Based on the simple idea of stability analysis (see Roth et al. (2002)) as intrinsic cluster validation, we will define structure as maximally stable cluster solutions. This allows us to elude hefty model assumptions, introducing on the other hand a dependence on the clustering algorithm. However, this dependence is not accidental, it rather reflects a de facto relation between the structure and our way to perceive it (clustering algorithm).

These considerations will quickly lead us to a simple algorithm first proposed in Laub et al. (2004) which we will henceforth call *Stability Component Analysis* (SCA). It will be illustrated by a small toy example and an application to USPS handwritten digits.

§. 5.2.

STABILITY COMPONENT ANALYSIS.

Projections onto the leading negative eigendirections were used to visually inspect the relevance of the structure coded by non-metricity. We now go an important step further beyond visualization towards a quantitative analysis of the relevance of negative eigendirections by automatically detecting, i.e. learning structure.

Loss index

The first step towards structure learning is to define a loss index that is minimized by the structure that we are interested in. As we are effectively most interested in grouping the pairwise data into n groups, we need to focus on stable n-modal clustering solutions.

Bimodal stability

Resampling stability has been shown to be a good criterion assessing the quality of a solution in unsupervised learning, see Roth et al. (2002) and Meinecke et al. (2002). Let us only consider $k = 2$, i.e. the stability index of bimodal clustering solutions. Note that the stability index is a particular choice of a projection index for projection pursuit (Huber, 1985), as it basically measures the probability of confusing the two estimated clusters. In the view of this instability index, interesting directions are thus defined to be the *eigendirections* that allow a stable bimodal clustering solution in the corresponding subspace. The rationale for choosing eigendirections is that they do not depend on parameters, which means that there is no further model selection step required.

Let us first consider only the stability of individual directions, under the strong assumption that one-dimensional subspaces are sufficient to discover interesting structure.

ALGORITHM. The following algorithm computes the subspace of maximal bimodal stability. Let D be some dissimilarity matrix and let X be the column matrix of the data projection onto a pseudo-Euclidean space (see subsection on page 28).

Elementary stability component analysis

1. Compute the bimodal stability for every column of X according to Roth et al. (2002).
2. Sort the instability index.
3. Choose the directions of maximal stability (minimal instability index) with respect to some threshold.

This yields the subspace of maximal bimodal stability in each of its directions.

This algorithm can be interpreted as stability component analysis, hence its name, since it sorts the components according to their decreasing stability as opposed to, say, decreasing variance with PCA.

We will now discuss how this algorithm can be used to ascertain structure coded by non-metricity as well as distinguish between such a structure and non-metricity as artifact of noise.

DETECTING STRUCTURE CODED BY METRIC VIOLATIONS.

Discriminate between metric violations of different nature

One of the goals in studying the information coded by non-metricity is to discriminate between "interesting" information from intrinsic non-metric data and artifacts due to non-metricity induced by noise. Stability component analysis is used to systematically evaluate the stability of bimodality along the eigendirections. For the purpose of visualization it is useful to sort the eigenvectors according to increasing values (e.g. like in Figure 4.1). On the assumption that stable structure is likely to be found in the directions of high variance, the expected curves of the instability index are given in Figure 5.1 (note that high stability means a low value of the instability index).

Structure due to intrinsic non-metricity will reflect in curve like in Figure 5.1, right, whereas non-metricity as mere artifact of noise will translate into a stability curves like in Figure 5.1, left; provided that the spectrum has been sorted as in Figure 4.1.

EXAMPLE: STRUCTURE VS. NOISE.

Illustration

We present a small toy example that highlights the differences between inherent non-metricity and non-metricity caused by noise. Two non-metric data matrices are constructed. The second data set contains a clear structure in the negative eigenspace, whereas in the first data set non-metricity is an artifact of noise. Figure 5.2 shows the spectrum of the associated pseudo-covariance

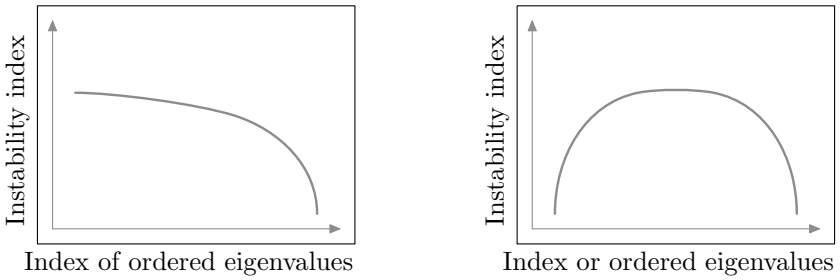


Figure 5.1. Instability indices for sorted eigenspectra. Left: noise, right: structure.

matrix C and the visualization of the data by projection onto the leading positive and negative eigendirections as indicated in the previous chapter. On the left are the figures corresponding to the first data set, on the right, the figures corresponding to the second data set (intrinsic non-metricity). The fourth row of Figure 5.2 shows the result obtained by SCA. In the first case, the instability index exhibits a shape like in the left panel of Figure 5.1 (superposed in light gray), indicating no presence of interesting information specifically coded by non-metricity. In the second case, we obtain a curve similar to that in the right panel of Figure 5.1, indicating the relevant structure in the positive *and negative* part of the spectrum.

This small example illustrates the relevance of negative eigendirections when non-metricity is an intrinsic property of the data. After embedding the non-metric data into a pseudo-Euclidean space SCA effectively and *automatically* selects the leading eigendirections based on the stability criterion.

§. 5.3.

APPLICATION.

The data set

To illustrate our procedure of structure learning in non-metric pairwise data with a real world example we obtain non-metric pairwise data from the USPS handwritten digits data set previously used.

The similarity matrix

The similarity matrix is obtained from binary image matching on the digits 0 and 7 of the USPS data set. Digits 0 and 7 have been chosen since they exhibit clear geometric difference. All images have been sorted according to decreasing sum of pixel value (1 to 256) thus separating the bold digits from the light ones. A total of 200 samples have been retained. The images have

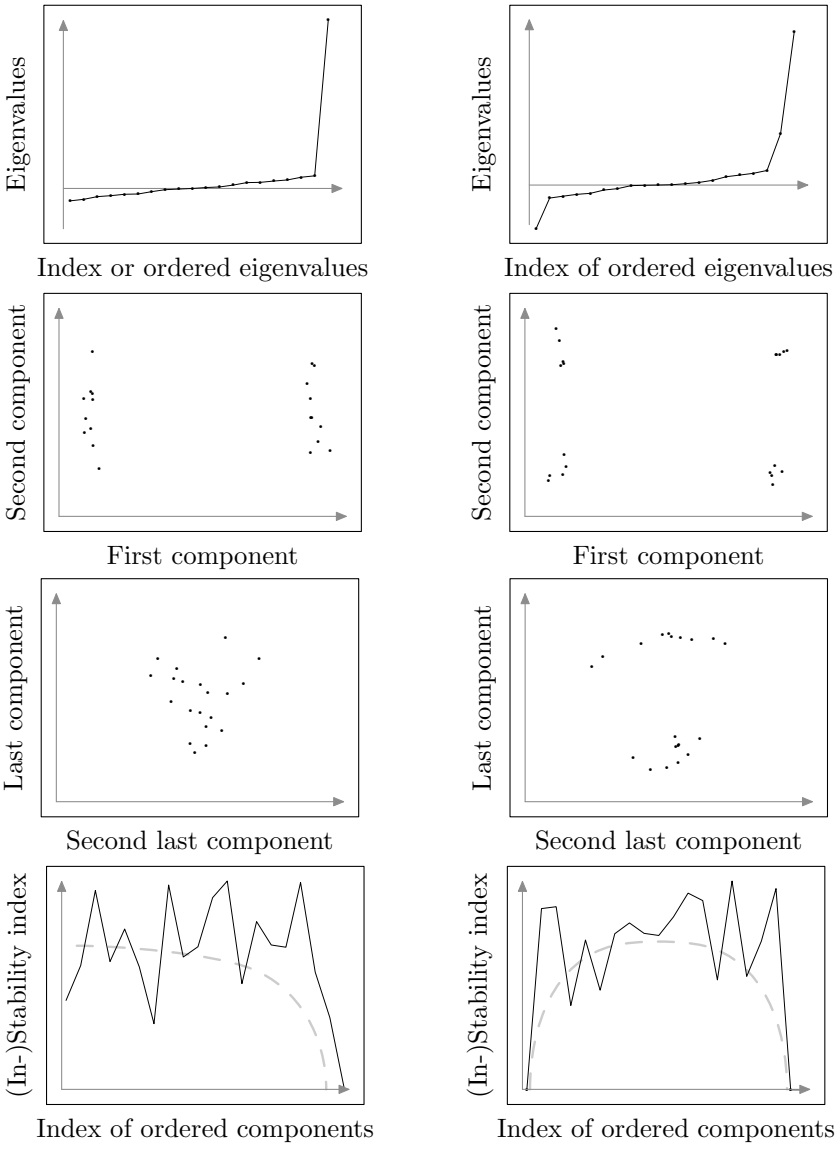


Figure 5.2. Left: Pairwise data where non-metricity is an artifact of noise. Right: Pairwise data with intrinsic non-metricity.

been normalized and discretized to have binary pixel values 0 and 1. Binary image matching is performed and the Simpson score (Equation 4.4) computed.

The Simpson score for every pair of images yields a similarity matrix which is converted to a dissimilarity matrix via $d_{ij} = s_{ii} + s_{jj} - 2s_{ij}$. The associated pseudo-covariance matrix C exhibits a strongly falling negative spectrum, corresponding to highly non-metric data (see Figure 5.4, right).

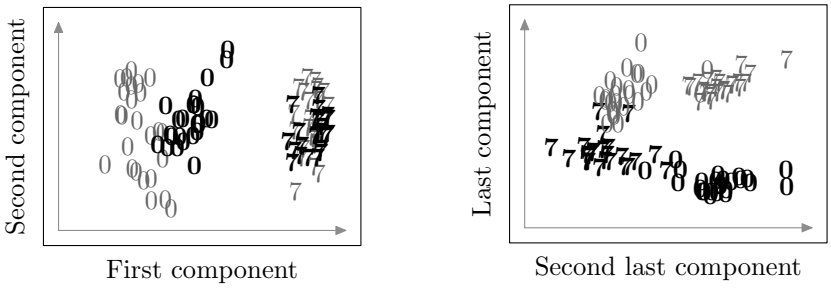


Figure 5.3. Visualization of the USPS data. The light digits are in gray. The leading positive eigendirection separates the 0's from the 7's (left) while the leading negative eigendirection separates the bold digits from the light ones (right).

Visualization

The data (a random subset of 100 digits) is visualized according to the procedure of Section 4.4. The information reflected in the leading positive eigendirections corresponds to the geometric distinction of 0's and 7's (Figure 5.3, left). The information reflected in the leading negative eigendirections corresponds to the categorical distinction of the bold and the light digits (Figure 5.3, right).

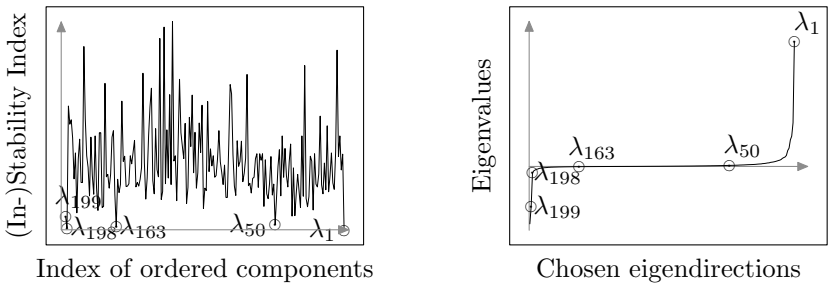


Figure 5.4. The instability index for bimodality as function of the ordered components (left) and the five chosen directions (right).

Note that in Figure 5.3 the second leading eigendirection is not an informative one (as will be seen by the chosen directions). The structure related to the separation between 0's and 7's is contained in the leading eigendirection alone, the second being only good for the purpose of visualization.

Results

SCA is used on the embedded data to search for stable directions. Figure 5.4, left, shows the instability index. While the shape is not as pronounced as in Figure 5.1, right, it is yet clearly visible. The five most stable eigendirections are:

$$[199, 1, 198, 163, 50].$$

Figure 5.4, right, shows the chosen eigendirection. Not astonishingly, the leading eigendirection (1) is among the chosen stable directions. As is shown by Figure 5.3, left, this corresponds to the geometric separation of the 0's and the 7's. The majority of machine learning algorithms will detect this structure.

The interesting new structure can be learned from the negative eigenspace. It corresponds to the leading negative eigendirections 198 and 199. Figure 5.3, right, shows that this indeed makes sense: the two last eigendirections separate the bold digits from the light ones.

Note that the last eigendirection is not 200 since the embedding of an $n \times n$ matrix is of dimension $n - 1$. In the matrix X, we can exclude the empty direction where the coordinates are zero for all vectors.

Our procedure further illustrates the fact that directions with high variance are not automatically stable directions. The second leading eigenvalues is not informative in the sense of stability, as is well seen in Figure 5.3, left. On the other hand, the algorithm selects two unexpected directions, namely 163 and 50. These directions contain stable structure which cannot be easily interpreted as for the leading positive and negative stable eigendirections and we are tempted to label them as outlier due to the non negligible variance of the instability index. As a matter of fact, these directions are no longer chosen when one departs from the assumption of unidimensionality (see discussion in the following section).

§. 5.4.

DISCUSSION.

Generalization of SCA

The presented stability component analysis admits a number of natural extensions which we will briefly discuss here.

The presented "basic version" of SCA considers the bimodal stability in unidimensional subspaces. It is capable to find structure which previously went unnoticed, since the information contained in the non-metric part of the data is not accounted for by the usual machine learning techniques.

A natural extension is to explore subspaces, say, pairs of directions to project

on and calculate the instability index for these $n \times n - 1$ ordered pairs. For adjacent eigendirections in the sense of an ordering given like in Figure 4.1 one obtains the curve for the instability index in Figure 5.5.

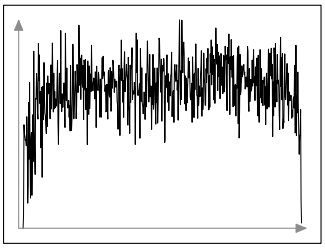


Figure 5.5. Instability index for bimodality for adjacent eigendirections (USPS data set). The structure discovered in the positive and negative subspaces is more pronounced than for the unidimensional subspaces, which shows that the assumption on unidimensionality is abusive.

Obviously the stability "on both ends" is much more pronounced as for the unidimensional case (Figure 5.4, left), which speaks in favor of this more complex evaluation of the stability.

In its most general version, SCA would operate on l-dimensional subspaces and compute a p-modal stability (i.e. k-means with $k = p$). However, for subspaces of more than one dimensions, one easily runs into computational problems because of the exponential number of possible combinations.

§. 5.5.

CONCLUSION.

We have presented a *simple* automated structure learning approach to assess relevant structure coded by non-metricity. It allows to unravel structure neglected by most exploratory learning algorithms.

This chapter shows that automated structure learning can extract problem relevant structure in the negative eigenspace which is associated to the structure coded by metric violations.

The structure learning algorithm proceeds by defining an index on the principle components obtained after embedding of the non-metric pairwise data into a pseudo-Euclidean space.

6. CONCLUSION

Summary

We have studied in this thesis several issues related to non-metric pairwise data, i.e. pairwise proximity data which, when formulated as dissimilarities, violate the requirements of a metric function. Our interest focused both on the nature of these violations and their consequences for subsequent data analysis. This both theoretical and empirical study yielded important new insights in the relationship between vectorial and pairwise representations when considered from a structural rather than geometrical point of view and in the mechanisms which are responsible for metric violations and which must be considered an integral part of the problem rather than an accidental perturbation.

The issue: non-metric pairwise data

There are two main data types in intelligent data analysis, namely the vectorial and the pairwise data. Only small subsets of these two data representations are mutually equivalent. In order to make pairwise data available to the powerful data analytical tools developed for the vectorial representation, they are embedded into a vector space, be it at the price of possibly large distortion. Two question naturally arose: can we find embeddings without distortion? What are the losses incurred when forcefully embedding pairwise data.

Unification of vectorial and pairwise representation for a certain class of cost function based learning algorithms

The first question has been answered. While it is not possible to embed non-metric pairwise data when considering geometric distortion, it has been shown that is still possible to find a set of vectors such that the *structure* is conserved. This is a great step forward since it associates representation and interpretation and shows that for a specific class of clustering algorithms the two data types coincide in as much they yield the same interpretation, i.e. clustering results.

While traditional techniques proceed in two independent steps, first embedding then clustering, by optimizing two unrelated cost functions, the framework of Constant Shift Embedding shows that we really must consider these as one and that, by doing so, we obtain optimal embeddings.

Understanding the semantics of metric violations

The second question was answered by showing that metric violations can carry valuable information about the data set. They can indeed form a structure on their own which is encoded, from an Euclidean point of view, in the negative part of the eigenspectrum of the associated pseudo-covariance matrix. While several authors allude to the danger of forcefully embedding pairwise data, this study is the first one to show why.

Several simple models for non-metric pairwise data have been presented. They allow for a deeper understanding of the processes that underlie metric violations and foster the intuition the experts needs when facing such data. It

has been shown how to extract the information coded by metric violations. The relevance of this information and the models have been illustrated by three worked through applications.

Automated structure learning

Chapter 4 on feature discovery, after visually appreciating the information coded by metric violations, raised the question whether there was a possible automation of this task. General variance interpretation requires great a priori knowledge which the experimenter may not have. In order to still profit from possible information coded by non-metricity a simple algorithm called Stability Component Analysis was developed. It was shown to efficiently work on artificial and real world data.

Quintessence

This thesis is not an exhaustive treatment of non-metric pairwise data and it does not solve all problems related to them. But it certainly has contributed its due part to their demystification. It allowed to cast a different, mathematically well funded look on metric violations and their consequences.

A. APPENDIX: BEYOND EIGENVALUES

In this appendix we briefly present an outlook on ongoing work and possible future research directions. In particular we have a brief look at measures of non-metricity relying on a direct measurement of metric violations rather than the spectrum of a pseudo-covariance matrix.

§. A.1. INTRODUCTION.

Beyond Euclideanness

In this thesis non-metricity has been investigated as implying non-Euclideanness and thus preventing the pairwise data to be embedded in the ubiquitous Euclidean spaces. However, Euclideanness is a strong assumptions on data, and one might extend the understanding of non-metricity to some "weaker" spaces.

Beyond eigenvalues

Furthermore, our approach to understanding the metric violations passed through the computation of the spectrum of the associated pseudo-covariance matrix. We would like to present here a more direct way of apprehending the violation of triangle inequality.

Return to the root of metricity

If the pairwise data is metric it can obviously be represented in a metric space. This might not be helpful if one wishes to visualize the data, but it may be of some theoretic implication. Recall the definition of metric dissimilarities. As our major concern is Equation 2.7, we will, for sake of simplicity, assume that Equation 2.4 to Equation 2.6 be fulfilled. In the following we will only be concerned with the triangle inequality. We will introduce two *direct* measures for its violations and discuss and illustrate a few of their properties.

§. A.2.

MEASURING TRIANGLE INEQUALITY VIOLATIONS.

Let $D = (d_{ij})$ be a symmetric dissimilarity matrix. (Note that we require D to be symmetric in order to have real spectra.)

The counting matrix T

Define the *counting matrix* $T = (t_{ij})$ to be

$$t_{ij}(D) = \sum_{\substack{k=1 \\ d_{ik}+d_{kj}<d_{ij}}}^{n} 1.$$

Properties

T simply counts the violations of the triangle inequality. It can be easily shown that T is positive, symmetric and reflective, so that T can itself be interpreted as some sort of—usually non-metric—dissimilarity matrix, and that $T \equiv 0$ (i.e. $t_{ij} = 0$ for all $i, j = 1, 2, \dots n$) if and only if D satisfies the triangle inequality. T is a non-linear non-injective function of D whose support is the set of D violating at least once the triangle inequality.

T is sensitive to small perturbation: $T(D + \epsilon) \neq T(D) + \epsilon$. Thus, noise corrupted data might yield positive counts for minor metric violation. Usually these counts will not exceed 1 and thus be still different from large metric deviation which cause large counts.

The amplitude matrix P

Define the *absolute amplitude matrix* $P = (p_{ij})$ to be

$$p_{ij}(D) = \begin{cases} \max\limits_{k=1\dots n} \left(|d_{ij} - d_{ik} - d_{kj}|\right) & \text{if } d_{ik} + d_{kj} < d_{ij} \\ 0 & \text{else.} \end{cases}$$

The amplitude matrix P contains the maximal absolute deviations from the triangle inequality.

Properties

P satisfies the same proprieties as T and can also be interpreted as some sort of dissimilarity matrix. P is *not* scale invariant since $P(\lambda D) = \lambda P(D)$. The absolute amplitude of the non-metricity does depend on the intrinsic scale of the data. However, $P(D + \epsilon) \sim P(D) + \epsilon$, so the influence of noise is not beyond its own scale, as opposed to T.

Note that $T \equiv 0$ if and only if $P \equiv 0$. This follows from the elementary property of T and P to be zero if and only if D satisfies the triangle inequality.

Consequence

Since T and P can be regarded at as dissimilarities, the same embedding procedure as for D can be applied to T and P as described in Section 4.4 of the previous chapter.

THE USPS DATA SET REVISITED.

Illustration

To illustrate how T and P work, we again visit a now well-known data set, namely the data set consisting of 100 USPS digits 0 and 7, bold and light. The dissimilarity matrix is computed via the Simpson score. It is given in Figure A.1, left. The counting matrix T and amplitude matrix P are calculated for this D (Figure A.1, middle and right).

Figure A.1. Distance matrix for the set of 100 *USPS digits* 0 *and* 7*, bold and light (left). In the middle, the corresponding counting matrix* T *and on the right* P*. Note the striking resemblance of their structure.*

We immediately notice their striking resemblance. By merely counting the number of triangle inequality violations we get a dissimilarity matrix with a very similar structure. The same holds for the amplitude matrix. This shows that indeed *the violations of the triangle inequality are entirely structure determining.*

Interpretation

In that sense, both T and P allow us to have a direct look at the violation induced structure. When the metric violations are due to noise and are not intrinsic, this will automatically be reflected in T and P. Figure A.2 shows an

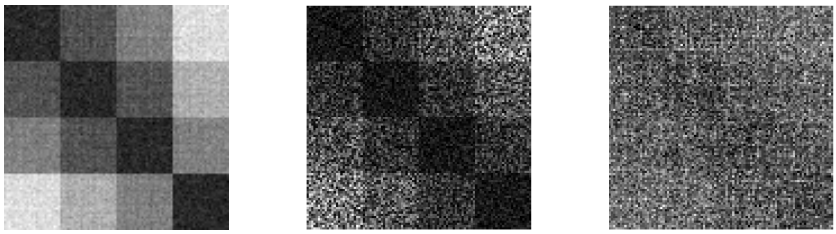


Figure A.2. Distance matrix for four artificially generated clusters (left). In the middle, the corresponding counting matrix T *and on the right* P*.* T *still somewhat resembles* D *because of its sensitivity to noise, while* P *can not be related to* D*.*

artificial data set of four clusters corrupted by some random noise. T is sensitive to noise and thus still keeps track of its origin, even though the resemblance

is not as strong as in the previous example for intrinsically non-metric data. It is P which shows the real difference, since P has no resemblance at all with D which shows that the metric violations of D are not structure determining. In order to further investigate how T and P capture structure by simply measuring the metric violations we recall that both can be interpreted themselves as dissimilarity matrices and can receive proper treatment. Figure A.3 and Figure A.4 show the spectrum of the associated pseudo-covariance matrix and the projection of the data on the leading positive and negative eigenvalues.

T and P as dissimilarity matrices

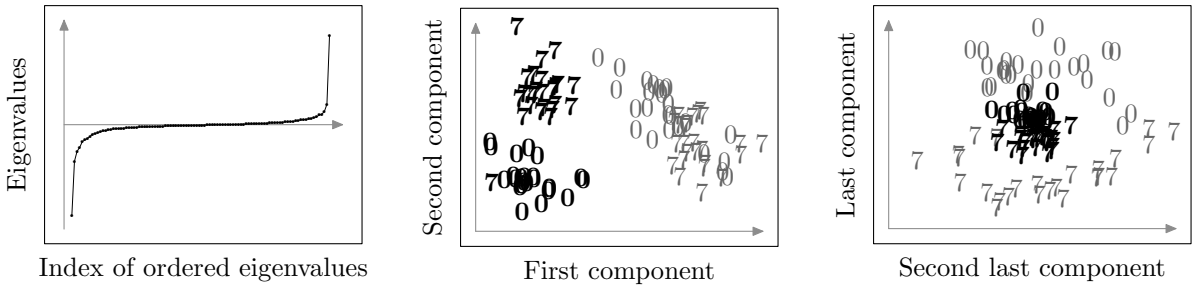


Figure A.3. Spectrum and projections when T is considered a dissimilarity matrix.

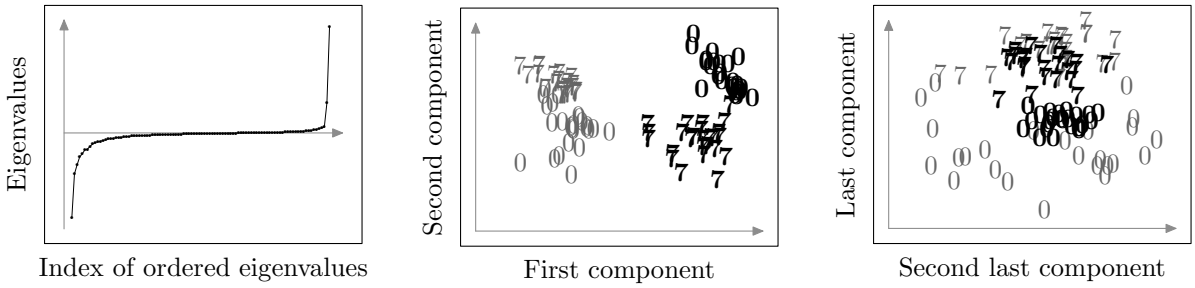


Figure A.4. Spectrum and projections when P is considered a dissimilarity matrix.

Both for T and P the leading eigendirections separates bold from light. This is not astonishing, since this is what the matrices measure: the metric violations induced by the encoding of this feature. Interestingly, the separation of 0 and 7 are (roughly) found in the leading negative eigendirection, which shows that these similarities now act as penalization. In a certain sense, they are the metric violations of the measured metric violations!

§. A.3.

DECOMPOSITION OF D INTO A METRIC AND A NON-METRIC PART.

Yet another look at non-metricity

We have seen in the previous chapter that different similarity or dissimilarity can be combined by subtraction or division to yield a non-metric D.

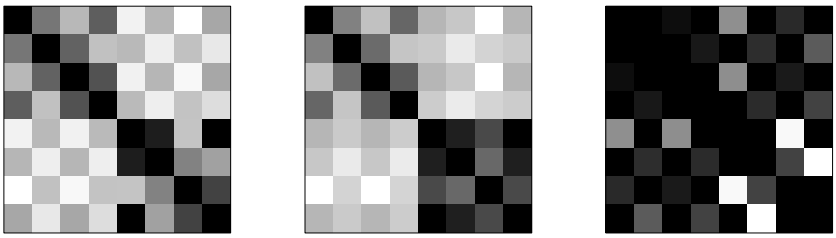
Here we want to briefly consider the converse approach, given a fixed D, find a an additive or multiplication decomposition of the dissimilarity matrix into a metric (possible Euclidean) and non-metric part.

Possible solutions

The problem of the additive decomposition—which is not unique!—into a metric part M and a non-metric part N can by solved e.g. by the constant shift procedure $M = D + N$ where $N = 2\lambda_n(C)(ee^t - I)$, $C = -\frac{1}{2}QDQ$, $e = (1, 1, \dots 1)$ and $\lambda_n(C)$ is the smallest eigenvalue of C. MDS solves the same problem by minimizing $\|D - M\|$ for some norm $\|\cdot\|$. Yet another way is to iteratively subtract T or P until all metric requirements are met. However, the author could not prove that this could be done in a finite number of steps. Usually metricity is achieved within one or two iterations. The results of this procedure which we will not formalize more is shown in Figure A.5 to Figure A.7.

THREE LITTLE EXAMPLES REVISITED.

Recall the three little examples I, II and III given respectively on pages 68, 70, and 72. The dissimilarities were computed and decomposed by iteratively subtracting T. We thus recover a metric matrix M and are left with an non-metric N.



D M N

Figure A.5. Decompostion of D from small example I. The block structure and the line structure are recovered.

Figure A.6 nicely shows the recovery from the block and line structure which were put into it by the from scratch construction. Figure A.5 exhibits the same

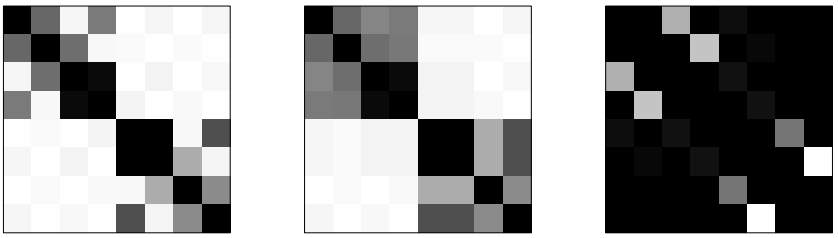


Figure A.6. Decompostion of D from small example II. The block structure and the line structure are recovered.

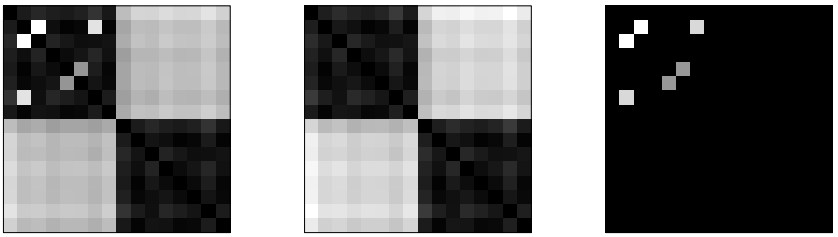


Figure A.7. Decompostion of D from small example III. The weighted distances are recovered in the non-metric part N.

structure but it is less visible. Figure A.7 permits to visualize the "outliers" due to the weighted measurements. These decompositions are yet another way to analyze and visualize metric violations.

§. A.4.

CONCLUSION.

This outlook presented some new considerations on non-metric pairwise data. The previously adopted point of view which focuses on violation of Euclideanness and negative spectra was abandoned in favor of a more generic one, measuring directly the metric violations, namely the number and severeness of triangle inequality violation. These two measures were illustrated and were proved to be able to capture the essence of non-metric pairwise data.

BIBLIOGRAPHY

S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403–410, 1990.

S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI–BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, 25:3389–3402, 1997.

A. Banerjee and J. Ghosh. Clickstream clustering using weighted longest common subsequences. *In Proceedings of the Web Mining Workshop at the 1 st SIAM Conference on Data Mining, Chicago, April 2001.*, 2002.

M. Blatt, S. Wiseman, and E. Domany. Clustering data through an analogy to the potts model. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press: Cambridge, MA, 1996.

B. Boeckmann, A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, M. J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider. The SWISS-PROT protein knowledgebase and its supplement TrEMBL. *Nucleic Acids Res.*, 31:365–370, 2003.

I. Borg and P. Groenen. *Modern Multidimensional Scaling*. Springer, New York, 1997.

P. Brucker. On the complexity of clustering problems. In M. Beckman and H. P. Kunzi, editors, *Optimization and Operations Research: Lecture Notes in Economics and Mathematical Systems*, pages 45–54. Springer, 1978.

A. Buja, D. F. Swayne, M. Littman, N. Dean, and H. Hofmann. Xgvis: Interactive data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, 2001.

F. Chung. Spectral graph theory. *CBMS Regional Conference Series in Mathematics*, 92, 1997.

F. Corpet, F. Servant, J. Gouzy, and D. Kahn. Prodom and prodom-cg: tools for protein domain analysis and whole genome comparisons. *Nucleid Acids Res.*, 28:267–269, 2000.

T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman & Hall, London, 2001.

I. Dagan, S. Marcus, and S. Markovitch. Contextual word similarity and extimation from sparse data. *Computer Speech and Language*, 9(2):123–152, 1995.

P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. *Procs. SODA*, 1999.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, second edition, 2001.

S. Dudoit and J. Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, 3(7):0036.1–0036.21, 2002.

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis*. Cambridge University Press, 1998.

B. S. Everitt and S. Rabe-Hesketh. *The Analysis of Proximity Data*. Arnold, London, 1997.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Inc., San Diego, 1990.

I. Gati and A. Tversky. Representation of qualitative and quantitative dimensions. *Journal of Experimental Psychology: Human Perception and Performance*, 8(2):325–340, 1982.

L. Goldfarb. A unified approach to pattern recognition. *Pattern Recognition*, 17:575–582, 1984.

L. Goldfarb. A new approach to pattern recognition. *Progress in Pattern Recognition*, 2:241–402, 1985.

R. L. Goldstone, D. L. Medin, and D. Gentner. Relational similarity and the nonindependence of features in similarity judgments. *Cognitive Psychology*, pages 222–262, 1991.

J. C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53:325–338, 1952.

J. C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 27:857–874, 1971.

T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer. Classification on pairwise proximity data. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11, pages 438–444. MIT Press: Cambridge, MA, 1999.

W. Greub. *Linear Algebra*. Springer Verlag, 1975.

T. Hofmann and J. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19 (1):1–14, 1997.

T. Hofmann, J. Puzicha, and J. M. Buhmann. Unsupervised texture segmentation in a deterministic annealing ework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):803–818, 1998.

R. A. Horn and C. A. Johnson. *Matrix Analysis*. Cambridge University press, Cambridge, 1995.

P. J. Huber. Projection pursuit. *The Annals of Statistics*, pages 435–475, 1985.

D. W. Jacobs, D. Weinshall, and Y. Gdalyahu. Classification with nonmetric distances: Image retrieval and class representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):583–600, 2000.

A. K. Jain, M. N. Murty, , and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

J. Kapur and H. Kesavan. *Enropy Optimization Principles with Applications*. Academic Press Inc, London, 1992.

H. Kasai, A. Bairoch, K. Watanabe, K. Isono, and S. Harayama. Construction of the gyrb database for the identification and classification of bacteria. *Genome Informatics*, pages 13–21, 1998.

T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 1995.

J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964a.

J. B. Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29:115–129, 1964b.

T. Lange, M. Braun, V. Roth, and J. M. Buhmann. Stability-based model selection. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press: Cambridge, MA, 2003.

J. Laub and K.-R. Müller. Feature discovery in non-metric pairwise data. *Journal of Machine Learning Research*, 5:801–818, 2004.

J. Laub, V. Roth, and K.-R. Müller. Structure learning in non-metric pairwise data. *Submitted*, 2004.

H. Lüthkepohl. *Handbook of Matrices*. Wiley & Sons, New York, 1996.

T. Manke, C. Dieterich, J. Laub, and M. Vingron. Associations and hierarchies in the human transcription factor network. *Submitted*, 2004.

K. V. Mardia. Some properties of classical multi-dimensional scaling. *Communications in Statistics. Theory and Methods*, 13:1233–1241, 1978.

F. Meinecke, A. Ziehe, M. Kawanabe, and K.-R. Müller. A resampling approach to estimate the stability of one-dimensional or multidimensional independent components. *IEEE Transactions on Biomedical Engineering*, 49: 1514–1525, 2002.

S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de–noising in feature spaces. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11, pages 536–542. MIT Press: Cambridge, MA, 1999.

K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, 247:536–540, 1995.

D. J. Navarro and M. D. Lee. Common and distinctive features in stimulus similarity: A modified version of the contrast model. *Psychonomic Bulletin & Review*, 2002.

E. Pękalska, P. Paclík, and R. P. W. Duin. A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research*, 2:175–211, 2001.

W. R. Pearson and D. J. Lipman. Improved tools for biological sequence analysis. *Proc. Natl. Acad. Sci*, 85:2444–2448, 1988.

J. Puzicha, T. Hofmann, , and J. Buhmann. A theory of proximity based clustering: Structure detection by optimization. *Pattern Recognition*, 33(4):617–634, 1999.

L. M. Rocha. Talkmine: A soft computing approach to adaptive knowledge recommendation. *Soft Computing Agents: New Trends for Designing Autonomous Systems*, pages 89–116, 2001.

L. M. Rocha and J. Bollen. Biologically motivated distributed designs for adaptive knowledge management? *Design Principles for the Immune System and other Distributed Autonomous Systems*, pages 305–334, 2001.

K. Rose, E. Gurewitz, , and G. C. Fox. A deterministic annealing approach to clustering. *Pattern Recognition Letters*, 11(9):589–594, 1990.

V. Roth, T. Lange, M. Braun, and J. M. Buhmann. A resampling approach to cluster validation. *Statistics–COMPSTAT*, pages 123–128, 2002.

V. Roth, J. Laub, J. M. Buhmann, and K.-R. Müller. Going metric: Denoising pairwise data. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 817–824. MIT Press: Cambridge, MA, 2003a.

V. Roth, J. Laub, M. Kawanabe, and J. M. Buhmann. Optimal cluster preserving embedding of non-metric proximity data. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 25(12):1540–1551, 2003b.

S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500), 2000.

C. Schäfer and J. Laub. Anneald k-means clustering and decision trees. In Weihs, editor, *Classification, the ubiquitous challenge. Proc. 28th Annual GfKl Conference*, Heidelberg-Berlin, 2005. Springer-Verlag.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

P. Soundararajan and S. Sarkar. Investigation of measures for grouping by graph partitioning. *Computer Vision and Pattern Recognition–CVPR2001*, pages 239–246, 2001.

Y. Takane, F. W. Young, , and de Leeuw J. Nonmetric individual differences multidimensional scaling: an alternating least squares method with optimal scaling features. *Psychometrica*, 42:7–67, 1977.

J. B. Tenenbaum, V. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

M. S. C. Thomas and D. Mareschal. Connectionism and psychological notions of similarity. *Proceedings of the 19th Annual Conference of the Cognitive Science Society*, 1997.

W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17:401–419, 1952.

W. S. Torgerson. *Theory and Methods of Scaling*. John Wiley and Sons, New York, 1958.

K. Tsuda, T. Kin, and K. Asai. Marginalized kernels for biological sequences. *Proc. ISMB*, 2002.

K. Tsuda, S. Uda, T. Kin, and K. Asai. Minimizing the cross validation error to mix kernel matrices of heterogeneous biological data. *Neural Processing Letters*, 19:63–72, 2004.

V.N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

G. Wahba. Support vector machines, reproducing hilbert spaces and the randomized gacv. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 69–88, Cambridge, MA, 1999. MIT Press.

E. W. Weisstein. Basis. In *From MathWorld–A Wolfram Web Resource.*, 2004.

G. Young and A. S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3:19–22, 1938.