Unsupervised and Semi-Supervised Learning
Series Editor: M. Emre Celebi

Y-h. Taguchi

Unsupervised Feature Extraction Applied to Bioinformatics

A PCA Based and TD Based Approach

Second Edition

Springer

Unsupervised and Semi-Supervised Learning

Series Editor

M. Emre Celebi, Computer Science Department, Conway, AR, USA

Springer's Unsupervised and Semi-Supervised Learning book series covers the latest theoretical and practical developments in unsupervised and semi-supervised learning. Titles – including monographs, contributed works, professional books, and textbooks – tackle various issues surrounding the proliferation of massive amounts of unlabeled data in many application domains and how unsupervised learning algorithms can automatically discover interesting and useful patterns in such data. The books discuss how these algorithms have found numerous applications including pattern recognition, market basket analysis, web mining, social network analysis, information retrieval, recommender systems, market research, intrusion detection, and fraud detection. Books also discuss semi-supervised algorithms, which can make use of both labeled and unlabeled data and can be useful in application domains where unlabeled data is abundant, yet it is possible to obtain a small amount of labeled data.

Topics of interest in include:

- Unsupervised/Semi-Supervised Discretization
- Unsupervised/Semi-Supervised Feature Extraction
- Unsupervised/Semi-Supervised Feature Selection
- Association Rule Learning
- Semi-Supervised Classification
- Semi-Supervised Regression
- Unsupervised/Semi-Supervised Clustering
- Unsupervised/Semi-Supervised Anomaly/Novelty/Outlier Detection
- Evaluation of Unsupervised/Semi-Supervised Learning Algorithms
- Applications of Unsupervised/Semi-Supervised Learning

While the series focuses on unsupervised and semi-supervised learning, outstanding contributions in the field of supervised learning will also be considered. The intended audience includes students, researchers, and practitioners.

** Indexing: The books of this series indexed in zbMATH **

Y-h. Taguchi

Unsupervised Feature Extraction Applied to Bioinformatics

A PCA Based and TD Based Approach

Second Edition

Springer

Y-h. Taguchi
Department of Physics
Chuo University
Tokyo, Japan

ISSN 2522-848X ISSN 2522-8498 (electronic)
Unsupervised and Semi-Supervised Learning
ISBN 978-3-031-60981-7 ISBN 978-3-031-60982-4 (eBook)
https://doi.org/10.1007/978-3-031-60982-4

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To all the scientists who have ever written at least one peer-reviewed paper. . . .

Foreword

Machine learning techniques serve as powerful tools in bioinformatics, specifically for predicting the structure and function of proteins, identifying disease causing mutations, biomarkers, potential drug-like molecules, and so on. However, it is not straightforward to relate the features with performance. On the other hand, a simple statistical analysis can provide insights to understand the relationship; for example, increase in long-range contacts slows down the folding of proteins, positive charged residues tend to dominate in DNA binding domains, etc. Hence, linear algebra has the capability to reveal complicated genomic structures in a more direct manner than machine learning.

Almost 10 years ago, Prof. Taguchi and I published a paper on predicting protein folding types using principal component analysis (PCA), one of the liner algebra methods. He has continued his research to investigate the applications of PCA on various biological problems. Recently, he successfully moved to tensors. These methods provide insights to understand the concepts due to the fact that the data are easily interpreted, and "trace back" the output from input features. It is amazing that such a simple strategy can be applied to a wide range of biological problems discussed in this book.

Prof. Taguchi has elegantly designed the book to understand the concepts easily. He has provided mathematical foundations on all important aspects followed by feature extractions. At the end of the book, he shows that PCA and tensors are powerful tools, which perform similar to machine learning techniques in the study of biological problems, namely, biomarker identification, gene expression, and drug discovery, evidenced with his numerous high-quality publications in reputed international journals.

In essence, this book is a valuable resource for students, research scholars and faculty members to simultaneously grasp the fundamentals and applications of PCA and tensors. Although the applications listed in this book are limited to

bioinformatics, the approach is extendable to other fields as well since they are general linear methods, which are easily understandable.

With these appreciations, I recommend this well-written book to the readers.

Chennai, India
25 March, 2019

M. Michael Gromiha

Preface to the Second Edition

Magic is most fun when you're looking for it.
Frieren, Frieren, Season 1, Episode 21

Since the publication of the first edition, we drastically improved the methods described in the first edition. The sections added are 5.7, 5.8, 5.9, 7.10, 7.11, 7.12, 7.13, 7.14, 7.15, 7.16, 7.17, 7.18, 8.1, 8.2, 8.3. In the sections added in Chaps. 5 and 7, we described newly proposed strategy that enables us to treat with more complicated integration of multiple profiles. In the sections added in Chap. 8, we have discussed why proposed method can work well. Based upon the theoretical discussion, we proposed the new strategy where we optimized the standard deviation used in Gaussian distribution by which P values are attributed to the features and the features associated with significant P values are selected. These methods are also implemented in the form of R packages as described in the Appendix C. We hope that this book can help many people who need to solve the problems described in this book.

Tokyo, Japan
March, 2024

Y-h. Taguchi

Preface to the First Edition

He stole something unexpected..., your heart.
Inspector Zenigata, Lupin III: The castle of Cagliostro, movie, Episode 1

This is a book about very classical mathematical techniques: principal component analysis and tensor decomposition. Because these two are essentially based upon linear algebra, one might think that these are no more than text book level matters. Actually, when I started to make use of them for the cutting-edge researches, many reviewers who reviewed my manuscripts complained about the usage of these old-fashioned techniques. They said, for example, "Why not using more modernized methods, e.g., kernel tricks?" or "Principal component analysis is a very old method for which no new findings can exist." In spite of these criticisms, I have continuously published numerous papers where I discussed how principal component analysis or tensor decomposition can be used for data science in a completely new way.

The principal reason why such old techniques can work pretty well is because of the topic targeted: feature selection in large p small n problem. Large p small n problem means that there are huge number of variables of which very small number of observations are available. In such situations, it is of course difficult to know what has happened in the system, because there are not enough number of points that cover the whole state space. This situation is also known as "the curse of dimensionality" which means the lack of enough number of observations compared with the number of dimensions. This problem remains unsolved over long period.

In this book, I apply principal component analysis and tensor decomposition in order to tackle this difficult problem. There are several reasons why these two can work well in this difficult problem. At first, these two are unsupervised methods. In contrast to the conventional supervised methods, unsupervised methods are more robust. Especially, it is free from overfitting that can easily occur when supervised methods are applied to small number of samples with large number of dimensions, because unsupervised methods do not learn from labeling from which supervised methods must learn. Second, unsupervised methods are more stable than supervised methods, because unsupervised methods are independent of labeling. Another advantage of principal component analysis and tensor decomposition is that

they consider the interaction between variables not after the features are selected but before they are selected.

The main purpose of this book is feature selection, which means selecting small or limited number of critical variables among huge number of variables. Although there have been numerous proposals for feature selection, there are very few fitted to apply to the large p small n problems. One typical approach among those not fitted to large p small n problems is a statistical test. When we would like to find features that satisfy some required properties, statistical test can compute the probability that the desired property can appear by chance. If some features are associated with small enough probability, we can regard that the feature is truly associated with this property. In large p small n problem, this strategy often fails. Smaller number of samples can increase the probability that the desired property can happen by chance. On the other hand, if the number of features are large, small probability can happen by chance; if the number of features is as many as 10^4, features associated with the probability as small as 10^{-4} can appear with the probability of 1 (i.e., almost always). Because of the same reason, even if we try to find the features best fitted with the desired property, it might be simply accidental.

The basic idea to resolve these difficulties using principal component analysis and tensor decomposition is as follows. First, before features are selected, whole data set is embedded into lower dimensional space. Because feature selection is performed within this lower dimensional space, it is not a large p small n problem any more. Thus, it is also free from "The curse of dimensionality." Then the dimension in which feature selection is performed is selected with variety of methods fitted to desired properties. As can be seen in the later parts of this book, this simple idea works surprisingly well.

In Chap. 1, I re-introduce basic concepts including scalar, vector, matrix, and tensor, from data science point of views. Chapters 2 and 3 introduce two embedding methods by which dimensions are reduced, principal component analysis as a part of matrix factorization and tensor decomposition, respectively. The following two chapters explain how we can make use of these two for the feature selection by applying them to synthetic data sets. The last two chapters are dedicated to the applications of two methods to bioinformatics where large p small n problems are very usual.

Although the application of the proposed methods is limited to genomic science, because general workframe of the methodologies is very universal, readers are expected to apply these two to their own problems in data science. I am happy to hear from their achievements when the methods proposed in this book are applied to various problems.

Tokyo, Japan Y-h. Taguchi
March, 2019

Acknowledgments

Unfortunately, because I have developed this method almost by my own, I have no persons to which I would acknowledge their contributions. On the other hand, I have many researchers who would like to make use of my own methods for their problems as applications. These researchers include Prof. Yoshiki Murakami who is a medical doctor and wrote many medical papers with me, Prof. Hideaki Umeyama who is a pharmacologist and a specialist about in silico drug design and has performed a university running project with me, and Prof. Mitsuo Iwadate who was once a member of Prof. Umeyama's lab. In addition to this, two Taiwanese professors, Hsiuying Wang at the Institute of Statistics, National Chiao Tung University, and Ka-Lok Ng at the Department of Bioinformatics and Medical Engineering, Asia University, have published a few papers with me. I would like to thank all of the researchers, including these above-mentioned professors, who wrote papers with me using the methods proposed in this book. At last, but not at least, I would like to thank all of my family members, my wife, Tomoko, and sons, Yuu and Koki, for their continuous mental supports for me. Without their help, I could not write this book. In addition to the acknowledgment above when the first edition was written, I would like to add two more persons to be acknowledged. Prof. Turki Turki who has written almost all papers published since the publication of the first edition with me and Prof. Sanjiban Sekhar Roy who has published a few papers with me and edited two volumes with me. Without the contributions from these two, I could not write the second edition at all.

Contents

Acronyms

AD	Alzheimer's disease
ALL	acute lymphoblastic leukemia
ALS	alternating least square
AY	amygdala
BAHSIC	backward elimination using Hilbert-Schmidt norm of the cross-covariance operator
BH	Benjamini Hochberg
BP	biological process
CC	cellular component
CHB	chronic hepatitis B
CHC	chronic hepatitis C
ChIP	chromatin immunoprecipitation
CP	canonical polyadic
DAVID	The Database for Annotation, Visualization and Integrated Discovery
DBTSS	DataBase of Transcriptional Start Sites
DEG	differentially expressed gene
DF	dengue fever
DHF	dengue hemorrhagic fever
DMS	differentially methylated site
DNA	deoxyribonucleic acid
FACS	fluorescence activated cell sorting
FDR	false discovery rate
FE	feature extraction
FN	false negative
FP	false positive
GEO	gene expression omnibus
GO	gene ontology
HC	hippocampus
HDAC	histone deacetylase
HOOI	higher orthogonal iteration of tensors
HOSVD	higher order singular value decomposition

HTS	high-throughput sequencing
KEGG	Kyoto Encyclopedia of Genes and Genomes
KO	knock out
LBDD	ligand-based drug design
LDA	linear discriminant analysis
limma	Linear Models for Microarray Data
LOOCV	leave one out cross validation
MF	matrix factorization
MF	molecular function
miRNA	microRNA
MPFC	medial prefrontal cortex
MSigDB	the molecular signatures database
NASH	nonalcoholic steatohepatitis
NP	non-deterministic polynomial-time
NSCLC	non-small cell lung cancer
OE	overexpression
PC	principal component
PCA	principal component analysis
PTSD	post-traumatic stress disorder
RFE	recursive feature elimination
RGB	red, green, and blue
RNA	ribonucleic acid
RPKM	reads per kilobase of exon per million
SAM	significance analysis of microarrays
SBDD	structure-based drug design
scRNA-seq	single cell RNA sequencing
SD	standard deviation
SE	septal nucleus
SNP	single-nucleotide polymorphism
SOTA	State-of-the-Art
ST	striatum
SVD	Singular value decomposition
TCGA	The cancer genome atlas
TD	Tensor decomposition
TF	transcription factor
TGE	transgenerational epigenetics
TN	true negative
TP	true positive
TSS	transcription start site
UDB	universal disease biomarker
UFF	unsupervised feature filtering
UPGMA	unweighted pair group method using arithmetic average
UMAP	Uniform manifold approximation and projection
UTR	untranslated region
VS	ventral striatum

Part I
Mathematical Preparations

In this part, we briefly introduce mathematical basics required for understanding the content of this book. Most of the part is usually taught in the first grade of undergraduate course of university. Thus, some reader might skip this part. It is tried to reintroduce basic mathematical concept from the data science point of views.

Chapter 1
Introduction to Linear Algebra

None can extinguish souls!
Momo Minamoto, Release the Spyce, Season 1, Episode 12

1.1 Introduction

Linear algebra is composed of simple arithmetic operations: addition, subtraction, multiplication, and division. In spite of their simpleness, it is often powerful enough to represent some complicated data set. In some sense, linear algebra is something like scissors. Although scissors can do only one thing, cutting, it can be used for various purposes if it is used by skilled persons. A piece of paper can be a beautiful art called as a cutting picture that looks like a very complicated sculpture. A skilled hairdresser can use scissors to change a female outlook so beautiful. Likewise, linear algebra can be used to understand very complicated data set that is difficult to understand otherwise, if you can make use of it so as to let it to demonstrate the maximum power. In this chapter, we prepare the knowledge that can be used in the later chapters for the application as data science technology.

1.2 Scalars

1.2.1 Scalars

Scalars are numbers that take real values. In the data science context, scalars are usually numbers that describe samples. Here samples correspond to some objects that will be targeted under the investigation. The examples of pairs of samples and associated scalars are:

Y-h. Taguchi, *Unsupervised Feature Extraction Applied to Bioinformatics*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-031-60982-4_1

- Person and weight
- Food and price
- Star and brightness

Thus, in contrast to the generic algebra, scalars are not always able to be added with each other; brightness cannot be added to price, price cannot be added to weight, and so on. Not only addition, but also division, multiplication, or subtraction are not always possible, either. Arithmetic is possible only between same scalars: brightness plus brightness, weight plus weight. In this sense, data science algebra is more restricted than usual algebra.

In the data science, it is critically important to remember that all scalars analyzed have origins in the real world; no scalars are purely ideal numbers. This is primarily distinct from simple mathematical numbers that do not always have counterpart in the real world. Scalars in data science always represent something that exists in the real world.

Exercise

1.1 List ten pairs of samples and associated scalars.

1.2.2 Dummy Scalars

In contrast to scalars that describe samples, samples are often associated with features that cannot be described with real values. Such examples are color. Although it is possible to artificially attribute real values to colors, e.g., using RGB (red, green, and blue) color model, it is empirically useless. In RGB color system, colors are represented as combinations of three scalars. For example, red corresponds to (1,0,0) and blue corresponds to (0,0,1). Formal addition of distinct colors, e.g., red plus blue, results in completely distinct third colors, (1,0,1), which corresponds to pink. Thus, it does not make sense. More severely, there are generally no ways to add distinct features. What comes if American is added to Japanese (in this case, feature is nationality)? In order to avoid this difficulty, dummy scalars are usually introduced. All features that cannot be described using real values are converted into 1 or 0. If a sample has the feature, corresponding dummy scalar takes 1 otherwise 0. In the example of colors, the number of scalars is as many as the number of colors. If all samples under the investigation can take one hundred colors, we have to prepare same number of dummy scalars and add 1 or 0 to them dependent on color association with each sample. All samples with red have dummy scalar, to which red color is attributed, of 1. Introduction of dummy scalars is critically important since its introduction enables us to deal with any features that cannot be easily represented by real values.

Exercise

1.2 List ten features that must be treated as dummy scalars.

1.2.3 Generating New Features by Arithmetic

Although distinct scalars cannot be added with each other, in the real application we need to generate new features from scalars. In order to perform arithmetic between distinct scalars, multipliers are introduced. Suppose that there are three distinct scalars, x, y and z. In order to enable addition among these, multipliers α, β and γ are multiplied to scalars as αx, βy, and γz. Now, it is possible to add them as $\alpha x + \beta y + \gamma z$. Multipliers have two functions. The first function is to make scalars nondimensional. Nondimensional scalars mean those without unit. For example, if one would like to add weight, price, and brightness, the multipliers of these should have unit of inverse of weight, price, and brightness. Then products of scalars and associated multipliers are nondimensional. In order to perform arithmetic between scalars, introduction of multipliers is essential. The second function of multipliers is to equalize the amount of scalars. If weight is measured in kg, it has values between 0 and 100. If price is defined in Japanese currency, yen, it typically has values between 0 and 1,000,000. Brightness can be measured by various units. If lumen is employed as unit, brightness typically takes values as large as several thousands. Without multipliers, individual contributions of distinct scalars to newly generated feature cannot be balanced. Thus, the introduction of multipliers is required in order to control contributions of scalars to generated feature. Once scalars are multiplied with multipliers, the product of scalars and multipliers can be arguments of any arithmetic functions, e.g., sin and log. Thus, new features can be generated not only by arithmetic but also using functions, e.g., $\log(\alpha x + \beta y + \gamma z)$.

In this context, dummy scalars can also be combined with usual scalars that take real values. In this sense, any of x, y and z can also be dummy scalars. Since dummy scalars are nondimensional, multipliers associated with dummy scalars are also nondimensional.

Exercise

1.3 Generate ten new features using three scalars x, y and z as well as three associated multipliers α, β and γ.

1.3 Vectors

1.3.1 Vectors

Vectors are composed of a set of scalars. For convenience, the elements of vectors are represented by adding suffix to scalars, e.g., x_j, where x is scalar and j is suffix that spans integers. By employing these notations, we are free from introducing the numerous characters to represent a set of many scalars.

In order to be free from representing vectors as a set of many scalars with suffix, we can introduce a vector notation, $\boldsymbol{x}$,

Table 1.1 An example of vector: foods vs. prices

Foods	Prices
Bread	100 Yen
Beef	1000 Yen
Pork	300 Yen
Fish	200 Yen

Table 1.2 Another example of vector: foods vs. weights

Foods	Weights
Bread	200 g
Beef	300 g
Pork	100 g
Fish	150 g

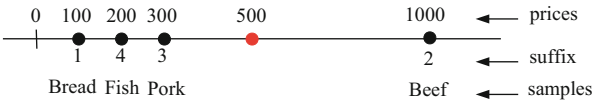

Fig. 1.1 A geometrical interpretation of vector $\boldsymbol{x} = (100, 1000, 300, 200)$. Individual components of the vector that correspond to prices of four samples are considered to be four coordinates of four points aligned along a line. Prices considered to be coordinates are displayed above the line, while suffix that corresponds to four samples is displayed below the line. A red point represents an imaginary sample with the price of 500 yen

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_M), \tag{1.1}$$

where M is the number of samples. In short, it is often represented as $\boldsymbol{x} \in \mathbb{R}^M$. This says that there are M samples, each of which a scalar x_j is attributed to. A typical example of $\boldsymbol{x}$ is that there are M foods, each of which prices are attributed to, e.g., (Table 1.1) where $M = 4$ and $\boldsymbol{x} = (100, 1000, 300, 200)$.

Exercise

1.4 Generate some vectors that represent a set of samples.

It is very usual that samples are accompanied with more than one scalar. For example, we can attribute weights to foods (Table 1.2).

Then, a set of foods is accompanied with additional vector, $\boldsymbol{y} = (200, 300, 100, 150)$.

1.3.2 Geometrical Interpretation of Vectors: One Dimension

It is often very useful to interpret the vectors geometrically. For example, $\boldsymbol{x} = (100, 1000, 300, 200)$ can be considered to be coordinates of four points aligned along a line (Fig. 1.1).

There are several advantages of the geometrical representation of vectors. At first, it can give samples the order that can be easily visually recognized. By simply

glancing the sequence of scalars, it is hard to recognize the rank order of scalars. Second, the distances between samples can be introduced. Then, from the prices, we can say that two pairs of samples, the pair of bread and fish and the pair of pork and fish, are equally separated. If we specifically define measure of distance, say Euclid distance, we can compute the distance between samples numerically as

$$\text{distance between bread and beef} = \sqrt{(100 - 1000)^2} = 900 \tag{1.2}$$

$$\text{distance between bread and fish} = \sqrt{(100 - 200)^2} = 100, \tag{1.3}$$

where Euclid distance between two points j and j' having coordinates of x_j and $x_{j'}$, respectively, can be defined as

$$\sqrt{\left(x_j - x_{j'}\right)^2}. \tag{1.4}$$

Using the numerical distances, we can quantitatively compare two pairs of samples on how far they are apart from each other. In this case, bread is nine times apart from beef than fish. These two points, the definition of rank order of samples and numerical distances between pairs of samples, will turn out to be critical for data science analysis.

An additional advantage of geometrical interpretation is that any points along the line automatically have prices. For example, if a point is placed on the line with the coordinate of 500 yen (a red point in Fig. 1.1), this point represents a sample with the price of 500 yen. This allows us to think about an imaginary sample with this price without specifying what it is. This is also a great advantage for data science, which must predict something unknown. With geometrical representation, we can discuss about samples with arbitrary scalars without specifying what it is. This abstraction is very important as can be seen later.

Exercise

1.5 Draw geometrical representation of Table 1.2.

1.3.3 Geometrical Interpretation of Vectors: Two Dimensions

As denoted in the Sect. 1.3.2, samples can be associated with more than one scalar (Tables 1.1 and 1.2). In this case, geometrical representation must also be altered from a line to a plane. Figure 1.2 shows geometrical representation of four foods according to the scalars shown in Tables 1.1 and 1.2.

Now, using two scalars simultaneously, the relationship among four foods becomes clearer. Beef is apart from other three, because it has the largest weight and highest price. As in the one dimension, any points in the plane are automatically associated with pairs of scalars: prices and weight. A red point in Fig. 1.2 represents an imaginary sample associated with a price of 500 yen and a weight of 250 g.

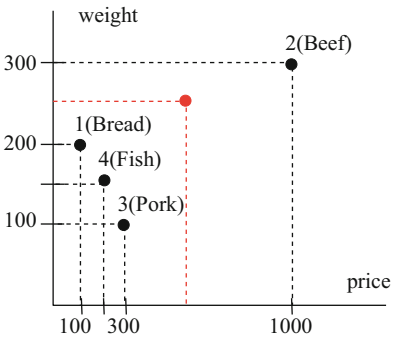

Fig. 1.2 A geometrical interpretation of Tables 1.1 and 1.2. Horizontal axis and vertical axis correspond to prices (Table 1.1) and weights (Table 1.2), respectively. A red point represents an imaginary sample with the price of 500 yen and the weight of 250 g

Table 1.3 Foods vs. prices with using dollar as price

Foods	Prices
Bread	1 dollar
Beef	10 dollars
Pork	3 dollars
Fish	2 dollars

If one thinks that there are nothing unclear, one might miss an important point: scale. In Fig. 1.2, length that corresponds to 100 yen does differ from length that corresponds to 100 g. Nevertheless, there are no reasons to make them equal to each other. When length of 100 yen is made to be equal to 100 g, the plot will be elongated toward horizontal direction. The problem is that there are no criteria to decide scale, since prices can never be related to weight.

One may wonder that it is not a problem, since numerical distance can be defined independent of scale. For example, the Euclidean distance between fish and pork in the plane shown in Fig. 1.2 can be defined as

$$\sqrt{(200-300)^2+(150-100)^2} \simeq 111 \tag{1.5}$$

that is independent of scale.

Exercise

1.6 Compute Euclidean distances of any pairs of samples (points) in Fig. 1.2.

Although it apparently seems to work, it actually does not. Suppose that we use dollar instead of yen for prices. For example, if we can assume that 1 dollar costs 100 yen, Table 1.1 now becomes Table 1.3.

Then, the Euclidean distance between fish and pork is not about 111 but

$$\sqrt{(2-3)^2+(150-100)^2} \simeq 50. \tag{1.6}$$

Now it is clear that there are many problems in two dimensional representations. At first, the distance cannot be determined independent of the unit of scalars. As soon as the foods are imported from Japan to the USA, the distances between foods

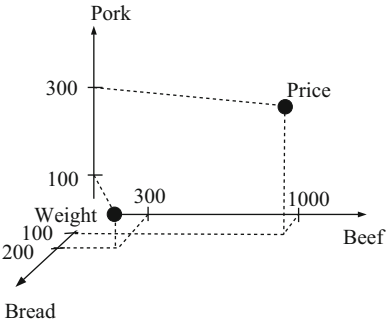

Fig. 1.3 An alternative geometrical interpretation of two vectors $\boldsymbol{x} = (100, 1000, 300, 200)$ for prices and $\boldsymbol{y} = (200, 300, 100, 150)$ for weights. Because of the limitation of the spatial dimension that we can recognize (up to three), the fourth scalars in $\boldsymbol{x}$ and $\boldsymbol{y}$ that represent Fish are omitted

might change. It does not make sense. In addition to this, in the system of dollar-gram unit, the prices are almost ignored on the computing distances. It also does not make sense.

Unfortunately, there are no definite ways to address this problem uniquely. How we should scale different scalars must be decided dependent upon what we would like to know from the data given. It is highly context dependent. Thus, we have to postpone this discussion later when we apply mathematics to real data set.

1.3.4 Geometrical Interpretation of Vectors: Features

In the previous sections, geometrical representations were applied to samples, i.e., four foods. In the Sect. 1.3.1, two vectors $\boldsymbol{x} = (100, 1000, 300, 200)$ for prices and $\boldsymbol{y} = (200, 300, 100, 150)$ for weights were defined, respectively. These two vectors can also be interpreted as a geometrical representation of two features, price and weight (Fig. 1.3). Excluding the omission of fish for easier visual recognition, Figs. 1.2 and 1.3 are mathematically equivalent. In spite of the mathematical equivalence, it is not very popular to interpret vectors as geometrical representation of not samples but features. This is primarily because we have to plot different scalars, i.e., prices and weights, on the common axes. In the Sect. 1.3.3, the ambiguity of scale was pointed out. The problem of scale is more visible in the geometrical interpretation of vectors for features (Fig. 1.3) than that for samples (Figs. 1.1 and 1.2). In the third (vertical) axis in Fig. 1.3 that corresponds to pork, 300 yen is more distant from origin than 100 g. It is apparent that this spatial relationship between price and weight of bread is not informative at all, since as soon as we use dollar (Table 1.3) instead of yen, the price (now it is "only" three dollars) becomes closer to the origin than the weight (100 g). Second, it is not recommended to plot distinct units (in this case, price and weight) along the same axis in physical sciences where this kind of coordinate representation was firstly developed (for example, energy and force can never be plotted on the same axis).

In spite of these difficulties, the emphasis of the equivalence between two geometrical representations (either that of samples or that of features) will turn out to be practically very useful for the main topics of this book.

Exercise

1.7 Draw geometrical representations of prices and weights using combinations of samples distinct from those used in Fig. 1.3, e.g., beef, pork, and fish.

1.3.5 Generating New Features by Arithmetic

As has been down in scalars (Sect. 1.2.3), new features can be generated from vectors, too, e.g., $\alpha \boldsymbol{x} + \beta \boldsymbol{y} + \gamma \boldsymbol{z}$, where α, β, and γ are multipliers similar to the cases in scalars and $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ are vectors. One distinction from generations of new features using scalars is that function must be applied to individual new features generated from scalars. Then, generating new feature with applying a function to vector should be denoted as like $\log(\alpha x_i + \beta y_i + \gamma z_i)$, which corresponds to the ith scalar that consists of new features in the form of vectors.

Exercise

1.8 Generate new features in the vector form, using scalars shown in Tables 1.1 and 1.2 with arbitrary multipliers (and if possible, with applying functions to scalars).

1.3.6 Dummy Vectors

As features that cannot be described with real values were treated as dummy scalars, vectors can also be composed of dummy scalars. In some sense, dummy scalars themselves could be interpreted as vectors. For examples, three colors in RGB representation, (1, 0, 0), (0, 0, 1) and (1, 1, 0), can be now geometrically interpreted in three dimensional vectors that consist of three integer scalars. They are also geometrical representations of features introduced in Sect. 1.3.4. Thus, from this point of views, i.e., unified treatment of dummy scalars with usual scalars that can be treated as real numbers, introduction of geometrical vector representation of features is critical, although it is rarely emphasized in the text books that introduce data science.

In the later part of this book, we try to select a part of features from all features for the practical reasons. Colors represented in geometrical vector representation are very useful for this purpose, since these allow us to select, for example, only the first

scalars of RGB representations. Such a decomposition of colors never be possible without vector representations.[1]

On the other hand, in contrast to vector representation of scalars that can be represented as real values, dummy vectors can be placed only at grid points whose coordinates are composed of integer. Of course, as can be seen in RGB representation of colors, dummy scalars are often allowed to be extended to take real values as well ((0.5, 0.5, 0) can make sense in RGB representation of colors), it is not always true. For example, if the dummy scalars represent whether sample is book, chair, or stick, although dummy scalars can be represented as (1, 0, 0), (0, 1, 0), (0, 0, 1), (0.5, 0.5, 0) does not make sense at all, since (0.5, 0.5, 0) means a sample associated with a feature composed of 50% book and 50% chair.

In contrast to vectors that can be represented as real numbers, e.g., prices and weights, not all points in the geometrical representation of dummy scalars do not have anything real. For example, the dummy vector that represents if a sample is book, chair, or stick cannot take (1, 1, 0) since no samples cannot be book and chair simultaneously.

Exercise

1.9 Think about dummy vectors assuming some.

1.4 Matrices

As vectors are composed of scalars, matrices, X, are composed of vectors, as

$$X = \left(\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \ldots, \boldsymbol{x}_M^T \right), \tag{1.7}$$

where M is the number of features, e.g., price, weight, and color. $\boldsymbol{x}^T$ represents transposition of a vector $\boldsymbol{x}$ where

$$\boldsymbol{x}_j = \left(x_{1j}, x_{2j}, \ldots, x_{Nj} \right) \tag{1.8}$$

corresponds to the vector of ith feature (M is the number of samples). When prices in Table 1.1 and weights in Table 1.2 are represented as matrix, it should be Table 1.4. In this case, a matrix X is

$$X = \begin{pmatrix} 100 & 1000 & 300 & 200 \\ 200 & 300 & 100 & 150 \end{pmatrix} \tag{1.9}$$

[1] Practically, employing only the first scalars in RGB representation is equivalent to the usage of red sunglass through which only red color can penetrate. Now, colors are transformed to real values that describe red color intensity of colors, although in this example only integers are allowed since colors are treated as example dummy scalars.