David B Skillicorn

Understanding High-Dimensional Spaces

– Springer Brief –

August 22, 2011

Springer

Preface

High-dimensional spaces arise naturally as a way of modelling datasets with many attributes. Such a dataset can be directly represented in a space spanned by the attributes, with each record of the dataset represented as a point in the space whose position depends on its attribute values. Such spaces are not easy to work with because of their high dimensionality: our intuition about space is not reliable, and measures such as distance do not provide as much information as we might expect.

High-dimensional spaces have not received as much attention as their applications deserve, partly for these reasons. Some areas where there has been substantial research are: images and video, with high-dimensional representations based on one attribute per pixel; and spaces with highly non-convex clusters. For images and video, the high dimensionality is an artifact of a direct representation, but the inherent dimensionality is much lower, and easily discoverable. Spaces with a few highly non-convex clusters do occur, but are not typical of the kind of datasets that arise in practice.

There are at least three main areas where complex high-dimensionality and large datasets arise naturally. The first is data collected by online retailers (e.g. Amazon), preference sites (e.g. Pandora), social media sites (e.g. Facebook), and the customer relationship data of all large businesses. In these applications, the amount of data available about any individual is large but also sparse. For example, a site like Pandora has preference information for every song that a user has listened to, but this is still a tiny fraction of all of the songs that the site cares about. A site like Amazon has information about which items any customer has bought, but this is a small fraction of what is available.

The second is data derived from text (and speech). The word usage in a set of documents produces data about the frequency with which each word is used. As in the first case, all of the words used in a given document are visible, but there are always many words that are not used at all. So such datasets are once again large (because easy to construct), wide (because languages contain many words), and sparse (because any document uses a small fraction of the possible words).

The third is data collected for a security, defence, law enforcement or intelligence purpose; or collected about computer networks for cybersecurity. Such datasets are

large and wide because of the need to enable as good solutions as possible by throwing the data collection net wide. This third domain differs from the previous two because of greater emphasis on the anomalous or outlying parts of the data rather than the more central and commonplace.

High-dimensional datasets are usually analyzed in two ways: by finding the set of clusters they contain; or by looking for the outliers – really two sides of the same coin. However, these simple strategies conceal subtleties that are usually ignored. A cluster cannot really be understood without seeing its relationships to other clusters "around" it; and outliers cannot be understood without understanding both the clusters that they are nearest too, and what other outliers are "around" them. The development of the idea of local outliers has helped with this latter issue, but is still weak because a local outlier is defined only with respect to its nearest non-outlying cluster.

In this book we introduce two ideas that are not completely new, but which have not received as much attention as they should have, and for which the research results are partial and scattered. In essence, we suggest a new way of thinking about how to understand high-dimensional spaces using two models: the *skeleton* which relates the clusters to one another, and *boundaries in empty space* which provides another perspective on outliers, and on outlying regions.

This book should be useful to those who are analyzing high-dimensional spaces using existing tools, and who feel that they are not getting as much out of the data as they could; also their managers who are trying to understand the path forward in terms of what is possible, and how they might get there. The book assumes either that the reader has a reasonable grasp of mainstream data mining tools and techniques, or does not need to get into the weeds of the technology but needs a sense of the landscape. The book may also be useful for graduate students and other researchers who are looking for open problems, or new ways to think about and apply older techniques.

Acknowledgements

of helpful people

Kingston, *David Skillicorn*
August 2011

Contents

Chapter 1
Introduction

Many organizations collect large amounts of data: businesses about their customers, governments about their citizens and visitors, scientists about physical systems, and economists about financial systems. Collecting such data is the easy part; extracting useful knowledge from it is often much harder.

Consider, for example, the tax collection branches of governments. Most governments create a record for each resident and business each year, describing their incomes, their outflows that can be used as deductions, their investments, and usually some demographic information as well. From this information they calculate the amount of tax that each resident and business should pay.

But what else could they learn from all of this data? It is a very large amount of data: there is one record per citizen and business (a tax return); and each record contains a large number of pieces of information, although of course most of these values are null or zero for most returns.

One kind of knowledge that governments spend a lot of effort to acquire is whether, who, and by how much individuals and businesses are defrauding the government by providing false information that results in them owing (apparently) less tax. A strategy for detecting this is to compare people or businesses of the same general kind, and see whether there are some that seem qualitatively different, without any obvious explanation of why. In other words, one way to discover tax fraud is by exploiting *similarity* among the income and outflows of individuals, expecting that similarity of record values should be associated with similarity of tax payable.

This strategy works quite well and is routinely used by tax departments, targeting one year dentists, say, and another year flight attendants. Of course, this means that tax cheats in other professions receive less scrutiny (until their turn comes up). It is attractive for governments to do this kind of assessment globally (for all taxpayers) every year, but there are several problems: pragmatically, the size of the data makes the necessary computations alarmingly large; but, more significantly, we do not yet understand clearly how to represent and analyze the structure of the space implicit in such a collection of data. If a particular taxpayer looks unusual within the set of taxpayers of the same general kind, is it because they are paying less tax than they should, or is it because of some other difference between them and the set to

which they are being compared. In other words, there are deep issues to do with the concept of "similarity", and these issues are complicated by the size and richness of the datasets that we would like to analyze.

The focus of this book is on ways to think about structure and meaning associated with such large, complex datasets; algorithms that can help to understand them; and further analyses that can be applied once a dataset has been modelled that provide payoffs in many different domains (including tax fraud detection).

The data that we will consider, at least initially, is *record* data, that is data that consists of a set of records, each of which contains a number of fields that hold values. Thus the data naturally forms a table or matrix. Throughout, n will be used to denote the number of records, and so the number of rows of the table or matrix, and m will denote the number of values in each record, which are called *attributes*. The values that a field can hold are often numeric, but there is no intrinsic reason why a field cannot hold a piece of text. So, in the taxation example, some fields, such as income, hold numbers; while other fields, such as occupation, holds strings. (Of course, in this example, all occupations could be given numeric codes and so converted from strings to numbers but this is not always feasible.)

The kind of datasets we are interested in have two properties:

- The datasets are wide, that is they have many attributes. Often they will also have many records as well but, in general, it is the number of attributes that creates the conceptual difficulties.
- The number of attributes reflect the inherent complexity of the data, and so of the system being modelled; rather than arising from a particular choice of representation of the system. For example, one way to represent images is to regard them as a record with one attribute for each pixel. While this can sometimes be useful, the apparent complexity of the representation does not necessarily match the real underlying complexity of the set of images. In other words, it is possible to choose representations that create apparent complexity that isn't really there.

The other critical aspect to the problems we will address is that the expected structure cannot be straightforwardly inferred from the problem domain. Returning to the taxation example, we can certainly imagine some ways in which the data might be manipulated to reduce apparent tax payable. For example, incomes might be altered to appear smaller than they are, and deductions inflated; this is quite natural to suspect and so to look for. A more sophisticated inspection tactic is based on Benford's Law [27] which describes the expected distribution of digits in certain kinds of real-world numbers – for example, the first digit in such a number is much more likely to be a 1 than a 9. When humans make up numbers, for example a deduction that didn't really exist, they tend to choose the first digit much more uniformly. Knowing this, numbers in a tax return can be scored by how unusual they are with respect to Benford's Law.

However, there are presumably many more subtle aspects of the content of tax returns that are even more useful for detecting tax fraud. There is usually no obvious *a priori* way to look for them – and, sadly, intuition about suspicious patterns has sometimes been quite unreliable.

If we cannot find these kinds of alterations in the data, and the implied structure and similarity of records by looking for them explicitly using predetermined models of the underlying mechanisms, how can we find them? What has turned out to be a powerful approach is to build models of the structure in the data *inductively*, that is letting the data reveal its own structure. Once the structure of the data is understood, subsequent questions about unusual parts of this structure become easier to answer.

This is not to say that there is not sometimes a role for *pattern-based* understanding of large datasets. For example, humans have been coming up with ways to alter financial accounts since financial accounts were invented, and so any auditor has a long list of possible frauds and the associated patterns visible in accounting data; tax and insurance investigators do too. The credit card industry is the leading example of this; they have accumulated so many examples of what people do to carry out credit-card fraud, and they turn popular examples into rules that are used to check new transactions. Nevertheless, new forms of creative accounting continue to be discovered, so the set of patterns is always growing and never, at any moment in time, exhaustive. So there is a role for inductive model building and pattern discovery to make sure that unsuspected structures are noticed. In these example settings, not all of these induced structures will be suspicious – but all need to be considered as potentially suspicious until proven otherwise.

1.1 A Natural Representation of Data Similarity

We now turn to the question of how to construct a measure for the similarity of each pair of records that accurately reflects intuitive ideas of similarity between the entities that the records describe. Such a measure should be reflexive (so that a record is similar to itself) and symmetric (so that if A is similar to B then B is also just as similar to A). It is less clear how similarity should behave transitively: if A is similar to B and B is similar to C, then how similar is A to C? Mostly a measure that obeys the triangle inequality is plausible and well-behaved, but this is not an obligatory requirement.

Any dataset has a natural representation, the space spanned by its attributes. Each attribute defines one axis of a space, and each record is represented by a point that is placed at the position corresponding to the values of its attributes. So, for a very simple dataset recording people's heights and weights, there would be two axes, one for height and one for weight, and each person would be represented by a point whose position is determined by the value of their particular height and weight. This is illustrated in Figure 1.1. In this representation, a number of properties of human heights and weights become visible. For example, heights and weights are reasonably well correlated; but the relationship between the two is slightly different for men and women; and the range of heights and weights is different for children and adults. All of these properties can be discovered inductively from the representation.

Unfortunately, real datasets are not often so well-behaved. Some of the problems that arise are:

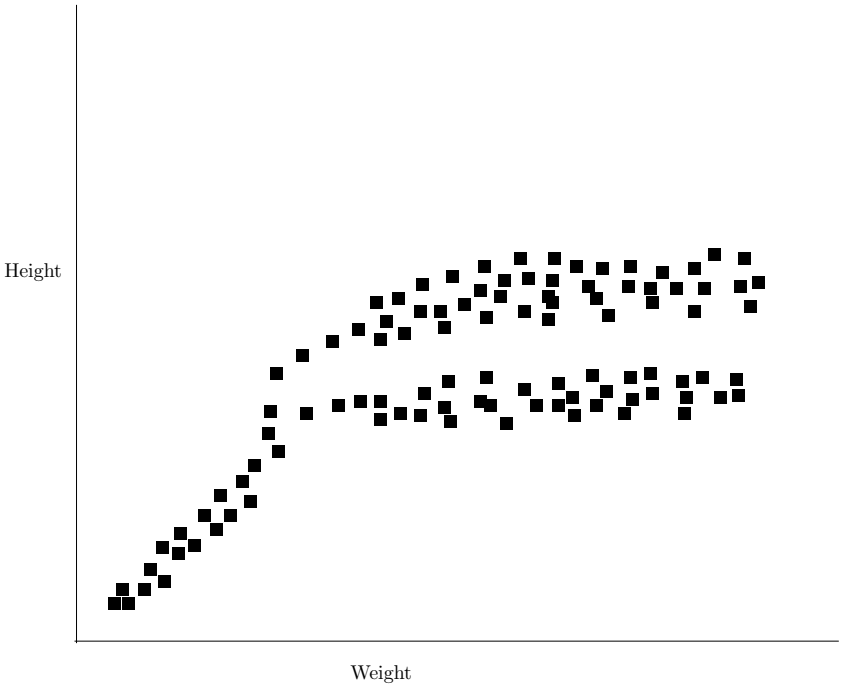

Fig. 1.1 Points derived from height and weight values for a population

- The units in which each attribute's values are expressed make a difference to the apparent similarity, but there is no natural way to choose them;
- The attributes are often collected for reasons unrelated to modelling, so that important ones are missed, irrelevant ones are collected, and some subsets of those collected may be measuring the same underlying property. This also distorts the apparent similarity.
- Algorithms that collect sets of unusually similar records into *clusters* have to be told what clusters look like and (usually) how many clusters there are, but this is often little better than a guess. Hence the results may depend heavily on the parameter choices rather than on the data itself.
- Many clustering algorithms silently cluster records that are not actually very similar, especially lumping small numbers of records in with a larger set of mutually similar records to which they are only modestly similar.

We will return to these issues in subsequent chapters.

There are two levels of understanding the structure of a dataset. The first is to understand where the data is and what it looks like. At its simplest, this could be just applying some clustering algorithm to the dataset, and evaluating the structure associated with the resulting clusters:

- How many clusters are there (but this often requires encoding assumptions about what a cluster is like and how many there should be into the clustering algorithm, so can often be a bit circular)?
- What are the clusters like? Do they have characteristic sizes (the number of records the contain), shapes (in the natural geometry, for example are they spherical or elliptical, or more spider-like), and densities (how similar are the members of each cluster)?

At a more sophisticated level, it might be useful to know if there are records that do not fit into any cluster in a plausible way, and how many of these unusual records there are. Concentrating on such unusual records is called *outlier detection*.

In general, though, a deeper understanding comes from seeing how the clusters and these few unusual records (which could be considered clusters of size one) are related to one another. Figure 1.2 shows two simple clusterings with the same number and size of clusters – but we would clearly consider the datasets described by each to be qualitatively different. This global structure, that includes both the clusters *and* their relationships to each other we will call the *skeleton* of the dataset.

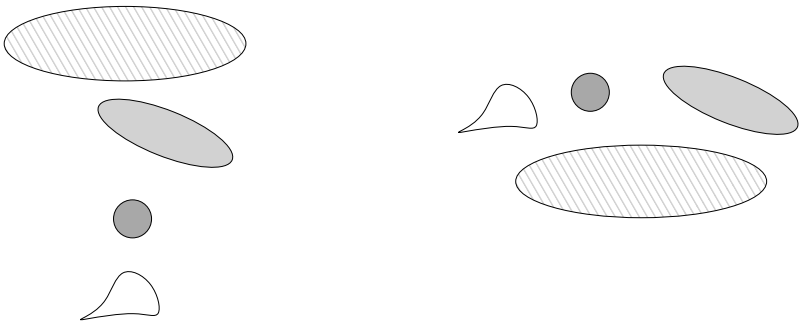

Fig. 1.2 Individual clusters are identical but the clustering is different

A part of understanding the skeleton is to understand what each cluster represents. This is a surprisingly difficult problem. A clustering algorithm or approach considers some set of the records to be similar to one another, but this does not immediately tell us what this mutual similarity is capturing – although we can be confident that, given appropriate attributes and algorithms, it is capturing something real about the system that the dataset describes. Sometimes it is possible to compute the centroid of each cluster which becomes a kind of artificial record that resembles closely all those in the cluster. Examining the values of the attributes of this artificial record can suggest the cluster's "meaning" but this is hit-and-miss in practice.

A neglected part of understanding the structure of a dataset, is to consider the places where data *isn't*, that is to consider the empty space in the natural representation. First of all, understanding the empty spaces is another way of understanding the relationships in the skeleton. For example, a record in empty space can be of vastly different significance depending on *where* in the space it is. A simple exam-

ple is shown in Figure 1.3. The three labelled points (cross, disk, and square) are all far from any of the clusters – and yet we are tempted to regard them as representing records of quite different kinds. Empty space, therefore, is not uniformly bland but rather has a structure all of its own. Some locations in this structure are more significant than others.

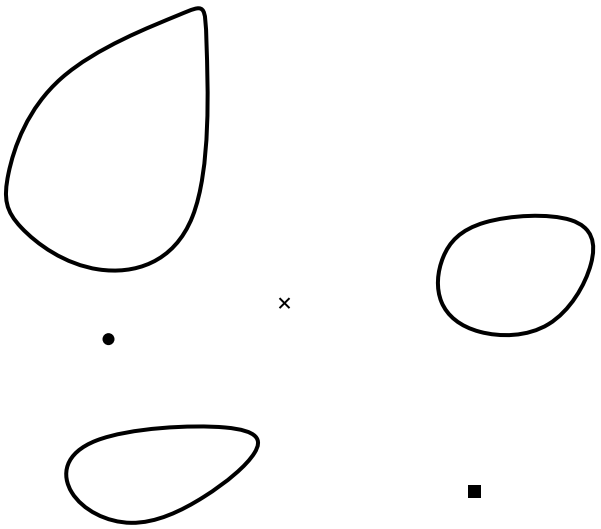

Fig. 1.3 Individual points with different significance

If the problem is to understand the structure of a single, once-for-all dataset, then all we need is this view of almost empty space – it provides a way to categorize isolated points that refines the skeleton structure induced by the clusters.

However, if the problem domain, and so dataset, is one where new data sometimes appears, either via some uncontrolled mechanism or because it is requested, then the structure of empty space becomes much more important and useful. A new record that appears in a space close to or inside an existing cluster brings little or no new information about the real-world system. A new record that appears in empty space, however, introduces new information and its meaning depends on where in the empty space it appears. For example, a new record might suggest that two of the existing clusters are actually parts of the same cluster; or that there is another previously unsuspected cluster; or that the data has a much greater extent than previously realized. Furthermore, if new data can be requested, then the structure of empty space suggests what kinds of new records will be most useful to request.

Overall, we might divide the significance of space into five different categories, depending on what the arrival of a new record in each of the five kind of regions indicates. These five categories are illustrated in Figure 1.4, against a background of a few representative clusters.

The five categories, and their meanings, are:

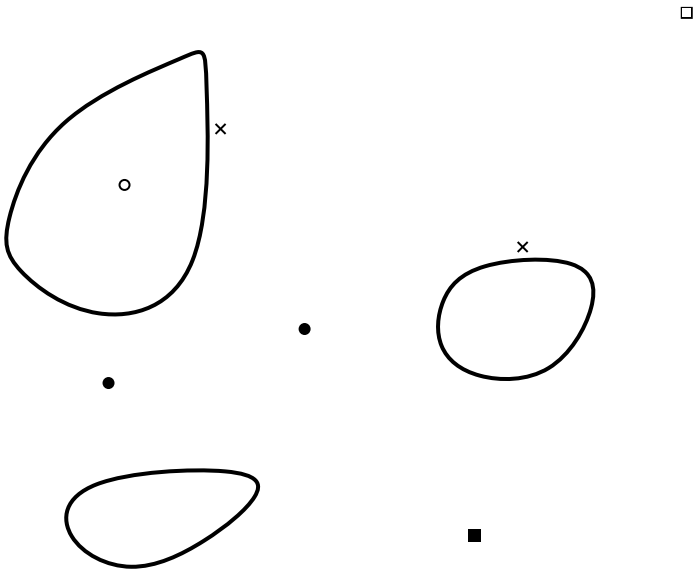

Fig. 1.4 Hierarchy of significance of isolated points, and so the regions they are in

- *normal* records, indicated by circles. These fall within existing clusters and so represent "more of the same".
- *aberrant* records, indicated by crosses. These lie on the outskirts of an existing cluster (whatever that means for a particular clustering technique) but their position is readily explainable by the finiteness of the data used for clustering. They also represent "more of the same" (or perhaps "more of almost the same").
- *interesting* records, indicated by solid circles. These lie in empty space between the existing clusters. Their presence suggests that the current model or understanding of the space is inadequate.
- *novel* records, indicated by solid squares. These also lie in empty space but on the "outside" of the entire clustering. They do not have the same implications about the structure of the model because they are so different from the data from which the model was built. They imply that the data collection was inadequate, rather than that the model built from it was inadequate.
- *random* records, indicated by squares. These lie so far from other data that the suspicion is that something has gone wrong with the data collection rather than that there is previously unsuspected data.

These five categories are a helpful way to understand the structure of empty space. Exactly how the come into play depends on the problem structure, as we shall see.

We have described them as properties of newly arriving records and how these records might be interpreted. These categories can also be understood as descriptions of certain regions of space. In other words, a region might meaningfully con-

sidered an interesting region. This is useful in some situations because these regions can be used to generate requests for the collection of new data that is expected to be particulary revealing. In other words, as well as providing a way to interpret particular kinds of answers, these categories can also define particular kinds of questions.

1.2 Goals

The primary and immediate goal of the kind of analysis we have been talking about is understanding: understanding what clusters are like and what they represent, understanding the relationships among clusters; understanding which records do not really belong to any cluster (or which form a cluster of size 1); and understanding the spaces in between clusters. Such analysis can be revealing about structures in the data and their meaning in the real-world system from which the data came.

However, the more important analysis, with even greater payoff, is to leverage these induced structures to tell us more about the individual records. A space is constructed by using all of the information in all of the records, and the resulting space therefore depends on the integration of all of this information. The structure of the space then becomes background information against which each record can be understood more deeply than just its own contents allows. This is tremendously powerful – the inductive approach produces emergent knowledge that is implicit in the entire dataset but invisible at the level of a single record. This emergent knowledge nevertheless allows deductions about individual records.

One important class of important structures that can be built on understanding of the dataset are *rankings*. In other words, the structure enables the records to be organized in new ways, the simplest of which is a linear ordering of all of the records based on some property of the structure.

A ranking does not require constructing a space from the dataset. The set of records could be ordered (sorted) according to the values of one particular attribute. For example, tax departments might investigate individuals starting with those with the largest income, using the rationale that discovering fraud in such people brings in the greatest amount of extra money. This idea can be extended to any function that combines the attributes; for example, a tax department might compute the sum of income and deductions for each taxpayer and use this to sort the list of taxpayers. Those at the top of the sorted list might be plausible targets for investigation because they make a lot of money; but for two people with the same income the new function ranks the one with the greater deductions higher.

The problem with constructing such a function is that it embodies a kind of pattern that has to be known in advance; someone has to decide which attributes are important, whether they are positively or negatively related to the goal property, and how they should be weighted and combined. These are not easy decisions in most settings.

The advantages of constructing a space and using it as the basis for ranking is that the properties of the space emerge from the properties of the data and so do not

have to be known in advance. In the simplest case, imagine the points corresponding to the records in some space and sweep a plane across the space from one side to the other, inserting each record into an ordered list as the plane encounters that record. If the orientation of the plane is chosen appropriately (a complex issue we will postpone for now) then the ordering will be derived from the data. In fact the plane describes a function on the attributes but the way in which they are combined and weighted has been inferred from the data, rather than constructed beforehand by an expert.

Another useful kind of ranking is from the "middle" of the data to the "outside" (or *vice versa*) where both "middle" and "outside" are intuitively obvious but practically rather difficult. Often the "middle" represents data that are common or "normal" while the "outside" represents data that are anomalous. So in situations where the anomalies are the records that deserve further attention (the tax example, many kinds of fraud, intrusion detection) such rankings focus attention on some of the records, those at one end of the ranking. On the other hand, if the records represent documents, the document closest to the "middle" is somehow the most representative and so might be the place to starting learning about whatever the set of documents describes.

So global rankings are useful because the records at both ends of the ranking are special in different ways, and the rankings allow us to find them and perhaps focus more attention on them.

It is sometimes the case that what is of interest is not a global ranking, but the ranking in the neighborhood of one particular record. As before, it is possible to address this without constructing a global space – in the natural space spanned by all of the attributes, find the closest neighbors of the given point (closest in the sense of, say, Euclidean distance). This has the same drawbacks as ranking without constructing a space – any metric that defines "neighbors" treats all of the attribute differences as of equal importance.

In a constructed space, there are two advantages when trying to find the neighbors of a given point. First, similarity measures take place in a space where the selection and weighting of attributes has been made globally; second, the skeleton makes it computationally easy to choose a smallish set of points that *could be* neighbors and compute similarity to them, rather than having to compute similarity to all neighbors and then discard those who are too far away.

One thorny issue that remains is that of *context*. Everything so far has assumed that the same space will do for every purpose; but often the person doing the analysis has extra domain knowledge or knowledge about the particular dataset that should be accounted for in the analysis. We have argued above that, in general, not enough is known about which attributes are most important and by how much, nor about how these choices depend on the structure in the real world. However, a particular person analyzing the data may know enough to discount a particular attribute, or to know that a particular discovered cluster represents a known problem with data collection, and it would be helpful if there were a way to include this knowledge in the construction of the space. Furthermore, it is often useful to be able to ask "what if?" questions about the data.