DEPARTAMENTO DE INTELIGENCIA ARTIFICIAL

Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid

PhD THESIS

Developments in probabilistic graphical models, circular distributions and theory of random forests with applications in neuroscience

Author

Pablo Fernández-González
MS Artificial Intelligence

PhD supervisors

Pedro Larrañaga
Professor of Computer Science and Artificial Intelligence

Concha Bielza
Professor of Statistics and Operational Research

2019

A mi madre, Inmaculada González Álvarez

Acknowledgements

I want to thank my thesis supervisors, Pedro Larrañaga, and Concha Bielza, for their efforts aiding in the completion of this dissertation. With them I have learned and grown a lot during these years of intense work.

I convey my gratitude as well to all members of the Computational Intelligence Group at the Technical University of Madrid, for creating a welcoming, inspiring and fruitful work environment.

Finally, I wholeheartedly thank my family and friends, for their support and positive influence during the good and bad times that crossed my path. Without their help I might not have reached this day.

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness through the Cajal Blue Brain (C080020-09; the Spanish partner of the Blue Brain initiative from EPFL), TIN2013-41592-P and TIN2016-79684-P projects, by the Regional Government of Madrid through the S2013/ICE-2845-CASI-CAM-CM project and by the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 604102 (Human Brain Project).

Abstract

In artificial intelligence, the discipline of machine learning has emerged as the flagship of the field of study. The era of big data, where increasingly large amounts of data are available to the public, requires of tools that summarize and manipulate it correctly. For this reason, substantial effort is invested nowadays in the development of new methods for learning and detecting patterns in the data. In this environment, techniques such as Bayesian networks and random forests enjoy success at a practical level. However, theoretical developments for the field in general and for many methods in particular are less abundant than desired, and the general consensus is still that we do not understand many aspects of why the best performing algorithms work. In this dissertation, we explore both the theoretical and practical branches of machine learning with a multi-focused approach that spans across various technologies.

In the purely theoretical side, we cover contributions to two branches: pure statistics and the theory of random forests.

In the first case we develop the univariate and bivariate truncated von Mises probability distributions for circular statistics. These distributions can be understood as a generalization of the well-known von Mises distribution that implies the addition of two or four new truncation parameters in the univariate and, bivariate cases, respectively. The contributions include the definition, properties of the distribution and maximum likelihood estimators for the univariate and bivariate cases. Additionally, the analysis of the bivariate case shows how the conditional distribution is a truncated von Mises distribution, whereas the marginal is a generalization of the non-truncated marginal distribution. We also show its performance modeling data of leaf inclination angles.

In the second case we tackle the problem of random forests for regression expressed as weighted sums of datapoints. We study the theoretical behavior of k-potential nearest neighbors under bagging and obtain an upper bound on the weights of a datapoint for random forests with any type of splitting criterion, provided that we use unpruned trees that stop growing only when there are k or less datapoints at their leaves. Moreover, we use the previous bound together with the new concept of b-terms (i.e., bootstrap terms), to derive the explicit expression of weights for datapoints in a random k-potential nearest neighbors selection setting, a datapoint selection strategy that we also introduce, and build a framework to derive other bagged estimators using a similar procedure. Finally, we derive from our framework the explicit expression of weights of a regression estimate equivalent to a random forest regression estimate with random splitting criterion and demonstrate its equivalence both theoretically and practically.

For the practical branch of this dissertation, we have two remaining works: A statistical analysis that uses the previously defined truncated von Mises distribution and a multidimensional Bayesian network classifier. In both cases, we study neuronal data in an effort to gain insights of neuroscientific value.

For the first work, we analyze branching angles of the basal dendrites of pyramidal neurons of layers III and V of the human temporal cortex. For this, we use the truncated von Mises distribution, showing that is able to describe more accurately the dendritic branching angles than previous proposals. Then, we perform comparative studies using this and other statistical methods to determine similarities and/or differences between branches and branching angles that belong to different cortical layers and regions, among other comparisons.

Finally, a class-bridge decomposable multidimensional Gaussian network is presented as an interpretable and high-performing model, to account for the morphological differences that exist between

different neurons when varying the species, gender, brain region, cell types and developmental stage of the animal of origin, and to tackle the problem of inference complexity in multidimensional classifiers. This work includes a structural learning algorithm that, for continuous nodes and discrete features, makes use of the CB-decomposability property to alleviate the inference complexity and uses it to learn topologically unrestricted complex network structures that take into account relationships between classes. The model is trained with data from NeuroMorpho (v5.7) and it is then used for accurate prediction of all classes simultaneously for new examples and, given its interpretability, to extract knowledge at a neuroscience level.

Resumen

En inteligencia artificial, la disciplina del aprendizaje automático se ha instaurado como el buque insignia del campo de estudio. La era del Big data, en la que volúmenes cada vez mayores de datos son accesibles por el público general, requiere de herramientas que sean capaces de concisarlos y manipularlos correctamente. Por este motivo, en la actuallidad se están invirtiendo notables esfuerzos para el desarrollo de nuevos métodos para el aprendizaje y detección de patrones en los datos. En este entorno, técnicas como las redes bayesianas y los bosques aleatorios atesoran éxito a nivel de aplicación. Sin embargo, desarrollos teóricos para el campo en general y para muchos métodos en particular son menos abundantes, y el consenso general es que aún no entendemos muchos aspectos de porqué funcionan los mejores algoritmos. En esta disertación, exploramos tanto la vertiente teórica como la práctica del aprendizaje automático con un enfoque multienfático que cubre varias tecnologías.

Para la vertiente más teórica, nuestras contribuciones abarcan dos ramas: Estadística pura y teoría de bosques aleatorios.

En el primer caso desarrollamos la distribución de probabilidad circular von Mises truncada univariante y bivariante. Estas distribuciones pueden ser entendidas como una generalización de la conocida distribucion von Mises, que implica la adicción de dos o cuatro nuevos parámetros en el caso de la univariante o bivariante, respectivamente. Las contribuciones incluyen la definición, propiedades de la distribución y estimadores de máxima verosimilitud para los casos univariante y bivariante. Adicionalmente, el análisis del caso bivariante muestra cómo la distribución condicionada es una distribución von Mises truncada, mientras que la marginal es una generalización de la marginal no truncada. También mostramos su rendimiento a la hora de modelar datos sobre los ángulos de inclinación de las hojas.

En el segundo caso abordamos el problema de bosques aleatorios para regresión expresados como sumas de puntos. Estudiamos el comportamiento teórico de los k-vecinos potenciales más cercanos bajo agregación de muestras *bootstrap* (*bagging*) y obtenemos una cota superior en los pesos de un punto para bosques aleatorios equipados con cualquier tipo de regla de corte (*splitting criterion*), si utilizamos árboles sin poda que dejan de crecer cuando hay k o menos puntos en sus hojas. Además, utilizamos la cota anterior junto con el nuevo concepto de b-terms (o términos de *bootstrap*) para derivar expresiones explícitas para los pesos de puntos del selector aleatorio de k-vecinos potenciales más cercanos, una estrategia de selección de puntos que también introducimos, y para construir un marco de trabajo que nos permite derivar otros estimadores que utilizan agregación de muestras *bootstrap* mediante un procedimiento similar. Finalmente, derivamos la expresión explícita de los pesos de un estimador de regresión equivalente a un estimador bosque aleatorio para regresión equipado con una regla de corte aleatoria y demostramos su equivalencia tanto a nivel teórico como práctico.

Para la vertiente más práctica de esta disertación, desarrollamos dos trabajos: Un análisis estadístico que emplea la distribución von Mises truncada anteriormente definida y un clasificador multidimensional con redes bayesianas. En ambos casos, estudiamos datos neuronales en un esfuerzo por adquirir conocimiento de valor neurocientífico.

Para el primer trabajo, analizamos ángulos de bifurcación de dendritas basales de neuronas piramidales de las capas III y V del cortex temporal humano. Para ello, utilizamos la distribución von Mises truncada, mostrando que es capaz de describir con mayor precisión los ángulos de bifurcación dendritica que anteriores propuestas. A continuación, realizamos estudios comparativos utilizando éste y otros métodos estadísticos para determinar similitudes y/o diferencias entre ramas y ángulos de bifurcación

que pertenecen a diferences capas corticales y regiones, entre otras comparativas.

Finalmente, presentamos un classificador gaussiano multidimensional clase-puente descomponible (*class-bridge decomposable multidimensional Gaussian network classifier*) como un modelo de alto rendimiento e interpretable, para procesar las diferencias morfológicas que existen entre diferentes neuronas cuando variamos la especie, el género, la región del cerebro, el tipo de célula y el estado de desarrollo del animal de origen, así como para tratar de avanzar en la resolución del problema de la complejidad de inferencia en clasificadores multidimensionales. Además, este trabajo incluye un algoritmo de aprendizaje de estructura que hace uso de la propiedad clase-puente descomponible para aliviar la complejidad de inferencia, que usamos para aprender estructuras de redes complejas no limitadas topológicamente que tienen en cuenta relaciones entre diferentes clases. El modelo es entrenado con datos de NeuroMorpho (v5.7) y después es utilizado para realizar predicciones precisas de todas las clases simultáneamente para nuevas muestras y, dada su interpretabilidad, para la extracción de conocimiento en neurociencia.

Contents

Acronyms

AIC Akaike information criterion

Bagging Bootstrap aggregating

BAN Bayesian augmented naive Bayes

BE Bayesian estimation

BIC Bayesian information criterion

CART Classification and regression trees

CBBP Cajal Blue Brain Project

CB-MGC Class-bridge decomposable multidimensional Gaussian classifier

CPT Conditional probability table

DAG Directed acyclic graph

DT Decision tree

GBN Gaussian Bayesian network

HBP Human Brain Project

k-PNN k-Potential nearest neighbors

MBC Multidimensional Bayesian network classifier

MGNC Multidimensional Gaussian Bayesian network classifier

MLE Maximum likelihood estimation

MSE Mean squared error

NB Naive Bayes

PGM Probabilistic graphical models

RF Random forest

TAN Tree augmented naive Bayes

TvM Truncated von Mises distribution

UPM Universidad Politécnica de Madrid

vM von Mises distribution

Part I

INTRODUCTION

Chapter 1

Introduction

In this dissertation we present works in multiple fields of machine learning. Our works travel from pure statistics to machine learning theory, with stops in algorithmic developments and statistical analysis. The chosen approach emphasizes the importance of multiple perspectives when analyzing statistical phenomena.

Directional statistics (Mardia [1975]) (also called circular statistics) is the field of study in statistics that concerns itself with observations of angular nature, or more generally, that include a periodicity property. For example, time and angular measurements (such as the first rain of the year, or the direction of the wind) require a different theory than classical statistics if we are to work with them in a similar manner to linear data. In neuroscience, circular measures arise when considering the branching angles of dendritic trees in neurons. In Bielza et al. [2014], the von Mises distribution (vM), a circular probability distribution, was used to model this phenomenon. In this dissertation we study the field of directional statistics and improve upon the proposal of the von Mises distribution by developing the truncated von Mises distribution (TvM), a more general alternative to the former that adds two parameters for limiting the support of the distribution. We then use this distribution in the field of neuroscience to model dendritic branching angles in humans.

Bayesian networks (BN) (Pearl [1988]) are a probabilistic knowledge representation framework that allows us to capture the conditional independence relationships that exist between variables of a domain, and build models that can perform multiples forms of reasoning on queries over those variables. Their strengths with respect to other proposals in machine learning are interpretability, the ability to sample from the learned distribution (generative model), handle hidden variables and perform well in tasks such as classification of newly found examples. Of the multiple approaches to build Bayesian network classifiers, the multidimensional approach to classification (Bielza et al. [2011]) has been considerably less studied than the single variable case (Minsky [1961]). The main caveat of multidimensional classification with Bayesian networks is the inference complexity, that scales exponentially with the complexity of the network's topology and the number of variables. For this reason, topological restrictions in structure are common (Bielza et al. [2011]). In this work we contribute to alleviate the complexity of building topologically unrestricted multidimensional Bayesian network classifiers (MBCs), with the introduction of a learning algorithm that, for continuous feature nodes and discrete classes, increasingly progresses from a collection of simple structures to the fully connected non-restricted case.

Finally, random forests (RF) (Breiman [2001]) are generally considered one of the best performing

techniques available in machine learning today. In the standard case, they are an ensemble of decision trees used for classification and/or regression problems. Each tree is trained with different versions of the data and on different subsets of the features, to produce multiple individual predictions that are finally combined to output a prediction. RFs display excellent accuracy and are relatively fast to train and use for an ensemble. They also enjoy a history of successful practical applications (Criminisi et al. [2012], Boulesteix et al. [2012]). However, theoretical efforts to fully characterize RFs have so far not uncovered a deep understanding of the model, and works often settle to analyze simplified versions of the original algorithm (Biau and Scornet [2016]). In Lin and Jeon [2006], a very important connection was unveiled between RFs and weighted k-potential nearest neighbors (k-PNN), a special type of nearest neighbors. But bootstrapping was not considered there. In this dissertation we present a step forward in the understanding of RFs by providing for the first time to the best of our knowledge, explicit calculations of the weighting schemes that a complete RF (bootstrapping included) is equivalent to. Moreover, we develop a framework for the calculation of these weights for the class of regression estimates that implement point selection strategies (such as selecting the k nearest neighbors associated values for predictions) that operate strictly within the k-PNNs of the prediction target.

1.1 Hypotheses and objectives

1.1.1 Hypotheses

We have the following hypotheses for this dissertation:

1. A truncated directional probability distribution can be used to model angular phenomena that occur in a restricted sector of the circle.
2. Neuron's branching patterns in humans can be properly modeled using a directional probability distribution that it is not forced to assume symmetry and full support on the circle.
3. Multidimensional Gaussian network classifiers can benefit from the CB-decomposable property and Gaussian nodes to produce highly complex and interpretable multidimensional classifiers.
4. A general random forest algorithm for regression can be expressed as a weighted sum of datapoints.

1.1.2 Objectives

The previous hypotheses are addressed with the following set of objectives:

1. To develop the truncated von Mises distribution, an extension to the von Mises distribution that incorporates the ability to restrict the support of the distribution and to produce non-symmetrical densities.
2. To apply the developed truncated von Mises distribution to real dendritic branching angles data from humans and achieve a superior modeling performance to that of the unrestricted von Mises distribution.
3. To build a multidimensional Gaussian network classifier with a structural learning algorithm that makes use of the CB-decomposable property that achieves the desired level of network complexity and interpretability with respect to the state of the art.

4. To develop the theory and methodology required to express a general random forest for regression as a weighted sum of datapoints. Then, provide a practical demonstration of this equivalence by building an estimator as a weighted sum of datapoints that behaves similarly to a random forest.

1.2 Document organization

This work is subdivided in five parts and nine chapters, following this summary:

Chapter 1

The reader can find here the objectives and hypotheses that produced the research leading to this dissertation. Followingly, the content of subsequent chapters is described.

Background

In this part we cover the relevant concepts and developments in literature that support this work. We trace back our knowledge dependencies to some levels prior to the production of our developments. Therefore, we cover from the basic description of some techniques to their state-of-the-art form in nowadays literature. It includes Chapters 2-4.

Chapter 2

We cover in a summarized way the field of directional statistics from its basic conception to a more informed position. We start with some essential statistics that are reformulated to fit the circular paradigm, then we introduce the von Mises distribution, give an interpretation of its parameters and discuss its properties, maximum likelihood estimation, characteristic function and moments.

Chapter 3

Here we introduce probabilistic graphical models. We introduce Bayesian networks from their historical development to more modern works. We formally define the model and discuss various aspects of defining, building and using the model. Namely, we describe the parameters and structure as the two elements that complete the definition of a Bayesian network and outline its importance and its precise role within the model. We then discuss inference as a query answering paradigm with multiple interpretations and objectives. Followingly, we detail the process of learning a Bayesian network and discuss some consolidated approaches in literature to attain this goal, as well as provide with references to important contributions. Finally, we assess the particular case of Bayesian networks for classification and detail both the unidimensional and multidimensional classification approaches and the relevant works that develop them.

Chapter 4

We introduce the reader to the ensemble approach for classification. We first discuss the idea of combining weak classifiers to produce a stronger one and review the literature that shows its growth from a

question to a subfield of study. We detail bagging as a technique of interest for the combination of classifiers and then explain two specific classifiers that are of interest for this dissertation: nearest neighbor classifiers and decision trees. We then cover the ensemble of decision trees, random forests, from their definition to their very important impact in literature.

Contributions

This part covers the contributions of this thesis, developed along Chapters 5-8.

Chapter 5

Here we present and develop the truncated von Mises distribution in the univariate and bivariate cases. This development consists of a series of results from definition to properties, maximum likelihood estimation and moments. For the bivariate case, we also discuss the conditional truncated von Mises distribution and the marginal truncated von Mises distribution. Finally, we present a real data study for leaf inclination angles using our proposed circular distribution.

Chapter 6

In this chapter we perform comparative studies of dendritic branching angles of pyramidal cells in the human cerebral cortex. We first discuss the methods that we employ, which can be separated into two categories: statistical tests and probability distributions. In the latter case, we use the truncated von Mises distribution introduced in Chapter 5. Our results follow with a set of comparative studies that examine the data from different perspectives. These perspectives are obtained by separating the data in comparative groups according to different criteria. Finally, we present our conclusions and discuss our findings.

Chapter 7

In here we develop the structural learning algorithm for the CB-decomposable multidimensional Gaussian classifiers. We first introduce the formalism and the key property this algorithm exploits: the CB-decomposability. We then present our structural learning algorithm in three steps of incremental network complexity. We then apply our procedure to train and test a model that captures the morphological differences between neurons and draw our conclusions.

Chapter 8

This chapter contains the theoretical developments that solve the problem of expressing a random forest model as a weighted sum of datapoints. We first familiarize the reader with the problematic and pending problems and the concept of k-potential nearest neighbors. Then our analysis shows the effect of bagging on a regression estimate equipped with a 1-PNN distance metric and discusses its differences and similarities with the 1-NN regression estimate. We continue with the main analysis of this work where we show how to obtain explicit expressions for the weights of the class of regression estimates that use for prediction a selection of datapoints strictly within the k-PNN set of the target. We also provide two particular cases of explicit weight calculation; the first is a regression estimate directly defined over the

k-PNN set, and the second is a regression estimate equivalent to a RF equipped with random splitting criterion. For this latter regression estimate, we produce practical results comparing it with a classically implemented version of the RF equipped with random splitting criterion, showing that the predictions of both are virtually identical.

Conclusions

In this last part, we summarize our work and present our conclusions. It comprises Chapter 9.

Chapter 9

We summarize the contributions contained in this dissertation and show the list of derived publications. Finally, we discuss future work and open lines of research that emerged from our research efforts.

Appendices

Appendix

It contains proofs of our results. Specifically, in Chapter 5 we have Lemma 5.2.1 for the analytical expression of the normalization constant in the truncated von Mises distribution, and Theorems 5.3.1 and 5.3.2 to account for the behavior of the truncated conditional and marginal distributions, respectively. In Chapter 8, we have Lemma 8.3.1 to account for the behavior of k-PNN under bootstrapping, Theorem 8.3.1 and Lemma 8.4.1 to establish the concept of bootstrap weights, Lemma 8.4.2 to calculate the numerical value of a b-term, Theorem 8.4.2 to calculate the explicit expression of the weights of the random k-PNN selection regression estimate, and Theorem 8.4.3 to write the random k-PNN selection regression estimate using the explicit expression of the weights obtained in Theorem 8.4.2.

Part II

BACKGROUND

Chapter 2

Directional statistics

2.1 Introduction

Directional statistics is a particular case of the statistical theory and methodology where the format of the observations meets the particular requirement of having a vectorial representation of fixed length (one by convention). It was first developed as such by Kanti V. Mardia (Jupp and Mardia [1989]) to properly handle circular and/or spherical observations, whose properties are not correctly addressed by conventional statistics. Kanti V. Mardia and Peter E. Jupp can be considered the essential authors and the main specialists in the field gathering a number of additional contributions such as Mardia and Jupp [2000].

All possible vectors of a fixed length in an n-dimensional space conform an n-dimensional sphere of that fixed radius. Distributions can be drawn out of the different configurations at which we can find the observations to be given as well as apply many other statistics to describe them. Directional statistics is also referred to as circular statistics as the unidimensional case conforms a circular space and then a circular observation can be regarded as a point in the perimeter of the circle. Circular distributions arising in this reformulation of classical statistics can easily appear as proper distribution models for a variety of phenomena in the application domain. Most classical examples include measurements of wind directions from a stationary point, time measurements where we are interested in the positions of the clock's hands rather than the absolute time, compass measurements, angles that javelin throwers produce respect to the ground line, and many others.

Circular statistics can be considered a transformation from classical statistics where the observations on the perimeter of a circle contrast with the infinite line of the classical approach. We will define the points in the perimeter of a circle of radius 1 (and refer to them from now on simply as points in the circle, unless stated otherwise) as the $\mathbb{O}$ set, which we can express in a Cartesian coordinate bi-dimensional space as $\mathbb{O} = \{(x, y) \in \mathbb{R}^2 \text{ such that } x^2 + y^2 = 1\}$ and use the classical $\mathbb{R}$ real set for the line.

When analyzing the points in the circle, a fundamental difference between both spaces ($\mathbb{R}$ and $\mathbb{O}$) is clear under observation: The circle space has a close perimeter, as it could be viewed as a line whose two extrema are connected, or differently said, the circle comprises a closed shape inside its perimeter. This fundamental difference allows the representation of periodic functions in a natural way and also implies the insufficiency of the classical statistics to compute correctly circular data and/or to summarize and

describe the observations properly.

2.1.1 Coordinate systems and the limitations of classical statistics

Points in the circle need to be represented and referred properly in $\mathbb{O}$. If we were to address the problem with unidimensional Cartesian coordinates, and attempt to address the fundamental difference by

$$x_w = x \mod 2\pi,$$

(where x_w denotes a wrapped variable), restricting our values to 2π with the modulus periodicity, we may find that the linear statistics used to summarize and describe our data fail to calculate the expected solution. As an example, problems may arise when trying to obtain a point that is at distance d from another. In the circle, the shortest path between two points is defined through the circumference with no distinction between the point we consider the reference and any other. Thus, if we compute the distance between $\frac{2\pi}{9}$ and $\frac{(2\pi)8}{9}$ (in radians), our linear statistics distance expression would calculate:

$$\left| \frac{(2\pi)8}{9} - \frac{(2\pi)}{9} \right| = \frac{(2\pi)7}{9},$$

yielding an incorrect solution since we were expecting to obtain $\frac{(2\pi)2}{9}$ (see Figure 2.1).

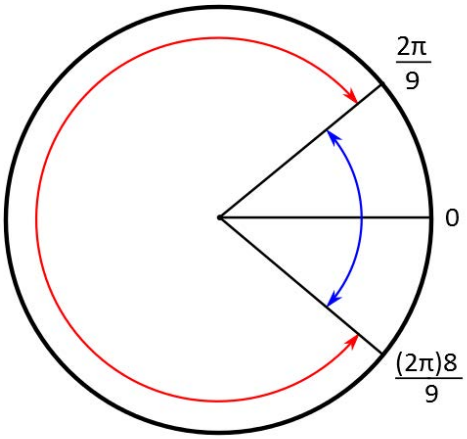


Figure 2.1: In radians, the incorrect distance of $\frac{(2\pi)7}{9}$ that the classical mean computed (red) compared to the correct solution of $\frac{(2\pi)2}{9}$ (blue).

This problem appears under the special consideration that the 0 value has, as it is considered to be "the beginning" of a circle. This example not only suggests that the distance notion has to be rewritten but also shows how classical Cartesian coordinates are not directly compatible with the notion of circle.

Further extending the drawbacks of the classical approach, another example arises when e.g., computing the sample mean of a set of observations. Let us consider a set of three observations $\theta_1 = 30°, \theta_2 =$

$0°, \theta_3 = 330° \in \mathbb{O}$ (in degrees) and use the classical sample mean $\hat{\mu}$

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n}\theta_i.$$

Here we obtain $\frac{(30°+0°+330°)}{3} = 120°$ (see Figure 2.2). The result given by the classical mean again does not acknowledge the closed nature of the circle. In the circle $0° = 0° + 360°k, k \in \mathbb{Z}$, so it is possible to say with care (specifying the k periodic values in both expressions) that $0° \geq 330°$ or otherwise exposed, $330°$ has a difference of $30° + 360°k$ with respect to $0°$ that is not acknowledged by the classical mean, thus yielding an incorrect result (it treats the circle as if it was cut at $0°$).

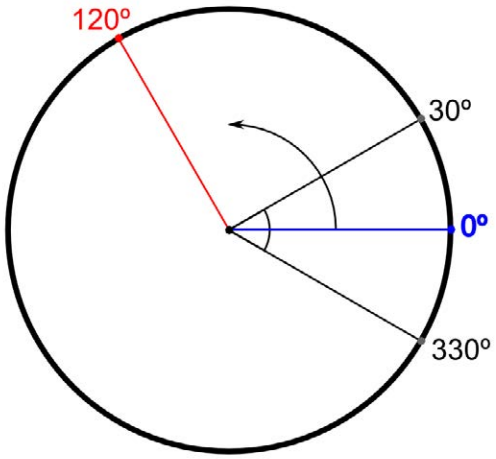


Figure 2.2: The incorrectly calculated mean of $0°, 30°$ and $330°$ using standard statistics (red) compared to the correct solution (blue).

We need therefore a coordinate system that will naturally address the properties of $\mathbb{O}$ over which we can define the statistics to properly describe and summarize our data.

The solution was found to be to consider the points in the circle as vectors of modulus one in $\mathbb{R}^2$ and refer to them by the angle they create with respect to a preferred angle and orientation, that is, using polar coordinates. Unless otherwise stated, points on circular statistics and on the $\mathbb{O}$ set are to be regarded as angular values.

Equipped with those considerations we can finally redefine the Cartesian coordinates to its circular analogue by means of:

$$\mathbf{x} = (\sin\theta, \cos\theta),$$

where θ is the angle created with respect to the initial direction and a reference angle that needs to be specified. Note that despite the representation uses a 2-dimensional coordinate system, the interdependence of the coordinates created by the use of only one argument (θ) prevents it to cover every point in the plane, and by means of the angular trigonometrical representation the set of covered points results to be only the allowed $\mathbb{O}$ perimeter set. We can see this by increasing the θ value and observing how the specified points under the coordinate system are "drawing" $\mathbb{O}$ and only $\mathbb{O}$. Also, it needs to be noted how periodicity is now naturally handled (as expected by definition) and how now $\forall\theta_1, \theta_2 \in \mathbb{O},\ \theta_1 + \theta_2 \in \mathbb{O}$, that is, we have a closed operation with respect to the $\mathbb{O}$ set as well as all the well known properties that

operations between angles satisfy in $\mathbb{O}$.

More formally, if we consider the new coordinate system as an embedding function C we have that $C : \mathbb{R} \to \mathbb{O}$, that is, C "shrinks" the $\mathbb{R}$ line (as we are referring to one dimensional quantities) into the subset of the points that belong to the circle in $\mathbb{O} \in \mathbb{R}^2$.

Another proposal is to regard the points in the circle's perimeter as complex numbers of the form: $z = e^{i\theta} = \cos\theta + i\sin\theta$ (see Figure 2.3). Both notations are commonly used.

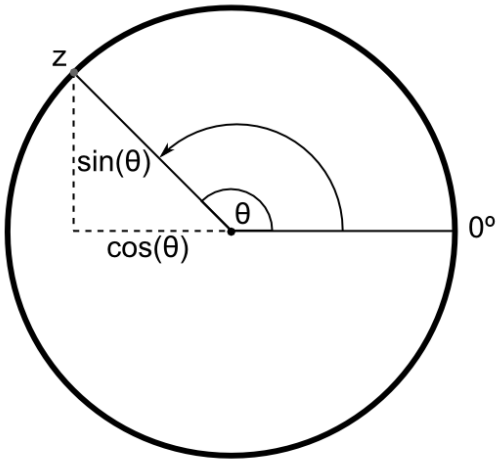


Figure 2.3: Both circular Cartesian and complex number coordinates approaches to reference the angle $\theta = \frac{3}{4}\pi$ in the circle once initial direction (counterclockwise) and reference angle (0 degrees) have been chosen.

Solving the problem of the coordinates is not enough as the distance example brought to observation. New statistics need to be defined in order to effectively study data on the circle.

The redefinition of the mean goes through the definition of two statistics. Let $\Theta = \{\theta_1, \theta_2, \cdots, \theta_n\}$ be a set of angular observations (note that if we were given the unitarian vectors as observations, the angles with respect to our reference system would be calculated to use them as the data). We define the mean components of the circular Cartesian coordinates as:

$$\overline{S} = \frac{1}{n}\sum_{i=1}^{n}\sin\theta_i,\ \overline{C} = \frac{1}{n}\sum_{i=1}^{n}\cos\theta_i$$

Then the mean angle is calculated as:

$$\overline{\theta} = \begin{cases} \arctan\frac{\overline{S}}{\overline{C}} & \text{if } \overline{C} \geq 0 \\ \\ \arctan\frac{\overline{S}}{\overline{C}} + \pi & \text{if } \overline{C} < 0 \end{cases} \tag{2.1}$$

This expression will give the same mean as the classical linear sample mean as long as the observations are in $[0°, 180°]$ (with a counterclockwise direction and a reference point of $0°$) where acknowledging or not if the line is closed on itself is simplified under appearances.

It can be noted that if we represent the point $(\overline{S}, \overline{C})$ in the plane it may not be in the circle as it could happen that it produces a non-unitarian vector. The length of this vector is called the mean resultant length. It can be calculated as

$$\overline{R} = \sqrt{\overline{S}^2 + \overline{C}^2} \tag{2.2}$$

or

$$\overline{R} = \frac{1}{n}\sum_{i=1}^{n}\cos(\theta_i - \overline{\theta}), \tag{2.3}$$

and additionally related to $\overline{C}$ and $\overline{S}$ by

$$\overline{C} = \overline{R}\cos\overline{\theta} \tag{2.4}$$

$$\overline{S} = \overline{R}\sin\overline{\theta} \tag{2.5}$$

where $\overline{\theta}$ is the mean angle (see Figure 2.4).

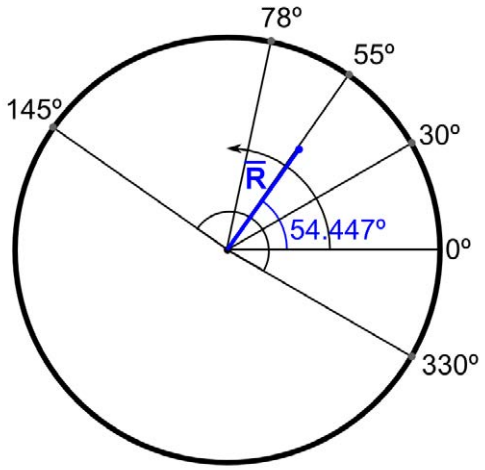


Figure 2.4: For angles $0^\circ, 30^\circ, 55^\circ, 78^\circ, 145^\circ$ and 330°, the correctly calculated mean and the mean resultant length. The calculated values were: $\overline{\theta} = 54^\circ 26' 49.2''$ and $\overline{R} = 0.5828$.

The $\overline{R}$ value has a meaning in the description of the set of observations as it results to be a measure of the concentration as opposed to the concept of variance in classical statistics. If we were in the position to place some observations on the circle and compute its mean resultant length, to maximize its expression we must place all of them at the same point. We can get more detailed insights about $\overline{R}$ by means of the following results:

Lemma 2.1.1. $\overline{R} \in [0, 1]$.

Lemma 2.1.2. If Θ can be expressed as $\Theta = \{\theta_1, \ldots, \theta_n, \theta_1 + \pi, \ldots, \theta_n + \pi\}$ then $\overline{R} = 0$.

Lemma 2.1.3. $\overline{R} = 1$ only when $\theta_1 = \theta_2 = \theta_3 = \cdots = \theta_{n-1} = \theta_n \in \Theta$ (all angles are equal).

Proofs of these results can be found in Mardia and Jupp [2000]. With this information, we define another statistic that was conceptually introduced before: the distance between two angles ϕ and θ as

$$d(\phi, \theta) = 1 - \cos(\phi - \theta).$$

So we are now in conditions to interpret $\overline{R}$ as the mean of the "1−distance to the mean" that each of our observations present. Thus, $\overline{R}$ only contains and uses the information of computing the average of the distances to the mean, which can be considered the nature of its concentration diagnosing capabilities.

Formally,

$$\frac{1}{n}\sum_{i=1}^{n} d(\theta_i, \overline{\theta}) = \overline{d} = \frac{1}{n}\sum_{i=1}^{n}(1 - \cos(\theta_i - \overline{\theta})). \tag{2.6}$$

Then, by using Equation (2.3),

$$\overline{d} = \frac{1}{n}\sum_{i=1}^{n} 1 - \frac{1}{n}\sum_{i=1}^{n}\cos(\theta_i - \overline{\theta}) = \frac{1}{n}\sum_{i=1}^{n} 1 - \overline{R}.$$

We obtain

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^{n} 1 - \frac{1}{n}\sum_{i=1}^{n} d(\theta_i, \overline{\theta}) &= \overline{R} \\ \frac{1}{n}\sum_{i=1}^{n}(1 - d(\theta_i, \overline{\theta})) &= \overline{R} \end{aligned}$$

as stated above.

It is now straightforward to introduce as a generalization of the mean restriction imposed in Equation (2.6), the statistic for computing the dispersion of a set of angles Θ about a given angle θ as:

$$D(\Theta, \theta) = \frac{1}{n}\sum_{i=1}^{n}(1 - \cos(\theta_i - \theta)).$$

This distance notion takes into consideration the periodicity of the circle, but its results are not expressing perimeter distances. Accounting the perimeter scaling, another notion of distance was found in this work to be:

$$d_2(\theta_1, \theta_2) = \arccos(\cos(\theta_1 - \theta_2)),$$

which can be considered the circular analogue to that on the line

$$d(x_1, x_2) = |x_1 - x_2|.$$

Lastly, it has been proposed as the circular analogue to the linear variance the statistic

$$\overline{V} = 1 - \overline{R} \in [0, 1]$$

although other proposals also exist.

2.2 The von Mises distribution

In this Section we will give a complete addressing of the von Mises distribution as its definition and properties intersect highly those of the truncated von Mises distribution of Chapter 5. Similarly to the line, probability distributions followed by a random circular variable (random variable that produces angular values or unitarian vectors) can also be subject to study and definition. Distributions on the

circle are angular l-periodic distributions (where $l \in \mathbb{R}$ and $\exists n \in \mathbb{N}$ such that $nl = 2\pi$), that is, periodic distributions whose period is multiple of 2π. They can be obtained mainly by two related procedures: natively defining them on $\mathbb{O}$ or wrapping them from distributions on the line.

A wrapped on the circle random variable is obtained from a random variable on the line by introducing the fundamental difference between both sets on its definition. In this case a random circular variable X_w is defined with respect to the line random variable X as:

$$X_w = X \mod 2\pi.$$

Using the complex numbers notation, it is defined as:

$$X_w = e^{iX}.$$

and the density function of the probability distribution associated to that variable can also be written in terms of the line density function as:

$$f_w(\theta) = \sum_{k=-\infty}^{\infty} f(\theta + 2\pi k).$$

The most significant example is the wrapped normal distribution:

$$f_{WN}(\theta; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} e^{\frac{-(\theta-\mu+2\pi k)^2}{2\sigma^2}}. \tag{2.7}$$

Native circular distributions are directly defined in the $\mathbb{O}$ domain, although one can establish a mapping between both line and circle's perimeter and therefore find or hypothesize the existence of their linear counterpart and vice-versa.

Let θ be a continuous random variable that follows a circular density distribution, then $f(\theta)$ satisfies:

1. $\int_a^{2\pi+a} f(\theta)d\theta = 1$, where $a \in \mathbb{R}$
2. $f(\theta + 2\pi k) = f(\theta),\ \forall k \in \mathbb{Z}$

That is, the properties that mostly differentiate both scenarios (linear and circular) are the redefinition of the integral coefficients to those of the circle (1.) and the periodicity of the density function (2.).

2.2.1 Definition

The von Mises probability distribution is natively defined as

$$f_{vM}(\theta; \mu, \kappa) = \frac{e^{\kappa \cos(\theta-\mu)}}{2\pi I_0(\kappa)} \tag{2.8}$$

where

1. $\mu \in [i, i + 2\pi], i \in \mathbb{R}$, is the location parameter as it defines where the mode of the distribution is going to be placed. In this case, the maximum value of the $\cos(.)$ function is reached at $\theta = \mu$, thus relating μ directly with the mode. The i value in this context enables the selection of the interval of length 2π where the distribution is going to be observed. Most common values in literature

are $i = 0$ or $i = -\pi$ and in this work, unless otherwise stated, the considered interval is $[0, 2\pi)$. Additionally, the μ parameter is commonly called the mean parameter as in this case as well as other well known cases such as the normal distribution, the mode and the mean have similar value (these distributions are called "mean-centered distributions" as the density tends to concentrate around it).

2. $\kappa \in [0, \infty)$ is the scale or concentration parameter, as opposed to the σ parameter on the normal distribution. It determines the concentration of the distribution around its highest value (in this case the mean). The higher κ is, the more concentrated around the mean the distribution becomes. In the special case where $\kappa = 0$ the distribution reduces to the uniform circular distribution: $f_{vM}(\theta; \mu, 0) = u(\theta) = \frac{1}{2\pi}$.

3. $I_0(\kappa) = \sum_{m=0}^{\infty} \frac{\kappa^{2m}}{2^{2m}(m!)^2}$ is the first kind modified Bessel function of order 0.

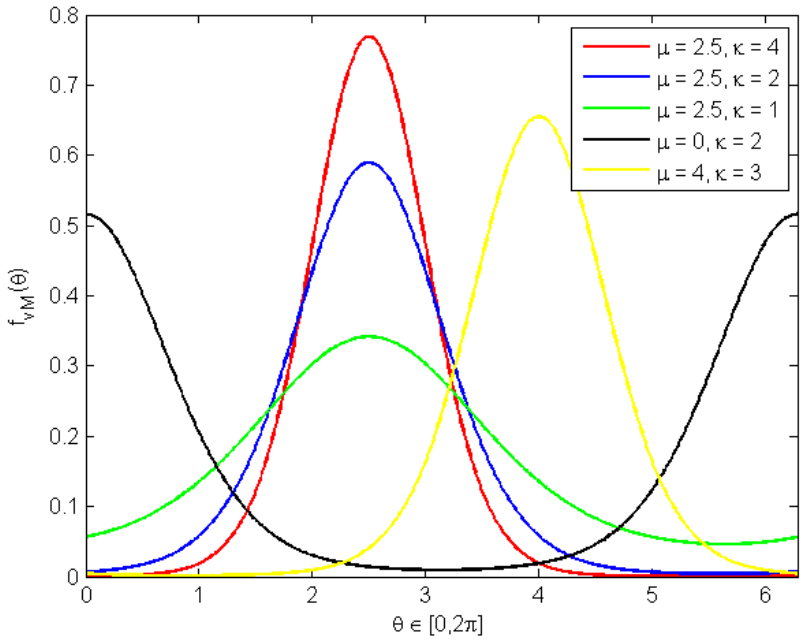


Figure 2.5: Example of different von Mises density functions with varying μ, κ parameters.

By manipulating the μ, κ parameters, the resulting von Mises function may differ in location and concentration from other von Mises distributions (see Figure 2.5), as suggested by the parameters definition.

2.2.2 Properties

The von Mises distribution is composed by the periodic function

$$f_{uvM}(\theta; \mu, \kappa) = e^{\kappa \cos(\theta - \mu)}, \tag{2.9}$$

which will be referred to as unnormalized von Mises distribution and its integral over any interval of length 2π $[i, i + 2\pi]$ is

$$\int_i^{i+2\pi} e^{\kappa\cos(\theta-\mu)}d\theta = 2\pi I_0(\kappa).$$

Therefore, analyzing Equation (2.9) allows us to observe and report many of the properties of the distribution. f_{uvM} can be subdivided into a continuous strictly increasing function $e^{(.)}$, a positive constant κ and a $\cos(.) \in [-1, 1]$ function.

With this we can conclude

$$f_{uvM}(\theta; \mu, \kappa) \in [e^{-\kappa}, e^{\kappa}]$$

Realizing now that $I_0(\kappa)$ is a positive strictly increasing function for $\kappa > 0$ allows us to say that

$$f_{vM}(\theta; \mu, \kappa) > 0 \quad \forall\theta, \mu, \kappa$$

which implies that its distribution function $F_{vM}(x) = \int_0^x f_{vM}(\theta; \mu, \kappa)d\theta$ for f_{vM} defined in $[0, 2\pi]$ and $x \in [0, 2\pi]$ is a strictly increasing function in $[0, 2\pi]$. In general, $F_{vM}(x) = \int_i^{x+i} f_{vM}(\theta; \mu, \kappa)d\theta > 0$ provided $x \in [i, i + 2\pi]$ (see Figure 2.6).

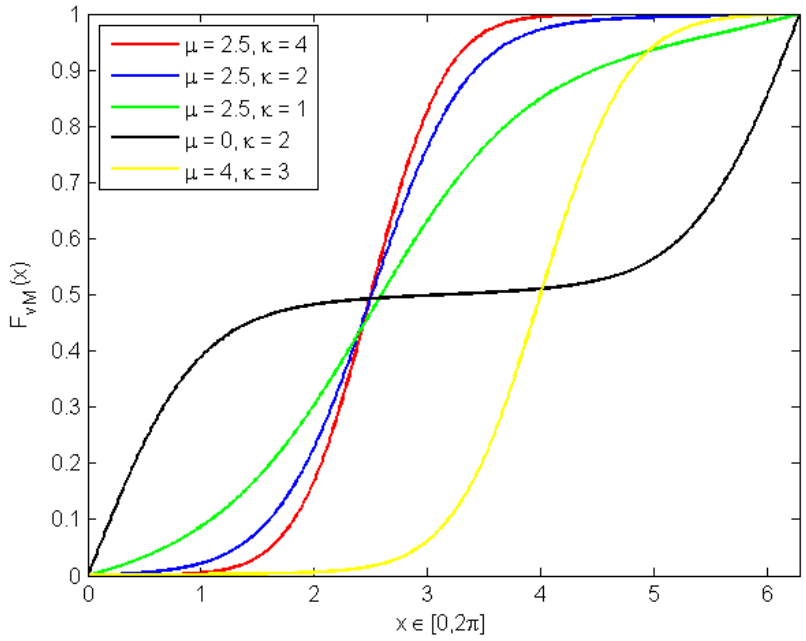


Figure 2.6: The von Mises distribution functions of the previously shown von Mises density functions.

The distribution is symmetrical with respect to the location parameter as:

$$f_{vM}((\mu + \theta) - \mu) = f_{vM}((\mu - \theta) - \mu)$$

$$f_{vM}(\theta) = f_{vM}(-\theta)$$

This behavior is obtained from the known even property of the $\cos(.)$ function where $\cos(-x) = \cos(x)$, as it takes the independent variable (θ) as input.

An interesting result comprehending both wrapped normal distribution and von Mises distribution is the increasing approximation capability as κ grows that both share: the von Mises distribution tends to converge to a corresponding wrapped normal distribution for large κ. More formally, the obtained results reported in Mardia and Jupp [2000] were:

$$\lim_{\kappa \to \infty} f_{vM}(\theta; \mu, \kappa) = f_{WN}\left(\theta; \mu, \sqrt{\frac{1}{\kappa}}\right)$$

where f_{WN} was defined in Equation (2.7).

The existence of the progressive approximation to the previous equality as κ grows is acknowledged in the literature and allows the use of f_{WN} instead of the von Mises distribution for different problems where it could be applied.

2.2.3 Maximum likelihood estimation

Inside the statistical inference scenario, we are interested in approximating the underlying probability distribution that a random variable follows by the information provided solely by the samples collected from it. In this section, we will develop for contextual purposes the maximum likelihood estimator of the von Mises distribution parameters. It can be found also in Mardia and Jupp [2000].

Given the data $\Theta = \{\theta_1, \theta_2, ...\theta_n\}$, the log-likelihood function

$$\ln \mathcal{L}(\mu, \kappa; \theta_1, \theta_2, \ldots, \theta_n) = \sum_{i=1}^{n} \ln f(\mu, \kappa; \theta_i)$$

is, for the von Mises distribution,

$$\ln \mathcal{L}(\mu, \kappa; \theta_1, \theta_2, \ldots, \theta_n) = \sum_{i=1}^{n} \kappa \cos(\theta_i - \mu) - n \ln(2\pi I_0(\kappa))$$

We seek to solve the system of log-likelihood equations created by:

$$\begin{cases} \frac{\partial \ln \mathcal{L}}{\partial \mu} = 0 \\ \frac{\partial \ln \mathcal{L}}{\partial \kappa} = 0 \end{cases}$$

These are two equations with two unknown variables. For the partial derivative with respect to μ we obtain:

$$\frac{\partial \ln \mathcal{L}}{\partial \mu} = \sum_{i=1}^{n} \kappa \sin(\theta_i - \mu) = 0$$

or

$$\frac{1}{n}\sum_{i=1}^{n}\kappa\sin(\theta_i-\mu)=0$$

We know by definition that $\kappa > 0$. Thus, in the case of the existence of a solution, it is independent of the κ value. Therefore

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}\sin(\theta_i-\mu) &= 0\\
\frac{1}{n}\sum_{i=1}^{n}(\sin(\theta_i)\cos(\mu)-\sin(\mu)\cos(\theta_i)) &= 0\\
\frac{\sin(\mu)}{\cos(\mu)}\frac{\frac{1}{n}\sum_{i=1}^{n}\cos(\theta_i)}{\frac{1}{n}\sum_{i=1}^{n}\sin(\theta_i)} &= 1\\
\tan(\mu) &= \frac{\overline{S}}{\overline{C}}\\
\hat{\mu} &= \arctan\left(\frac{\overline{S}}{\overline{C}}\right)
\end{aligned}$$

That is, the μ parameter reaches a critical point at the definition of the sample mean (Equation (2.1)).

Now we proceed with the partial derivative with respect to κ as:

$$\frac{\partial \ln\mathcal{L}}{\partial\kappa} = \sum_{i=1}^{n}\cos(\theta_i-\mu)-n\frac{I_1(\kappa)}{I_0(\kappa)}=0$$

or

$$\frac{1}{n}\sum_{i=1}^{n}\cos(\theta_i-\mu)=\frac{I_1(\kappa)}{I_0(\kappa)},$$

given the equation for the Bessel function derivative, stated as

$$\frac{\partial I_n(x)}{\partial x} = \frac{n}{x}I_n(x)+I_{n+1}(x). \tag{2.10}$$

At this point we can observe that we are dealing with the definition of $\overline{R}$ in Equation (2.3) as we have

$$\hat{R}=\frac{I_1(\kappa)}{I_0(\kappa)} \tag{2.11}$$

Equation (2.11) is commonly referred to in the literature (for example in Mardia and Jupp [2000]) as the maximum likelihood estimator of $\overline{R}$.

If we now consider the system of log-likelihood equations

$$\begin{cases} \hat{\mu} = \arctan(\overline{S}/\overline{C}) \\ \\ \frac{1}{n}\sum_{i=1}^{n}\cos(\theta_i - \mu) = \frac{I_1(\kappa)}{I_0(\kappa)} \end{cases}$$

we can find the estimator

$$MLE(\mu) = \hat{\mu} = \arctan(\overline{S}/\overline{C}),$$

as its expression is independent of all remaining parameters (κ) in the system and depends solely on the sample data.

The estimator of κ, also independent, introduces the non trivial problem of obtaining the inverse function of

$$A(\kappa) = \frac{I_1(\kappa)}{I_0(\kappa)}.$$

However, in this case we can consider to calculate $\overline{R}$ by Equation (2.2) and (2.3) and approximate numerically its value with $A(\kappa)$ by assessing it for different κ values.

2.2.4 Characteristic function

The characteristic function of a random variable is widely used in literature as a tool to handle the underlying probability distribution followed by that variable. Among its interesting properties we have that a probability distribution is uniquely determined by its characteristic function, which can then be used to refer uniquely to such distribution when performing studies over it and its existence for any probability distribution.

The general expression of the characteristic function of a circular random variable X is defined as the sequence of complex numbers given by the expression:

$$\Phi_X(t) = \mathbb{E}[e^{itX}],$$

where $t \in \mathbb{Z}$ follows the sequence $t = -\infty, \ldots, -1, 0, 1, \ldots \infty$.

For the von Mises density function in $[0, 2\pi]$ we have:

$$\begin{aligned} \Phi_{X_{vM}}(t) = \mathbb{E}[e^{itX}] &= \frac{1}{2\pi I_0(\kappa)} \int_0^{2\pi} e^{itx} e^{\kappa\cos(x-\mu)} dx \\ &= \frac{1}{2\pi I_0(\kappa)} \int_0^{2\pi} \left(\cos(tx) + i\sin(tx)\right) e^{\kappa\cos(x-\mu)} dx \\ &= \frac{\int_0^{2\pi} \cos(tx) e^{\kappa\cos(x-\mu)} dx}{\int_0^{2\pi} e^{\kappa\cos(x-\mu)} dx} + \frac{i\int_0^{2\pi} \sin(tx) e^{\kappa\cos(x-\mu)} dx}{\int_0^{2\pi} e^{\kappa\cos(x-\mu)} dx} \end{aligned}$$

The second addend is 0, $\forall t \in \mathbb{Z}$, when the distribution is symmetrical with respect to the mean. As it is always the case and considering Equation (2.10), we can simplify the former expression by

$$\Phi_{X_{vM}}(t) = e^{it\mu}\frac{I_t(\kappa)}{I_0(\kappa)}$$

where $I_t(\kappa)$ is the modified Bessel function of the first kind and order t. Note that $\Phi_{X_{vM}}(-t) = \Phi_{X_{vM}}(t)$.

2.2.5 Moments

The moments of a probability distribution are descriptors associated to power values of its population and can be derived from the characteristic function associated to that distribution. More precisely, the t-th trigonometric moment (with $t \in \mathbb{Z}$) m_t in the circle is calculated as the expectation

$$m_t = \mathbb{E}\left[\left(e^{iX}\right)^t\right] = \mathbb{E}[e^{itX}].$$

It can be immediately noticed that the sequence of all possible moments for t is equivalent to the characteristic function of that random variable.

Unlike distributions in the line, an important result acknowledged in Mardia and Jupp [2000] reveals that any circular distribution is completely determined by its characteristic function, implying that any circular distribution has well defined moments for every value of t. This result appears to arise from a practical fundamental difference of the closed space of the circle with respect to the line and that is the lack of the infinite extension in the domain of any distribution function, which frees us from needing it in the circular expectation operators and calculation definitions.

We can derive the moments of the von Mises distribution about the a direction by:

$$m_{t_{vM}} = \mathbb{E}[e^{it(X-a)}]$$

Without considering $m_0 = 1$, the first moment about the 0 direction for the von Mises distribution is

$$m_{1_{vM}} = \frac{\int_0^{2\pi} \cos(x) e^{\kappa \cos(x-\mu)} dx}{\int_0^{2\pi} e^{\kappa \cos(x-\mu)} dx}$$

Or equivalently:

$$\begin{aligned} m_{1_{vM}} &= \mathbb{E}[e^{iX}] \\ &= \mathbb{E}[\cos X + i \sin X] \\ &= \mathbb{E}[\cos X] + i\mathbb{E}[\sin X] \end{aligned}$$

Now applying the population versions of Equation (2.4) and (2.5) we can follow with:

$$\begin{aligned} m_{1_{vM}} &= R\cos(\mu) + iR\sin(\mu) \\ &= Re^{i\mu} \\ &= \frac{I_1(\kappa)}{I_0(\kappa)} e^{i\mu} \end{aligned}$$

which constitutes the final expression for the first moment. For the second moment we have

$$\begin{aligned} m_{2_{vM}} &= \frac{\int_0^{2\pi} \cos(2x) e^{\kappa \cos(x-\mu)} dx}{\int_0^{2\pi} e^{\kappa \cos(x-\mu)} dx} \\ &= \frac{I_2(\kappa)}{I_0(\kappa)} e^{i2\mu} \end{aligned}$$

where $I_2(\kappa)$ is the modified Bessel function of the first kind and order 2.

Since our distribution location is controlled by μ parameter, for location independent descriptions it is interesting to consider the moments about the real μ direction as:

$$m'_{1_{vM}} = \frac{\int_0^{2\pi} \cos(x-\mu) e^{\kappa \cos(x-\mu)} dx}{\int_0^{2\pi} e^{\kappa \cos(x-\mu)} dx}$$

which results in:

$$m'_{1_{vM}} = \frac{I_1(\kappa)}{I_0(\kappa)}$$

And

$$m'_{2_{vM}} = \frac{\int_0^{2\pi} \cos(2(x-\mu)) e^{\kappa \cos(x-\mu)} dx}{\int_0^{2\pi} e^{\kappa \cos(x-\mu)} dx}$$

which results in:

$$m'_{2_{vM}} = \frac{I_2(\kappa)}{I_0(\kappa)}.$$

We can generalize the notion of moments about the 0 direction for the von Mises distribution as

$$m_{t_{vM}} = \frac{I_{|t|}(\kappa)}{I_0(\kappa)} e^{it\mu}$$

where $|.|$ is the absolute value operator.

And for the moments about the μ direction we have:

$$m'_{t_{vM}} = \frac{I_{|t|}(\kappa)}{I_0(\kappa)}.$$

Chapter 3

Probabilistic graphical models

3.1 Introduction

Part of the problem in artificial intelligence is focused around systems that can perform reasoning under uncertainty. Probabilistic graphical models (PGMs) use a graphical representation of the domain of knowledge in the form of a graph around a set of variables. Abscense/presence of arc may help to derive conditional independencies. The idea for a graphical model can be traced back to various sources, but in the fields more directly related to its current form, we see first conclusive evidence of its adoption in the works of Vorob'ev [1962], Goodman [1970] and Haberman [1970] in the field of statistics and in Warner et al. [1961], Gorry and Barnett [1968] and De Dombal et al. [1972] in the field of artificial intelligence.

Bayesian networks (Pearl [1988]) consolidated the popularity and theoretical foundations of PGMs. They were proposed as a general framework for probabilistic reasoning capable of overcoming the strong limitations and assumptions of contemporary models. This was accompanied by early successful applications of the framework, for which we can highlight perhaps Heckerman and Nathwani [1992a] and Heckerman and Nathwani [1992b].

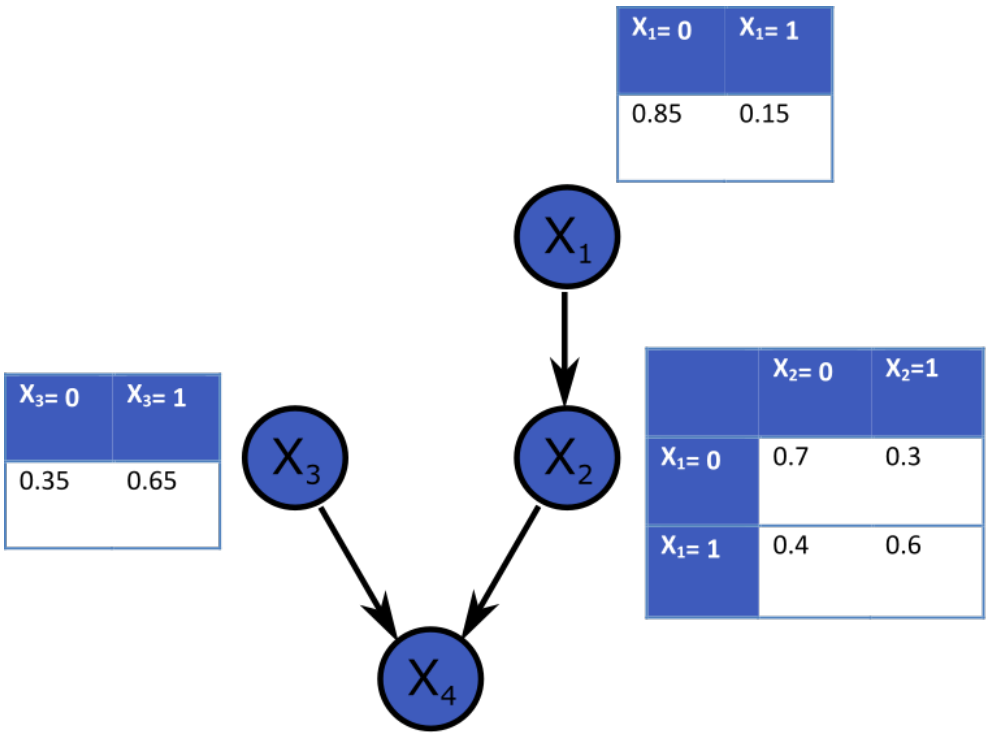
Bayesian networks support multiple classes of problems such as classification, regression, clustering, variable selection and sampling and can perform inference and multi-type reasoning (i.e., diagnostic, predictive, abductive, causal and more) for different queries to the variables of the model. They can be used as a probabilistic knowledge base where interepretability and readability of the model are possible, and decisions made by the model can be explained in a comprehensive way. This distinguishes it from other machine learning methods and makes it a preferred option in sensitive domains where the explanations of answered questions are as important as the answer itself.

From the 90s to 00s, most notable works can be found in Lauritzen [1996], Jensen [1996], Castillo et al. [1997] and Jordan [1998]. From then on, in Neapolitan et al. [2004], Cowell et al. [2006], Darwiche [2009], Korb and Nicholson [2010] and Russell and Norvig [2016]. In later years, BNs have been used successfully in the neuroscience domain in Lopez-Cruz et al. [2011], López-Cruz et al. [2014], Smith et al. [2006] and Jung et al. [2010]. See a review in Bielza and Larrañaga [2014b]. Additionally, steady progress has been made in the multidimensional classification paradigm using Bayesian networks. The reader is directed to Bielza et al. [2011] for a complete survey.

3.2 Bayesian networks

Formally, a Bayesian network is a pair $\mathcal{B} = (\mathcal{G}, \Theta)$ over a set of random variables $\mathcal{X} = \{X_1, X_2, \ldots, X_d\}$ where $\mathcal{G} = (V_\mathcal{X}, A_\mathcal{X})$ is a directed acyclic graph, $V_\mathcal{X}$ is a collection of vertices, $A_\mathcal{X} \subseteq V_\mathcal{X} \times V_\mathcal{X}$ is a collection of arcs between vertices of $V_\mathcal{X}$ and Θ, in the context of Bayesian networks, is a set of conditional probability distributions paired with the structure $\mathcal{G}$.

Vertices of the Bayesian network represent the random variables in $\mathcal{X}$ and the directed arcs represent probabilistic dependence relationships between the variables. Probability distributions in Θ are defined as $\theta_{x_i|\mathbf{pa}(x_i)} = p(x_i|\mathbf{pa}(x_i))$, that is, conditional probability distributions of variable X_i given a value $\mathbf{pa}(x_i)$ of the set of variables $\mathbf{Pa}(X_i) \in \mathcal{X}$. In here, $\mathbf{Pa}(X_i)$ stands for the set of parent variables of X_i in $\mathcal{G}$ (that is, variables of the graph with connected arcs that end in X_i) (Figure 3.1).



X_1= 0	X_1= 1
0.85	0.15

X_3= 0	X_3= 1
0.35	0.65

	X_2= 0	X_2=1
X_1= 0	0.7	0.3
X_1= 1	0.4	0.6

	X_4=0	X_4=1
X_3=0, X_2=0	0.14	0.76
X_3=0, X_2=1	0.33	0.67
X_3=1, X_2=0	0.32	0.68
X_3=1, X_2=1	0.8	0.2

Figure 3.1: A Bayesian network with four nodes and the conditional probability tables associated with each node.

The joint probability distribution can be used to model different classes of problems. However, the computation of joint probability distributions is considered intractable in the general case. With Bayesian

networks, it is possible to factorize a joint probability distribution as follows:

$$p(X_1, \ldots, X_d) = \prod_{i=1}^{d} p(X_i|\mathbf{Pa}(X_i)) \tag{3.1}$$

Equation (3.1) can be seen as a substantial reduction in the size and complexity with respect to the joint distribution case, where the need to store every possible d-tuple of values would pose a significantly bigger problem even for a relatively small number of nodes. Bayesian networks reduce this complexity by exploiting the conditional independence relationships between the variables in the domain.

3.2.1 Parameters

We have already seen that the vertices or nodes of a Bayesian network can be seen as variables that are conditionally distributed on their parent variables. In order to calculate the value that corresponds to a specific assignment of values, we must first make the distinction between discrete and continuous nodes.

3.2.1.1 Discrete nodes

Discrete nodes have an associated discrete probability distribution taking values in a finite numerable domain. The output of these nodes are probabilities as opposed to densities in the continuous case. Notice that a node X_i encodes multiple probability distributions, one per each parent $\mathbf{Pa}(X_i)$ distinct configuration. In order to represent this information, a conditional probability table (CPT) is generally used.

A CPT (Figure 3.1) can be regarded as a table that has for rows the distinct assignments of all parent variables of X_i, and as columns all values of X_i. If we consider $Val(X_i)$ the set of values that the variable X_i can take, then

$$\sum_{x_i \epsilon Val(X_i)} p(x_i|\mathbf{pa}(X_i)) = 1$$

is satisfied for each assignement $\mathbf{pa}(X_i)$ for the parents of X_i, $\mathbf{Pa}(X_i)$. This is, for each valid joint assignment of values for the conditional variables, a categorical probability distribution is defined that assigns probability values to the different values the random variable X_i takes.

Notice also that the CPT grows exponentially in size with each new parent addition, as all possible configurations of that new parent must then be taken into account. This shows that the number of parameters needed for a discrete node in a Bayesian network can be calculated as the product

$$(|Val(X_i)| - 1)(|Val(\mathbf{Pa}(X_i))|),$$

where we can define $|Val(\mathbf{Pa}(X_i))|$ as the cardinality of the set containing the total number of joint distinct value assignments of the parents of X_i. The -1 in the expression comes from the fact that for a categorical distribution on a random variable with $v_1, \ldots, v_m$ values, after all probabilities but $p(v_i)$ are specified, $p(v_i)$ can be trivially inferred as $p(v_i) = 1 - \sum_{j \neq i} p(v_j)$.

Then, for a complete discrete Bayesian network, the total number of parameters can be calculated as:

$$\sum_{i=1}^{d} (|Val(X_i)| - 1)(|Val(\mathbf{Pa}(X_i))|). \tag{3.2}$$

3.2.1.2 Continuous nodes

Continuous nodes have an associated continuous probability distribution, taking infinite values on a continuous domain. The CPT representation does not adapt well to the change of variable nature. A common workaround is discretization, aiming to produce usable CPTs by grouping ranges of values into a finite size group of categories (Garcia et al. [2013]). However, discretization techniques incur in loss of information, making it an undesirable option in certain classes of problems. Additionally, different discretization strategies significantly affect the parameter values of the resultant Bayesian network.

Another most commonly used option is to introduce in our model the assumption that the distributions associated with the nodes belong to a certain parametric family of distributions. Of all available parametric forms, by far the most commonly used is the Gaussian family. Bayesian networks composed solely of linear Gaussian nodes are called Gaussian Bayessian networks (GBNs).

In the case of GBNs, a node X_i that is linear Gaussian with parents $\mathbf{Pa}(X_i) = \{Pa_{i1}, Pa_{i2}, \dots, Pa_{il}\}$, $l \in \mathbb{N}$ has an associated Gaussian probability distribution given by

$$f(X_i|\mathbf{Pa}(X_i)) \sim \mathcal{N}(\beta_0 + \beta_1 Pa_{i1} + \cdots + \beta_l Pa_{il}, \sigma_i^2)$$

where $\beta_0, \beta_1 \dots \beta_l$ are the linear regression coefficients of X_i over $\mathbf{Pa}(X_i)$ and σ_i^2 is the variance of X_i. This shows that a GBN can be computed as a product of gaussian distributions. Indeed, the joint probability density $f(X_1, \dots, X_n)$ is factorized as:

$$f(X_1, \dots, X_d) = \prod_{i=1}^{d} f(X_i|\mathbf{Pa}(X_i)).$$

GBNs satisfy certain properties that set them apart from discrete BNs. For example, it can be proven that a GBN defines a multivariate Gaussian distribution and viceversa (Koller and Friedman [2009]). This, parameter wise, implies that instead an exponential increase in the parameter count when we add a new node (discrete joint probability distribution case), we have a quadratic increase for a multivariate gaussian distribution (since $\mathcal{N}(\boldsymbol{\mu}; \boldsymbol{\Sigma})$ is defined by a vector of means $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$). For our chosen factorization, however, the parameter count of a GBN is given by:

$$2d + \sum_{i=1}^{d} |\mathbf{Pa}(X_i)| \tag{3.3}$$

GBNs work better in practice when the underlying data distribution of the problem is not too far off from the assumption of gaussianity. Otherwise, the model may suffer from poor quality fitting.

3.2.2 Inference with Bayesian networks

One of the most interesting properties of Bayesian networks is its ability to perform multiple forms of probabilistic reasoning. Once our model is built, it can answer different types of queries regarding its knowledge domain (that is, its variables).

Typically, we have information on a subset of the total variables of the network $\mathbf{E} \subset \mathcal{X}$ (evidence variables), and we formulate our query over another subset $\mathbf{Q} \subset \{\mathcal{X} \setminus \mathbf{E}\}$ (query variables). Then, a

conditional probability query can be written as:

$$p(\mathbf{Q}|\mathbf{E} = \mathbf{e}) = \frac{p(\mathbf{Q}, \mathbf{e})}{p(\mathbf{e})} \tag{3.4}$$

In order to calculate the probabilities of Equation (3.4), we can proceed by renormalizing the random vector of marginal probabilities

$$p(\mathbf{q}_1, e), \ldots, p(\mathbf{q}_{|Val(\mathbf{Q})|}, e)$$

so that $\sum_{i=1}^{|Val(\mathbf{Q})|} p(\mathbf{q}_i, e) = 1$, which implies that, for each possible query answer, the joint distribution must be calculated in order to sum out the remaining variables of the network (that is, those that are nor evidence nor query). However, working with the joint distribution is intractable in the general case.

Unfortunately, exact inference in the general case is also intractable, with a $\mathcal{NP}$-hard result shown first in Cooper [1990]. However, the complexity of inference is intimately tied to the structural properties of the Bayesian network, and for specific cases, even for large networks, exact inference can be carried out in polynomial time.

In standard literature, most prominent solutions to the computation of exact inference are the algorithm of variable elimination (Zhang and Poole [1994], Huang and Darwiche [1996] and Dechter [1999]) and clique trees (Shafer and Shenoy [1990]). Both are capable of taking advantage of the structural properties of the network to lower the complexity of inference in some scenarios.

The algorithm of clique trees has the advantages over variable elimination of answering multiple queries using the same data structure and reusing the computations performed for previous queries. Additionally, it allows for dynamical introduction or deletion of evidence prior to each query computation, making it a recommended choice in the general case over the standard implementation of variable elimination when multiple queries are intended. However, in clique trees we are forced to store intermediate computations that in variable elimination can be discarded, resulting in an increase of memory space. Additionally, since the structure is fixed it is possible to miss on some computational savings that occur in some specific cases of evidence and query subsets. Particularly, networks displaying context-specific independence (Boutilier et al. [1996]) would often be computed suboptimally with respect to variable elimination, since the precomputed structure would not be able to recognize this type of shortcut as available for this type of query + evidence.

Overall, both algorithms can be considered versions of a broader class of algorithms that we may call variable elimination algorithms.

Approximate inference approximates the queried probabilities while trying to avoid the explosion in computational requirements. Generally, this approach can be subdivided in Monte Carlo algorithms for inference, such as likelihood weighting (Henrion [1988]) or Gibbs sampling (Neal [1993]), and search based methods for high probability instantiations, as in Cooper [1984], Peng and Reggia [1987], Henrion [1990] and Henrion [1991]. Unfortunately, approximate inference was also proven to be an NP-hard problem in Dagum and Luby [1993] .

3.2.2.1 MAP queries

For the purposes of this dissertation, our interest will be placed on a subtype of queries known as Maximum a posteriory queries (MAP queries) rather than conditional probability queries. In MAP queries we seek to answer the query with the most likely assignment of the query variables given the evidence. The

natural product of this type of queries is a unique value assignment to each of the query variables.

More specifically, our interest lies on a simpler case of MAP that is regarded in some literature as the MPE (Most Probable Explanation) problem. This occurs when $\mathbf{Q} \cup \mathbf{E} = \mathcal{X}$, that is, all variables are covered between query and evidence and thus there is no need for the marginalization computations. The MPE problem is considered to be easier in the general case than the MAP problem. However, in both cases we do not leave the exponential complexity category.

We can slightly modify the original formulation of the conditional probability query shown earlier to answer a MPE query by computing

$$\arg\max_{Val(\mathbf{Q})}(p(\mathbf{q}_1, \mathbf{e}), \ldots, p(\mathbf{q}_{|Val(\mathbf{Q})|}, \mathbf{e})).$$

That is, we select the value assignment that yields the highest probability rather than keeping the marginal distribution of the query variables given the evidence.

Variable elimination can be adapted to compute MPE queries simply by swapping the summations for maximization operators, clique trees max product algorithm can also be tweaked by computing max-marginals instead of sum-marginals at each clique of the tree. Our interest in MPE lies within the use of this type of query for classification in Bayesian network classifiers, as we will see in Section 3.2.4.

3.2.3 Learning Bayesian networks

Early efforts in constructing a Bayesian network model typically involved the presence of an expert, whose primary task was to manually identify the most fitting structure and parameters for the network of the domain variables $\mathcal{X}$. This approach is largely considered deprecated nowadays as even for networks of modest size, the building time would scale to hours and required the additional assistance of a knowledge engineer. Another reason for its diminished use is the abundance of data, which allows for automation of the learning procedures. In this setting, the data $\mathcal{D}_n$ is regarded as a collection of samples that belong to an unknown probability distribution $\mathcal{D}$ (and were sampled independently), and our task is that of finding a model that best fits the observed cases of the unknown distribution $\mathcal{D}$. It is also possible to have different goals in mind when building a model, that is, we may be interested in focusing on the performance of the model in a subset of $\mathcal{X}$ variables. Depending on our goals when building the model, we may be discussing density estimation, classification tasks or knowledge extraction as the main categories on which our priorities can be sorted. In all cases, the procedure amounts to the definition of a loss function that we want to minimize, which allows us to compare different candidate models and select the best ones.

For example, let $Q_1 \in \mathcal{X}$ be a single target variable to predict, the $0/1$-loss function, commonly referred to as the classification error, is used to direct our learning procedure towards unidimensional classification. Using MPE queries, for evidence $\mathbf{e}$, we can write a prediction as

$$c_{\tilde{\mathcal{D}}}(\mathbf{e}) = \arg\max_{q_1} \tilde{\mathcal{D}}(q_1|\mathbf{e}),$$

where $\tilde{\mathcal{D}}$ is a probability distribution approximation of $\mathcal{D}$, produced by a BN model trained with $\mathcal{D}_n$. Then the $0/1$-loss function can be written as:

$$\mathbb{E}_{(\mathbf{e},q_1)\sim\tilde{\mathcal{D}}}\left[\mathbb{1}_{[c_{\tilde{\mathcal{D}}}(\mathbf{e})\neq q_1]}\right] \tag{3.5}$$

which can be read as the probability over $\mathcal{D}$ that the network selects the wrong label. For $\mathbf{Q} = \{Q_1, \ldots, Q_b\} \subset \mathcal{X}$ or multiple variables for prediction, the $0/1$-loss function can still be used (some times referred to as global accuracy), although it becomes an exponentially stricter criterion since the number of possible value assignments to the query variables is now $\prod_{i=1}^{b} |Val(Q_i)|$. For this case, a less strict and commonly used criterion is the Hamming loss, which computes the average amount of mistakes per prediction. If we consider $c_{\tilde{\mathcal{D}}}(\mathbf{e}) = (c_1, \ldots, c_b)$ the MPE resultant assignment for the variables in $\mathbf{Q}$, then the Hamming loss is given by the expression:

$$\mathbb{E}_{(\mathbf{e},\mathbf{q})\sim\tilde{\mathcal{D}}}\left[\frac{1}{b}\sum_{i=1}^{b}\mathbb{1}_{[c_i \neq q_i]}\right]. \tag{3.6}$$

However, regardless of the loss function to minimize, in order to obtain a complete model, both the parameters and the structure of a Bayesian network must be estimated.

3.2.3.1 Estimating parameters

Parameter estimation is a relatively easy operation in Bayesian networks with a fixed structure, both in complexity and conceptualization. Most typically two approaches are considered, maximum likelihood estimation (MLE) and Bayesian estimation (BE). In both cases, well defined closed forms for discrete and Gaussian Bayesian networks are available.

Maximum likelihood estimation of the parameters tries to find the most likely assignment for the parameters given the structure of the Bayesian network and the data. For this, we make use of the likelihood function. Formally, if we are given a dataset $\mathcal{D}_n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and a Bayesian network $\mathcal{B} = (\mathcal{G}, \Theta)$ with $\mathcal{X} = \{X_1, X_2, \ldots, X_d\}$ the likelihood function is defined as:

$$\mathcal{L}(\langle \mathcal{G}, \theta \rangle | \mathcal{D}_n) = \prod_{i=1}^{n}\prod_{j=1}^{d} p(x_{ij} | \mathbf{pa}(x_j)_i, \mathcal{G}), \tag{3.7}$$

where for a given $\mathbf{x}_i \in \mathcal{D}_n$ we have $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})$ and $\mathbf{pa}(x_j)_i$ outputs the values in $\mathbf{x}_i$ that the parent's variables of variable X_j take for that instance.

Thus, we are looking for the values that maximize the likelihood function, that is, for a set of parameters estimates $\tilde{\Theta}$ we seek the assignment:

$$\tilde{\theta} = \arg\max_{\theta} \mathcal{L}(\langle \mathcal{G}, \theta \rangle | \mathcal{D}_n).$$

Alternatively, it is commonly found in literature as:

$$\tilde{\theta} = \arg\min_{\theta} -\log(\mathcal{L}(\langle \mathcal{G}, \theta \rangle | \mathcal{D}_n)) \tag{3.8}$$

since it is an equivalent but easier to work with the form of Equation (3.7).

Bayesian estimation, on the other hand, tackles the problem by introducting prior distributions to the parameters. For this problem, we have a prior distribution on the parameters $f_P(\theta)$ and we update our beliefs with the new evidence that the dataset $\mathcal{D}_n$ provides. This corresponds to computing the posterior distribution $f_{P|\mathcal{D}_n}(\theta|\mathcal{D}_n)$ and finding the configuration of parameters that maximizes it. Formally, its corresponding optimization function is given by:

$$\tilde{\theta} = \arg\max_{\theta} f_{P|\mathcal{D}_n}(\theta|\mathcal{D}_n).$$

Maximum likelihood estimation is the most commonly used method in literature for parameter estimation. Its advantage over Bayesian estimation is its simplicity. In fact, maximum likelihood estimation can be understood as a particular case of Bayesian estimation where no prior information is given. For Bayesian estimation, we have the ability to operate in online settings, where we could transform previous data into our prior knowledge and update our networks with new arriving examples. In the limit, both cases are proven to converge to the "closest" approximation to the true underlying distribution $\mathcal{D}$ that the chosen Bayesian network structure $\mathcal{G}$ is capable of producing.

3.2.3.2 Obtaining the structure

Algorithms for the estimation of the structure of a Bayesian network are and have been historically one of the most active research topics in the field. The problem is by no means trivial: For a given set of d nodes $\mathcal{X}$, the number of possible graphs is given by Robinson's recursive formula (Robinson [1973])

$$\begin{aligned} r(0) &= 1, \\ r(d) &= \sum_{i=1}^{d} (-1)^{i+1} \binom{d}{i} 2^{i(d-1)} r(d-i), \end{aligned}$$

which shows a superexponential growth (according to $2^{\mathcal{O}(d^2)}$) (see Figure 3.2). Additionally, relationships between variables are encoded in a Bayesian network in an ambiguous way. That is, for a given structure, others exist that encode the same set of conditional independence relationships between the variables. The class of networks that encode a given set of conditional independence relationships is called an I-equivalence class. If our goal is to recover a specific structure then our data can only take us as far as the I-equivalence class.

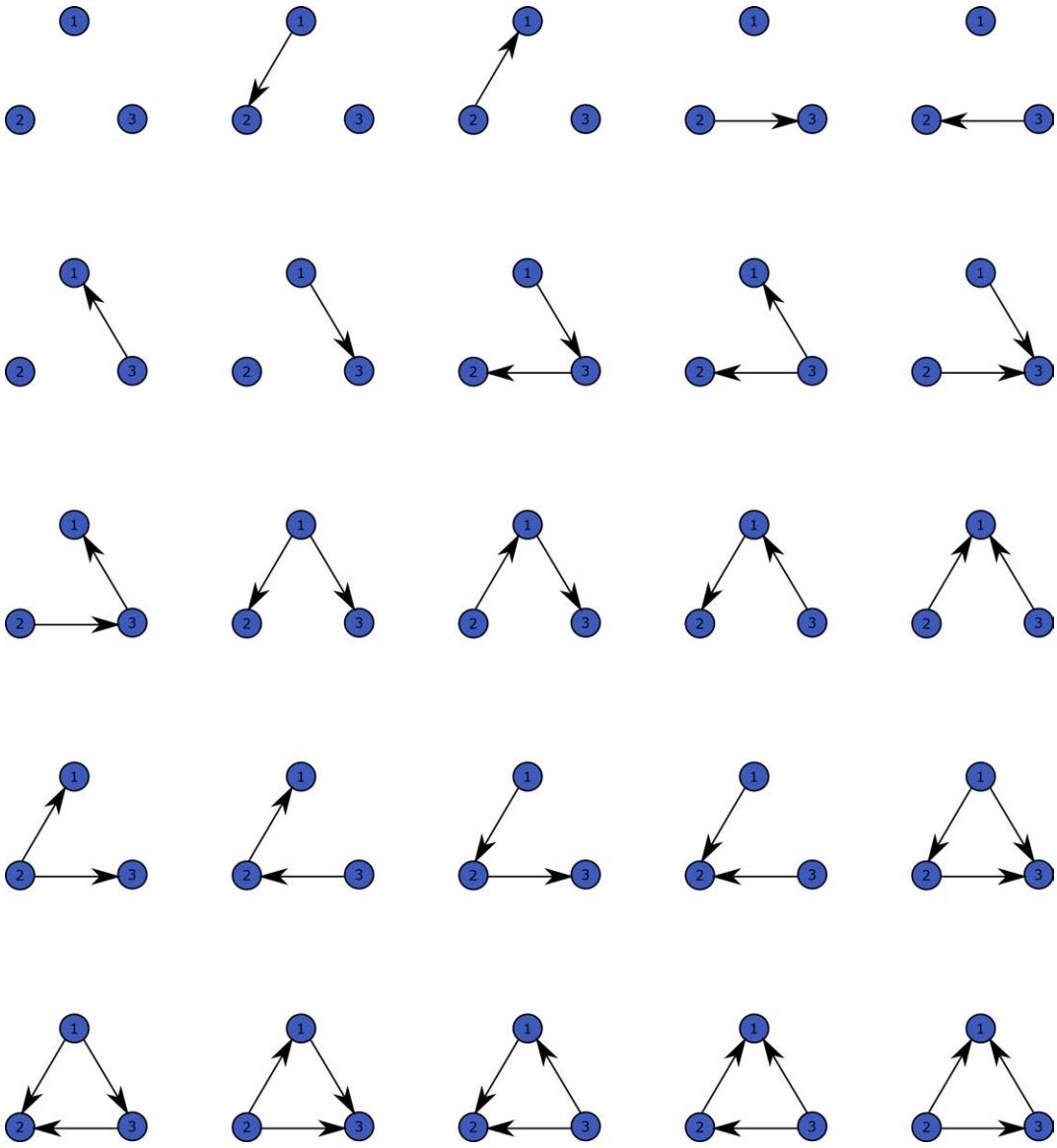


Figure 3.2: All possible Bayesian network structures for a three nodes network.

From the numerous proposals to reconstruct/find the optimal structure, two main categories stand out: constraint-based and score + search.

Constraint- based structural learning

In this approach, our goal is to find the set of conditional independences that best approximates the relationships between the variables in $\mathcal{X}$. That is, queries for this type of problem are of the form:

$$(X_i \perp \{X_j, X_k\} | X_l)$$

which can be read as "X_i is independent of X_j and X_k given X_l". We use independence testing for different subgroupings of variables in $\mathcal{X}$. The specific way in which an independence test answers a query is not required for the algorithms to work. In all cases, once the network dependencies have been identified, the output is an undirected acyclic graph that best encodes the found and non-found dependencies, that is, the I-equivalence class that the final Bayesian network must implement. For hereon, all that is left is

for a second procedure to assign directions to the arcs so that the final result is a valid Bayesian network that best approximates $\mathcal{D}$. Computationally wise, in the general case there is an exponential growth with respect to the number of variables that we must include in a query in order to detect all independencies of the network. In practice, oftentimes a threshold parameter i_g, that controls the indegree of the network, (that is, the maximum number of parents any node can have) is used. For a fixed i_g, however, computation can be carried out in polynomial time. Most prominent work in literature for this approach is the PC algorithm (Spirtes et al. [1993]).

In this dissertation, however, we concern ourselves with the next approach.

Score + search structural learning

Here, we approach the problem in a different way: We are equipped with a score function that can assess the "fitness" of a given network structure with respect to the data $\mathcal{D}_n$, and our goal is to find networks in the space of DAGs for variables in $\mathcal{X}$ that maximize our scoring function. Since the space of DAGs is super-exponential on $\mathcal{X}$, some algorithms attempt to reduce the search space by making use of properties within the scoring functions (such as local decomposability, see below) as well as within the network structure.

For the scoring function, we can find many proposals in literature: An intuitive choice is to use the likelihood function. Indeed, we can regard the output of the likelihood function, that is, the probability of the data given the model, as the score of a candidate network. This score, for a network $\mathcal{B} = (\mathcal{G}, \Theta)$ can be constructed as

$$\log(\mathcal{L}(\langle \mathcal{G}, \tilde{\theta} \rangle | \mathcal{D}_n))$$

where $\tilde{\theta}$ are the maximum likelihood parameters of $\mathcal{G}$ as shown in Equation (3.8).

The likelihood score offers a very interesting property: The score of the likelihood function can be traced back, in the computation of the likelihood function, to local computations on the variables and their parent configurations. In the log-likelihood function, these are expressed as addends in the total sum. Interestingly, this decomposability allows us to assess local changes to a network using a previous one as a reference, as only a subset of the sums in the likelihood of the previous candidate would change. A score that exhibits this property is called a decomposable score. In literature, most commonly used scores for Bayesian networks belong to this category.

The likelihood score alone, however, is not considered to be a good score for candidate networks. The problem arises from the following property: Let us consider two candidate structures $\mathcal{G}_1 = \{V_\mathcal{X}, A_\mathcal{X}^{(1)}\}$ and $\mathcal{G}_2 = \{V_\mathcal{X}, A_\mathcal{X}^{(2)}\}$ with maximum likelihood parameters $\tilde{\theta}_1$ and $\tilde{\theta}_2$, respectively, for the variables $\mathcal{X}$. Then, if $A_\mathcal{X}^{(1)} \subset A_\mathcal{X}^{(2)}$ we have that $\log(\mathcal{L}(\langle \mathcal{G}_1, \tilde{\theta}_1 \rangle | \mathcal{D}_n)) \leq \log(\mathcal{L}(\langle \mathcal{G}_2, \tilde{\theta}_2 \rangle | \mathcal{D}_n))$. That is, the likelihood score shows a preference for complex networks over simpler ones, and if one candidate includes all the arcs in the same way as another, and some additional ones, the score is guaranteed to be at least equal. In practice, almost all search procedures using this score will converge to fully connected networks. The only case where this does not hold is on the unlikely event that an exact conditional independence between a subset of variables of $\mathcal{X}$ is detected in the data, without noise.

In order to correct the previous problem, the Bayesian information criterion (BIC) (Schwarz et al. [1978]) includes a penalty term on both the sample size and the number of parameters in $\mathcal{G}$. Its expression

is given by:

$$BIC(\mathcal{G}|\mathcal{D}_n) = \log(\mathcal{L}(\langle \mathcal{G}, \tilde{\Theta} \rangle | \mathcal{D}_n)) - \frac{\log(n)}{2} Dim[\mathcal{G}].$$

$Dim[\mathcal{G}]$ is the number of free parameters in the model (see Equation (3.2) for discrete networks and Equation (3.3) for Gaussian networks). This penalty biases the score towards simpler structures. However, for large n, it can be proven that the optimal candidate structure $\mathcal{G}^*$ maximizes the score, and that all structures that do not belong to the I-equivalence class of $\mathcal{G}^*$ score strictly lower than those that do. For each pair of structures that belong to the same I-equivalence class, the scores in both the likelihood and the BIC score are the same (this is referred to as the equivalence property). Similarly, this score is proven to be decomposable.

The BIC score is widely used and some other popular scores can be understood as variations of BIC. Most notably, Akaike information criterion (AIC) (Akaike [1974]) is a well known variation of BIC where instead of using $\frac{\log(n)}{2} Dim[\mathcal{G}]$ for the penalty term, we use $2Dim[\mathcal{G}]$. The AIC score has the property over BIC to be an estimator of the Kullback-Leibler divergence between the true distribution and our candidate model, however, unlike BIC, it does not converge in probability to the true model. The minimum description length (Schwarz et al. [1978] and Rissanen [1987]) is defined as the opposite of the BIC, sharing similar properties. Other notable mentions are the deviance information criterion (Spiegelhalter et al. [2002]) defined as a generalization of the AIC score for hierarchical modeling and the Hannan-Quinn information criterion (Sin and White [1996]), which can also be viewed as a variation to BIC and AIC scores with a different penalty term.

Once we are equipped with a scoring function, we need a procedure to navigate the space of DAGs, finding and proposing candidate models. As we have examined before, brute force search of the DAG space, or random generation of candidates, is not expected to yield good results even for a target network with a relatively small number of variables. In the general case, the complexity of our algorithms never goes below that of a *NP*-hard problem. For this reason, most search algorithms employed in structural learning are heuristic algorithms. Here, we do not attempt to examine the complete search space, rather, to build a "route" towards a local optimum in the score function with a polynomial number of candidate model evaluations.

Most heuristic search algorithms define three search steps: arc addition, arc deletion and arc reversal. With this, it is possible to navigate part of the search space considering neighboring structures that differ from a given one in one of these operations, and rely on a decomposable score that allows for local changes between different candidate models to easily assess the change that applying those operations would produce (in fact, only one local score addend changes for addition or deletion and two for reversal). This direction was fully developed in Chickering et al. [1995] and Buntine [1991]. If the score satisfies the equivalence property, searching in the space of undirected acyclic graphs is also possible.

In literature, structural search pioneering efforts can be traced to Chow and Liu [1968] for learning tree-restricted BN topologies. In Cooper and Herskovits [1992], a widely used and first structurally unbounded algorithm is published. The K2 algorithm only required a node order for the variables to be provided, but it is not robust as it achieves different networks for different orderings. Still, some works on the optimal detection of orderings are Larrañaga et al. [1996] and Tabar [2017] and Aghdam et al. [2019] of recent date. The greedy equivalent search (GES) (Chickering [2002]) algorithm searches in the space of equivalence classes. It was shown that this algorithm correctly recovers the structure if the data was sampled from a PGM (with or without directed arcs) when $n \to \infty$. Another widely used method is

the hill-climbing structural learning algorithm (Tsamardinos et al. [2006]) which, adapting from the well known hill-climbing optimization technique, performs a local greedy search in the space of DAGs.

3.2.4 Bayesian network classifiers

In a supervised classification setting, we use two differentiated sets of variables, the class variables $\mathcal{C} \subset \mathcal{X}$ and the feature variables $\mathcal{X}_f \subset \mathcal{X}$. We are given a dataset of annotated examples $\mathcal{D}_n = \{(\mathbf{x}_1, \mathbf{c}_1), \dots (\mathbf{x}_n, \mathbf{c}_n)\}$ and our goal is to maximize the quality of new predictions for variables in $\mathcal{C}$, using the information provided by the feature variables $\mathcal{X}_f$. In our case, $\mathcal{X}_f \cup \mathcal{C} = \mathcal{X}$ and $\mathcal{D}_n$ is a dataset with no missing values.

A Bayesian network classifier, in our case, uses MPE queries to answer new predictions and treats the information obtained from the feature variables as evidence for the query. Equations (3.5) and (3.6) illustrate typical target functions of our building algorithms. In general, a Bayesian network classifier (BNC) offers interpretability and explainability over other models while maintaining a competent performance in metrics like misclassification error.

3.2.4.1 Naive Bayes and extensions

The most popular Bayesian classifier is also the simplest. A naive Bayes classifier (NB) (Minsky [1961]) encodes the assumption that the features are conditionally independent given the class variable (a single class variable). It has a fixed structure where arcs are set to go from the class variable to the feature variables (see Figure 3.5).

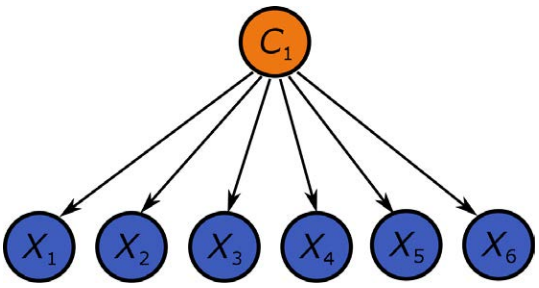


Figure 3.3: A naive Bayes classifier structure.

The factorization of the joint probability distribution offered by a naive Bayes, for $\mathcal{C} = \{C_1\}$ and $\mathcal{X}_f = \{X_1, \dots, X_{d-1}\}$ is:

$$p(C_1, X_1, \dots, X_{d-1}) = p(C_1) \prod_{i=1}^{d-1} p(X_i|C_1).$$

This can be seen as an attractive decomposition in terms of simplicity. For inference purposes, the particular case of naive Bayes using MPE rule is defined as:

$$c^* = \arg\max_{c_1} p(C_1 = c_1|\mathbf{x}).$$

This equation shows that for each MPE query, we must only examine as many cases as values in $Val(C_1)$ for a discrete network. In general, inference can be carried out in linear time for the naive Bayes case.

Since the structure is fixed, there is no need for a structural learning stage. Parameter estimation is performed in two possible ways: as described for the general case, but applied to a simple structure, or in a discriminative way, finding the parameters that yield the lowest missclassification rate. In general, the naive Bayes model is considered to be a non-demanding computational method with surprisingly good performance for its simplicity and strong assumptions. They have a history of success in the early stages of artificial intelligence (Gorry and Barnett [1968] and Warner et al. [1961]), with the notable case in De Dombal et al. [1972] where the model significantly outerperformed experts in diagnosing acute abdominal pain. On the other hand, naive Bayes has shown to be incapable of capturing complex patterns in the data. The XOR problem cannot be solved by a naive Bayes classifier (Ling and Zhang [2002]) and the decision boundary of a naive Bayes classifier is a hyperplane in the binary case (Minsky [1961]) (that is, when the class variable has two possible labels), and is a sum of polynomials in the arbitrary case (Duda et al. [2012] and Varando et al. [2015]). For this, nowadays, its use alone is often discarded in favor or lower-bias models.

The naive Bayes classifier has spawned numerous research directions and numerous extensions to the base model. Most known works include the tree augmented naive Bayes (Friedman et al. [1997]) commonly known as TAN, that relaxes the constrains on the feature subgraph to allow for tree type structures (Figure 3.4).

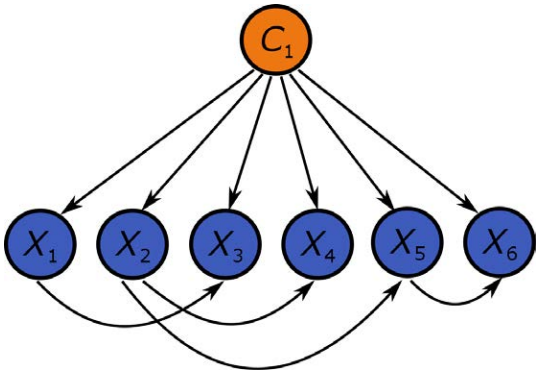


Figure 3.4: A tree-augmented naive Bayes classifier structure.

The k-dependence Bayesian classifier (Sahami [1996]) imposes less restrictive constrains by only allowing acyclic structures in the feature nodes within a bounded indegree $i_g = k$, that is, the number of parents for a feature node on the resultant network can be at most $k + 1$ (the class adds one parent). The Bayesian network augmented naive Bayes (Friedman et al. [1997]), commonly known as BAN, allows an unrestricted graph in the feature variables. For other extensions of the naive Bayes classifier and more, the reader is directed to Bielza and Larrañaga [2014a].

3.2.4.2 Multidimensional BN classifiers

In the general case, we are not bounded by a single class variable or a fixed structure. Depending on the number of class variables, the problem can be considered as a unidimensional classification problem (or simply a classification problem) when $|\mathcal{C}| = 1$ or a multidimensional classification problem (Bielza et al. [2011]) if $|\mathcal{C}| = s$, $s > 1$. The structure of a general classifier, however, is still bounded by some strict restrictions. Namely, a class variable should not have feature variables as parents, and the classifier

should not contain feature variables that, for structural reasons, cannot affect the classification outcome in any possible case.

Therefore, in all cases our MPE queries are tasked with returning the label assignment $\mathbf{c}$ that maximizes the posterior probability

$$\mathbf{c}^* = (c_1^*, \ldots, c_s^*) = \arg \max_{c_1,\ldots,c_s} p(C_1 = c_1, \ldots, C_s = c_s|\mathbf{x}).$$

In multidimensional BN classifiers (MBCs) (Van Der Gaag and De Waal [2006]), then, we seek to answer simultaneously a label assignment to multiple class variables. In here, class variables may have other class variables as parents, but since inference complexity scales with structural complexity, many works using restricted structures exist. In general multidimensional problems, the simplest case is to consider an empty structure in the class subgraph although no dependence relationship between class variables is modeled (Godbole and Sarawagi [2004] and Zhang and Zhou [2005] for binary classes in a multi-label fashion). An important subsequent work (Read et al. [2009]) used chain type structures to build a multidimensional classifier. In this setting, after a class variable is selected as part of the chain, it becomes part of the evidence, along with the features, to select a new member of the chain among the remaining class variables. For BNs specific results, other popular topologically restricted model proposals are tree-tree MBC (Van Der Gaag and De Waal [2006]) (MBCs that follow a tree structure both in class and feature subgraphs), polytree-polytree MBC (De Waal and Van Der Gaag [2007]) (MBCs that follow a polytree structure both in class and feature subgraphs), a special DAG-DAG MBC (Rodríguez and Lozano [2008]) (MBCs that follow a bounded indegree DAG structure both in class and feature subgraphs) and general structures in Bielza et al. [2011]. In Chapter 7 of this disseration, we focus on the class bridge-decomposable proposal, first presented in Bielza et al. [2011] and further developed in Borchani et al. [2010].

In order to learn the structure of an MBC, algorithms fall within three categories: Filter, wrapper and hybrid (Bielza et al. [2011]). In all cases, we refer to greedy score+search algorithms. A filter approach allows for a faster computational time by scoring the network independently of the classification performance, looking for a good structure according to some other criteria. A wrapper approach is computationally expensive, but yields better results for classification. Wrapper algorithms assess how changes in arc inclusion, deletion or reversal affect the misclassification error of the resultant network, requiring a MAP/MPE query at each step. Hybrid strategies use for some parts of the network a filter score and for others a wrapper score, somewhat averaging the pros and cons of both approaches.

In a more general view, many strategies that involve the treatment of multiple class variables have been proposed without the explicit dependence of a Bayesian network classifier, and have been adapted to the particular case of the BN domain. A multi-label classification complete survey is provided in Gibaja and Ventura [2014].

Chapter 4

Ensembles of classifiers and random forests

4.1 Combining classifiers

The idea of combining classifiers in machine learning can be traced back to Kearns [1988] and Kearns and Valiant [1993] with the question "*Can a set of weak learners create a single strong learner?*". The question was answered positively in Schapire [1990] with the creation of the first version of boosting. Boosting (Schapire [1990], Freund [1995], Freund and Schapire [1997] and Schapire [2003]) is an ensemble of learners that generally works by iteratively training a population of classifiers, each using a dataset that emphasizes the mistakes made by the previous members of the population. Decisions are made by combining the predictions of the learners, typically by majority voting in classification and averaging in regression. Of all boosting algorithms (Zhou [2012]), the most notorious is the AdaBoost algorithm (Freund and Schapire [1997] and Friedman et al. [2000]), which improves on previous versions by adapting to the weak learners. In Friedman et al. [2000], AdaBoost adjusts the distribution for the next learner by minimizing

$$EE(h|f,\mathcal{D}_n) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_n}[e^{-f(\mathbf{x})h(\mathbf{x})}], \tag{4.1}$$

where $EE(h|\mathcal{D}_n, f)$ is the Adaboost exponential loss function, $f(\mathbf{x})$ is the true distribution (in practice, our labeled examples) and $h(\mathbf{x})$ is a weak learner from the set $\mathcal{H} = \{h_1, \ldots, h_t\}$ and then combines the t predictions of $\mathbf{x}$ by additive weighting

$$\sum_{i=1}^{t} w_i h_i(\mathbf{x}), \tag{4.2}$$

where predictions by learners h_i are weighted according to some weighting scheme $w_1, \ldots, w_t$.

That is, after a base learner is trained and its error is measured, the probabilities of the samples that comprise the next dataset for the next weak learner to learn are updated using Equation (4.1). When all learners are trained, their predictions are combined using Equation (4.2). The weights paired to each learner depend on the individual error of each learner. In Friedman et al. [2000], this is calculated with the following expression:

$$w_i = \frac{1}{2}\log\left(\frac{1-err_i}{err_i}\right),$$

where err_i is the error of base learner h_i evaluated in its training dataset.

Adaboost in its most known form and many of its variants have been interpreted as procedures that perform gradient descent over the hypotheses space using a convex cost function (Mason et al. [2000]). Furthermore, adding random noise to classification has been shown to drastically decrease performance in boosting algorithms that fit this description (Long and Servedio [2010]). However, non-convex optimization algorithms for boosting have been proposed with successful response to this problem (Cheamanunkul et al. [2014]).

In general, boosting learners are viewed as high accuracy, overfitting resistant easy to implement procedures for classification and/or regression. With the mentioned caveat of vulnerability to noisy data in many of its variants.

4.1.1 Bagging

In the ensemble category of learners (also called metalearners), two clearly defined directions can be identified. The first one corresponds to boosting and its variants and can be thought of as the "sequential" approach to learners combination. The second one is bagging, and conversely, can be thought of as the "parallel" approach to the combination of learners. In this dissertation we focus on this approach in Chapter 8.

Bagging (Breiman [1996]), originating from Bootstrap AGGregatING, is a method for combining classifiers whose main steps are, as implied, bootstraping (Efron and Tibshirani [1994]) and aggregation. Given a dataset $\mathcal{D}_n$, we seek to train t learners, each on a bootstrapped version sampled from the original data.

Formally, we have a population of learners $\mathcal{H} = \{h_1, \ldots, h_t\}$ and $B(\mathcal{D}_n) = \{\mathcal{D}_1^*, \ldots, \mathcal{D}_{n^n}^*\}$ the set of all bootstrap variations of $\mathcal{D}_n$. Our goal is to compute

$$h^*(\mathbf{x}, \mathcal{D}_n) = \mathbb{E}_{B(\mathcal{D}_n)}[h(\mathbf{x}, B(\mathcal{D}_n), \mathcal{D}_n)]. \tag{4.3}$$

This sampling is typically done with replacement, which allows for the inclusion of repeated examples. The differences in training data produce differences in the final models that comprise the population of learners. Then, predictions from all models are combined typically by voting in classification and averaging in regression. In practice and in the general case, Equation (4.3) is evaluated using Monte Carlo simulation.

The theoretical understanding of why bagging works has produced significant literature. In Breiman [1996] and Dietterich [2000], it is shown that an ensemble strength can be characterized in terms of the individual accuracy of its members and the overall diversity of the population. An extreme case of why uncorrelated classifiers affects the outcome is the case where all members of the ensemble are copies of a single learner. In this setting, its clear that no improvement can be achieved by using an ensemble and the system degenerates to the task of training that single learner. On the other hand, if errors in the learners are independent of each other and the probability of each learner to be correct is $p > 1/2$, then for each learner predicting an instance incorrectly, a majority of others is expected to predict it correctly, effectively lowering the probability of error of the ensemble beyond that of the base learners (Figure 4.1).

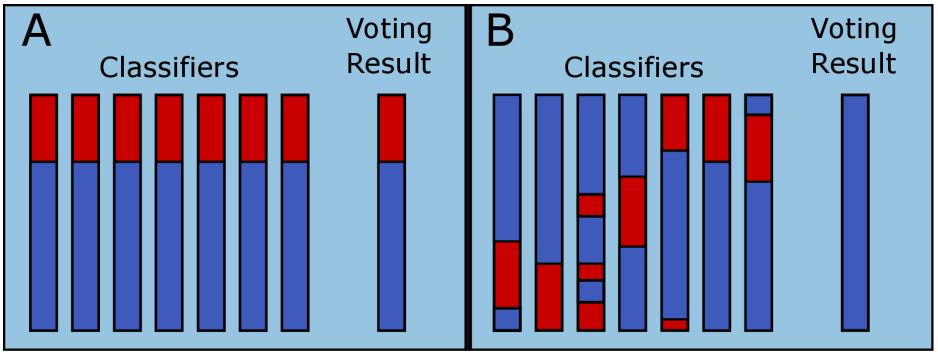


Figure 4.1: An intuitive view of why diversity and accuracy influence performance in an ensemble of classifiers. The colored bars represent the correct (blue) to incorrect (red) classifications of a population of classifiers and their distribution in a dataset. The colors of the "Voting result" bar are calculated by a majority vote. In case A, we show the less diverse error distribution scenario, where all classifiers are identical and so is the result. In case B, we can see how we can take advantage of uncorrelated errors to "compensate" for the mistakes of other classifiers, greatly reducing the error. Notice, however, that if the colors were reversed (more specifically, if the correct labels were the minority) the ensemble would instead amplify the error in case B.

In Friedman and Hall [2007] a smooth estimator is decomposed into terms of linear and superlinear orders (i.e., quadratic or cubic) and the effects of bagging analyzed. It was concluded that the linear order term of the estimator is roughly unaffected, but variance is reduced for the terms of superlinear orders. This is further detailed in Buja and Stuetzle [2000], which uses U-statistics to study the effects of bagging on variance, square bias and mean squared error (MSE), arriving at broadly similar conclusions. In Büchlmann and Yu [2002], bagging is characterized as a softhresholding function that is specially effective in reducing the MSE on non-smooth, unstable predictors, such as decision trees, whose decision boundary is comprised exclusively of hard cuts. Indeed, while practical success of bagging is a well documented fact (Breiman [2001], Valentini et al. [2003], Chen and Yu [2007], Biau and Devroye [2010], Zhang et al. [2010] and Syarif et al. [2012]), as we will see in Section 4.4, random forests is perhaps the most successful use of bagging.

In this dissertation we concern ourselves with the effect of bagging, specially in the cases where the weak learners are nearest neighbors (NN) and decision tree (DT) predictors.

4.2 Nearest neighbors

The nearest neighbors (NN) predictor (Fix and Hodges Jr [1951] and Fix and Hodges [1952]) is one of the oldest machine learning methods used to predict new examples from data. It is still used today as its simplicity, easy implementation and intuitive understanding make it an attractive proposal. A clear advantage of NN over other methods, besides its simplicity, is that it does not require training time and prediction is done in polynomial time. Additionally and perhaps due to its simplicity, it is considered a well understood method. For these reasons, it has not been deprecated from machine learning literature and continues to see practical use.

Formally, a k-NN predictor is a function $h : \mathcal{X} \to \mathcal{Y}$ that outputs a prediction over a domain $\mathcal{Y}$ using a distance metric $\|.\|_p$ and the variables of $\mathcal{X}$. In all cases, NN predictor uses the y-associated values of the $k \in \mathbb{N}$ neighboring datapoints to the target $\mathbf{x}_0$ to output a prediction, where "neighboring" is defined with respect to $\|.\|_p$. The properties of $\mathcal{Y}$ determine the type of problem, that is either classification or regression.

Let $(\mathbf{x}'_1, y'_1), (\mathbf{x}'_2, y'_2), \ldots, (\mathbf{x}'_n, y'_n)$ be an order for the data such that $||x'_1 - \mathbf{x}_0||_p < ||x'_2 - \mathbf{x}_0||_p < \ldots < ||x'_n - \mathbf{x}_0||_p$. Then, if we define $y_{knn} = \{y'_1, \ldots, y'_k\}$ as the set that contains the y-associated values of the k "closest " datapoints, we can write a prediction for classification as:

$$y^* = \arg\max_y \sum_{y'_i \epsilon y_{knn}} \mathbb{1}_{[y'_i = y]}$$

This can be read as the selection of the most popular label among the k selected instances.

For regression we have:

$$y^* = \frac{1}{k} \sum_{y'_i \epsilon y_{knn}} y'_i$$

which is simply the average of the y-associated values of the k selected instances.

NN predictors work well when the target function does not deviate largely from the assumption that datapoints close in distance have also close y-associated values. This can be intuitively seen for $k = 1$, as the decision boundary of the NN algorithm is a Voronoi diagram (Figure 4.2).

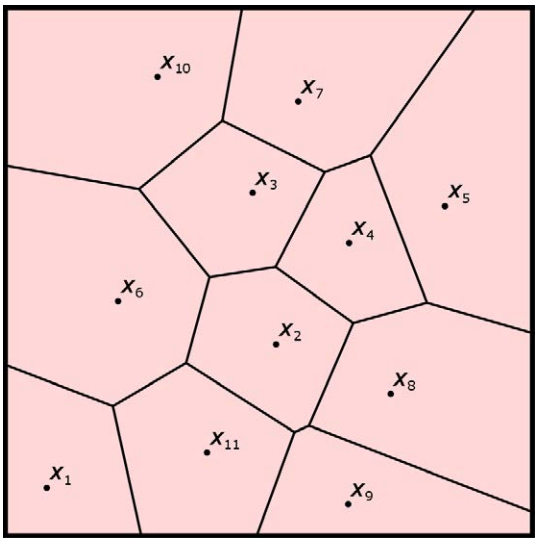


Figure 4.2: Voronoi cells representing the decision boundary of a 1-NN classifier/regressor in a 2-dimensional feature space. In this setting, if we were to select an arbitrary location of the diagram as the coordinates for a datapoint to be predicted, the y-associated value of the datapoint included in the Voronoi cell would be used for the prediction.

The first theoretical results on the NN algorithm are presented in Cover et al. [1967], where it is shown that the risk of a 1-NN predictor converges to double the Bayes optimal error under mild conditions. In Devroye et al. [1996] we can see a more detailed analysis and other results of interest. In Biau and Devroye [2010], the bagged NN is explored together with a known variation of interest in this dissertation, the k-potential nearest neighbors. For relatively recent convergence results, we have Biau et al. [2010]. Finally, in even more recent studies, Gottlieb et al. [2014] proposed a finite sample bound, with similarities to Vapnik-Chervonenkis bounds (or VC bounds, a well-known measure of the expressive power of a learning model)

In Chapter 8, we will examine the case of bagging the NN predictor.

4.3 Decision trees

Formally, a decision tree is a function $h : \mathcal{X} \to \mathcal{Y}$ that outputs a prediction over a domain $\mathcal{Y}$ using a splitting rule, a stopping criterion, a prunning criterion (in some cases) and the variables of $\mathcal{X}$. A decision tree is built using an associate tree structure that sorts new arriving examples based on the splitting rule (also referred to as splitting criterion). At each bifurcation of the tree, a Boolean criterion is applied to a direction of the feature space and datapoints are separated into different paths depending on the outcome. The final result can be interpreted as a recursive application of "IF THEN ELSE" rules that encode the pattern of our predictive model (Figure 4.3). At the leaves of the tree, a predictive value is assigned to new arriving examples based on the type of problem. As in Section 4.2, the properties of $\mathcal{Y}$ determine the type of problem: If $\mathcal{Y}$ is a discrete set of values we use decision trees for classification; if it is continuous, for regression.

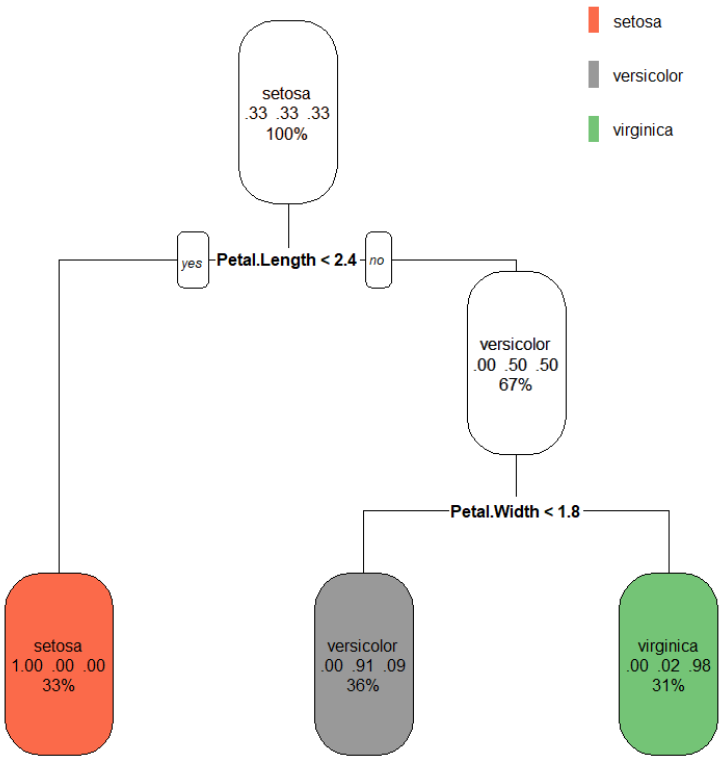


Figure 4.3: Example of a decision tree for classification for the well-known `iris` dataset (Fisher [1936]). It displays the classification of three different species of iris flowers based on various length measurements of each population. At each node, we see the most popular label at that level, the proportion of each label in the data and the proportion of the data that reaches the node.

4.3.1 Building decision trees

Decision trees can be built with the aid of human experts. However, for machine learning purposes, we are interested in algorithms that can build decision trees automatically from data. Algorithms for this endeavor must decide on the different aspects that compose a decision tree. The main design choices that characterize the different existing proposals are: The type of value assignment to instances at the leaves, the type of splitting criterion, the type of stopping criterion and the type of pruning criterion.

The type of value assignment defines the way instances are labeled at the leaves. The available choices are typically separated for classification and regression trees. A splitting rule typically consists of the recursive application of a splitting criterion to subsequently smaller partitions of the dataset until the terminal nodes (leaves) are reached, leaving a subset of instances that are used for the final value assignment. The stopping criterion halts the depth exploration (branching) of the tree when a local criterion on the resultant partitions of the split is satisfied. Finally, the pruning criterion seeks to reduce the dimensions of the final tree by merging terminal nodes.

We review now the most relevant proposals in literature.

4.3.1.1 Value assignment to a prediction

For value assignments, we follow a similar approach as in the NN case.

Let us define $\mathcal{U} = \{u_1, u_2, \ldots\}$ as the set of leaves of a decision tree, $u_i(\mathcal{D}_n) \subset \mathcal{D}_n$ as the partition of the dataset that only contains the datapoints in leaf u_i. Then, for a given leaf u_i and a datapoint $\mathbf{x}_0$ to predict, the label assignment y^* for classification can be written as:

$$y^* = \arg\max_y \sum_{(\mathbf{x}_i, y_i) \in u_i(\mathcal{D}_n)} \mathbb{1}_{[y_i = y]}.$$

This amounts to select the most popular label in $u_i(\mathcal{D}_n)$. For regression, we have:

$$y^* = \frac{1}{|u_i(\mathcal{D}_n)|} \sum_{(\mathbf{x}_i, y_i) \in u_i(\mathcal{D}_n)} y_i.$$

Thus, the most used value assignment is simply the average value of the y-associated values of the datapoints in the leaves.

4.3.1.2 Types of splitting criteria

At any given node, we can define $\mathcal{S} = \{s_1, s_2, \ldots\}$ as the set of all candidate splittings, and for each $s_i \in \mathcal{S}$ we have s_{i1} and s_{i2} as the two resultant subsets of the data after splitting it at s_i. Then, all splitting criteria can be regarded as scoring functions that assess the candidates in $\mathcal{S}$. In order to select the next cut point s^*, our task is generally to identify the candidate with the minimum (or maximum) score. Most notable splitting criteria are:

1. Misclassification error:

$$MCE(s_p) = \min_y \frac{1}{n_{s_{p1}}} \sum_{(\mathbf{x}_i, y_i) \in s_{p1}} \mathbb{1}_{[y_i \neq y]} + \min_y \frac{1}{n_{s_{p2}}} \sum_{(\mathbf{x}_i, y_i) \in s_{p2}} \mathbb{1}_{[y_i \neq y]},$$

   where $n_{s_{p1}}$ and $n_{s_{p2}}$ are the number of datapoints contained in partitions s_{p1} and s_{p2}, respectively. Our task is then to find s^* such that

$$s^* = \arg\min_{s_p \in \mathcal{S}} MCE(s_p),$$

   This criterion simply measures the error in classification that we commit by choosing certain split. It is considered a simple split and can be replaced by either the Gini impurity or the information gain splitting criteria in most cases.

2. Gini impurity: Let us define $p(i, \mathcal{D}_n)$, in this context, as the proportion of datapoints with y-associated label i in a dataset $\mathcal{D}_n$. Then the Gini impurity (Breiman et al. [1984]) can be written as:

$$GI(s_p) = \left(1 - \sum_{i=1}^{l} p(i, s_p)^2\right) - \left(\frac{n_{s_{p1}}}{n_{s_p}}\left(1 - \sum_{i=1}^{l} p(i, s_{p1})^2\right) + \frac{n_{s_{p2}}}{n_{s_p}}\left(1 - \sum_{i=1}^{l} p(i, s_{p2})^2\right)\right),$$

where l is the total number of labels of y in $\mathcal{D}_n$, $n_{s_p} = n_{s_{p1}} + n_{s_{p2}}$ and s^* is obtained with

$$s^* = \arg\min_{s_p \in \mathcal{S}} GI(s_p).$$

It can be interpreted as an impurity measure that measures the divergences of the label probability distributions. It is minimized (with score 0) when the partition of the data to split contains only members of one class.

3. Information gain: Used in the ID3 and C4.5 algorithms (Quinlan [1993]), it can be written as:

$$\begin{aligned} IG(s_p) \quad = \quad & \sum_{i=1}^{l} p(i, s_p) \log_2(p(i, s_p)) \\ & - \left(\frac{n_{s_{p1}}}{n_{s_p}} \sum_{i=1}^{l} p(i, s_{p1}) \log_2(p(i, s_{p1})) + \frac{n_{s_{p2}}}{n_{s_p}} \sum_{i=1}^{l} p(i, s_{p2}) \log_2(p(i, s_{p2}))\right), \end{aligned}$$

and our target s^* can be expressed as

$$s^* = \arg\min_{s_p \in \mathcal{S}} IG(s_p).$$

It seeks to separate the data based on the definition of entropy in information theory. While it stems from a different field of study that the Gini index, some reports suggest that differences between the Gini index and information gain are small (Raileanu and Stoffel [2004]) and can, in most cases, be used interchangeably.

4. Sum of squared errors: For regression, the most commonly used splitting criterion is to minimize the sum of squared errors (Breiman et al. [1984]):

$$SSE(s_p) = \frac{1}{n_{s_p}} \sum_{(\mathbf{x}_i, y_i) \in s_p} (y_i - \overline{y}_p)^2 - \left(\frac{1}{n_{s_{p1}}} \sum_{(\mathbf{x}_i, y_i) \in s_{p1}} (y_i - \overline{y}_{p1})^2 + \frac{1}{n_{s_{p2}}} \sum_{(\mathbf{x}_i, y_i) \in s_{p2}} (y_i - \overline{y}_{p2})^2\right)$$

where $\overline{y}_p, \overline{y}_{p1}$ and $\overline{y}_{p2}$ are the means of the y-values in s_p, s_{p1} and s_{p2}, respectively. Thus we are looking for s^* such that

$$s^* = \arg\min_{s_p \in \mathcal{S}} SSE(s_p).$$

It can be seen as a variance reduction technique, grouping datapoints by the similarity of their y-associated values.

5. Random split: This splitting criterion simply picks a random possible split as s^*.

4.3.1.3 Types of stopping criteria and pruning

The stopping criterion is a condition that regulates the termination of the recursive exploration of splits. A common instance of this is to define a fixed maximum depth for the tree to grow, always producing trees of that depth or less. Another existing solution is to apply a pruning algorithm to a fully built tree (Breiman et al. [1984] and Quinlan [1993]). Decision trees that suffer from excessive size run the risk of overfitting, while on the contrary small trees may suffer from underfitting or high bias. Pruning large decision trees can lead to some improvements in generalization error. This is typically done in a path from the leaves to the root, but algorithms that travel the tree in the opposite direction also exist. After a decision tree is built until it has few instances per leaf, the algorithm searches to remove leaves that do not contribute positively to the performance of the model.

Finally, another alternative is to limit the number of instances that we allow at the leaves of a tree. In Breiman et al. [1984], CART trees use a parameter $k \in \mathbb{N}$ to stop splitting when the leaves contain k or fewer instances. Typically, $k = 1$ is used for classification and $k = 5$ for regression. This stopping criterion enjoys a very important theoretical property that will be discussed at length in Chapter 8. It is our preferred choice for this dissertation.

4.3.2 Decision trees in literature

Literature on decision tree is extensive and reveals different stages of development for the model. Modern decision trees are shown in Breiman et al. [1984] and Hastie et al. [2009]. Most prominent algorithms for learning decision trees are the CART algorithm (Breiman et al. [1984]) and the ID3 and C4.5 algorithms (Quinlan [1993]). In the CART algorithm, trees are grown using the Gini splitting criterion and stopping mechanism is equipped by a type of pruning known as cost-complexity pruning. It works for both classification and regression, in which case the sum of squared errors splitting criterion is employed. ID3 grows trees aggressively until all instances in the leaf are of the same class (or there is no information gain), using the information gain splitting criterion. C4.5 improves over the previous algorithm by using gain ratio as splitting criteria (Quinlan [1993]), thresholding on the number of instances for stopping criteria and including error-based pruning. For other decision tree building algorithms the reader is directed to Lim et al. [2000]. All algorithms reviewed here and most algorithms in literature greedily search for the optimal split, while exhaustive search for the optimal tree has been proven to be an NP-hard problem even in restricted settings (Hancock et al. [1996], Laurent and Rivest [1976] and Naumov [1991]).

4.4 Random forests

Random forests (RFs) (Breiman [2001]) are an ensemble of randomized classification or regression trees. Each tree is randomized by the use of bootstrapping in the training set and by a mechanism known as random subspace selection (RSS). RSS introduces variability between decision trees by sampling, at each node in the building process of the tree, p out of d features (with $p \leq d$) that are then used to search for the optimal splitting point. A conventional value for a RF is $p = \sqrt{d}$ in classification, and $p/3$ for regression. Predictions are then combined by voting in classification or averaging in regression.

Formally, we can model the randomization of each tree using a random variable Φ. Then a RF is a predictor formed by a set of trees $\mathcal{T} = \{h(\mathbf{x}, \phi_1, \mathcal{D}_n), \ldots, h(\mathbf{x}, \phi_t, \mathcal{D}_n)\}, t \in \mathbb{N}$, and we can express a

prediction from the RF estimate as:

$$f_{RF}(\mathbf{x}, \mathcal{D}_n) = \mathbb{E}_{\Phi}[h(\mathbf{x}, \Phi, \mathcal{D}_n)]. \tag{4.4}$$

In here, Φ contains all sources of randomness in the construction of each tree, namely, bootstrapping of the data, the splits to perform and the random subspace method choices. We then calculate our expectation with respect to the population of DT learners generated by Φ.

In practice and up to this point, Equation (4.4) is evaluated using Monte Carlo simulation. That is, algorithmically generating a number $t \in \mathbb{N}$ of trees and computing the prediction of each tree independently of the others. We now depict a general algorithm (see Algorithm 4.1), compatible with many versions of a RF using binary trees, for the prediction of a datapoint $\mathbf{x}_0$ in more detail.

Algorithm 4.1 Calculate RF

Require: $\mathcal{D}_n$, t, $\mathbf{x}_0$, C_s // The data, the total number of trees, the point to predict and the splitting criterion, respectively.

Initialize $P = \varnothing$ // Where P is the the list of predictions made by the trees
for $i = 1$ to t **do**
  $\mathcal{D}_i^*$:= sample_with_replacement$(n, \mathcal{D}_n)$ // Bootstrap sampling
  R_s := push$(\mathcal{D}_i^*)$ // A stack to keep track of the splitted subregions
  $R_f := \varnothing$ // A list with the final leaves of the tree
  while !is_empty(R_s) **do**
    R_a := pop(R_s)
    if stopping_criterion_fulfilled(R_a) // Here we can plug any of the different stopping criteria of a decision tree **then**
      R_f := add(R_a, R_f)
    else
      $\mathcal{S}$:= select_directions_to_split(R_a) // Here we implement random subspace method or other schemes to select the directions to split
      (R_{a1}, R_{a2}) := Split$(R_a, \mathcal{S}, C_s)$ // Here we find the best splitting point in R_a according to the subset of directions $\mathcal{S}$ and the splitting criterion C_s, and cut the dataset into two parts.
      R_s := push(R_{a1})
      R_s := push(R_{a2})
    end if
  end while
  u := Locate_leaf_for_prediction$(\mathbf{x}_0, R_f)$ // After the tree is trained, we can assign a leaf to the datapoint to predict
  p_i :=predict(u) // In order to obtain a prediction, here we can plug any of the value assignment options for a decision tree
  P := add(p_i, P) // We store the predictions made by this tree
end for
Output: Combine_predictions(P) // We can combine the predictions here using any scheme suited for RFs. Typically voting for classification and averaging for regression

RF is one of the most successful methods in machine learning (Howard and Bowles [2012]). Many versions of it exhibit state-of-the-art performance, can handle well large datasets even at low sample sizes, can be used to estimate variable importance, have a relatively low number of parameters for an ensemble, generally have high accuracy and run in polynomial times. There are numerous reports of success in

practical applications, such as Svetnik et al. [2003], Prasad et al. [2006], Cutler et al. [2007], Díaz-Uriarte and Alvarez de Andrés [2006] and Shotton et al. [2011]. The reader is directed to Criminisi et al. [2012] and Boulesteix et al. [2012] for two application-focused state-of-the-art reviews. The theoretical aspects of RFs, however, still remain under active investigation, as it is considered a not-well-enough understood model. In this dissertation we concern ourselves with the theoretical developments of RFs and review some of them in detail in Chapter 8.

Since its inception, and specially given its success and popularity, many extensions of RFs are available. While one could argue that using any combination of all design choices detailed in Section 4.3 for the base trees would produce at least some different types of RFs, we focus on prominent results in literature. In Geurts et al. [2006], the extra-tree algorithm randomly samples a subset of all split points that can be performed and a given node and then searches for the optimal split in the fashion of Breiman's CART trees. In the original work (Breiman [2001]), a variant where the split can consist of linear combinations of features is proposed. In Ziegler et al. [2010], a fast version of the algorithm, known as random jungle, was implemented as a response to concerns in parameter tuning procedures, previously studied in Díaz-Uriarte and Alvarez de Andrés [2006], Bernard et al. [2008] and Genuer et al. [2010].

Motivated by the need to perform theoretical studies on the model, simplified versions of RFs have been proposed in literature. Centered forest (Breiman [2004]) is a type of RF that ignores bootstraping, sets $p = 1$ for RSS and splits the data at the center of the range of the selected coordinate, with a stopping criterion of k or less datapoints per leaf. They were studied in Biau et al. [2008], Scornet [2016] and Biau [2012]. A similar approach, but swapping centered splits for empirical median splits, is discussed in Scornet [2016]. In Lin and Jeon [2006], RF omits bootstrapping for analysis purposes. In Cutler and Zhao [2001] the PERT-perfect trees can also be thought of as a simplification, since they switch the adaptive splitting criteria in the original CART trees for a purely random non-adaptive one. In Arlot and Genuer [2014], it is shown how a simplified version of a RF model can be viewed as a kernel estimate, also exploring a connection between RFs and kernel estimation that was first mentioned in Breiman [2000].

Finally, some extensions seek to augment the functionality of the base algorithm. In Winham et al. [2013] trees are weighted according to their accuracy in prediction. In a related approach, Bernard et al. [2012] defined tree building process designed so that newly created trees perform better where the previous ones were lacking. In Saffari et al. [2009], Denil et al. [2013], Lakshminarayanan et al. [2014] and Yi et al. [2012] RFs are equipped with online learning capabilities, that is, the ability to incorporate newly generated instances of data to the existing training set to further improve prediction capabilities. In Ishwaran et al. [2008], Yang et al. [2010] and Ishwaran et al. [2011], the extension is to the domain of survival analysis.

Overall, the literature surrounding RFs is quite extensive and many alternatives to the types of RFs presented here can be found. For an interesting and more complete overview, Biau and Scornet [2016] summarizes well the current situation.

Part III

CONTRIBUTIONS

Chapter 5

Univariate and bivariate truncated von Mises distributions

5.1 Introduction

The von Mises distribution has received undisputed attention in the field of directional statistics (Jupp and Mardia [1989]) and in other areas like supervised classification (López-Cruz et al. [2015]). Thanks to desirable properties such as its symmetry, mathematical tractability and convergence to the wrapped normal distribution (Mardia and Jupp [2000]) for high concentrations, it is a viable option for many statistical analyses. However, angular phenomena may present constraints on the outcomes that are not properly accounted for by the density function of the von Mises probability distribution. Thus, a truncated distribution with the capabilities of the von Mises distribution is strongly suggested. Additionally, there is hardly any literature in this direction, and to the best of our knowledge, only one paper, Bistrian and Iakob [2008], proposes a definition of the truncated von Mises distribution.

In this chapter, we propose a new definition of a truncated probability distribution, whose parent distribution is the von Mises distribution, for angular values. The univariate and bivariate cases of this distribution are explicitly developed.

Section 5.2 introduces the definition for the univariate case and derives some properties of the distribution, calculates the maximum likelihood estimators of the parameters and studies the distribution moments. Section 5.3 addresses the definition of the bivariate truncated von Mises, maximum likelihood estimation of the parameters and the definition and study of the conditional and marginal truncated distributions. Section 5.4 shows a real data application where this distribution successfully models the data. Finally, Section 5.5 discusses the summary and conclusions.

5.2 Univariate truncated von Mises distribution

Definition 5.2.1. The truncated von Mises distribution is presented as a four-parameter generalization of the non-truncated case for truncation parameters a, b as

$$f_{tvM}(\theta;\mu,\kappa,a,b) = \begin{cases} \frac{e^{\kappa\cos(\theta-\mu)}}{\int_a^b e^{\kappa\cos(\theta-\mu)}d\theta} & \text{if } \theta \in \mathbb{O}_{a,b} \\ \\ 0 & \text{if } \theta \in \mathbb{O}_{b,a} \end{cases}$$

where $\mu \in \mathbb{O}$ is the location parameter, $\kappa > 0$ the concentration parameter, $\mathbb{O}$ is the circular set of points ($\mathbb{O} : (x, y)$ such that $x^2 + y^2 = 1$), $\mathbb{O}_{a,b} \subset \mathbb{O}$ is obtained by selecting the points in the circular path from $a \in \mathbb{O}$ to $b \in \mathbb{O}$ in the preferred direction (counterclockwise) and $\mathbb{O}_{b,a}$ is its counterpart w.r.t. $\mathbb{O}$.

Our proposed definition differs from Bistrian and Iakob [2008] in the circular definition of the truncation parameters, not bounded to a linear definition involving the location parameter. The additional developments covered in this article can also be considered a novelty.

To illustrate the differences with the non-truncated case for these parameters, Figure 5.1 represents multiple examples of truncated von Mises distributions.

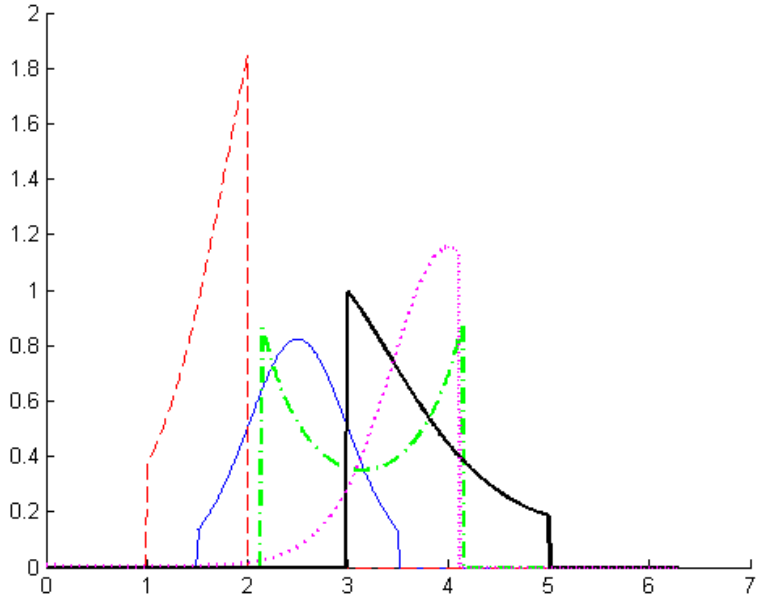


Figure 5.1: Several truncated von Mises distributions. Symmetrical function with maxima not at the extrema (thin continuous line), strictly increasing function (dashed line), strictly decreasing function (thick continuous line), unique critical point that is a minimum (dash-dot line) and two critical points, a maximum and a minimum (dotted line).

It is a well-known result (Abramowitz and Stegun [1964]) that $2\pi I_0(\kappa) = \int_0^{2\pi} e^{\kappa\cos(\theta-\mu)}d\theta$, where $I_0(\kappa)$ is the modified Bessel function of the first kind and order 0, that is,

$$I_0(\kappa) = \sum_{m=0}^{\infty} \frac{x^{2m}}{(m!)^2 2^m}.$$

The above expression suffices for truncation parameters a, b such that $\mathbb{O}_{a,b} = \mathbb{O}$. However, it is necessary

to calculate the general case for non-restricted truncation parameters. Taking $w = \lfloor \frac{n}{2} \rfloor + \mod \frac{n}{2} - 1$, we have obtained:

Lemma 5.2.1. $\int_a^b e^{\kappa \cos(\theta-\mu)} d\theta = I(b;\mu,\kappa) - I(a;\mu,\kappa)$, where

$$I(\theta;\mu,\kappa) = \sum_{n=0}^{\infty} \frac{\kappa^n}{n!} \left(\sin(\theta-\mu) \sum_{i=0}^{w} \left(\cos^{n-2i-1}(\theta-\mu) \prod_{j=0}^{2i} (n-j)^{-(-1)^j} \right) + \frac{((-1)^n+1) \prod_{j=0}^{w} (n-j)^{-(-1)^j} (\theta-\mu)}{2} \right).$$

$I(\theta;\mu,\kappa)$ is the distribution function of the positive support of the truncated von Mises density. (Note then that while truncation parameters are circular quantities, the values for the integration coefficients are linear)

Proof. See Appendix A. □

5.2.1 Maximum likelihood estimation

Provided we have a sample of observations $\theta_1, \theta_2, \ldots, \theta_n$ from a truncated von Mises distribution (1), we obtain:

$$\begin{aligned} \ln L(\mu,\kappa,a,b;\theta_1,\theta_2,\ldots,\theta_n) &= \sum_{i=1}^{n} \ln \left(\frac{e^{\kappa \cos(\theta_i-\mu)}}{\int_a^b e^{\kappa \cos(\theta-\mu)} d\theta} \right) \\ &= \sum_{i=1}^{n} \kappa \cos(\theta_i-\mu) - n \ln \left(\int_a^b e^{\kappa \cos(\theta-\mu)} d\theta \right) \end{aligned}$$

where $\ln L(\mu,\kappa,a,b;\theta_1,\theta_2,\ldots,\theta_n)$ is the log-likelihood function for the truncated von Mises distribution.

We now seek to solve the system of four log-likelihood equations created by the four parameters of the distribution. For parameters μ, κ, we have

$$\frac{\partial \ln L}{\partial \mu} = 0$$
$$\frac{\partial \ln L}{\partial \kappa} = 0.$$

As parameters a, b, define the region of the greater-than-zero density, we find that all $\theta_1, \ldots, \theta_n$ observations necessarily lie within the subset $\mathbb{O}_{a,b}$. This, together with the $-n \ln \left(\int_a^b e^{\kappa \cos(\theta-\mu)} d\theta \right)$ sub term of (3), allows us to isolate the estimators

$$\mathbb{O}_{\hat{a},\hat{b}} = \underset{a,b}{\operatorname{argmax}} (\max(\{A(\mathbb{O}_{\theta'_1,\theta'_2}), \ldots, A(\mathbb{O}_{\theta'_{n-1},\theta'_n}), A(\mathbb{O}_{\theta'_n,\theta'_1})\})),$$

where $A(\mathbb{O}_{\theta_1,\theta_2})$ is the angle between θ_1 and θ_2, and $\{\theta'_1, \ldots, \theta'_n\}$ is the sample sorted in ascending

order. Intuitively, the truncation parameters are separated by the largest angle and are contiguous in a sorted finite circular sample.

From this result, we can say that the truncation parameters of the truncated von Mises distribution have population-only dependent maximum likelihood estimators. For parameters μ and κ, interdependency is a consequence of the possibly non-symmetrical shape of the distribution. If we observe the expressions

$$\frac{1}{n}\sum_{i=1}^{n}\sin(\theta_i-\mu)-\frac{e^{\kappa\cos(a-\mu)}-e^{\kappa\cos(b-\mu)}}{\int_a^b e^{\kappa\cos(\theta-\mu)}d\theta}=0$$
$$\frac{1}{n}\sum_{i=1}^{n}\cos(\theta_i-\mu)-\frac{\int_a^b\cos(\theta-\mu)e^{\kappa\cos(\theta-\mu)}d\theta}{\int_a^b e^{\kappa\cos(\theta-\mu)}d\theta}=0,$$

$e^{\kappa\cos(a-\mu)}-e^{\kappa\cos(b-\mu)}=0$ holds if a,b are symmetrical w.r.t. μ, reducing the location parameter estimator to that of the non-truncated case (Mardia and Jupp [2000]), the circular sample mean $\hat{\mu}$. As no population-only dependent expressions of the parameters μ and κ were found, optimization techniques to maximize the log-likelihood function for those parameters are needed.

5.2.2 Moments

The moments in circular statistics are particular values of the characteristic function. The r-th moment about a direction d can be written as

$$m_{r_{tvM}}=\mathbb{E}[e^{ir(X-d)}].$$

The first moment about the 0 direction for the truncated von Mises is calculated as

$$m_{1_{tvM}}=\frac{\int_a^b\cos(\theta)e^{\kappa\cos(\theta-\mu)}d\theta}{\int_a^b e^{\kappa\cos(\theta-\mu)}d\theta}+\frac{i\int_a^b\sin(\theta)e^{\kappa\cos(\theta-\mu)}d\theta}{\int_a^b e^{\kappa\cos(\theta-\mu)}d\theta},$$

and we can relate (5) to the first moment about the μ direction, denoted as $m'_{1_{tvM}}$ as

$$m_{1_{tvM}}=e^{i\mu}m'_{1_{tvM}}. \tag{5.1}$$

Notice that if $\cos(a-\mu)=\cos(b-\mu)$, then $m'_{1_{tvM}}=\frac{\int_a^b\cos(x-\mu)e^{\kappa\cos(x-\mu)}d\theta}{\int_a^b e^{\kappa\cos(x-\mu)}d\theta}=R$, the mean resultant length of μ and thus $m_{1_{tvM}}=e^{i\mu}R$.

An alternative expression for $m_{1_{tvM}}$ can be found by considering equations $\mathbb{E}[\cos(x)]=R'\cos(\mu')$ and $\mathbb{E}[\sin(x)]=R'\sin(\mu')$, where R' and μ' are the sample mean resultant length and sample mean, respectively. We can then state

$$m_{1_{tvM}} = \mathbb{E}[\cos(x)]+i\mathbb{E}[\sin(x)]=R'\cos(\mu')+iR'\sin(\mu')=R'e^{i\mu'}. \tag{5.2}$$

Thus, merging Equations (5.1) and (5.2), we obtain

$$e^{i(\mu'-\mu)}R' = m'_{1_{tvM}},$$

which can be seen as a valuable expression as it contains the sample mean (μ') and the location parameter of the distribution (μ).

5.3 Bivariate truncated von Mises distribution

The non-truncated bivariate von Mises distribution was first proposed by Singh [2002] and extended and developed in Mardia et al. [2008] and Mardia and Voss [2014]. It is a unimodal/bi-modal function on the torus $f_{btvM} : \mathbb{O} \times \mathbb{O} \to \mathbb{R}$ obtained by replacing the quadratic and linear terms of the normal bivariate distribution with their circular analogues. This distribution is known as the "sin variant bivariate von Mises distribution" and is defined for dependent pairs of angular variables. It is expressed for variables θ_1 and θ_2, as

$$f(\theta_1, \theta_2) = Ce^{\kappa_1 \cos(\theta_1-\mu_1)+\kappa_2 \cos(\theta_2-\mu_2)+\lambda \sin(\theta_1-\mu_1)\sin(\theta_2-\mu_2)},$$

where $\kappa_1, \kappa_2 \geq 0, \lambda \in \mathbb{R}, \mu_1, \mu_2 \in \mathbb{O}$ and C is the normalization constant. We propose the density function for the truncated case as a nine-parameter function with density defined as follows:

Definition 5.3.1. The density function for the truncated case is a nine-parameter function with density

$$f_{btvM}(\theta_1, \theta_2; \boldsymbol{W}) = \begin{cases} \frac{f_{ubvM}(\theta_1,\theta_2;\boldsymbol{W})}{\int_{a_1}^{b_1}\int_{a_2}^{b_2} f_{ubvM}(\theta_1,\theta_2;\boldsymbol{W})d\theta_2 d\theta_1} & \text{if } \theta_1 \in \mathbb{O}_{a_1,b_1}, \theta_2 \in \mathbb{O}_{a_2,b_2}, \\ 0 & \text{otherwise} \end{cases}$$

where $\boldsymbol{W} = \{\lambda, \mu_1, \mu_2, \kappa_1, \kappa_2, a_1, b_1, a_2, b_2\}$ is the parameter vector and $f_{ubvM}(\theta_1, \theta_2; \boldsymbol{W}) = e^{\kappa_1 \cos(\theta_1-\mu_1)+\kappa_2 \cos(\theta_2-\mu_2)+\lambda \sin(\theta_1-\mu_1)\sin(\theta_2-\mu_2)}$ is the unnormalized bivariate von Mises distribution. Parameters μ_1, μ_2 and κ_1, κ_2 are analogous to parameters μ and κ, respectively, in the univariate truncated case. Truncation parameters a_1, b_1, a_2 and b_2 are similar to the univariate truncation parameters. The $\lambda \in \mathbb{R}$ parameter accounts for the dependency between the variable components (Figure 5.2). If $\lambda = 0$, then θ_1 and θ_2 are independent and each is distributed as a univariate von Mises distribution. Also, if θ_1, θ_2 are independent, then $\lambda = 0$.

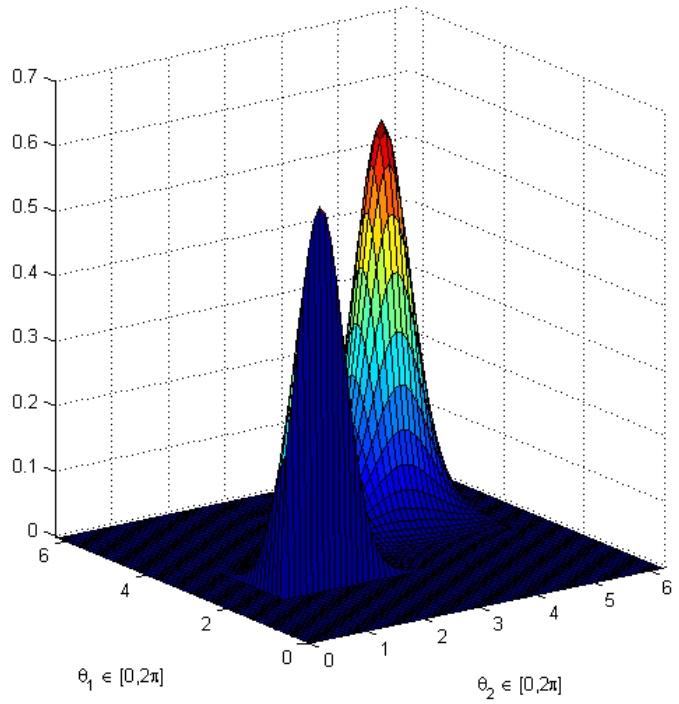


Figure 5.2: Example of the bi-dimensional von Mises distribution showing truncated bi-modality.

A desirable property of a joint distribution is having closed distributions under marginalization and conditioning, i.e., the marginal and conditional distributions should also follow the univariate distribution. Particularizing for the von Mises family, the bivariate von Mises distribution presents closed distributions only under conditioning as shown by Singh [2002]. We want to find out whether this also holds for the truncated case.

5.3.1 Maximum likelihood estimation

The maximum likelihood estimator for the bivariate distribution takes data of the form $\{(\theta_{1i}, \theta_{2i})\}$ $i = 1, \ldots, n$. The resulting log-likelihood function is

$$
\begin{aligned}
& \ln L(\boldsymbol{W}; (\theta_{11}, \theta_{21}), \ldots, (\theta_{1n}, \theta_{2n})) \\
= & \sum_{i=1}^{n} \ln \left(\frac{e^{\kappa_1 \cos(\theta_{1i}-\mu_1)+\kappa_2 \cos(\theta_{2i}-\mu_2)+\lambda \sin(\theta_{1i}-\mu_1)\sin(\theta_{2i}-\mu_2)}}{\int_{a_1}^{b_1}\int_{a_2}^{b_2} e^{\kappa_1 \cos(\theta_1-\mu_1)+\kappa_2 \cos(\theta_2-\mu_2)+\lambda \sin(\theta_1-\mu_1)\sin(\theta_2-\mu_2)} d\theta_2 d\theta_1} \right) \\
= & \sum_{i=1}^{n} (\kappa_1 \cos(\theta_{1i} - \mu_1) + \kappa_2 \cos(\theta_{2i} - \mu_2) + \lambda \sin(\theta_{1i} - \mu_1)\sin(\theta_{2i} - \mu_2)) \\
& -n \ln \left(\int_{a_1}^{b_1} \int_{a_2}^{b_2} e^{\kappa_1 \cos(\theta_1-\mu_1)+\kappa_2 \cos(\theta_2-\mu_2)+\lambda \sin(\theta_1-\mu_1)\sin(\theta_2-\mu_2)} d\theta_2 d\theta_1 \right).
\end{aligned}
$$

Thus we have

$$
\frac{\partial}{\partial \mu_1} \ln L(\boldsymbol{W}; (\theta_{11}, \theta_{21}), \ldots, (\theta_{1n}, \theta_{2n})) = 0,
$$

that is,

$$\sum_{i=1}^{n} \kappa_1 \sin(\theta_{1i} - \mu_1) - \lambda \cos(\theta_{1i} - \mu_1) \sin(\theta_{2i} - \mu_2) - \frac{n\left(\int_{a_2}^{b_2} f_{ubvM}(a_1, \theta_2) - f_{ubvM}(b_1, \theta_2) d\theta_2\right)}{\int_{a_1}^{b_1}\int_{a_2}^{b_2} f_{ubvM}(\theta_1, \theta_2) d\theta_2 d\theta_1} = 0,$$

where $f_{ubvM}(\theta_1, \theta_2)$ is the following unnormalized bivariate truncated von Mises function

$$f_{ubvM}(\theta_1, \theta_2) = e^{\kappa_1 \cos(\theta_1 - \mu_1) + \kappa_2 \cos(\theta_2 - \mu_2) + \lambda \sin(\theta_1 - \mu_1) \sin(\theta_2 - \mu_2)}.$$

Similarly, the partial derivate w.r.t. μ_2 gives

$$\sum_{i=1}^{n} \kappa_2 \sin(\theta_{2i} - \mu_2) - \lambda \cos(\theta_{2i} - \mu_2) \sin(\theta_{1i} - \mu_1) - \frac{n\left(\int_{a_1}^{b_1} f_{ubvM}(\theta_1, a_2) - f_{ubvM}(\theta_1, b_2) d\theta_1\right)}{\int_{a_1}^{b_1}\int_{a_2}^{b_2} f_{ubvM}(\theta_1, \theta_2) d\theta_2 d\theta_1} = 0.$$

For κ_1 we have

$$\frac{\partial}{\partial \kappa_1} \ln L(\boldsymbol{W}; (\theta_{11}, \theta_{21}), \ldots, (\theta_{1n}, \theta_{2n})) = 0,$$

that is,

$$\frac{1}{n}\sum_{i=1}^{n} \cos(\theta_{1i} - \mu_1) - \frac{\int_{a_1}^{b_1}\int_{a_2}^{b_2} \cos(\theta_1 - \mu_1) f_{ubvM}(\theta_1, \theta_2) d\theta_2 d\theta_1}{\int_{a_1}^{b_1}\int_{a_2}^{b_2} f_{ubvM}(\theta_1, \theta_2) d\theta_2 d\theta_1} = 0. \tag{5.3}$$

Similarly, the partial derivate w.r.t. κ_2 gives

$$\frac{1}{n}\sum_{i=1}^{n} \cos(\theta_{2i} - \mu_2) - \frac{\int_{a_1}^{b_1}\int_{a_2}^{b_2} \cos(\theta_2 - \mu_2) f_{ubvM}(\theta_1, \theta_2) d\theta_2 d\theta_1}{\int_{a_1}^{b_1}\int_{a_2}^{b_2} f_{ubvM}(\theta_1, \theta_2) d\theta_2 d\theta_1} = 0. \tag{5.4}$$

At this point, we can see that both Equations (5.3) and (5.4), involving κ_1, κ_2 parameters, respectively, preserve their analogy with the univariate case. Their second addend corresponds to the definition of the estimators of $\mathbb{E}[\cos(\theta_1 - \mu_1)]$ and $\mathbb{E}[\cos(\theta_2 - \mu_2)]$, respectively.

For the parameter λ we obtain

$$\frac{\partial}{\partial \lambda} \ln L(\boldsymbol{W}; (\theta_{11}, \theta_{21}), \ldots, (\theta_{1n}, \theta_{2n})) = 0,$$

that is,

$$\frac{1}{n}\sum_{i=1}^{n} \sin(\theta_{1i} - \mu_1) \sin(\theta_{2i} - \mu_2) - \frac{\int_{a_1}^{b_1}\int_{a_2}^{b_2} \sin(\theta_1 - \mu_1) \sin(\theta_2 - \mu_2) f_{ubvM}(\theta_1, \theta_2) d\theta_2 d\theta_1}{\int_{a_1}^{b_1}\int_{a_2}^{b_2} f_{ubvM}(\theta_1, \theta_2) d\theta_2 d\theta_1} = 0,$$

which analogously corresponds to the estimator of $\mathbb{E}[\sin(\theta_1 - \mu_1) \sin(\theta_2 - \mu_2)]$.

As in the univariate case, the truncation parameters has the following isolated estimators

$$\begin{aligned}\mathbb{O}_{\hat{a}_1,\hat{b}_1} &= \underset{a_1,b_1}{\operatorname{argmax}}(\max(\{A(\mathbb{O}_{\theta'_{11},\theta'_{12}}),\ldots,A(\mathbb{O}_{\theta'_{1n-1},\theta'_{1n}}),A(\mathbb{O}_{\theta'_{1n},\theta'_{11}})\}))\\ \mathbb{O}_{\hat{a}_2,\hat{b}_2} &= \underset{a_2,b_2}{\operatorname{argmax}}(\max(\{A(\mathbb{O}_{\theta'_{21},\theta'_{22}}),\ldots,A(\mathbb{O}_{\theta'_{2n-1},\theta'_{2n}}),A(\mathbb{O}_{\theta'_{2n},\theta'_{21}})\})),\end{aligned}$$

while as yielded by the above calculations, the expressions regarding the non-truncation parameters exhibit interdependency.

5.3.2 Conditional truncated von Mises distribution

The density of the conditional truncated von Mises distribution is defined as:

Definition 5.3.2. The conditional truncated von Mises distribution has density

$$f_{ctvM}(\theta_2|\theta_1;\lambda,\mu_1,\mu_2,\kappa 2,a_2,b_2) = \begin{cases} \frac{e^{\kappa_2\cos(\theta_2-\mu_2)+\lambda\sin(\theta_1-\mu_1)\sin(\theta_2-\mu_2)}}{\int_{a_2}^{b_2} e^{\kappa_2\cos(\theta_2-\mu_2)+\lambda\sin(\theta_1-\mu_1)\sin(\theta_2-\mu_2)}d\theta_2} & \text{if } \theta_2 \in \mathbb{O}_{a_2,b_2}. \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

It is a six-parameter distribution where the parameters hold the same meaning as in the bivariate case, with the simplification of parameters κ_1, a_1, b_1 for $f_{ctvM}(\theta_2|\theta_1)$ (or κ_2, a_2, b_2 for $f_{ctvM}(\theta_1|\theta_2)$). Worthy of note, however, is that $\theta_1 \in \mathbb{O}_{a_1,b_1}$ in $f_{ctvM}(\theta_2|\theta_1)$ since otherwise, by the definition of the conditional distribution $\left(f_{ctvM}(\theta_2|\theta_1) = \frac{f_{btvM}(\theta_2,\theta_1)}{f_{tvM}(\theta_1)}\right)$, $f_{ctvM}(\theta_2|\theta_1)$ is not defined.

Theorem 5.3.1. A conditional truncated von Mises distribution corresponds to the univariate truncated von Mises distribution

$$f_{ctvM}(\theta_2|\theta_1;\lambda,\mu_1,\mu_2,\kappa_2,a_2,b_2) =$$
$$f_{tvM}\left(\theta_2;\mu_2+\arctan\left(\frac{\lambda\sin(\theta_1-\mu_1)}{\kappa_2}\right),\sqrt{\kappa_2^2+(\lambda\sin(\theta_1-\mu_1))^2},a_2,b_2\right),$$

which completely specifies the behavior and properties of the conditional distribution and is analogous to the non-truncated conditional case (Singh [2002]).

Proof. See Appendix A. □

5.3.3 Marginal truncated von Mises distribution

We can define the density function of the marginal truncated von Mises distribution as:

Definition 5.3.3. The density function of the marginal truncated von Mises distribution can be written as

$$f_{mtvM}(\theta_1;\boldsymbol{W}) = \begin{cases} \frac{\int_{a_2}^{b_2} e^{\kappa_1\cos(\theta_1-\mu_1)+\kappa_2\cos(\theta_2-\mu_2)+\lambda\sin(\theta_1-\mu_1)\sin(\theta_2-\mu_2)}d\theta_2}{\int_{a_1}^{b_1}\int_{a_2}^{b_2} e^{\kappa_1\cos(\theta_1-\mu_1)+\kappa_2\cos(\theta_2-\mu_2)+\lambda\sin(\theta_1-\mu_1)\sin(\theta_2-\mu_2)}d\theta_2 d\theta_1} & \text{if } \theta_1 \in \mathbb{O}_{a_1,b_1} \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

It is a nine-parameter distribution that shares all the parameters with the bivariate truncated von Mises distribution. In the original publication, Singh [2002] studied the distribution and reported the "frontiers"

of bi-modality (for $\mu = 0$) as

$$\frac{I_1(\kappa_2)}{I_0(\kappa_2)} = \frac{\kappa_1\kappa_2}{\lambda^2}$$

where the distribution is unimodal if $\frac{I_1(\kappa_2)}{I_0(\kappa_2)} \geq \frac{\kappa_1\kappa_2}{\lambda^2}$, and bimodal with two equal maxima otherwise. Additionally, the modes were calculated to be symmetrical w.r.t μ_1 and at the distance value θ_1^* that solves the equation (for $\mu_1 = 0$):

$$\frac{A\left(\sqrt{\kappa_2 + \lambda^2 \sin^2(\theta_1^*)}\right)}{\sqrt{\kappa_2 + \lambda^2 \sin^2(\theta_1^*)}} \cos(\theta_1^*) = \frac{\kappa_1}{\lambda^2},$$

where $A(x) = \frac{I_1(x)}{I_0(x)}$. In order to generalize this analysis to cover the truncated case in Equation (5.6), we need to account for the contribution made by the parameters μ_2, a_2 and b_2 to the shape of the distribution. Contrary to the non-truncated case, a truncated marginal distribution that exhibits two maxima may have only one global maximum, and the distribution is not necessarily centered around the mean (Figure 5.3). Therefore, our analysis determines the different parameter configurations that produce the whole range of behaviors, focusing on bi-modality/unimodality.

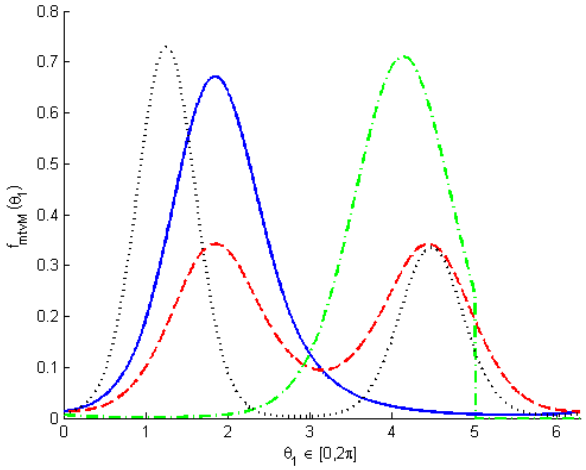


Figure 5.3: Several truncated marginal distributions showing unimodality (continuous line), two equal maxima (dashed line), truncated unimodality (dash-dot line) and two distinct maxima (dotted line)

If, without loss of generality, we take $\theta_{1'} = \theta_1 - \mu_1$, we can postulate the following theorem:

Theorem 5.3.2. All different behaviors w.r.t. the unimodality/bi-modality of the marginal truncated von Mises distribution can be accounted for as follows

1. $f_{mtvM}(\theta_{1'})$ is unimodal with mode (maximum) in μ_1, if and only if $T(\lambda, \mu_2, \kappa_1, \kappa_2, a_2, b_2) < 0$ and $\cos(b_2 - \mu_2) = \cos(a_2 - \mu_2)$.

2. $f_{mtvM}(\theta_{1'})$ is bi-modal with equal maxima, if and only if $T(\lambda, \mu_2, \kappa_1, \kappa_2, a_2, b_2) > 0$ and $\cos(b_2 - \mu_2) = \cos(a_2 - \mu_2)$. Also in this case, a minimum is found at $\theta_{1'} = 0$.

3. $f_{mtvM}(\theta_{1'})$ presents two differentiated maxima if and only if one of the two following cases applies:

(a) $\cos(b_2 - \mu_2) < \cos(a_2 - \mu_2)$ and $f'_{umtvM}(\theta_{1'}; \lambda, \mu_1, \mu_2, \kappa_1, \kappa_2, \mu_2, a_2, b_2)$ has exactly two zero points in $\theta_{1'} \in [-\frac{\pi}{2}, 0]$

(b) $\cos(b_2 - \mu_2) > \cos(a_2 - \mu_2)$ and $f'_{umtvM}(\theta_{1'}; \lambda, \mu_1, \mu_2, \kappa_1, \kappa_2, \mu_2, a_2, b_2)$ has exactly two zero points in $\theta_{1'} \in [0, \frac{\pi}{2}]$

4. $f_{mtvM}(\theta_{1'})$ is unimodal with mode not at μ_1 if the parameters do not match any of the above cases,

where $T(\lambda, \mu_2, \kappa_1, \kappa_2, a_2, b_2)$ is the test function and is defined as

$$T(\lambda, \mu_2, \kappa_1, \kappa_2, a_2, b_2) = -\frac{\kappa_1}{\lambda^2} + \frac{\int_{a_2}^{b_2} \sin^2(\theta_2 - \mu_2) e^{\kappa_2 \cos(\theta_2 - \mu_2)} d\theta_2}{\int_{a_2}^{b_2} e^{\kappa_2 \cos(\theta_2 - \mu_2)} d\theta_2},$$

and $f'_{umtvM}(\theta_{1'}; \lambda, \mu_1, \mu_2, \kappa_1, \kappa_2, \mu_2, a_2, b_2)$ is the unnormalized truncated marginal von Mises derivative function.

Proof. See Appendix A. □

5.4 Real data application

5.4.1 Leaf angle inclination

The data in Bowyer and Danson. [2005] was collected during a safari along the Kalahari Transect, southwest Botswana in 2001. It contains measurements of leaf inclination angles of four different woody plant species (*Acacia erioloba*, *Grewia flava*, *Acacia leuderitzii* and *Acacia mellifera*) across three different regions (Mabuasehube, Tsabong and Tshane). The measurements were taken using a clinometer.

In order to formally test the goodness-of-fit of the estimated distributions, we transform the data by means of the random variable $U = 2\pi \frac{[I(\theta,\mu,\kappa) - I(a,\mu,\kappa)]}{\int_a^b e^{\kappa \cos(\theta-\mu)} d\theta} \mod 2\pi$ that is applied over the sorted sample $\theta_1, \ldots, \theta_n$. If the data distribute according to the truncated von Mises distribution, then the above random variable has a uniform distribution. As shown in Mardia and Jupp [2000], the modified Rayleigh statistic $S^* = (1 - \frac{1}{2n})2nR^2 + \frac{nR^4}{2}$, where n is the sample size and R the mean resultant length, distributes as a χ^2_2 distribution.

1. For the first study, the whole dataset containing a total of 741 samples was observed (Table 5.1, Figure 5.8). A visual inspection of the plot clearly shows that the truncated von Mises distribution performs better. Formally, for the truncated case we have $S^* = 2.8887$, which corresponds to $p-$value $\in (0.2, 0.3)$. For the non-truncated case, $S^* = 25.5028$, with is a clear rejection $p-$value < 0.001. From these results we conclude that the truncated distribution is significantly better for these data. Truncation parameters conform the circular interval $\mathbb{O}_{0,\frac{\pi}{2}}$, which indicates no angle greater than $90°$ was measured in this study.

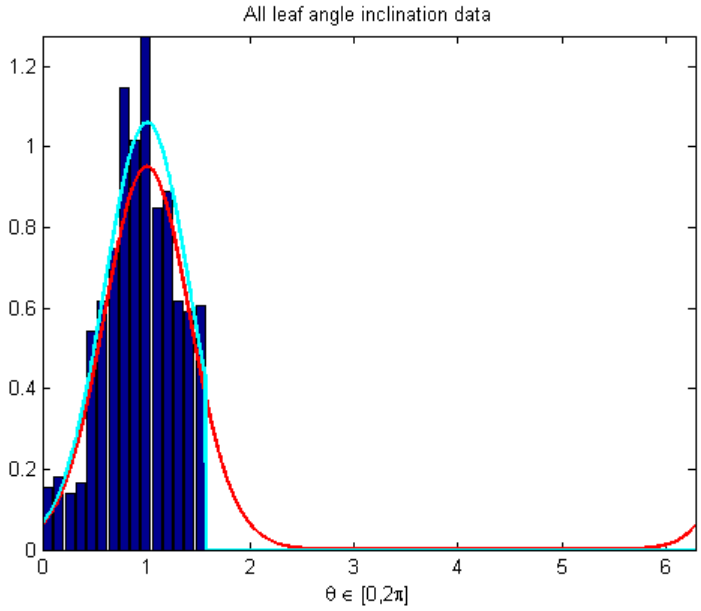


Figure 5.4: The study distribution and data representation of the entire dataset. The estimated truncated von Mises distribution (lighter line) clearly has higher density values than its associated von Mises distribution (darker line). The data are grouped by value intervals in order to observe its relative frequency (bars).

Table 5.1: Parameter values obtained after conducting the first study

	μ	κ	a	b	No. Samples
All data	1.0063	5.9602	0	1.5708	741

2. For the second study, we grouped the data by plant types without regards for region. This yielded four different distributions. A visual inspection shows that the univariate distributions are clearly better than the non-truncated von Mises distribution at describing the resulting data (Table 5.3, Figure 5.9), except for the case of *A. erioloba*. The goodness-of-fit tests (Table 5.2) revealed that the non-truncated distribution is rejected in all cases but in *A.erioloba*, whereas the truncated distribution hypothesis was more strongly accepted than that of the non-truncated distribution in all cases. Thus we can conclude that, for this study, the truncated distribution models the data better.

Table 5.2: Modified Rayleigh statistic values for the second study

	Truncated von Mises S^*	Non-truncated von Mises S^*
A. Erioloba	3.014	3.5534
Grewia flava	0.0038	20.6273
A. Leuderitzii	2.6073	10.1990
A. Mellifera	1.3157	7.3046

Truncation parameters were consistently found to be in $\mathbb{O}_{0,\frac{\pi}{2}}$ except for *A.erioloba*, which also presented a significantly higher concentration parameter than in any of the other estimations. The irregularities in *A.erioloba* could partially be explained by the small sample size, which causes the estimations to be less reliable. On the whole, the remaining studies show few variations in the location-concentration parameters, which closely resemble the ones obtained in the first study.

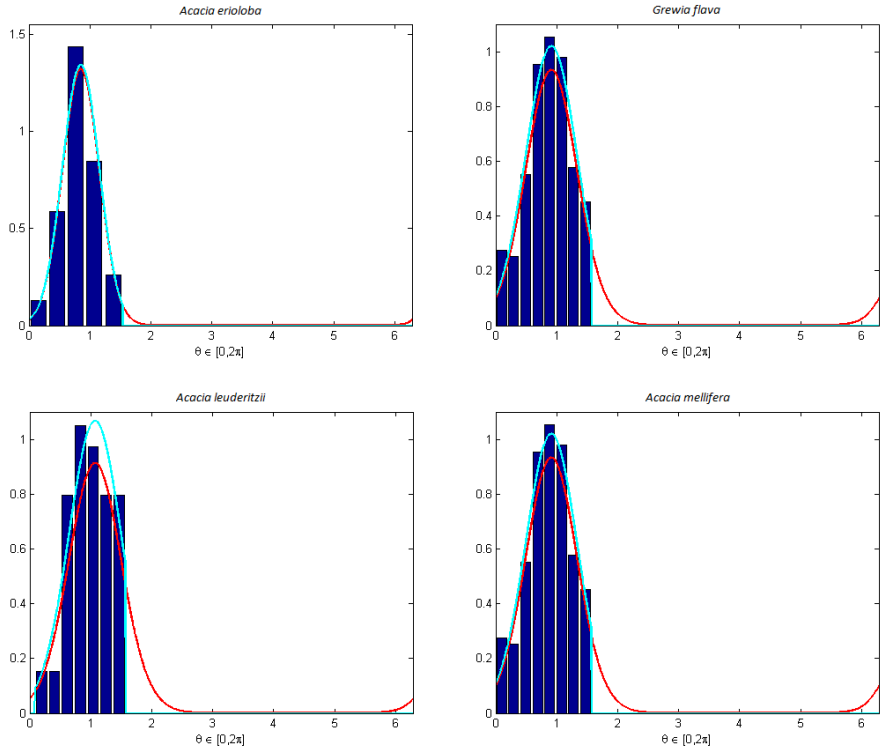


Figure 5.5: Studies of each type of plant.

Table 5.3: Parameter values yielded after conducting the second study

	μ	κ	a	b	No. Samples
A. Erioloba	0.8516	11.1894	0	1.5359	100
Grewia flava	1.1261	5.2668	0	1.5708	254
A. Leuderitzii	1.0706	5.5138	0	1.5708	184
A. Mellifera	0.9125	5.7396	0	1.5708	203

3. For the third study, fitted univariate truncated distributions for each plant in each region. Since not all plants were measured in all regions, this procedure produced eight different univariate truncated von Mises estimations. The distributions are generally observed to clearly differ from their associated non-truncated von Mises distribution, except in the first of the eight plots (Table 5.5, Figure 5.10). The goodness-of-fit tests (Table 5.4) are also consistent with previous studies. All truncated von Mises hypotheses were accepted, while around half of the non-truncated distributions were rejected. Thus, there is a strong suggestion that the truncated von Mises distribution properly models the underlying behavior that yielded the data.

Table 5.4: Parameter values yielded after conducting the third study

	Truncated von Mises S^*	Non-truncated von Mises S^*
A. erioloba, Mabuasehube	3.014	3.5534
Grewia flava, Mabuasehube	1.1543	8.9599
A. leuderitzii, Tsabong	2.0981	7.3115
Grewia flava, Tsabong	0.2050	3.8702
A. mellifera, Tsabong	0.1199	4.2131
Grewia flava(2), Tsabong	0.1165	9.7290
A. leuderitzii, Tshane	0.7002	2.8717
A. mellifera, Tshane	1.0525	10.2656

For this study, each distribution was estimated from a relatively small sample size ranging from 50 to 104 samples, which may have caused estimations to be less precise than desired. The concentration parameter shows the highest variability across the different cases (from 4.4078 to 11.1894 across the whole study or even from 4.8340 to 7.4245 in the case of *A. leuderitzii*). With more data it might be possible to distinguish if the variations in the concentration parameter are clearly influenced by the region of the plant species or the small sample size. Regarding the location parameter, there are few variations in the parameter value on the whole, *A. mellifera* being the species that experienced the highest variations with respect to one of the measurements in the first study. Truncation parameters remained consistently within the $\mathbb{O}_{0,\frac{\pi}{2}}$ interval.

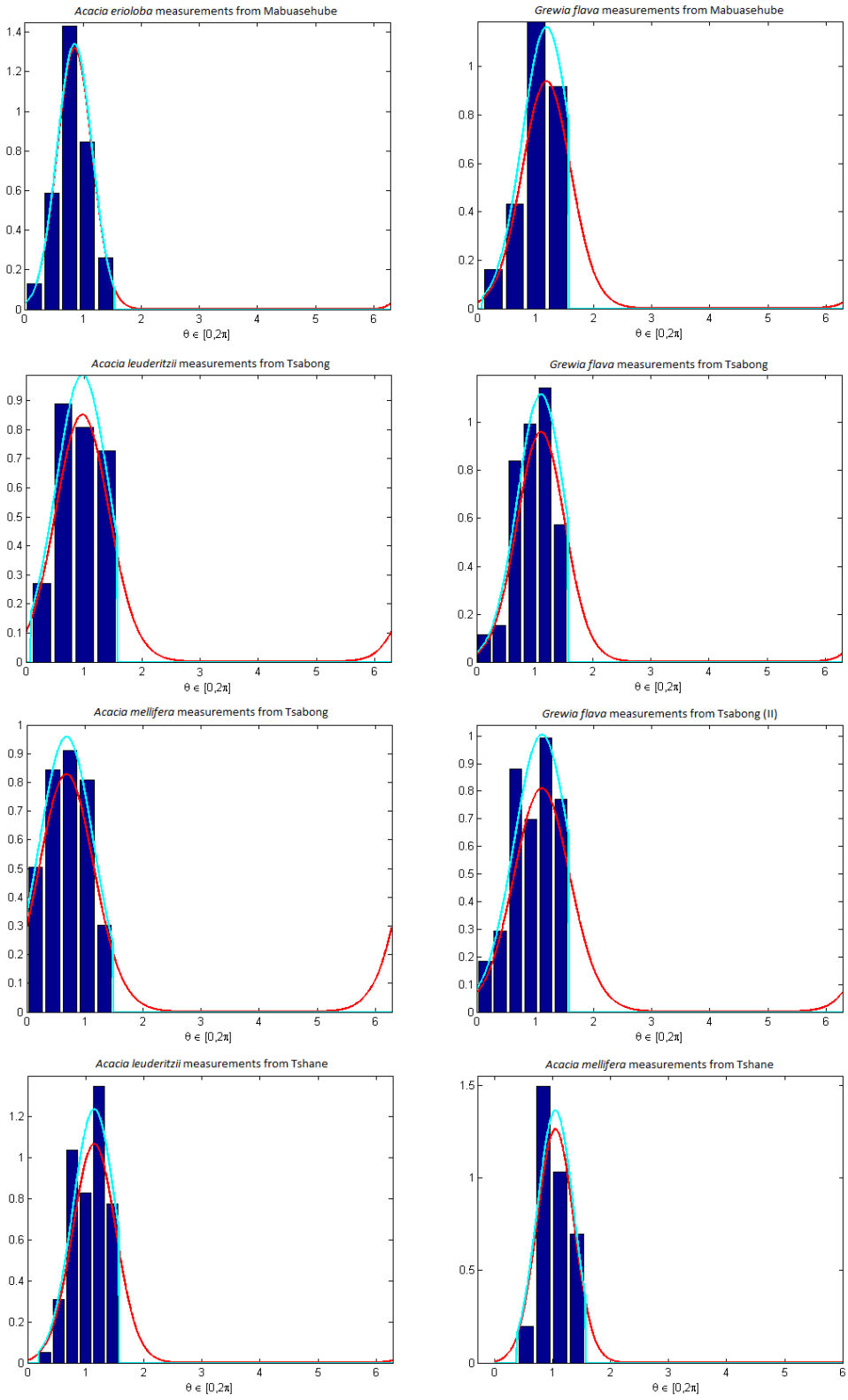


Figure 5.6: Studies of each type of plant in each region.

Table 5.5: Parameter values yielded after conducting the third study

	μ	κ	a	b	No. Samples
A. erioloba, Mabuasehube	0.8516	11.1894	0	1.5359	100
Grewia flava, Mabuasehube	1.1882	5.8142	0.0873	1.5708	50
A. leuderitzii, Tsabong	0.9712	4.8340	0.0873	1.5708	100
Grewia flava, Tsabong	1.1082	6.0832	0	1.5708	100
A. mellifera, Tsabong	0.6844	4.5884	0	1.4835	100
Grewia flava (2), Tsabong	1.1091	4.4078	0	1.5708	104
A. leuderitzii, Tshane	1.1474	7.4245	0.1920	1.5708	84
A. mellifera, Tshane	1.0525	10.2656	0.4014	1.5708	103

5.5 Summary and conclusions

In this chapter we developed the theoretical framework of the univariate and bivariate truncated von Mises distribution. To do this, we gave

1. The definition of a truncated von Mises distribution in the circle $\mathbb{O}$. The circular distribution is defined by means of the $\mathbb{O}$ subset, as the periodicity and properties of the circle have to be naturally acknowledged for.

2. The successfully determined expressions of the maximum likelihood estimators. For both univariate and bivariate cases, solely sample-dependent maximum likelihood estimators of the truncation parameters were found, while the other parameters showed interdependency.

3. The resulting moments of the univariate case and existing interrelationships.

4. The bivariate case and studies of the shape and behavior of marginal and conditional distributions. We determined that every conditional truncated von Mises distribution is a univariate truncated von Mises distribution. For the case of the marginal distribution, we concluded that only for parameter $\lambda = 0$ does the distribution behave like a truncated univariate von Mises distribution. When $\lambda \neq 0$, the resultant marginal distribution is potentially bi-maximal and not a von Mises distribution. The modality behavior of this distribution has been accounted for in Theorem 3.2.

This work has been published as Fernandez-Gonzalez, P., C. Bielza, and P. Larrañaga, “Univariate and bivariate truncated von Mises distributions”, *Progress in Artificial Intelligence*, pp. 1-10, 2017

Chapter 6

Dendritic branching angles of pyramidal neurons of the human cerebral cortex

6.1 Introduction

The design principles that govern the geometry of neurons are a major topic to those researchers interested in the generation of realistic mathematical models of neuronal morphologies. The study of pyramidal cells is of particular importance as they are the most abundant neurons in the cortex (estimated to represent 70-80% of the total neuronal population), where they are the main source of excitatory (glutamatergic) synapses. Furthermore, the dendritic spines of pyramidal cells constitute the main target of excitatory synapses in the cerebral cortex (DeFelipe and Farinas [1992]). Thus pyramidal cells are considered the principal building blocks of the cerebral cortex and it is thought that unraveling the morphology, connectivity and functional organization of this type of neurons is critical for better understanding cognitive functions.

There are considerable differences in the structure of pyramidal cells when considering the size and complexity of their dendritic arborization -the complexity of a dendritic arbor is evaluated as the total length of its dendritic branches along with the number and distribution of their branching points-, in the density of dendritic spines on their dendritic branches and in the total number of dendritic spines. These differences are found not only between cortical areas but also between different species and these differences are thought to be critical for the functional specialization of the cortical areas (reviewed in Jacobs et al. [2001], Elston [2007], Elston et al. [2011], Defelipe [2011], Eyal et al. [2014] and Mohan et al. [2015]). In a previous study, we found that the dendritic branching angles of layer III pyramidal neurons in several regions of the frontal, parietal, and occipital cortex of the adult mouse follow similar principles despite the differences in the structure of these neurons in the different cortical regions examined (Bielza et al. [2014]). We found that 90% of these angles fell within a range of 20 to 97 degrees. These are similar values to the results obtained for the dendritic branching angles of pyramidal cells from layers II-VI of the juvenile rat somatosensory cortex (angles ranged from 10-104 degrees) in Leguey et al. [2016]. Since the dendritic spines length is relatively short ($< 2\mu m$), it follows that dendritic branching of pyramidal cells determine the connectivity of the pyramidal cell. Therefore, the finding that branching angles are designed in accordance with the rules of mathematical functions and that they show common design principles suggests certain predictability in the synaptic connections of

pyramidal cells in all cortical areas of the mouse and rat. In this chapter, we are interested in extending these studies to the human cerebral cortex to find out if the branching angles follow similar rules using a novel branching angles dataset. In particular, our aim is to try to find a statistical distribution that properly models branching angles in human pyramidal neurons and analyze possible differences and/or similarities between branching angles in different cortical layers. More specifically, we examined layers III and V of the temporal cortex in different antero-posterior regions. We proposed the truncated von Mises distribution as the distribution to model the behavior of the dendritic branching angles. Previous work (Bielza et al. [2014]) used a different although related distribution, the von Mises distribution (see Section 2.2) as the preferred distribution to model branching angles in mice. However, the von Mises distribution alone failed to acknowledge if all the angular measurements were contained within a reduced circular interval (as it was noted in the previous study) and was forced to assume that the angles were symmetrically distributed. The truncated von Mises distribution (that is a generalization of the von Mises distribution, see Chapter 5) is able to approximate efficiently within a reduced interval non-symmetrical data, thus appearing as a more accurate analysis tool for modeling the branching angles behavior.

The rest of the chapter is organized as follows. Section 6.2 details the different techniques chosen for the development of this work. Section 6.3 contains the results of all the data analysis. More concretely, in subsections 6.3.1 and 6.3.2 we perform goodness-of-fit tests according to groups obtained by different criteria (i.e., branch order or branch order together with maximum branch order), with results that clearly improve those of the von Mises distribution. Additionally, we perform hypothesis tests on different statistics related to the parameters of the distribution (such as the mean and the concentration around the mean), to further analyze the underlying patterns of the data.

In subsection 6.3.3 we group the data in pairs of angles of contiguous branch orders and use the bivariate truncated von Mises distribution as analysis tool.

In subsections 6.3.4 and 6.3.5 we are interested in analyzing the differences between angular measurements that belong to different layers as well as the differences between angular measurements that belong to the same layer, but in a different region. We perform tests for a common distribution (i.e. tests that try to diagnose if two datasets could have been drawn from the same probability distribution. We will refer to them as similarity tests) between different subgroups of the data for this purpose.

In subsection 6.3.6, we analyze some results found in this study in a comparison with the data of previous studies in mice (Bielza et al. [2014]) and rats (Leguey et al. [2016]). Our interest lies in finding similarities/differences of branching angles data between species, and for this we perform tests for a common distribution of the three datasets.

Finally, Section 6.4 contains the discussion of the findings and conclusions obtained throughout this study.

6.2 Methods

6.2.1 Data acquisition and preparation

Tissue was obtained from the anterolateral temporal gyri (Brodmann's areas 21 and 38; see Garey [1994]) of patients with pharmaco-resistant temporal lobe epilepsy (Department of Neurosurgery, 'Hospital de la Princesa', Madrid, Spain). This brain tissue was removed as part of surgical treatment of five male patients (28-48 years old, mean 36.6 years old) and had been used in previous studies (Kastanauskaite

[2009], Arion [2006] and Sola RG [2005]). The five patients used in this study had normal IQs and each had a different history of medications and treatment -they were treated with a variety of anti-epileptic drugs that affect GABAergic transmission and other neurotransmitter systems. Furthermore, the disease severity was variable (with daily, weekly or twice monthly seizures) as was the disease duration (from 10 to 29 years). However, as described below, in all cases the neocortical tissue used in the present study was histologically normal and without abnormal spiking activity. In each case, video-EEG recording from bilateral foramen ovale electrodes was used to localize the epileptic focus in mesial temporal structures. Subdural recordings with a 20-electrode-grid (lateral neocortex) and with a 4-electrode-strip (uncus and parahippocampal) were used at the time of surgery to further identify epileptogenic regions. After surgery, the lateral temporal neocortices of all patients and the mesial temporal structures from all patients except one were available for standard neuropathological assessment. In the latter case, most mesial structures were absorbed during surgical removal and, therefore, could not be examined. The lateral neocortices were histologically normal in all cases. However, alterations were found in the hippocampal formations of three out of the four patients that could be examined; these three patients showed hippocampal sclerosis, whereas no apparent alterations were found in the hippocampal formation of the remaining patient. Furthermore, only neocortical tissue that showed no abnormal spiking -as characterized by normal ECoG activity- was used in this study (see Arion [2006]). Surgically resected tissue was immediately immersed in cold 4% paraformaldehyde in 0.1 M phosphate buffer, pH 7.4 (PB). After 2-3 h, the tissue was cut into small blocks (0.5 x 8 x 8 mm) which were flattened (e.g., Welker and Woolsey 1974) and post-fixed in the same fixative for 24 h at 4°C. Horizontal sections (250 microns) were obtained using a Vibratome. By relating these sections to coronal sections, we were able to identify, using cytoarchitectural differences, the section that contained each cortical layer, allowing the subsequent injection of cells (e.g., Elston and Rosa [1997]). Sections were prelabeled with 4,6-diamidino-2-phenylindole (DAPI; Sigma, St Louis, MO), and a continuous current was used to inject individual cells with Lucifer yellow (8% in 0.1; Tris buffer, pH 7.4; LY) in cytoarchitectonically identified layers III and V of the anterolateral temporal cortex (see results section for further details). Neurons were injected until the individual dendrites of each cell could be traced to an abrupt end at their distal tips and the dendritic spines were readily visible, indicating that the dendrites were completely filled. After injection of the neurons, the sections were first processed with a rabbit antibody to Lucifer yellow produced at the Cajal Institute [1:400,000 in stock solution: 2% BSA (A3425; Sigma), 1% Triton X-100 (30632; BDH Chemicals), 5% sucrose in phosphate buffer (PB)] and then with a biotinylated donkey anti-rabbit secondary antibody (1:200 in stock solution, RPN1004; Amersham Pharmacia Biotech), followed by a biotin-horseradish peroxidase complex (1:200 in PB, RPN1051; Amersham). 3,3'-Diaminobenzidine (D8001; Sigma Chemical Co.) was used as the chromogen, allowing the visualization of the entire basal dendritic arbor of pyramidal neurons. Finally, sections were mounted in 50% glycerol in PB. Possible changes in the size of the sections due to processing of the tissue was evaluated by measuring the cortical surface and thickness in adjacent sections before and after intracellular injections and processing of the tissue, using Neurolucida 11.07 and StereoInvestigator 11.02.1 from MicroBrightField (MBF, VT, USA). We found no shrinkage in the surface area of the sections and a decrease in thickness of only approximately 7% was observed. Therefore no correction factors were included. Neurons were reconstructed in three dimensions using Neurolucida (MicroBrightField) as previously described in detail (for further methodological details, see Elston et al. [2001] and Benavides-Piccione et al. [2006]).

We refer to branch order of a branching angle as the number of branchings (including itself) that exist

between the branching angle and the root of the dendrite. As an example, a branching angle with branch order 4 comes after 3 preceding branching angles from the root of the dendrite, which is the branch order 1. We refer to maximum branch order or tree order of a dendrite as the total amount of branch orders of a dendrite, or the branching angle at the highest order that can be found in the dendrite.

The dataset included: 57, 37 and 87 cells from layer IIIAnt (1452 measurements), VPost (1328 measurements) and IIIPost (2430 measurements), respectively. More precisely, the dataset for layer IIIPost contained measurements of 7 branch orders (300, 477, 430, 198, 39, 5 and 3 from order 1-7, respectively) extracted from a total of 57 neurons. The dataset for layer VPost contained measurements of 8 branch orders (247, 381, 373, 226, 82, 14, 4 and 1 from order 1-8, respectively) extracted from a total of 37 neurons. Finally, the data set for layer IIIAnt contained measurements of 7 branch orders (470, 732, 714, 375, 114, 24 and 1 from order 1-7, respectively), extracted from a total of 87 neurons. In this data, branch orders above five suffer from very low number of observations and thus we will restrict our analysis to the first five branch orders. The 3D reconstructions of these cells will be available in another publication (Benavides-Piccione, Kastanaukaite, Rojo and DeFelipe, in preparation).

6.2.2 Univariate truncated von Mises distribution

The statistical analysis of branching angles requires directional statistics, as conventional statistics do not address well the circular properties. In this field, the von Mises distribution (Mardia [1975]) is the most known distribution and the analog of the Gaussian distribution in the line. This distribution has properties such as symmetry and positive support on all the values in a circle ($[0°, 360°)$) which are necessary simplifications of the data in many case studies. As it is found that in neuroscience, such simplifications may hinder the accuracy and reliability of the complex behaviors it studies, we propose for the first time to use the univariate truncated von Mises distribution (see Section 5.2 of Chapter 5).

6.2.3 Bivariate truncated von Mises distribution

For the case of events that are defined by two angular measurements (θ_1, θ_2). We propose, for analogous reasons as the univariate case, the bivariate truncated von Mises distribution (see Section 5.3 of Chapter 5).

6.2.4 Statistical tests

Test of goodness-of-fit a univariate truncated von Mises distribution. We tested if the angular data, under different groupings, can be properly modeled with a truncated von Mises distribution. As considered in Mardia and Jupp [2000], we transformed the data $\theta_1, \ldots, \theta_n$ by means of the angular variable $U_{tvM}(\theta_i) = 2\pi F_{tvM}(\theta_i)$ where $F_{tvM}(.)$ is the probability distribution function of the truncated von Mises distribution. Then, we tested circular uniformity (i.e., the circular distribution where every observation is equally likely to occur) using a modified Rayleigh statistic (Cordeiro and De Paula Ferrari [1991]) that distributes according to a χ^2_2 distributes under the null hypothesis to obtain the final $p-$value for the fit. If the data distributes following a truncated von Mises distribution, the previous transformation generated a uniform distribution from the data.

Test of goodness-of-fit to a univariate von Mises distribution. A similar procedure is used for the von Mises distribution. The difference between both cases is the probability distribution function that is used.

In this case, $F_{vM}(\theta)$ is the probability distribution function of the von Mises distribution, and therefore the angular variable for this case is $U_{vM}(\theta_i) = 2\pi F_{vM}(\theta_i)$.

Two sample tests for common distribution (similarity). We tested the hypothesis of similarity between two datasets, i.e., if two datasets can be considered to be drawn from the same probability distribution. We used the non-parametric Watson's two sample U^2 test (Watson [1962]), that does not assume any underlying probability distribution. This test was used to perform the comparisons between layer IIIPost and layer VPost, and layer IIIAnt and layer IIIPost. See Supplementary material, Tables 9, 10 at http://cig.fi.upm.es/thesis/phd/Supplementary_Material_thesis_Pablo_2019.pdf.

Tests for mean comparison. We use Watson's large sample (where "large"stands for samples greater or equal to 25) non-parametric test (Watson [1983]) to test the null hypothesis of the same mean direction. The test does not assume any underlying probability distribution. It was used with three different subgroups of the data as we were interested in testing if the means of the data, grouped by branchings or branchings together with maximum branch order, follow any noticeable tendency. It was additionally used for comparisons between layers IIIPost and VPost and for the comparisons of branch order 1 mean values. See Supplementary material, Tables 1, 2 and 4.

Tests for the concentration comparison. Wallraff's test for common concentration (Wallraff [1979]) was useful for comparisons between layer IIIPost vs. layer VPost, and layer IIIAnt vs. layer IIIPost. It is a non parametric test with no assumptions regarding data generating distributions. See Supplementary material Table 4.

Tests of independence. We used two different tests to verify or reject the hypothesis of independence (i.e., if positive or negative significant correlation between two random variables exists). First, we used a randomized version of the Rothman's test for independence (Rothman [1971]), a test that does not assume any underlying probability distribution for the two tested datasets. See Supplementary material, Table 8. Finally, we used a permutations tests over the λ parameter (that we previously estimated using the maximum likelihood method from the datasets) which tested the null hypothesis of $\lambda = 0$

Test-based diagrams. We used two different forms of visualization for the comparison of test results. The first type of diagram, the test-based diagram, was originally proposed in (Bielza et al. [2014]) and consists of a space of nodes that are connected or not by edges depending on the non-rejection or rejection result of the test, respectively. In this diagram, every node that appears is pairwise tested w.r.t. all the other nodes. These diagrams are shown in Figures 2D and 3. The second type of diagram, the test-based tree, is first proposed here as a form to easily visualize comparisons between two cortical brain layers or two datasets whose data is organized in a tree-like structure that includes branch orders. It consists of trees where the branch order in the graphic corresponds to the branch order of the conducted test. If the space between the branches is subdivided and labeled with a number, the number that labels each subdivided area indicates the maximum branch order of the data of the conducted test. Finally, the green or red color of the area between the branches indicates the non-rejection or rejection of the hypothesis of the conducted test, respectively. These diagrams are shown in Figures 6.4A, 6.4B, 6.5A and 6.5B.

6.3 Results

In the present work, a total of 181 3D reconstructed basal dendritic arbors of intra-cellularly injected cells from the human temporal cortex were included in the branch angle analysis. The cells were located in layers III and V of the temporal cortex (at a distance of 2-3 cm from the temporal pole), corresponding

to Brodmann's area 21 and in layer III of the temporal pole proper, corresponding to Brodmann's area 38. For simplicity, we will refer to layer III anterior neurons to those located in the temporal pole as layer IIIAnt neurons, while those located at 2-3 cm will be referred as layer IIIPost and layer VPost neurons, respectively (Figure 6.1).

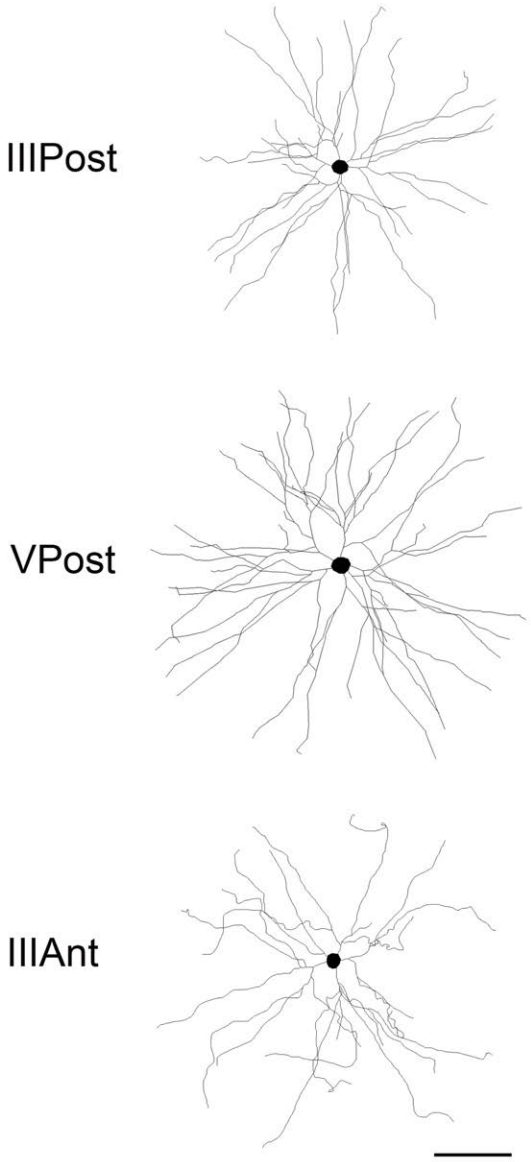


Figure 6.1: Schematic drawing examples of basal dendritic arbors of pyramidal neurons from layers III and V of the temporal cortex at a distance of 2-3 cm from the temporal pole (IIIPost and VPost, respectively) and layer III of the temporal pole proper (IIIAnt). Scale bar 100 μm.

We first analyzed the distribution of angles of each dendritic branch order (Figure 6.2A; see material and methods for details). In general, the inspection of the rose diagrams showed that the underlying distribution for the data should be unimodal with a slight deviation from symmetry with respect to the mean (Figure 6.2B). Also, we noticed that all observations in the three datasets were contained within a circular interval that goes from $0°20'58''$ to $170°16'59''$, which covers less than half of a circle. The truncated von Mises distribution has two parameters (called a and b) that set the inferior and superior

limits of the circular interval where observations can occur, leaving a potentially non symmetrical distribution inside. This capability makes it especially attractive for this case and it's the justification of its choosing, together with its capability to capture unimodality.

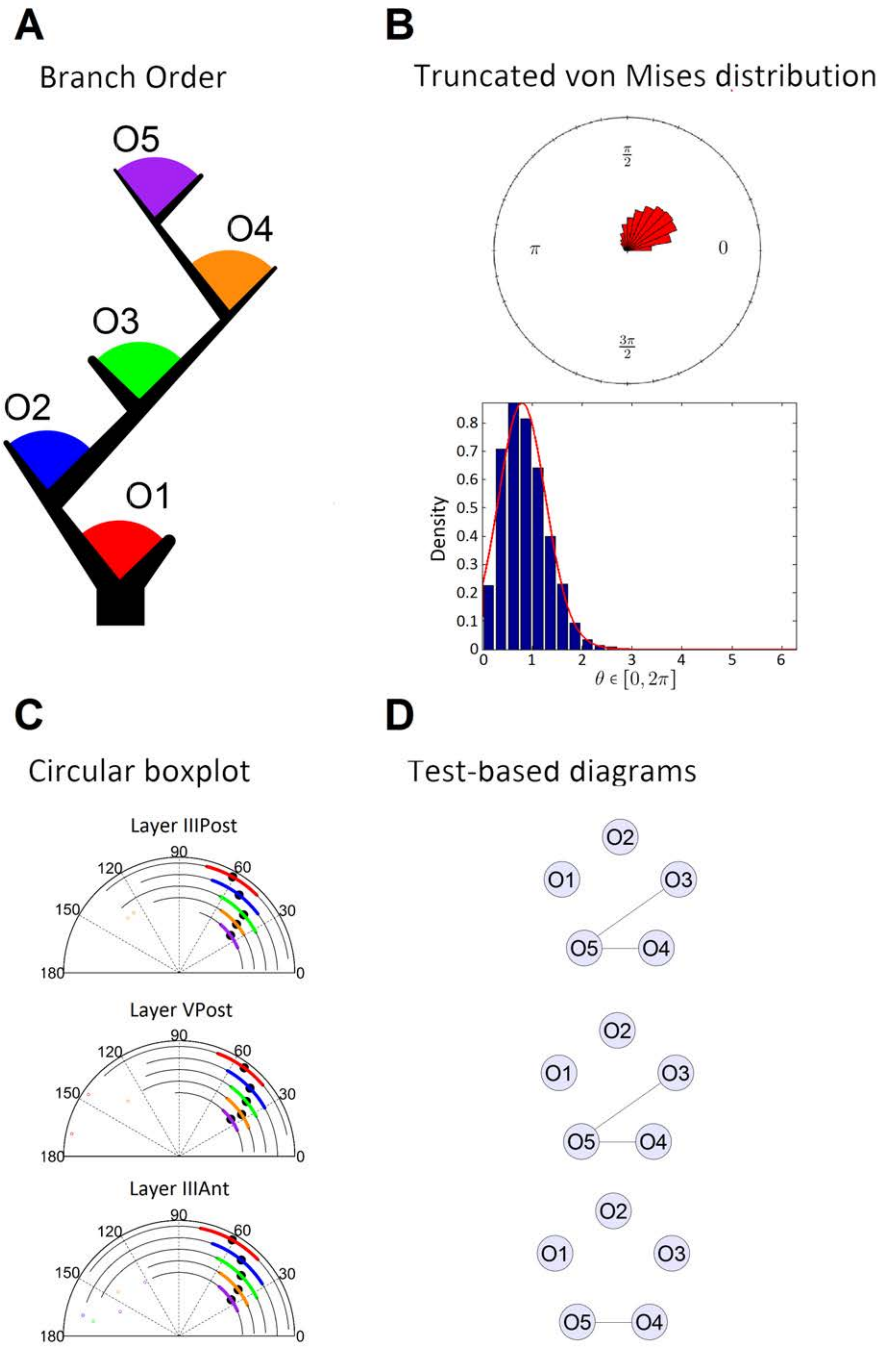


Figure 6.2: **A** Color codes for the branch orders represented in a denritic tree. **B** Rose diagram (top) and truncated von Mises distribution (bottom) plots of the combined data of layers IIIPost, VPost and IIIAnt. The bars in both plots represent the frequency of the data. The red curve in the bottom plot is the estimated truncated von Mises density function. **C** Circular boxplots of the first five branch orders. In the different subdivisions of the semi-circle we find the data summarized in different ways. The colored curves cover the circular interval from the lower quartile ($Q1$) to the upper quartile ($Q3$). The longer black thin curve covers all the values inside $[Q1 + (V) * CIQR, Q3 - (V) * CIQR]$, where $CIQR = Q3 - Q1$ and V is 2.5 or 1.5 depending of the concentration of the data (2.5 for all our cases). The black dot represents the Fisher's median statistic, and the isolated colored dots indicate outliers. **D** Test-based diagrams illustrating the similarity comparisons of the data groups selected in **C**. Each node represents a data group and two nodes are connected when the hypothesis of same probability distribution is not rejected (conversely, not connected if rejected). See Section 6.2 section for more details.

6.3.1 Study of branching angles by branch order

We compared angles of different branch order in layers IIIPost, VPost and IIIAnt. We will use the circular boxplots proposed in Abuzaid et al. [2012] and used in Bielza et al. [2014] as an efficient way to visualize information about the observations.

As seen in Figure 6.2C, the median angular values tend to decrease as the order increases for the three groups. This is also true for the mean angular values, decreasing as the branch order increases (see Supplementary Table 1, rows 1-10). Thus, angles in higher branch orders are smaller than those of lower branch orders. Also, it was noticed that angles of layer VPost are smaller in all branch orders than the corresponding ones in layers IIIPost. See Supplementary Table 2, rows 1-5.

Regarding the concentration of the angles around the mean, angles in general showed a tendency, when compared between layers, to be similar (Supplementary Table 3). The comparison between layer IIIPost and layer IIIAnt deviated the most from these results, suggesting that the angles in layer IIIAnt may be slightly lower concentrated (see Supplementary Table 3, rows 1-5). Intuitively, a lower concentration around the mean in layer IIIAnt branching angles implies that it is more likely to find an observation far distant from the mean in layer IIIAnt than in layer IIIPost.

Regarding the boundaries of the branching angles, the minimum angles variation (i.e., the variation of the lowest angles per bifurcations) seemed clearly lower, with a circular variance of 0.0014 radians for layer IIIPost branch orders, 0.0043 radians for layer VPost and 0.0003 radians for layer IIIAnt, than the maximum angles variation (the variation of the highest angles per bifurcations), with a circular variance of 0.163 radians for layer IIIPost, 0.193 radians for layer VPost and 0.038 radians for Layer IIIAnt (see Supplementary Tables 5-7 for the a and b truncation parameters that correspond with the minimum and maximum angular values).

Test-based comparisons showed that each branch order resulted significantly different from all the other branch orders except in the comparisons with the branch order 5 (Figure 6.2D), which could not be rejected for branch orders 3 and 4 in Layer IIIPost, branch orders 3 and 4 in layer VPost and branch order 4 in Layer IIIAnt. All other cases presented a complete absence of links between the nodes in the test-based diagram (i.e., all tests results were rejections). Comparisons with branch order 5 may be interpreted with caution due to the small number of observations available.

The goodness-of-fit tests for the truncated von Mises distribution and the von Mises distribution revealed modest results, with the truncated von Mises scoring 3/5 non-rejections for Layer IIIPost, 3/5 non-rejections for layer VPost and 3/5 non-rejections for Layer IIIAnt (Table 6.1, rows 1-5). The von Mises distribution scored 3/5 non-rejections for Layer IIIPost, 2/5 non-rejections for layer VPost and 1/5 non-rejections for Layer IIIAnt (Table 6.1, rows 1-5). These results show a slightly better performance for the truncated von Mises distribution in this case (the estimated parameter values of the truncated von

Mises distribution, obtained in the tests, can be found in the Supplementary Tables 5-7, rows 1-5).

Table 6.1: Goodness-of-fit values for the truncated von Mises distribution (TvM) and the von Mises distribution (vM) for the three datasets and the two different studies. The numerical value in each cell represents the $p-$value of the goodness-of-fit test. The notation OX is read as "branch order X"(for example, O3 is the branch order 3, this notation is used for the study in "Data acquisition and preparation") and the notation MaxXOY is read as "Maximum branch order X, branch order Y"(for example, Max2O1 is the branch order 1 of dendrites with maximum branch order 2, this notation is used for the study in "Univariate truncated con Mises distribution"). If a cell contains the symbol * it indicates that the test hypothesis was not rejected, whereas if the * symbol is missing, the opposite occurred.

	Layer IIIPost		Layer VPost		Layer IIIAnt	
	TvM	vM	TvM	vM	TvM	vM
O1	*0.6268	*0.6465	*0.4353	0.0393	*0.9663	*0.6428
O2	*0.5562	*0.9626	0.0872	*0.1482	0.0458	<0.001
O3	0.0813	0.0137	0.0370	0.0038	*0.1124	<0.001
O4	0.0688	0.0061	*0.1849	<0.001	*0.2141	<0.001
O5	*0.8735	*0.8476	*0.5509	*0.1693	0.0220	<0.001
Max1O1	*0.3985	*(0.1,0.2)	*0.7195	<0.001	*>0.95	(0.01,0.05)
Max2O1	*0.3985	0.0524	*0.8388	<0.001	*0.4316	0.0654
Max2O2	*0.5142	0.0575	*0.4207	0.0488	*0.2275	<0.001
Max3O1	*0.8434	*0.4830	*0.4697	*0.1870	*0.3770	*0.2551
Max3O2	*0.9504	*0.7647	*0.4966	0.0177	*0.653247	0.0172
Max3O3	*0.2021	*0.2718	*0.1983	0.0280	*0.2477	<0.001
Max4O1	*0.7246	*0.7626	*0.9129	*0.3953	*0.8469	*0.6671
Max4O2	*0.4771	*0.4926	*0.8063	*0.9781	*0.2547	0.0734
Max4O3	*0.6594	0.0079	*0.7752	0.0010	*0.2928	<0.001
Max4O4	*0.2578	0.0213	*0.2962	<0.001	*0.2030	<0.001
Max5O1	*0.7556	*0.1723	*0.9230	*0.8568	*0.9666	*0.5508
Max5O2	*0.7343	*0.3677	*0.6352	<0.001	*0.4883	0.0622
Max5O3	*0.5558	*0.1008	*0.8770	0.0027	*0.6385	<0.001
Max5O4	*0.1101	0.0294	*0.8498	*0.1210	*0.6153	0.0205
Max5O5	*0.9778	0.0043	*0.9602	*0.4863	0.0572	<0.001

6.3.2 Study of branching angles by branch order grouped according to their maximum branch order

Then, we compared angles of different branch orders originating from dendritic trees of similar complexity (i.e. different dendritic trees were grouped according to their maximum branch order). The analysis showed that the previously observed tendencies for the median (Figure 6.3), the tests for the mean (see Supplementary Table 1, rows 11- 30 and Table 2, rows 6-20) and the concentration around the mean (see Supplementary Table 3, rows 6-20) hold also for this study.

It was found that mean values of the first branch order angles increase with respect to the maximum branch order (Supplementary Table 4), this was discovered by comparing only the first branch order of dendritic trees with different maximum tree orders. In the case of the boundaries of the branching angles, it seems that the angles of the highest branch order cover a relatively small interval of angles in

each maximum branch order subgroup, although it is not clear that the interval of angles decreases with the branch order as the mean does. The observed variance on the maximum angles was higher than the variance on the minimum angles in all cases also for this study (see Supplementary Tables 5-7, rows 6-20 for parameter values).

The similarities between branch orders resulted to be scarce, with the majority of the comparisons producing test rejections (Figure 6.3). For this case, the layer with more non-rejected comparisons was layer V and the lowest $p-$values (closer to similarity) were generally found between first and second order branchings.

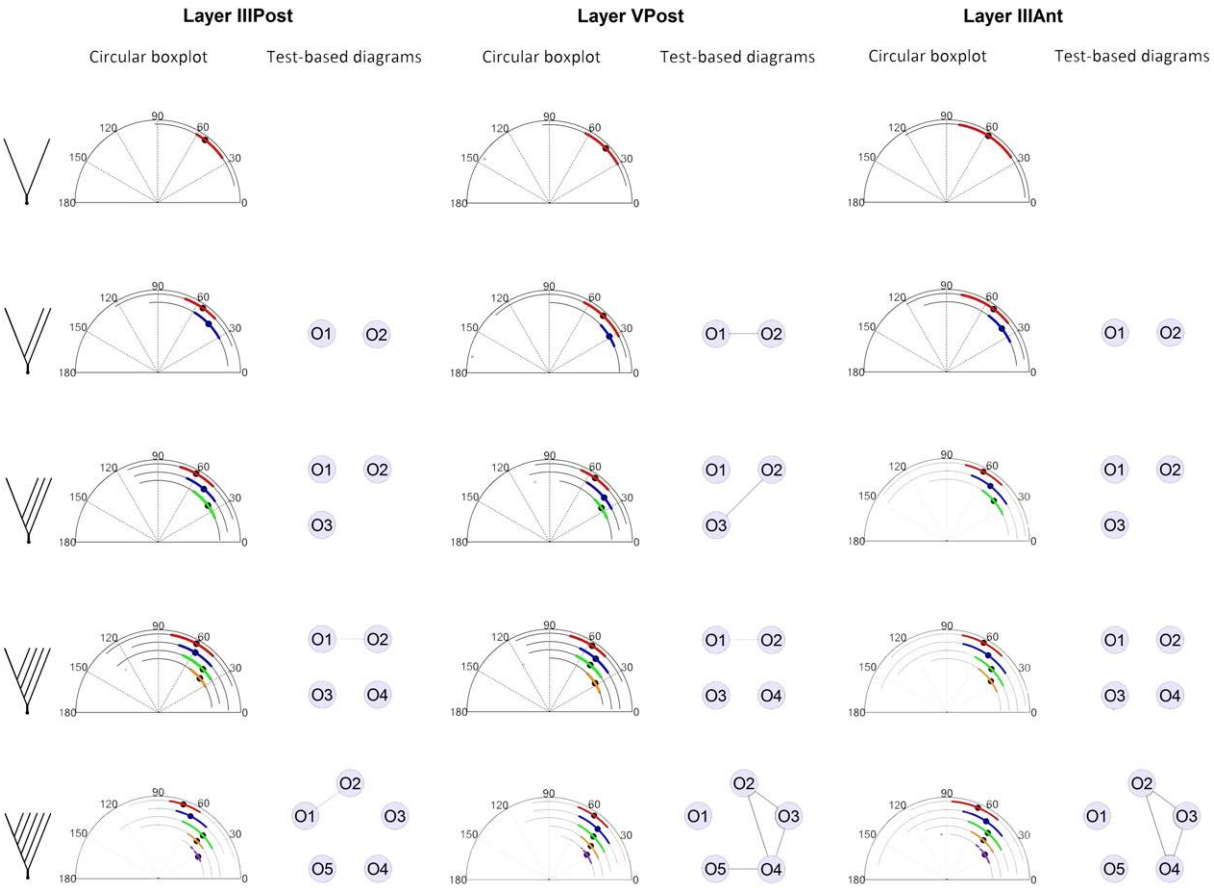


Figure 6.3: *Circular boxplots* and associated test-based diagrams coming from basal dendritic trees of pyramidal neurons grouped according to their branch complexity.

When performing the goodness-of-fit tests, we obtained very good results for the truncated von Mises distribution with 15/15 non-rejections for Layer IIIPost, 15/15 non-rejections for layer VPost and 14/15 non rejections for layer IIIAnt. The von Mises distribution scored 9/15 non-rejections for layer IIIPost, 7/15 non-rejections for layer VPost and 3/15 non-rejections for layer IIIAnt (Table 6.1, rows 5-19). This shows that the truncated von Mises distribution clearly outperforms the von Mises distribution in all cases (the estimated parameter values of the truncated von Mises distribution, obtained in the tests, can be found in the Supplementary material, Tables 5-7, rows 6-20). These results strengthen our belief in that grouping the data by maximum branch order and branch order is a more appropriate way to study branching angles in dendrites. It could partially shed light on why the results of grouping the data merely by branch orders are less informative.

6.3.3 Comparison of pairs of angles of contiguous orders

The data was further compared in pairs of contiguous branching angles to explore the possibility that angles of the first branching may somehow influence the angles of the second branch order, using a bivariate truncated von Mises distribution. We only used the data of layer IIIAnt since bivariate estimations require higher sample size than the univariate case. We studied if there was a measurable dependency between pairs of contiguous branch orders when fitting the distribution. We performed a Rothman's test for independence over the data of contiguous branch orders (see Supplementary Table 8). We also performed a permutation test (results not included) for $\lambda = 0$ in our fitted models, where λ is the parameter in the bivariate truncated von Mises distribution that measures the level of dependency between the two random variables (if its value is 0, both variables are considered independent). Tests results showed independence in almost all cases.

6.3.4 Comparison between layer IIIPost neurons and layer VPost neurons

Next step was to compare angles per branch order between layer III and V. This comparison showed statistical differences with only 1/5 tests not rejected, which is the corresponding to the branch order five comparison between the two layers (Figure 6.4A, see Supplementary Table 9, rows 1-5). Then we grouped the angles additionally by maximum branch order. In this case, we found a majority of differences (test rejections) with only 5/15 tests not-rejected. More precisely, the tests that produced a non-rejection result correspond to the first branching of dendrites of maximum branch order one, three, and four, and the branch orders three and five of the dendrites of maximum branch order five (Figure 6.4B, see Supplementary Table 9, rows 6-20). We found that, in general, angles in the first order are the most similar of all the orders compared in the same maximum branch order group and the overall most similar (i.e., they obtained generally higher $p-$values in the tests). We concluded that layers IIIPost and VPost can be considered statistically different.

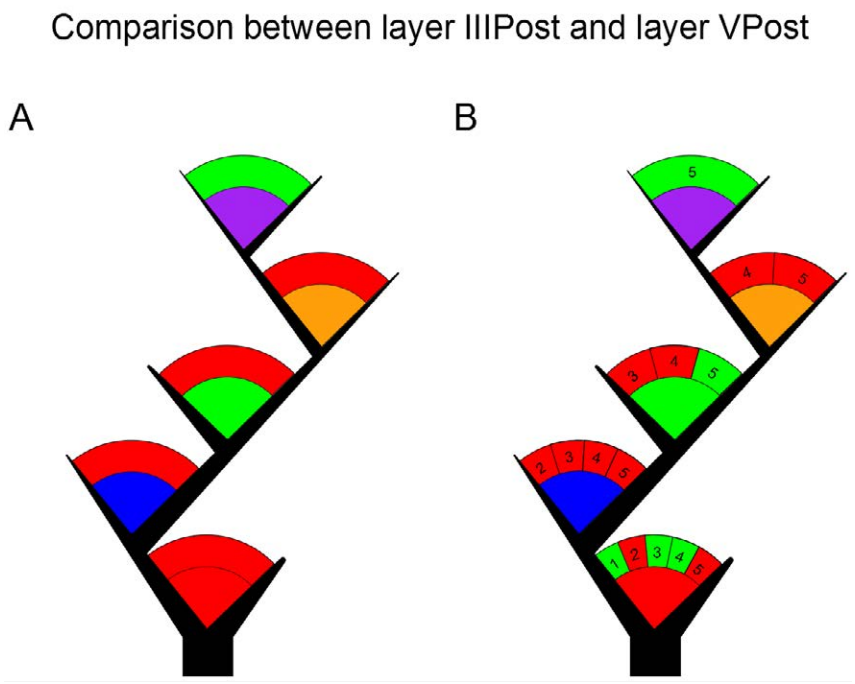


Figure 6.4: **A** Test-based tree illustrating pairwise comparisons between the branch orders in layers IIIPost and VPost. If the arc that appears above the branch order color code is red, the test produced a rejection result. If the arc is green, the result was non-rejection. **B** Comparisons of branch order angles grouped according to their maximum branch order. The numbers in the arc above the branching color codes indicate the maximum tree order and each of the subdivisions of the arc corresponds to a test. As an example, the first branch order in the graphic shows the information of five tests performed to the first branch order of dendrites with maximum tree order one, two, three four and five.

6.3.5 Comparison between layer IIIPost neurons and layer IIIAnt neurons

Similarly, we compared angles per branch order between neurons from different antero-posterior regions of the temporal cortex. We found that only 1/5 tests were not rejected (Figure 6.5A, see Supplementary Table 10, rows 1-5), which is the corresponding to the branch order five comparison. When we also grouped angles additionally by maximum branch order, and we found that non-rejections were a clear majority with 12/15 tests passed. As in the previous study in Section 6.3.2, the angles in the first branch order could be generally considered more similar (i.e. higher $p-$values), while the least similar angles were located around the branch or-der two, with two tests rejected for maximum branch orders three and four (Figure 6.5B, see Supplementary Table 10, rows 6-20). We conclude not enough statistical evidence was gathered to consider layers IIIAnt and layer IIIPost to be significantly different from each other.

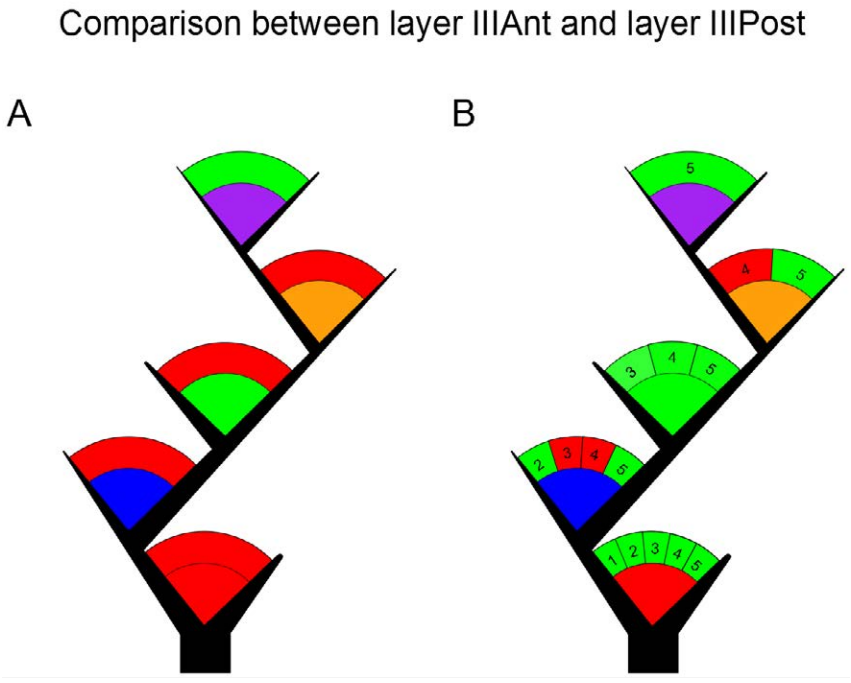


Figure 6.5: **A** Test-based tree illustrating pairwise comparisons between the branch orders in layers IIIAnt and IIIPost. If the arc that appears above the branch order color code is red, the test produced a rejection result. If the arc is green, the result was non-rejection. **B** Comparisons of branch order angles grouped according to their maximum branch order. The numbers in the arc above the branching color codes indicate the maximum tree order and each of the subdivisions of the arc corresponds to a test. As an example, the first branch order in the graphic shows the information of five tests performed to the first branch order of dendrites with maximum tree order one, two, three four and five.

6.3.6 Comparison between layer IIIAnt and IIIPost neurons and layer III neurons from mice and rats

We use the data from Leguey et al. [2016] for the rat neuronal data, selecting only the layer III subset. For the mouse data, we use the data from Ballesteros-Yáñez et al. [2010] selecting only the layer III subset of the wild-type mice data subset. We first compared angular ranges eliminating 5% of the lowest values and 5% of the highest values. The remaining 90% of the angular vales showed remarkable range similarities as they ranged from 13 to 98 degrees in humans (IIIAnt and IIIPost data combined), 17 to 92 degrees in rats, 20 to 97 degrees in mice. However, a two sample Watson test for similarity (same distribution) between layers III neurons of human, rat and mouse reveals significant differences between the three species (Supplementary Table 11). We further expanded our comparison between human and mouse cortical areas and performed comparisons between the layer IIIAnt and IIIPost for humans and the data for mice grouped according to seven different cortical areas, which included: primary motor cortex, secondary motor cortex, prelimbic/infralimbic cortex, primary somatosensory cortex, secondary somatosensory cortex, primary visual cortex and secondary visual cortex. Results show in more detail the dissimilarity between both datasets with only 1/14 non-rejected tests. More specifically, we found Layer IIIPost similar to primary somatosensory cortex (see Supplementary Table 12).

6.3.7 Comparison between different humans under various groups of data

We now split the data into five different groups according to the different humans that generated the data. The different labels that identify them are H153, H155, H213, H263 and H264. The first comparison was between the data grouped only by different humans. The results show a majority of test rejections (9/10) with the only exception between the data of H155 and H153 (Supplementary Table 13). Subsequently, we analyze first order branch angle only of those groups, with the goal to locate the source of the diversity among individuals. We found that for the first branch order only, the data is remarkably different from the first study, showing a majority of non-rejections for similarity (8/10). We then continued to test other branch orders, and found that for branch order two, results are similar to the global study with 9/10 rejections for the same pairs of combinations, leaving the comparison of H153 and H155 as the only non-rejected case (Supplementary Table 14). Finally, we compared the number of branching angles per dendrite for all different humans, with resulted in a mixed combination between rejections (i.e. the number of nodes per dendrite does not follow a similar distribution in the comparison) and non-rejections (5/10 in both cases) (Supplementary Table 15).

6.4 Discussion

In this Chapter the main objective was to analyze the branching angles of human layers III and V pyramidal neurons with the aim of trying to find a statistical distribution that properly models branching angles in human pyramidal neurons, and to find out possible differences and similarities between branching angles in different cortical layers of the temporal cortex. Furthermore, we compared the branching angles of human layer III pyramidal neurons with data obtained in previous studies in layer III of the rat somatosensory cortex (Leguey et al. [2016]) and in several cortical areas of the mouse (Bielza et al. [2014]). The main conclusions are the following:

1. The truncated von Mises distribution seems to improve the results of the von Mises distribution to model branching angles, with excellent results in modeling the data.

2. Moreover, we found that branch orders nearer to the soma have the widest angles and that they gradually decrease as the branch order increases in all groups. This was more evident when angles are selectively grouped according to the maximum branch order of their dendritic trees in all groups, suggesting that bigger trees tend to require wider first order angles to grow.

3. The variations between the minimum branching angles, per branch order and maximum tree order, were clearly lower than the variation of the maximum angles, which could imply that the highest branch order angles vary less than, for example, first order angles, which perhaps is related to the fact that first order angles have to allow the dendrite to grow, while the last branch order angles are the only ones that do not have to.

4. Branch orders are shown to be statistically different from each other, which seems to be a further evidence that in the process of building a dendrite, different branch orders follow different patterns (i.e., they have to be modeled separately at least until general variation rules between branchings are found).

5. Independence tests have shown that no measurable dependency is observed between branching orders. In this direction, future work could be to consider other forms of dependency or other ways of splitting the data where such supposed dependencies could be observed.

6. Regarding comparisons between layers III and V, angles in layer VPost were found to be clearly smaller than the angles in layer IIIPost, whereas the concentration of the angles was similar in all cases for both layers. The similarity tests showed that the design principles behind the formation of branching angles differ somehow between the layers IIIPost and VPost, as they can be considered statistically different. Layer IIIAnt branching angles presented slightly lower concentrated angles than layer IIIPost. The similarity tests showed that they cannot be concluded to be statistically different by examining the data. These results are in line with previous studies of pyramidal neurons in layer III of the mouse cerebral cortex (Bielza et al. [2014]).

7. Importantly, the general rules above summarized were similar for pyramidal cells in human, rat and mouse. Furthermore, the range of the angular branching angles showed remarkable similarities between the three species.

8. The five individuals examined showed significant differences in the mean branching angles among them except in one of the comparisons. However, significant differences in the branching angles for branch order 1 was only found in two of the ten comparisons, whereas for branching order 2 all were different except in one comparison. Thus, the differences between individuals are mainly due to branching angles other than for branch order 1.

Therefore, taking into consideration all these results together, we can deduce that there are common design principles that govern the geometry of dendritic branching angles of pyramidal neurons in different layers, cortical areas and species. These results were unexpected as major differences in the structure of pyramidal cells are observed between these neurons in the human, rat and mouse in terms of

the size and complexity of their dendritic arborization, in the density of dendritic spines on their dendritic branches and in the total number of dendritic spines. Thus, the present results further suggest that the branching dendritic angles do not seem to be related to the overall complexity of the dendritic arbors and number of dendritic spines, or if they are related, these differences must be due to relatively small variations in the branching angles. For example, these angles are in general wider in humans compared to rats and mice. Indeed, we found that the distribution of the branching angles of layer III pyramidal cells between the three species were statistically different in spite of the similarities of the ranges. However, when we compared the data between human layer IIIAnt and IIIPost with the data for mice grouped according to seven different cortical areas that were available (primary motor cortex, secondary motor cortex, prelimbic/infralimbic cortex, primary somatosensory cortex, secondary somatosensory cortex, primary visual cortex and secondary visual cortex), we found that Layer IIIPost was similar to primary somatosensory cortex. Thus, further similarities or differences between different species may be found by examining additional cortical regions and layers. Intuitively, the differences between the human and mouse regarding different cortical regions would be expected, given the different functional specializations. Conversely, we do not know why there are similarities between pyramidal cells of human and mouse in areas as different as the posterior temporal cortex of humans and the primary somatosensory cortex of mouse. Therefore, further studies are necessary to include more detailed comparisons between branch orders as the mean angle per area and the range of angles alone do not provide enough information to fully address the issue. In addition, it will be necessary to compare not only between human, rat and mouse pyramidal neurons to try to generalize the results, but also between pyramidal cells of other species as significant morphological differences do exist between other species (reviewed in Jacobs et al. [2001], Elston [2007], Elston et al. [2011], Defelipe [2011], Eyal et al. [2014] and Mohan et al. [2015]), and it is possible that certain morphological features might be related to the dendritic branching angles of particular branch orders in particular cortical layers, areas or species.

Finally, the neocortex tissue of the five patients examined was histologically normal, despite the fact that these individuals were epileptic. This tissue was removed to gain access to the epileptic focus that was located in the mesial structures. In previous studies, it has been shown that the biopsy material obtained during neurosurgical treatment for epilepsy represents an excellent opportunity to study the microanatomy of the human brain because the resected tissue can be immediately immersed in the fixative. Thus, this tissue is lacking possible post-mortem time-induced changes that may occur at both the neurochemical and anatomical levels, which is the major problem when using brain tissue from autopsies. Certainly, this is why the quality of the immunocytochemical staining at both the light and electron microscopy levels in human biopsy material has been shown to be comparable to that obtained in experimental animals (e.g., del Río and DeFelipe [1994] and Alonso-Nanclares et al. [2008]). Therefore, these biopsies are of great value since, for obvious ethical reasons, it is as close to a 'normal'sample of brain tissue as is possible to obtain for studying the human brain. However, a major drawback is that epileptic patients are heterogeneous in terms of their disease history and it is possible that the different medical characteristics of the epileptic patients (i.e., differences in the medication, severity of the disease, onset and duration, etc.) may modify the brain tissue, but we do not have enough cases to analyze this possibility. Interestingly, the five cases examined showed significant differences in the mean branching angles among them except in the comparison between two individuals that were 28 and 41 years old at the time of neurosurgery (H153 and H155, respectively). It is not known whether this represents "normal" interindividual variability or whether the differences observed were due to the different medical condi-

tions. Nevertheless, these two "similar" cases have a rather different medical history regarding the age at onset (9 years old for case H153, 17 years old for H155); the duration (19 years for case H153, 24 years for H155), the seizure frequency (daily for H153, weekly for H155); and the pathology observed in the mesial structures (no apparent hippocampal alterations in H153, hippocampal sclerosis in H155). Thus, we are inclined to think that the differences between individuals may simply be due to interindividual variability. Further studies would be necessary to ascertain the range of variability between pyramidal cells of the human cerebral cortex.

This work has been published as Fernandez-Gonzalez, P., R. Benavides-Piccione, I. Leguey, C. Bielza, P. Larrañaga, and J. DeFelipe, "Dendritic branching angles of pyramidal neurons of the human cerebral cortex", *Brain Structure and Function*, vol. 222, issue 4, pp. 1847-1859, 2017.

Chapter 7

Gaussian Bayesian networks for multidimensional classification of morphologically characterized neurons in the NeuroMorpho repository

7.1 Introduction

Neurons's morphology differences have been observed between different animals, but also within the same species. The developmental stage and the location in the brain can also show morphological variations between cells (Jacobs et al. [2014]). In order to statistically analyze these differences, a multidimensional classifier using an interpretable statistical model is one of the most appealing approaches. To build a model that can effectively predict class labels such as in which species, gender and developmental stage an animal is and to which cell types the sample neuron belongs, given a set of morphological descriptors of the neuron, could be considered a big step towards neuron morphology understanding. Specially if the selected model has the property of interpretability, allowing us to extract knowledge directly from it.

For this work, a class-bridge decomposable multidimensional Gaussian Bayesian network classifier (CB-MGC) is proposed and trained with the neurons dataset of (Ascoli et al. [2007]). This classifier is not bounded to a prefixed structure (naive Bayes, tree-like structures in the class variables, etc.) and also handles variables of continuous (Gaussian) and discrete nature. It is influenced by the works of Pérez et al. [2006] and Borchani et al. [2010]. The classifier's strengths are its interpretability, the capability to capture dependencies between the class variables, the exploitation of the class-bridge decomposability property and its ability to handle feature variables of continuous nature straightforwardly, without the need to discretize the data. Its weakness may be the assumption of Gaussianity in the continuous nodes, where features whose distribution strongly deviates from the Gaussian distribution could hinder the model's performance. However, this is acceptable in this case as the data features tend to distribute according to Gaussian distributions. The definition and properties of this model will be detailed in Section 7.2.

Section 7.3 presents a structural learning algorithm that uses the class-brigde decomposability to incrementally build a complex network structure while saving computational costs in the process. Section

7.4 shows the results and final network with a focus on the implications of the obtained relationships in the final model. Finally, Section 7.5 summarizes the main findings and discusses the conclusions and implications of this work.

7.2 Multidimensional Gaussian Bayesian network classifiers

A multidimensional Gaussian network classifier (MGNC) is a Bayesian network $\mathcal{B} = (\mathcal{G}, \Theta)$ over a set $\mathcal{X}_f = \{X_1, \ldots, X_m\}$, $m \in \mathbb{N}$ of continuous random variables and a set $\mathcal{C} = \{C_1, \ldots, C_s\}$, $s \in \mathbb{N}$ of discrete class random variables where $\mathcal{X}_f$ is assumed to be jointly distributed as a multidimensional Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ is a vector of means and $\boldsymbol{\Sigma}$ is the covariance matrix of the variables in $\mathcal{X}_f$. $\mathcal{X}_f$ and $\mathcal{C}$ are referred to as the set of feature variables and the set of class variables, respectively. MGNCs are additionally constrained to satisfy $\mathbf{Pa}(C_i) \cap \mathcal{X}_f = \varnothing$, that is, no arcs from feature variables to class variables are permitted. Multidimensional classifiers have been studied initially in Van Der Gaag and De Waal [2006], and extended in Borchani et al. [2010] and Bielza et al. [2011].

In concordance with the literature, MGNCs can be additionally described by considering three different subgraphs in its structure:

- $A_{\mathcal{C}} \subseteq V_{\mathcal{C}} \times V_{\mathcal{C}}$ is the set of arcs connecting solely the class variables and $V_{\mathcal{C}}$ is the set of vertices representing the class variables. The associated subgraph, that contains as nodes all the class variables and is induced by $V_{\mathcal{C}}$, is denoted as $\mathcal{G}_{\mathcal{C}} = (V_{\mathcal{C}}, A_{\mathcal{C}})$
- $A_{\mathcal{X}_f} \subseteq V_{\mathcal{X}_f} \times V_{\mathcal{X}_f}$ is the set of arcs connecting solely the feature variables and $V_{\mathcal{X}_f}$ is the set of vertices representing the feature variables. The associated subgraph, that contain as nodes all the feature variables and is induced by $V_{\mathcal{X}_f}$, is denoted as $\mathcal{G}_{\mathcal{X}_f} = (V_{\mathcal{X}_f}, A_{\mathcal{X}_f})$
- $A_{\mathcal{C}\mathcal{X}_f} \subseteq V_{\mathcal{C}} \times V_{\mathcal{X}_f}$ is the set of arcs that go from the class variables to the features variables. The associated subgraph comprehends all nodes of the network as is denoted as $\mathcal{G}_{\mathcal{C}\mathcal{X}_f} = (V, A_{\mathcal{C}\mathcal{X}_f})$

For this type of models, classification using a 0-1 loss function (see Section 3.2.4.2) amounts to solving the most probable explanation problem (that is, the search of the class labels that maximize the probability of the class variables given the evidence of the feature variables). When calculating the MPE in a MGNC, it is possible to use Equation (3.25) to compute the MPE by considering that $p(\mathbf{c}|\mathbf{x}) \propto p(\mathbf{c}, \mathbf{x})$, where $p(c_i|\mathbf{pa}(c_i))$ is computed as a classical discrete probability in a BN and for the feature nodes, $f(x_i|\mathbf{pa}(x_i))$ follows a Gaussian distribution $\mathcal{N}(\mu_i, \sigma_i^2)$, where

- $\mu_i = \mu_{i|pc_i} + \sum_{j=1}^{d_{p_i}} \beta_{ij|pc_i}(x_j - \mu_{j|pc_i})$
- $\sigma_i^2 = \dfrac{\left|\boldsymbol{\Sigma}_{X_i, PX_{i|pc_i}}\right|}{\left|\boldsymbol{\Sigma}_{PX_{i|pc_i}}\right|}$

where $pc_i = \mathbf{pa}_{V_{C_i}}(x_i)$ is the set of class parents of X_i, $\mathbf{pa}_{V_{\mathcal{X}_{fi}}}(x_i)$ is the set of feature parents of X_i (px_i or PX_i for notation conciseness), $d_{p_i} = |\mathbf{pa}_{V_{\mathcal{X}_{fi}}}(x_i)|$ is the number of feature parents of X_i , $\beta_{ij|pc_i}$ is a regression coefficient defined as:

$$\beta_{ij|pc_i} = \frac{\sigma_{ij|pc_i}}{\sigma^2_{j|pc_i}}$$

where $\sigma_{ij|pc_i}$ is the covariance of X_i and X_j conditioned to the class parents of X_i, $\sigma^2_{j|pc_i} = \sigma_{jj|pc_i}$ and $\mathbf{\Sigma}_{L|pc_i}$ is the covariance matrix of the set of variables L conditioned to the class parents of X_i.

A GBN possesses several desired properties such as the less demanding number of parameters to model a continuous distribution and the possibility to compute them independently from the structure of the GBN (Geiger and Heckerman [1994]). The computation of the MPE, however, concerns only the class variables, that is, the discrete part of the network, and therefore no complexity alleviation was found for inference by assuming Gaussianity in the feature nodes. This is a well-known problem as when learning an unrestricted class structure the MPE problem is exponential in the number of variables. This issue renders the inference intractable even for a relatively small set of class variables.

7.2.1 Class-bridge decomposability property

In order to tackle the inference problem, CB-decomposable MGNCs are considered, extending previous works Bielza et al. [2011] and Borchani et al. [2010], defined over discrete feature variables. A MGNC is a CB-decomposable MGNC if it satisfies the following two properties:

- $\mathcal{G}_{\mathcal{C}} \cup \mathcal{G}_{\mathcal{C}\mathcal{X}_f}$ can be partitioned as $\mathcal{G}_{\mathcal{C}} \cup \mathcal{G}_{\mathcal{C}\mathcal{X}_f} = \bigcup_{i=1}^{r}(\mathcal{G}_{C_i} \cup \mathcal{G}_{(CX)_i})$, where $\mathcal{G}_{C_i} \cup \mathcal{G}_{(CX)_i}$, for $i = 1, \ldots, r$ are complete subgraphs of the original graph, that is, maximal connected components[1].

- $Ch(V_{C_i}) \cap Ch(V_{C_j}) = \varnothing$ with $i, j = 1, \ldots, r$ and $i \neq j$, where $Ch(V_{C_i})$ stands for the set of children variables of V_{C_i}, the subset of class variables in $\mathcal{G}_{C_i}$ (i.e non-shared children property).

Then the MPE problem for a CB-decomposable MGNC is transformed into

$$\arg \max_{c_1,\ldots,c_n} p(C_1 = c_1, \ldots, C_n = c_n|\mathbf{x})$$
$$\propto \prod_{i=1}^{r} \max_{\mathbf{c}^{\downarrow V_{C_{i\in I_i}}}} \left(\prod_{C\in V_{C_i}} p(c|\mathbf{pa}(c)) \prod_{X\in Ch(V_{C_i})} p(x|\mathbf{pa}_{V_{\mathcal{C}}}(x), \mathbf{pa}_{V_{\mathcal{X}_f}}(x)) \right)$$

where $\mathbf{c}^{\downarrow V_{C_{i\in I_i}}}$ is the projection of the vector $\mathbf{c}$ to the coordinates in V_{C_i} and I_i stands for the sample space associated with V_{C_i} (that is, $I_i = Val\,(V_{C_i})$ in shorter notation). Intuitively, this breaks the MPE problem into r smaller MPE problems (Figure 7.1). Given the exponential nature of the total of possible label combinations with respect to the number of class variables, this effectively alleviates the computational burden as well as the sample size needed for the classification problem. It is also possible to see this property in the factorization of the network, as each component is identified as a subset of the network factors whose class variables form a closed group (that is, no other reference is found to them in the rest of the factors of the network).

[1] A graph is "connected" if for every pair of its vertices there is an path, without regard for the direction of the arcs, that links them together.

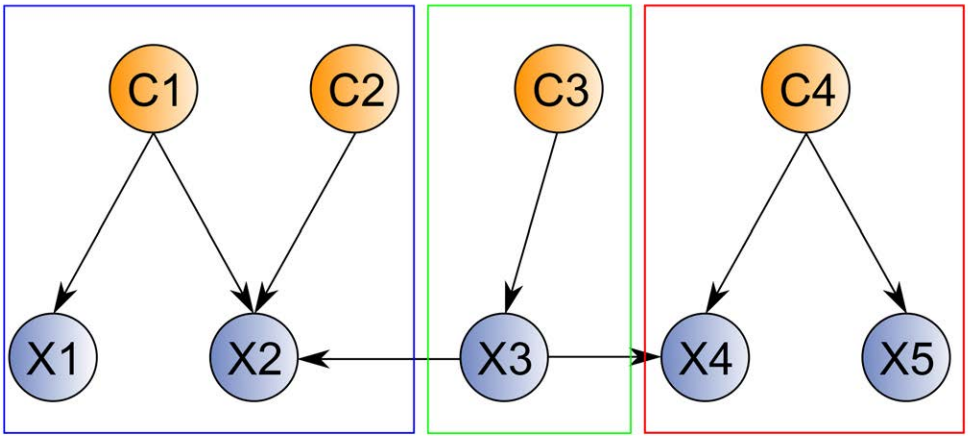


Figure 7.1: A Bayesian network structure showing three maximal connected components.

7.3 Structural learning algorithm

The proposed learning algorithm can be characterized as a three-step learning algorithm with a greedy forward search approach. That is, arcs are initialized to the empty set for the three different subgraphs $A_{\mathcal{C}} = \varnothing, A_{\mathcal{X}_f} = \varnothing$ and $A_{\mathcal{C}\mathcal{X}_f} = \varnothing$, obtaining an initial network with no arcs and all nodes present. Then, it follows with the addition of arcs to the different parts of the network (filling bridge, feature and class subgraphs in that order) judging their contributions using the global accuracy score in all three steps. The addition of arcs is designed to exploit the CB-decomposablility property to scale to complex network structures without unnecessary computational burden. This is accomplished by controlling the number of maximal connected components at any given moment, only reducing it after less costly arc insertions have taken place. This effectively allows it to learn a relatively complex structure before computational complexity becomes an issue. The final stage of computation only increases complexity when its unavoidable to do so, and it is capable of producing topologically unrestricted class subgraphs.

Once our model is built, we evaluate the inference performance of the trained model using the Hamming score HS (which is simply $1 - HL$, where HL is the Hamming loss) and global accuracy metrics (see Section 3.2.3). From an algorithmical perspective, however, we can define these metrics in a data-dependent, more comprehensive way, respectively, as follows:

$$HS = \frac{1}{n}\sum_{i=1}^{n}\frac{|T_i \cap P_i|}{|T_i \cup P_i|},$$

where, T_i is the set of true values for the variables in $\mathcal{C}$ in the ith instance in dataset $\mathcal{D}_n$, and P_i is the set of predicted labels for variables in $\mathcal{C}$ by the classifier for the ith instance, and

$$GA = \frac{1}{n}\sum_{i=1}^{n}\delta_{T_i}^{P_i},$$

where $\delta_{T_i}^{P^i}$ is a function that outputs 1 if $T_i = P_i$ and 0 otherwise.

The algorithm was tested using a train/split fashion where $\frac{1}{3}$ of the total dataset instances were used and randomly chosen for testing.

7.3.1 Learning the bridge subgraph

The algorithm first focuses on building a naive Bayes subgraph $NB_i(C_i, \mathcal{X}_e)$, with $\mathcal{X}_e \subset \mathcal{X}_f$ for each class variable $C_i, i = 1, \ldots, s$ of the network, over which a sequential feature subset selection process is carried out. First, the features are grouped according to their separation power with respect to C_i by means of a Kruskal-Wallis test (Kruskal and Wallis [1952]). In order to do this, each feature data is partitioned into subgroups according to the class label. Then, the Kruskal-Wallis test is used to measure whether the population of subgroups originate from the same probability distribution. Since features with lower $p-$values are considered to be more relevant for classification, they are sorted in ascending value. Then, the sequential feature subset selection technique is applied, which adds arcs from the C_i variables to X_j variables if an accuracy improvement is detected.

Finally, it eliminates shared children in order to obtain an initial CB-decomposable MGNC structure with the maximum number of r maximal connected components, where $r = s$ since each naive Bayes graph is a maximal connected component. In order to do this, it compares the $p-$values obtained in Kruskal-Wallis test for classes C_i and C_l and variable X_j and removes the arc that has a higher associated $p-$value. If $p-$values are equal, arc removal is chosen randomly. Algorithm 7.1 outlines the procedure.

Algorithm 7.1 Learning bridge subgraph

Require: $\mathcal{C}, \mathcal{X}_f, s, m$
for $i = 1$ to s **do**
  Select class variable C_i
  Initialize the set of features as $\mathcal{X}_i = \varnothing$
  for $j = 1$ to m **do**
    Obtain $p-$value from $kwpval(X_j, C_i)$ // The Kruskal-Wallis test after separating feature X_j according to the values of C_i
  end for
  Sort features according to ascending $p-$values
  for $j = 1$ to m **do**
    if $Acc(NB_i(C_i, \mathcal{X}_i)) < Acc(NB_i(C_i, \mathcal{X}_i \cup X_j))$ **then**
      $NB_i(C_i, \mathcal{X}_e) := NB_i(C_i, \mathcal{X}_i \cup X_j)$
    end if
  end for
end for
Compare all the children of all NB_i
for all $NB_a(C_a, \mathcal{X}_a), NB_b(C_b, \mathcal{X}_b)$ such that $\mathcal{X}_a \cap \mathcal{X}_b \neq \varnothing$ **do**
  for all $X_p \in \mathcal{X}_a \cap \mathcal{X}_b$ **do**
    $p-$values comparison for feature X_p and clases C_a and C_b
    if $kwpval(X_p, C_a) > kwpval(X_p, C_b)$ **then**
      Remove arc from C_a to X_p in NB_a
    else if $kwpval(X_p, C_a) < kwpval(X_p, C_b)$ **then**
      Remove arc from C_b to X_p in NB_b
    else
      Arc removal chosen randomly
    end if
  end for
end for
Output $\mathcal{G}_{\mathcal{C}\mathcal{X}_f} = \bigcup_{i=1}^{s} NB_i$

7.3.2 Learning the feature subgraph

The second step is to obtain the feature subgraph, for which a maximum number of iterations (parameter t), of arc insertions attempts, is defined. This decision was adopted to avoid the computational burden of examining all possible arc insertions. First, the algorithm calculates the global accuracy that corresponds to the concatenation of the individual class predictions of all existing maximal connected components.

Arc insertions between two unselected features in the previous process are not permited, while the other cases are allowed. This may allow unselected features in the previous process to become part of the model structure. When an arc insertion occurs, the parent feature is added to the component. For each arc insertion between a pair of nodes $X_i \rightarrow X_j$ the accuracy is recalculated. It is important to note that because of the CB-decomposability property, at this step only the MPE values for the class of the component containing the children node need to be recalculated. If there is a global accuracy improvement, the arc insertion is kept, otherwise is discarded. Because accuracy is used as the metric for the arc insertions, this is a wrapper structural learning step.

7.3.3 Learning the class subgraph

For the final graph, the algorithm tries to identify the existing dependencies between class variables and attempt to merge the r maximal connected components. It does this, like in the previous step, in a wrapper fashion. The algorithm starts by considering all possible pairwise components mergings. For each component, all single arc insertions between classes that belong to different components are evaluated, in both directions. If an improvement in accuracy exists, the arc insertion process continues updating the merged component class subgraph by further arc insertions. This process finishes when no improvement in accuracy is observed. Similarly, the merging component process finishes when no component merging improves accuracy or when the number of components has been reduced to one. It is important to notice that when two components are merged, the MPE values only need to be reevaluated for those two components, leaving the remaining nodes outside. This process of local computations guarantees that the computational burden of the MPE increases exponentially only when an arc insertion produces a network topology that cannot be separated in smaller maximal connected components and involves a higher number of class variables. If there are only two components and are merged, the MPE is computed similarly to a classic exact inference approach involving all class variables. Algorithm 7.2 outlines the method.

Algorithm 7.2 Calculate class subgraph

Require: $\mathcal{C}, \mathcal{X}_f, s, m$ and output of Step 2

Initialize AccImprovement = true, ComponentAccImprovement = true, $\mathcal{R}_c = \{R_1, \ldots, R_s\}$ where each $R_i \in \mathcal{R}_c$ is a GN (initially it is the list of components obtained in Step 2)
while AccImprovement and $|\mathcal{R}_c| > 1$ **do**
  $lR := \varnothing$ //Where the candidate mergings between components are stored
  for all possible R_i, R_j component mergings where $i, j = 1, \ldots, s$ and $i \neq j$ **do**
    $R_{ij} := R_i \cup R_j$
    $aR_{ij} = R_{ij}$
    while ComponentAccImprovement **do**
      Evaluate all possible single arc insertions $C_{R_ik} \rightarrow C_{R_jh}$, $C_{R_ik} \leftarrow C_{R_jh}$ from class nodes of different components in R_{ij}
      if exist arc insertions that improve component accuracy **then**
        select best arc and update R_{ij}
      else
        ComponentAccImprovement = false
      end if
    end while
    if $aR_{ij} \neq R_{ij}$ **then**
      $lR := lR \cup R_{ij}$
    end if
  end for
  if $lR \neq \varnothing$ **then**
    select the best merging of components, R_{ab}, contained in lR, $\mathcal{R}_c := \mathcal{R}_c \setminus \{R_a, R_b\} \cup R_{ab}$
  else
    Accimprovement = false
  end if
end while
Return the obtained CB-MGC = $\bigcup_{i=1}^{|\mathcal{R}_c|} R_i \in \mathcal{R}_c$

It should be noted that the class subgraph is not bounded to any network topology or any subset of all possible networks, which itself offers a great appeal with respect to restricted methods. This learning algorithm operates by scaling the complexity of the network topology through a path that minimizes the computational burden of calculating the MPE at each step, by exploiting the CB-decomposability property.

7.4 Classification of neuron's morphological features

The data was obtained from NeuroMorphov5.7.org, more specifically, the available data from Ascoli et al. [2007]. In its raw form, the dataset contained information about 10880 3D reconstructed neurons, that were later processed with the L-measure tool (Scorcioni et al. [2008]) to extract a total of 215 features describing the neurons morphology. Initially, the dataset was composed of seven class labels (species, gender, brain region, cell type level 1, cell type level 2, development and neocortex) with missing data, which shows that the initial problem is a multidimensional semisupervised classification problem. Another difficulty was that some class labels were heavily imbalanced, with the most extreme case represented by a rabbit's neuron, with only one instance for the species class. Hence, a prepro-

cessing step was conducted combining data imputation (using a 1-NN algorithm) with the elimination of class values that did not reach a critical l number of instances (l can be regarded as a parameter to the final model that shapes the data that the learning algorithm receives). This number was set to be $l = 200$. Preprocessing further continued as for the classifier to optimize its performance, features must not significantly deviate from Gaussianity and data fitting to a Gaussian distribution should be possible under all data subsets originated from conditioning the feature to each class variable. With this, dataset pruning further continued to reach a final count of 5136 instances, 6 classes (the neocortex class variable was left out as most of its values were missing) and a total of 158 features (57 were either too different from Gaussian distributions or had subpopulations with zero variance).

A more detailed description of the class labels can be found in Table 7.1. They conform a class cardinality space of 1440 possible label combinations. The features cover many measurement perspectives of the same celular body and offer a vast amount of information of the morphology of a neuron. For a more detailed description or the morphological details captured by the features, the reader is directed to Scorcioni et al. [2008].

Table 7.1: Class labels in the final dataset

Specie	Gender	Brain region	Cell type level 1	Cell type level 2	Development
drosophila	female	anterior olfactory nucleus	axonal terminal	ganglion cell	adult
human	male	basal forebrain	interneuron	granulle cell	young
monkey		hippocampus	principal cell	medium spiny cell	
ray		neocortex		pyramidal cell	
		optic lobe		tangential cell	
		retina			

The algorithm is now applied, trainning a CB-decomposable multidimensional classifier with the goal of finding relationships in the data that can help us understand and predict how neuron morphology changes across the different class labels. This algorithm was programmed using Matlab (version R2015a) and the Bayes net toolBox package (Murphy et al. [2001]) together with the structural learning package (Leray and Francois [2004]).

As seen in Figure 7.2, six components have been obtained that noticeably differ from each other after the first two steps. The parameter t for arc insertion attempts was fixed at $t = 250$ although it can be observed, in the scarcity of feature to feature arcs, that most of the arc insertions did not improve the final accuracy of the model and hence only a small subset produced definite arc inclusions. The software L-measure generally reports the minimum, maximum average and standard deviation values as descriptive features of some measured aspect of the neuron. In some cases, it can be observed how these values tend to appear together in the components (for example the "parent_daughter_ratio", that measures the ratio between the diameter of a dendrite or axonal segment and its segment prolongations after a bifurcation has taken place, can be seen three times the the component with class variable "development") which is perhaps indicative of a statistical dependency existing between that measured aspect and the class variable connected to the features that describe it. Its also worth noticing that after computing step 2, the same node can appear in two different components, but as child/parent of the feature variables (for example, "taper_1_avg" in the component with class variable "gender" and as a parent of "diameter_sd"

in the component whose class variable is "cell_type_level 2"). When components are merged in Step 3, intersecting features are merged together.

Figure 7.2: The six components after learning the brigde and feature subgraphs.

In Figure 7.3 the found dependencies between classes in the final network, after computing step 3, are visualized. Species represents the major discriminant variable between the morphological features of two neurons, as it conditions all but the development variable. This supports the common intuition that two animals from different species differ more in their morphology than, for example, two animals of the same species but of different genders. Along with intuition also seems to be the dependency between brain_region and cell_type_level 2 as different areas of the brain tend to have different neuron subpopulations. The gender dependency of development suggests that morphological differences between individuals of different genders vary with time (intuitively, this may correspond with the stages of sexual differentiation in the transition from young to adult that some species experience, or a sexual homogenization passing from adult to old). Moreover, cell_type_level 2 seems to be the most dependent of all classes, which also seems intuitive as it is measured at the smallest granularity, that is, "it is the closest to an individual neuron" or the one that has potentially less variability. These findings significantly improve the confidence on those previously hypothesized relationships between these classes.

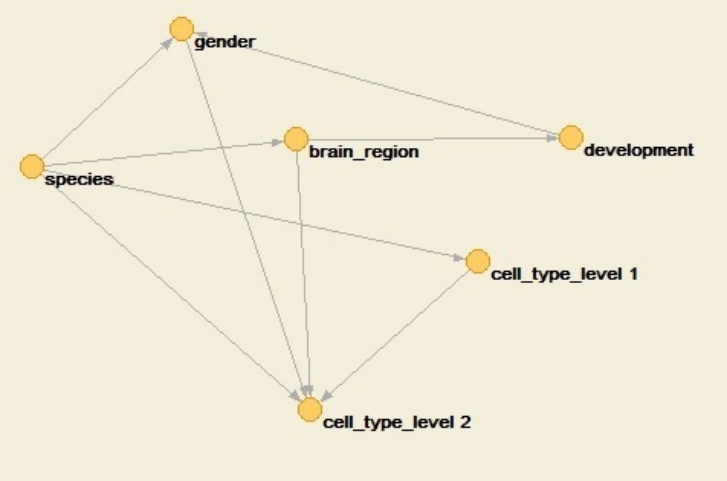


Figure 7.3: Final class subgraph depicting the dependencies found in the data

The final model performance as a multidimensional classifier was measured by the Hamming score and global acuracy metrics, and found values 0.7666 and 0.2288, respectively.

7.5 Conclusions and future lines of research

This new learning algorithm for a multidimensional classifier effectively models and predicts multiple classes provided a set of features. Also, it can be effectively used to build a model that predicts multiple classes of a neuron given a set of morphological descriptors. It is worth noticing that the obtained class subgraph could not have been obtained under common restrictions for multidimensional classifiers, such as independent classes, sequential (chain) dependencies or tree structures. Therefore, this learning algorith produces more expressive models that offer a superior performance in terms on interpretability. This was achieved by the continuous usage of the CB-decomposability property through the learning process, allowing it to scale from a simple to a complex network topology without computing the MPE problem with more variables than necessary. It also succeeds in the objective of extracting useful knowledge out of the data in the field of neuroscience, which we believe validates the application of our model to real life problems and our choice of this model for this problem.

In regards to future work and improvements, the 1-NN data imputation method can be substituted by a method based on the structural learning process (such as an expectation-maximization method). As in its current form, the structural learning algorithm does not explore the addition of arcs between class variables and feature variables that belonged to different components when merged, an investment in computational power that could lead to significant improvements in the classifiers accuracy. Also, the addition of arc removal operations can be considered.

An earlier version of this work has been published as Fernandez-Gonzalez, P., C. Bielza, and P. Larrañaga, “Multidimensional classifiers for neuroanatomical data”, *ICML Workshop on Statistics, Machine Learning and Neuroscience (Stamlins 2015)*, pp. 0-6, 2015. The present work has been published as Fernandez-Gonzalez, P., P. Larrañaga, and C. Bielza, “Bayesian Gaussian networks for multidimensional classification of morphologically characterized neurons in the NeuroMorpho repository”, In *Actas de la 17a Conferencia de la Asociación Española para la Inteligencia Artificial*, pp. 39-48, 2016

Chapter 8

Random forests for regression as a weighted sum of k-potential nearest neighbors

8.1 Introduction

Random forests is a powerful machine learning ensemble method that has achieved state-of-the-art performance in classification and regression tasks. It is computationally fast, produces high accuracy results, has a low parameter count for an ensemble and can handle small sample sizes even with a high number of features. As such, it has earned a wide interest in the research community that spawned a significant amount of papers (Biau and Scornet [2016]). It operates by training multiple decision or regression trees each on bootstrapped samples of the data and combining their predictions most typically by voting (classification) or averaging (regression). In the process of building each tree, a randomly selected subset of the total number of features is used at each time the data is split to search for the locally optimal splitting point (also referred to as the cutoff in continuous variables). To determine the optimal splitting point, a splitting criterion is required. In the random forest literature, the two most used splitting criteria for classification are the Gini impurity and the information gain. For regression, it is the predicted squared error/sum of squared errors.

In this chapter, we focus on RFs (Breiman [2001]) for regression. Initially, we have a training dataset $\mathcal{D}_n = \{(\mathbf{x}_1, y_1), \ldots, \mathbf{x}_n, y_n)\}$ of n i.i.d. samples from a $(d+1)-$dimensional random vector $(\mathcal{X}, \mathcal{Y})$ taking values in $\mathbb{R}^d \times \mathbb{R}$. Our goal is to estimate the regression function $f(\mathbf{x}) = \mathbb{E}[\mathcal{Y}|\mathcal{X} = \mathbf{x}]$ for any $\mathbf{x} \in \mathbb{R}^d$ using $\mathcal{D}_n$. In doing so, we attempt to minimize the mean squared error $MSE = \mathbb{E}[\hat{f}(\mathbf{x}) - f(\mathbf{x})]^2$, where $\hat{f}(\mathbf{x})$ is the regression function estimate of $f(\mathbf{x})$. In this context, we refer to the RF regression estimate as $\hat{f}_{RF}(\mathbf{x})$.

While RF desirability has been displayed at a practical level, sound mathematical understanding of the method is still a lacking subject. For the case of RFs with regression trees, the problem stems from the intricate relationships between bagging (Boostrap + AGGregatING) (Breiman [1996] and Büchlmann and Yu [2002]) and the splitting criteria together, which renders individual regression trees and conventional statistical analyses insufficient for describing the ensemble. In the direction of mathematical understanding of the model, some early works include Breiman [2001], that offered a widely known

upper bound on the generalization error based on the strength (individual classifier's performance) and correlation (similarity of response of the individual classifiers for given inputs) of the members of the ensemble. More recently, Lin and Jeon [2006] showed that when regression trees are grown without pruning and with a fixed parameter k that regulates the tree growth by stopping whenever there are k or fewer examples in a node, the regression function estimate given by a RF algorithm can be viewed as a weighted sum of datapoints:

$$\hat{f}_{RF_1}(\mathbf{x}_0) = \sum_{i=1}^{n} w_i(\mathbf{x}_0) y_i, \tag{8.1}$$

where y_i is the response value associated with datapoint $\mathbf{x}_i$, $w_i(\mathbf{x}_0)$ is a weight that scales the contribution of y_i to the final prediction and $\mathbf{x}_0$ is the target datapoint to be predicted. Additionally, an equivalence relationship between RFs and a special type of nearest neighbors (Fix and Hodges Jr [1951] and Fix and Hodges [1952]) called k-potential nearest neighbors (k-PNNs) was found out. It was shown that if we omit bootstrapping, the regression function estimate given by RFs can be expressed as

$$\hat{f}_{RF_2}(\mathbf{x}_0) = \sum_{\mathbf{x}_i \in P_k(\mathbf{x}_0|\mathcal{D}_n)} w_i(\mathbf{x}_0) y_i, \tag{8.2}$$

where $P_k(\mathbf{x}_0|\mathcal{D}_n)$ is a set containing the k-PNN datapoints of $\mathbf{x}_0$ in $\mathcal{D}_n$. In this setting, different splitting criteria determine different $w_i(\mathbf{x}_0)$ values for each datapoint, and different $w_i(\mathbf{x}_0)$ functions. This work established the foundations for a path towards a sound understanding of the model. Another work, Biau and Devroye [2010], extended Lin and Jeon [2006] and achieved consistency results on a regression estimate that uses the 1-PNN, as well as further understanding of the bagging technique when applied to the well-known nearest neighbors algorithm. Both Equation (8.2) from Lin and Jeon [2006] and the bagging and 1-PNN analyses of Biau and Devroye [2010] have been sources of inspiration for the work of this chapter.

While Equation (8.1) shows that the regression function estimate of RFs can be expressed in terms of the weights, an explicit expression for the weights is still unknown for any splitting criterion. Moreover, RFs equipped with non adaptive splitting criteria (i.e., that do not depend on the Y values) such as random splitting, while being studied and widely regarded as a simpler case of RF (Cutler and Zhao [2001], Geurts et al. [2006]), still lack an explicit expression of these weights. In this direction, while literature concerning bagged regression estimates as weighted sums of datapoints is relatively abundant for some selected regression estimates (Stone [1977], Samworth [2012], Caprile et al. [2004]), the general consensus is that the bagged form of a regression estimate cannot be computed analytically for most cases and Monte Carlo simulation must be used instead (Steele [2009]).

An explicit expression for the weights for a given splitting criterion would propose an alternative to the need of training stage for a RF model building algorithm, shifting all computational burden to the estimation of regression values of new examples and completely eliminating trees (effectively overcoming the Monte Carlo computational approach). Additionally, an equivalence between RFs and other more understood models could provide additional insights that could help us understand the unknown theoretical underpinnings of RFs. To the best of our knowledge, these weights are more directly discussed in Biau and Devroye [2010], characterized as "nonnegative Borel measurable functions of" $\mathbf{x}_0$ that sum to 1, but no method for the explicit, analytical expression of these weights can be found in the literature.

Together with this, Lin and Jeon [2006] and Biau and Devroye [2010] analyses left some open questions: In Lin and Jeon [2006], the k-PNN equivalence was discovered, but bootstrapping was discarded

as a simplification on the RF models in order to make the analysis affordable. Thus, the question of the k-PNNs relationship with RFs equipped with bootstrapping remains unsolved. In Biau and Devroye [2010], no analysis was performed on the bagged 1-PNN regression estimate and results for $k > 1$ were not considered, both remaining as open problems.

In this Chapter we propose a framework for the analysis and explicit calculation of the weights corresponding to general RFs, using bootstrapping and different splitting criteria, effectively answering all previously exposed concerns.

In Section 8.2 we review in detail the concept of k-PNNs and outline some of its most interesting properties.

In Section 8.3 we solve the problem of determining the influence of bootstrapping on bagged estimators (including RF) in terms of weights using k-PNNs. We call these weights bootstrapped weights and obtain results for $k = 1$.

In Section 8.4 we analyze the addition of splitting criteria to our previous developments. We first derive an upper bound on the final weights for any type of splitting criterion, and follow it by the proposal of a regression estimate called random k-PNN selection. We then extend the results of Section 8.3 for arbitrary k by means of a proposed notation on the bootstrap variations, denominated b-terms. Additionally, we use this notation to derive explicit weights for the random k-PNN selection regression estimate. Finally, we introduce a framework to derive bagged estimators for the general case of a splitting criterion and with it, we obtain another regression estimate that corresponds with a RF that uses random splitting criterion and stops at k datapoints in its leaves.

In Section 8.5 we validate the predictive behavior of both the random k-PNN selection regression estimate and the obtained RF-equivalent regression estimate with some practical experiments, to illustrate the results of our work.

Finally, in Section 8.6 we summarize our work and present our conclusions.

8.2 k-potential nearest neighbors

Intuitively, a datapoint $\mathbf{x}_i$ in the feature space $\mathbb{R}^d$ is considered a k-potential nearest neighbor (k-PNN) (Lin and Jeon [2006]) of another, $\mathbf{x}_0$, if the hyperrectangle defined by $\mathbf{x}_i$ and $\mathbf{x}_0$ as opposing vertices ($\mathbf{x}_0$ not included) contains k or less datapoints in the feature space. Formally:

Definition 8.2.1. Let $R(\mathbf{x}_0, \mathbf{x}_i)$ denote the set of datapoints contained in the hyperrectangle defined by $\mathbf{x}_0$ and $\mathbf{x}_i$ as opposing vertices ($\mathbf{x}_0$ not included) in the feature space in $\mathcal{D}_n$. Then $\mathbf{x}_i$ is a k-PNN of $\mathbf{x}_0$ in $\mathcal{D}_n$ if and only if $|R(\mathbf{x}_0, \mathbf{x}_i)| \leq k$ $(\mathbf{x}_i \in P_k(\mathbf{x}_0|\mathcal{D}_n))$.

$|.|$ denotes the cardinality of a set and $k \in \mathbb{N}$. Additionally, we now define $F_k(\mathbf{x}_0)$ as the set of datapoints of $\mathcal{D}_n$ that have exactly k datapoints contained in the hyperrectangle that goes from each of them to $\mathbf{x}_0$. That is, $F_k(\mathbf{x}_0) = \{\mathbf{x}_i \in \mathcal{D}_n \text{ such that } |R(\mathbf{x}_0, \mathbf{x}_i)| = k\}$.

Figure 8.1 shows an example of the k-PNN points of a point $\mathbf{x}_0$ for two different values of k ($k = 1, 2$). Note that $|P_k(\mathbf{x}_0)|$ (we use $P_k(\mathbf{x}_0)$ instead of $P_k(\mathbf{x}_0|\mathcal{D}_n)$ when the context is clear) can be clearly more than k. For a more precise study of the cardinality of the k-PNNs, the number of the 1-PNNs to be expected for uniform and arbitrary finitely bounded densities in $\mathbb{R}^d$ have been studied in Lin and Jeon [2006] and Biau and Devroye [2010], respectively.

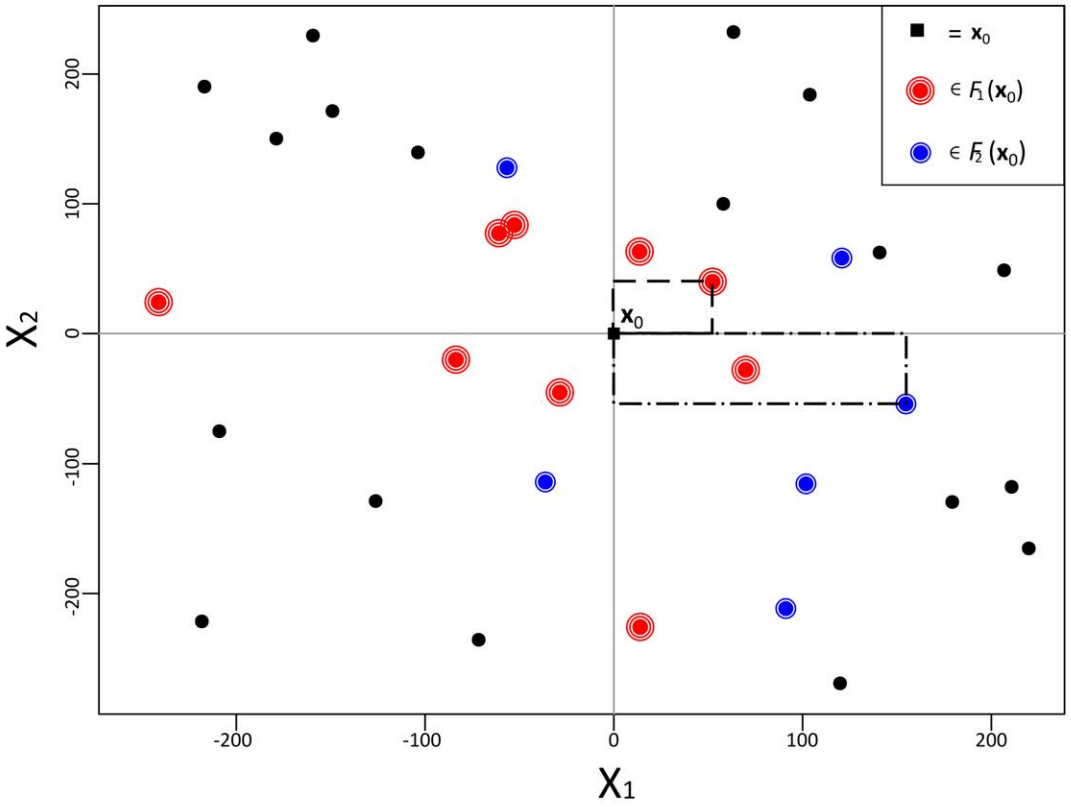


Figure 8.1: $\mathbf{X} = (X_1, X_2)$ feature space plot where we outline the datapoints in $F_1(\mathbf{x}_0)$ (red) and in $F_2(\mathbf{x}_0)$ (blue), with $\mathbf{x}_0 = (0, 0)$. The number of concentric circles around a datapoint represents the number of times the datapoint is selected as a k-PNN in the plot for $k = 1, 2$. Notice how the dashed rectangular area does not contain any other datapoint for the datapoints in $F_1(\mathbf{x}_0)$ and the dashed dot rectangular area contains just one for the datapoints in $F_2(\mathbf{x}_0)$

k-PNNs have a number of interesting properties:

k-PNNs correspond to a special case of nearest neighbors where the distance value is defined as the number of datapoints selected by all monotone distances (Lin and Jeon [2006]). A monotone distance satisfies the following property: Given datapoints $\mathbf{x}_a$ and $\mathbf{x}_b$ and the hyperrectangle defined by both as opposing vertices, any point $\mathbf{x}_c$ inside the hyperrectangle would be considered "closer" to $\mathbf{x}_a$ or to $\mathbf{x}_b$ than $\mathbf{x}_a$ to $\mathbf{x}_b$, (for example, all p-norm $||.||_p$ distances are monotone distances).

With this we can define the PNN distance between datapoints $\mathbf{x}_0$ and $\mathbf{x}_i$ as a function in $\mathbb{N}$ that outputs the number of datapoints inside the hyperrectangle defined by both as opposing vertices.

The particular case 1-PNN has received special attention and is commonly referred to as the layered nearest neighbors in the literature. It was initially proposed as an example of scale invariant metric in Devroye et al. [1996]. In Biau and Devroye [2010], Biau and Devroye showed that the layered nearest neighbors are closely related to the notions of maximum (Barndorff-Nielsen and Sobel [1966]) and dominance (Bai et al. [2005]) in high dimensional spaces. A point $\mathbf{x}_a$ dominates another $\mathbf{x}_b$ if $x_{ai} \geq x_{bi}$ for all $i = 1, \ldots, d$ and a point is a maximum if no point dominates it. The relationship between k-PNNs and dominance is the following: If we consider each quadrant separately and apply absolute value to the coordinates, 1-PNNs or layered nearest neighbors are precisely the points that do not dominate any other point. For arbitrary k, while not explicitly mentioned in Biau and Devroye [2010], the k-PNNs are the

points that dominate k or fewer points.

Finally, k-PNNs exhibit a property that links them directly to RFs that grow non-pruned trees stopping at leaves with k or less datapoints. Regression tree cuts at splitting points define hyperrectangular partitions of the feature space,and the number of possible partitions in a RF that include a point $\mathbf{x}_0$ with k or fewer datapoints, is finite and determined by the distribution of the datapoints. As proven in Lin and Jeon [2006] we have that, for a fixed dataset (i.e., without bootstrapping) the datapoints $\mathbf{x}_i$ that can be selected with $|R(\mathbf{x}_0, \mathbf{x}_i)| \leq k$, are the k-PNNs of $\mathbf{x}_0$ ($P_k(\mathbf{x}_0)$), that is, the voting points of a RF (as in Equation (8.2)).

8.3 Bagging and k-PNN

Biau et al. (Biau et al. [2010]) analyzed the regression estimate resulting from bagging the 1-NN regression estimator. It was shown that the bagged 1-NN takes the form of a weighted NN estimator where each point contributes to the regression estimate of $\mathbf{x}_0$, $\hat{f}^*_{1-NN}(\mathbf{x}_0)$, according to

$$\hat{f}^*_{1-NN}(\mathbf{x}_0) = \sum_{i=1}^{n} v_i(\mathbf{x}_0) y_i,$$

where all $\mathbf{x}_i$ datapoints are here sorted by increasing distance to $\mathbf{x}_0$ in the feature space, the $*$ symbol denotes a bagged estimator and $v_i(\mathbf{x}_0)$ is the probability that the i-th NN of $\mathbf{x}_0$, $\mathbf{x}_i$ in $\mathcal{D}_n$, is the closest neighbor in a bootstrapped dataset. The set of v_i's is in this case a decreasing sequence given by the expression

$$v_i(\mathbf{x}_0) = \left(1 - \frac{i-1}{n}\right)^n - \left(1 - \frac{i}{n}\right)^n. \tag{8.3}$$

We will refer to the set $V_{NN} = \{v_1(\mathbf{x}_0), \ldots, v_i(\mathbf{x}_0), \ldots, v_n(\mathbf{x}_0)\}$ as the bootstrap weights for the NN regression estimate.

Our interest now lies in understanding how bootstrap weights behave in a similar setting but using the set of k-PNN points instead of the k-NNs. We start by understanding that similarly to the previous case, each point must be weighted by an additional $v_i(\mathbf{x}_0)$ factor where $v_i(\mathbf{x}_0)$ is the probability that $\mathbf{x}_i$ is a k-PNN of $\mathbf{x}_0$ in a bootstrapped dataset ($\mathbf{x}_i$'s are not sorted here). Our bootstrap weights $v_i(\mathbf{x}_0)$ would appear in the bagged version of

$$f_{k-SA}(\mathbf{x}_0) = \sum_{\mathbf{x}_i \in P_k(\mathbf{x}_0)} y_i, \tag{8.4}$$

that is, $f^*_{k-SA}(\mathbf{x}_0)$.

We will refer to Equation (8.4) as "select all" point selection strategy, hence the SA subindex. Notice that Equation (8.4) is not an estimator of $\mathbb{E}[\mathcal{Y}|\mathcal{X} = \mathbf{x}]$. The normalized version, $\hat{f}_{k-PNN}(\mathbf{x}_0) = \frac{1}{|P_k(\mathbf{x}_0)|} \sum_{\mathbf{x}_i \in P_k(\mathbf{x}_0)} y_i$ is a regression estimate and is studied in Biau and Devroye [2010] for $k = 1$ as the layered NN estimate.

In order to calculate $\hat{f}^*_{k-SA}(\mathbf{x}_0)$, additional results and definitions are needed, with the final solution, for arbitrary k, provided in section 8.4. We now continue with the following lemma and the case $k = 1$:

Lemma 8.3.1. Let us define the set $Rm(\mathbf{x}_i) = R(\mathbf{x}_0, \mathbf{x}_i) \setminus \{\mathbf{x}_i\}$, where $\mathbf{x}_0$ and $\mathbf{x}_i$ are datapoints and $\mathbf{x}_0$ is our prediction target. Then $\mathbf{x}_i$ is a k-PNN of $\mathbf{x}_0$ for all bootstrap variations such that $|\mathcal{D}_j^* \cap Rm(\mathbf{x}_i)| \leq k-1$ where $\mathcal{D}_j^* \in B(\mathcal{D}_n)$ and $B(\mathcal{D}_n) = \{\mathcal{D}_1^*, \mathcal{D}_2^*, \dots \mathcal{D}_{n^n}^*\}$ is the set of all bootstrap variation selections of $\mathcal{D}_n$

Proof. See Appendix B.

Intuitively, Lemma 3.1 establishes for a datapoint $\mathbf{x}_i$ such that $|R(\mathbf{x}_0, \mathbf{x}_i)| = p$, $p \in \mathbb{N}$ to be a k-PNN with $p > k$, then $p - k$ points of $R(\mathbf{x}_0, \mathbf{x}_i)$ need not to appear in a considered bootstrap variation, or differently said, $\mathbf{x}_i$ is a k-PNN only in the fraction of the bootstrap variations that satisfy the requirement of Lemma 3.1 (for $p \leq k$, the only difference is that $\mathbf{x}_i$ is already a k-PNN in $\mathcal{D}_n$). Therefore, we only need to be concerned with the bootstrap variations that alter $P_k(\mathbf{x}_0)$.

8.3.1 The 1-PNN case

For the remainder of this section and for purposes of simplicity, we will analyze the case of bootstrap weights for 1-PNN. Notice that in this case we need $Rm(\mathbf{x}_i) = \varnothing$ for $\mathbf{x}_i$ to be a 1-PNN.

For purposes of explanation, let us consider a dataset plot (Figure 8.2) and analyze both the cases of using 1-NN and 1-PNN point selection strategies of $\hat{f}_{1-NN}(\mathbf{x}_0)$ (Biau et al. [2010]) and $f_{k-SA}(\mathbf{x}_0)$ (Equation (8.4)), respectively. Using 1-NN as our criterion and in a continuous feature space, we can arrange all datapoints in a ranking type hierarchy (from lowest to highest Euclidean distance to $\mathbf{x}_0$), where the point to be selected as 1-NN is always the highest ranked that appears in the bootstrapped variation. In other words, the i-th ranked point will be selected as the 1-NN in the bootstrap variations that do not include the first $i - 1$ ranked points.

For 1-PNNs, distances between points are discrete (PNN distance) and multiple point selections occur in the general case (that is, multiple 1-PNNs for a given $\mathbf{x}_0$ are expected). The result is a seemingly complex hierarchy where some points are linked to certain others by a "+1 PNN distance" relationship that determines the bootstrap requirements for a datapoint to be selected as a 1-PNN (Figure 8.2 and Figure 8.3).

We are now prepared for the following theorem:

Theorem 8.3.1. Let m be the minimum value of k for which all datapoints are m-PNN. Then f_{1-SA}^* can be written as

$$f_{1-SA}^*(\mathbf{x}_0) = \sum_{i=1}^{m} \left(\sum_{\mathbf{x}_j \in F_i(\mathbf{x}_0)} v_j(\mathbf{x}_0) y_j \right), \tag{8.5}$$

where the set of bootstrap weights V_{SA} are of the form:

$$v_j(\mathbf{x}_0) = \left(1 - \frac{|R(\mathbf{x}_0, \mathbf{x}_j)| - 1}{n}\right)^n - \left(1 - \frac{|R(\mathbf{x}_0, \mathbf{x}_j)|}{n}\right)^n$$

where $|R(\mathbf{x}_0, \mathbf{x}_j)| = i$.

Proof. See Appendix B.

Notice that as expected from multiple point selections, we do not necessarily have $\sum_{i=1}^{n} v_i(\mathbf{x}_0) = 1$, and in most of the cases $\sum_{i=1}^{n} v_i(\mathbf{x}_0) > 1$. Generalization of this theorem for $k > 1$ will be provided in Section 8.4.

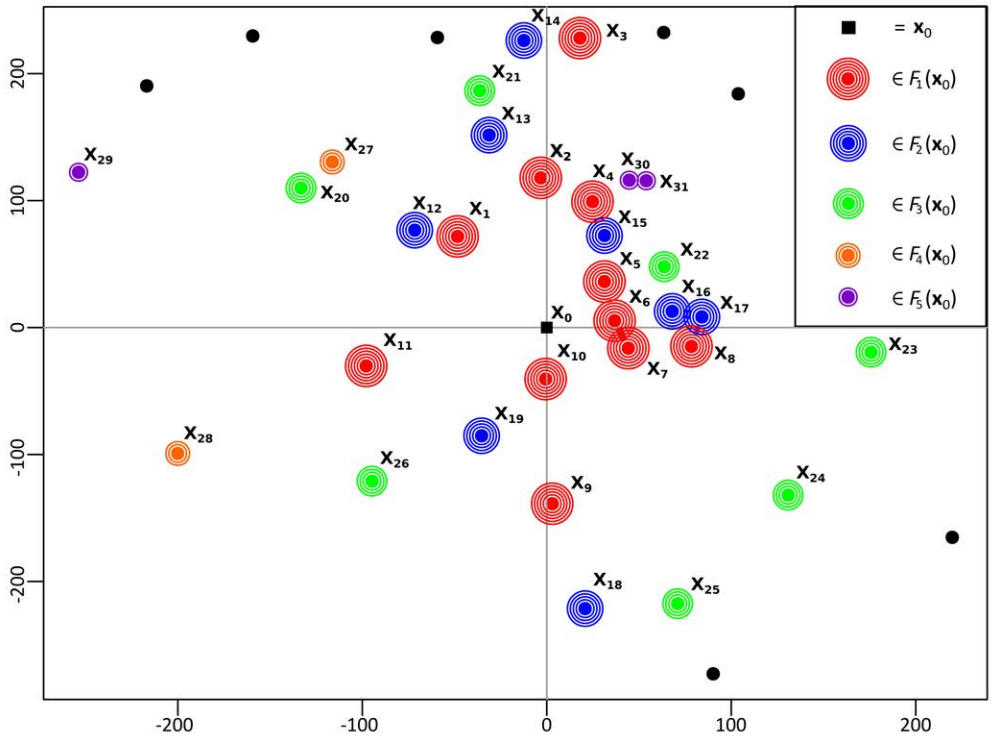


Figure 8.2: Feature space plot showing the outlining of $F_k(\mathbf{x}_0)$ of $\mathbf{x}_0 = (0, 0)$ for values of k from 1 to 5.

Our achievement here, described in terms of hierarchies for bootstrap variation requirements, is that it makes the bootstrap weights of 1-PNNs accessible for calculation as they are effectively expressed in Theorem 3.1 and by Lemma 3.1, in the same way as the resulting V_{NN} from bagging the 1-NN regression estimate. Its importance lies in that it describes the voting points (1-PNN) variations under bootstrapping, thus making it a useful tool for RF analysis.

8.4 Regression estimates as a weighted sum of k-PNNs

We are now interested in obtaining the final weights, that is, the set of weights W_{RF}, that accounts for both bootstrapping and splitting criterion in a RF algorithm, where $w_i(\mathbf{x}_0) \in W_{RF}$ is the probability that $\mathbf{x}_i$ is selected in a RF algorithm.

In Lin and Jeon [2006] it was shown that the splitting criteria can be viewed as weight redistributors for the obtained k-PNNs, corresponding to some particular solutions for the weights in Equation (8.2). Also, for a fixed dataset, the splitting criterion can be interpreted as a selector of k points from $P_k(\mathbf{x}_0)$.

We add the following result to the previous considerations by understanding the relationship between bootstrap weights and the final weights:

Lemma 8.4.1. Let $\hat{f}_{RF}(\mathbf{x}_0)$ be a RF regression estimate that uses bootstrapping and unpruned trees that stop at $k = 1$ datapoints in the leaves. Let $\mathbf{x}_i$ be a datapoint in $\mathcal{D}_n$ and $\mathbf{x}_0$ the datapoint to predict. Then

$$v_i(\mathbf{x}_0) = \left(1 - \frac{|R(\mathbf{x}_0, \mathbf{x}_i)| - 1}{n}\right)^n - \left(1 - \frac{|R(\mathbf{x}_0, \mathbf{x}_i)|}{n}\right)^n$$

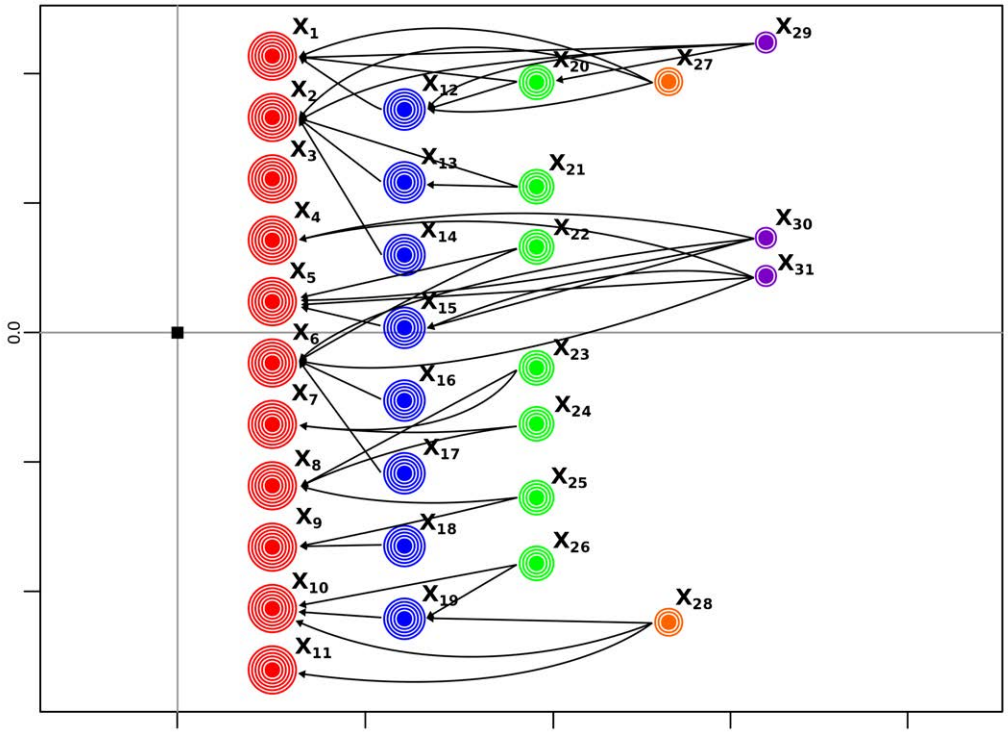


Figure 8.3: Graphical representation of the hierarchical precedence order for 1-PNN for the datapoints in Figure 8.2. In the plot, original points are arranged in a hierarchy that shows the precedence relationships for the 1-PNN. Similarly, we can see that the $k-1$ points connected by the arrows from another $\mathbf{x}_i$ are the points that need not to be included in a bootstrapped sample for $\mathbf{x}_i$ to be selected as a 1-PNN. For example, $\mathbf{x}_{29}$ will be selected as 1-PNN if and only if $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_{12}$ and $\mathbf{x}_{20}$ are not included in the bootstrapped sample.

is an upper bound of the final weight $w_i(\mathbf{x}_0)$ (that is, $v_i(\mathbf{x}_0) \geq w_i(\mathbf{x}_0)$).

Proof. See Appendix B.

Lemma 4.1 holds independently of the chosen split. Thus, it also holds for all RFs growing unpruned trees stopping when there is $k = 1$ datapoint in the leaves. Intuitively, we can think of the splitting criterion as a second point selection strategy applied after the selection of the k-PNNs for a given $\mathbf{x}_0$, which causes only up to k k-PNNs to be selected. In the general case, splitting criteria can be viewed as some form of weight shrinking procedure of bootstrap weights V_{SA} (that is, the bootstrap weights for the $f^*_{k-SA}(\mathbf{x}_0)$ regression estimate), that bounds the resulting $w_i(\mathbf{x}_0)$ to $\sum_{i=1}^n w_i(\mathbf{x}_0) = 1$. That is, a normalized regression estimator.

8.4.1 Analysis of point selection strategies using weighted b-terms

Now let us consider a regression estimate that applies a random selection over the k-PNNs, that is, k random k-PNN points are uniformly selected among the existing k-PNNs for each bootstrapped sample. We have

$$\hat{f}_{RkS}(\mathbf{x}_0) = \frac{1}{k} \sum_{\mathbf{x}_i \in P_k(\mathbf{x}_0)} \mathbb{1}_{[\mathbf{x}_i \in D(P_k(\mathbf{x}_0),k)]} y_i, \tag{8.6}$$

where $D(P_k(\mathbf{x}_0), k)$ is a set containing k uniform draws without replacement of datapoints from $P_k(\mathbf{x}_0)$. We call this criterion, random k-PNN selection (hence the RkS subindex on the regression estimate).

In a setting with a fixed dataset, the probability of selection for each k-PNN is equal to $\frac{1}{|P_k(\mathbf{x}_0)|}$. However, when considering bootstraping, different bootstrap variations have different number of k-PNNs. Classical analysis suggests we write the final weight of a point $\mathbf{x}_j$ under bootstrapping as

$$w_j(\mathbf{x}_0) = \frac{1}{n^n} \sum_{i=1}^{n^n} \mathbb{1}_{[\mathbf{x}_j \in P_k(\mathbf{x}_0|\mathcal{D}_i^*)]} \frac{1}{|P_k(\mathbf{x}_0|\mathcal{D}_i^*)|} \tag{8.7}$$

A first look at Equation (8.7) can be regarded as disappointing from a computational perspective, since it seems we are burdened with the need to calculate each individual $|P_k(\mathbf{x}_0|\mathcal{D}_i^*)|$ value.

Here we present an analysis framework that exploits the hierarchy of the PNNs and combinatorics regarding the bootstrap variations to arrive at a better calculation scenario. We introduce now the concept of b-term (bootstrap term).

Definition 8.4.1. A b-term $b_i = \mathbf{x}_a \dots \mathbf{x}_b \dots \neg\mathbf{x}_c \dots \neg\mathbf{x}_d$ written as a list of datapoints of $\mathcal{D}_n$, denotes the proportion of bootstrap variation selections of $\mathcal{D}_n$ that include the datapoints $\mathbf{x}_a, \dots, \mathbf{x}_b$ and do not include ($\neg$) datapoints $\mathbf{x}_c, \dots, \mathbf{x}_d$.

Also, we define $S(b_i)$ as a function that outputs the numerical value associated with a b-term b_i. To further understand b-terms and function $S(.)$, we present here some of their properties (proofs of these properties are not included for the sake of brevity):

1. ***Commutativity:*** Writing order is commutative. That is, $b_i = \mathbf{x}_a \dots \mathbf{x}_b \neg\mathbf{x}_c \dots \neg\mathbf{x}_d = \mathbf{x}_a \dots \neg\mathbf{x}_c \dots \neg\mathbf{x}_d \dots \mathbf{x}_b$.

2. ***Reduction by contradiction:*** For a b-term of the form $b_i = \mathbf{x}_a \dots \mathbf{x}_b \neg\mathbf{x}_a \dots \neg\mathbf{x}_d$ we have $S(b_i) = 0$. This can be interpreted as "no bootstrap variation selection can include and not include a point" ($\mathbf{x}_a$ in the example) .

3. ***Reduction by default:*** For an "empty" b-term b_i we have $S(b_i) = 1$. That is, without restrictions (empty b-term) all bootstrap variations are included.

4. ***Equivalence class:*** Let us define $E_{[l_p,l_m]}$ as the set of all possible b-terms in $\mathcal{D}_n$ that have $l_p \in \mathbb{N}$ included datapoints restrictions and $l_m \in \mathbb{N}$ non-included datapoints restrictions. Then for all $b_i, b_j \in E_{[l_p,l_m]}$ we have $S(b_i) = S(b_j)$.

5. ***Sum:*** We define the sum of two b-terms b_i, b_j as $b_i + b_j$ and $S(b_i + b_j) = S(b_i) + S(b_j)$.

6. ***Subtraction:*** Similarly, we define the subtraction of two b-terms b_i, b_j as $b_i - b_j$ and $S(b_i - b_j) = S(b_i) - S(b_j)$.

7. ***Concatenation:*** We define the concatenation of b-terms b_i, b_j as $b_t = b_i b_j$. That is, another b-term containing all b_i and b_j datapoint restrictions.

8. ***Concatenation of the sum:*** $(b_i + b_j)(b_a + b_b) = b_i b_a + b_i b_b + b_j b_a + b_j b_b$. That is, the concatenation of the sum works in the fashion of a classical product operation.

9. ***Reduction by sum:*** For the sum of b-terms, restrictions of different type over the same datapoint can be canceled. That is, $\mathbf{x}_a\mathbf{x}_d b_i + \neg\mathbf{x}_a\mathbf{x}_d b_i = \mathbf{x}_d b_i$; since $(\mathbf{x}_a + \neg\mathbf{x}_a)$ covers all possible cases for datapoint $\mathbf{x}_a$.

10. ***Reduction by subtraction:*** Similar rules apply for defining and using the subtraction of b-terms. That is, $\mathbf{x}_d b_i - \neg\mathbf{x}_a\mathbf{x}_d b_i = \mathbf{x}_a\mathbf{x}_d b_i$.

11. ***Reduction by redundancy:*** Redundancy is canceled in b-terms. That is, $b_j = \mathbf{x}_a\mathbf{x}_a b_i = \mathbf{x}_a b_i$.

12. ***Constant extraction:*** Constants multiplying b-terms can be computed outside the $S(.)$ function. That is, $S(Ab_i) = AS(b_i), A \in \mathbb{Z}$.

We can now use b-terms to write the bootstrap weights of Equation (8.3) (bagged 1-NN with datapoints sorted by increasing Euclidean distance) as

$$v_i(\mathbf{x}_0) = S(\neg\mathbf{x}_1\neg\mathbf{x}_2 \ldots \neg\mathbf{x}_{i-1}\mathbf{x}_i),$$

and accounting for the decomposability showed in property 9, and property 6, we can rewrite

$$v_i(\mathbf{x}_0) = S(\neg\mathbf{x}_1 \ldots \neg\mathbf{x}_{i-1}) - S(\neg\mathbf{x}_1 \ldots \neg\mathbf{x}_{i-1}\neg\mathbf{x}_i),$$

The following lemma can now be introduced:

Lemma 8.4.2. The numerical value of a b-term $b_i = \overbrace{\mathbf{x}_a \ldots \mathbf{x}_b}^{l_p}\overbrace{\neg\mathbf{x}_c \ldots \neg\mathbf{x}_d}^{l_m} \in E_{[l_p,l_m]}$ can be calculated as

$$S(b_i) = \sum_{i=0}^{l_p} \binom{l_p}{i} (-1)^i \left(1 - \frac{i + l_m}{n}\right)^n \tag{8.8}$$

Proof. See Appendix B.

It turns out that this notation allows us to express the final weights in a more accessible way than the direct computation of Equation (8.7). To illustrate this, let us break down Equation (8.7) into its pieces: We can see that $v_j(\mathbf{x}_0) = \frac{1}{n^n}\sum_{i=1}^{n^n} \mathbb{1}_{[\mathbf{x}_j \in P_k(\mathbf{x}_0|\mathcal{D}_i^*)]}$ (as these are all the cases where $\mathbf{x}_j$ is a k-PNN) and then see that $\frac{1}{|P_k(\mathbf{x}_0|\mathcal{D}_i^*)|}$ models the inclusion of the random k-PNN selection for each bootstrapping case. However, by Lemma 3.1, it is clear that this expression computes many unnecessary cases (as it iterates over all possible bootstrap variations without regard for changes in $P_k(\mathbf{x}_0|\mathcal{D}_i^*)$). Since b-terms cover subsets of the total bootstrap variation cases (those who satisfy the b-term), it is possible to cover the set of bootstrap variations for which $\mathbf{x}_j$ is a k-PNN using b-terms or sums of b-terms. For example, lets consider in isolation datapoints $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_{12}$ of Figure 2, with $\mathbf{x}_0$ as our prediction target and $k = 1$. Clearly, $\mathbf{x}_1, \mathbf{x}_2 \in F_1(\mathbf{x}_0)$ and $\mathbf{x}_{12} \in F_2(\mathbf{x}_0)$ since $\mathbf{x}_1$ is in the way. Then, we can simply write the bootstrap weights of $\mathbf{x}_{12}$ using b-terms as: $v_{12}(\mathbf{x}_0) = \neg\mathbf{x}_1\mathbf{x}_{12}$. Now, we need to account for the inclusion of the random 1-PNN selection using the b-terms notation. It turns out that by the property of reduction by sum, it is possible to expand a b-term into the different cases where the random 1-PNN selection takes different values for selecting a given datapoint. Continuing with our example, its possible to do $\neg\mathbf{x}_1\mathbf{x}_{12} = \neg\mathbf{x}_1\mathbf{x}_{12}\mathbf{x}_2 + \neg\mathbf{x}_1\mathbf{x}_{12}\neg\mathbf{x}_2$ which effectively accounts for the cases where $\mathbf{x}_2$ is present and absent. Then, the final weight of $\mathbf{x}_{12}$ can be expressed as: $w_{12}(\mathbf{x}_0) = \frac{1}{2}S(\neg\mathbf{x}_1\mathbf{x}_{12}\mathbf{x}_2) + S(\neg\mathbf{x}_1\mathbf{x}_{12}\neg\mathbf{x}_2)$. This shows how weighted sums of b-terms can be used to express the final weights W_{R1S}.

Formally, we define for $b_i \in E_{[l_p,l_m]}$:

$$P_{R1S}(\mathbf{x}_i, b_i) = L_{R1S}(\mathbf{x}_i, b_i)S(b_i) \tag{8.9}$$

where

$$L_{R1S}(\mathbf{x}_i, b_i) = \frac{1}{l_p}.$$

and for which following property is verified:

$$P_{R1S}(\mathbf{x}_i, b_i + b_j) = P_{R1S}(\mathbf{x}_i, b_i) + P_{R1S}(\mathbf{x}_i, b_j).$$

Using the function $P_{R1S}(\mathbf{x}_i, b_i)$, it is possible to pair each b-term in isolation or in a sum of b-terms with the weight obtained by $L_{R1S}(\mathbf{x}_i, b_i)$. Thus, final weights W_{R1S} of any datapoint can be expressed using this function as long as the necessary b-terms are known.

Notice that in here, it was possible to define the random 1-PNN selector as a function $L_{R1S}(.,.)$ requiring only the local information of the b-term to output its corresponding value.

The $S(b_i)$ function, on the other hand, accounted for the k-PNN hierarchy and the selectability and bootstrap variations all together. In this sense, b-terms can be looked at as bootstrap variations themselves and hence our target is to find all relevant bootstrap variations for the calculation of $w_i(\mathbf{x}_0)$.

8.4.2 Random k-PNN selection regression estimate

Here we solve the problem of calculating the expression of the bagged version of the regression estimate in Equation (8.6), that is $\hat{f}^*_{RkS}(\mathbf{x}_0)$, and the explicit form of its final weights W_{RkS}. Considering our work so far, all that remains open is to find regularized way to write the expression that we obtain by expanding the b-terms of $v_i(\mathbf{x}_0)$ using the reduction by sum property until all datapoints are considered. For this, we add to the previous work on b-terms the following definition:

Definition 8.4.2. We define the restricted concatenation operator

$$[\mathbf{x}_a\mathbf{x}_b \ldots \neg\mathbf{x}_c\neg\mathbf{x}_d, \ldots, \mathbf{x}_f\mathbf{x}_g \ldots \neg\mathbf{x}_h\neg\mathbf{x}_i]$$
$$(\mathbf{x}_j\mathbf{x}_k \ldots \neg\mathbf{x}_m\neg\mathbf{x}_n \ldots)$$

as a special type of concatenation operator which specifies in brackets $[.,\ldots,.]$ to which other b-terms the expression $(\mathbf{x}_j\mathbf{x}_k \ldots \neg\mathbf{x}_m\neg\mathbf{x}_n \ldots)$ is concatenated.

For Definition 4.2, let us consider the example

$$(\mathbf{x}_1 + \neg\mathbf{x}_1)(\mathbf{x}_2 + \neg\mathbf{x}_2)([\mathbf{x}_1\neg\mathbf{x}_2, \neg\mathbf{x}_1\neg\mathbf{x}_2](\mathbf{x}_3 + \neg\mathbf{x}_3)).$$

This results in:

$$\mathbf{x}_1\mathbf{x}_2 + \mathbf{x}_1\neg\mathbf{x}_2(\mathbf{x}_3 + \neg\mathbf{x}_3) + \neg\mathbf{x}_1\mathbf{x}_2 + \neg\mathbf{x}_1\neg\mathbf{x}_2(\mathbf{x}_3 + \neg\mathbf{x}_3),$$

where only the b-terms $\mathbf{x}_1\neg\mathbf{x}_2$ and $\neg\mathbf{x}_1\neg\mathbf{x}_2$ are concatenated with $(\mathbf{x}_3 + \neg\mathbf{x}_3)$.

Then, the following theorem holds:

Theorem 8.4.1. Let us consider a datapoint $\mathbf{x}_0$ as our prediction target. Weights W_{R1S} for the $\hat{f}^*_{R1S}(x_0)$

regression estimate have the form

$$w_i(\mathbf{x}_0) = P_{R1S}(\mathbf{x}_i, z_i(\mathbf{x}_0))$$

where

$$\begin{aligned} &z_i(\mathbf{x}_0) \\ &= (\mathbf{x}_i)(Req_1(\mathbf{x}_i)) \prod_{\mathbf{x}_j \in Ind(\mathbf{x}_i)} [Req_1(\mathbf{x}_j)](\mathbf{x}_j + \neg\mathbf{x}_j), \end{aligned}$$

$Req_1(\mathbf{x}_i) = \{\neg\mathbf{x}_a \neg\mathbf{x}_b \ldots \neg\mathbf{x}_s\}$ with $Rm(\mathbf{x}_i) = \{\mathbf{x}_a, \mathbf{x}_b \ldots, \mathbf{x}_s\}$ and $Ind(\mathbf{x}_i) = \mathcal{D}_n \setminus R(\mathbf{x}_0, \mathbf{x}_i)$ as the complementary set of points of $R(\mathbf{x}_0, \mathbf{x}_i)$ w.r.t. the data $\mathcal{D}_n$.

Proof. See Appendix B (Proof of Theorem 4.2).

The expression of $z_i(\mathbf{x}_0)$ shows "a sum expansion of $v_i(\mathbf{x}_0)$, where each added datapoint is restricted to be concatenated to the b-terms that can be expressed as $b_t Req_1(.)$ (that is, the b-terms that contain their $Req_1(.)$ set)". This guarantees that we only consider the inclusion or non inclusion of the datapoint in the subset of cases where it is relevant for the final weight calculation, while ignored otherwise.

Our goal is now to generalize the previous results for arbitrary k. With the b-terms notation, this turned out to be a natural step forward. We start with the introduction of the following lemma, that generalizes Lemma 4.1 for arbitrary k.

Lemma 8.4.3. Let $\hat{f}^*_{RF}(\mathbf{x}_0)$ be a RF regression estimate that uses bootstrapping, unpruned trees and stops at arbitrary $k \in \mathbb{N}$ datapoints in the leaves. Let $\mathbf{x}_0$ be our prediction target and $\mathbf{x}_i$ another datapoint. Then

$$v_i(\mathbf{x}_0) = S\left(\mathbf{x}_i \left(\sum_{c \in Req_k(\mathbf{x}_i)} r_{ic}\right)\right)$$

is an upper bound of $w_i(\mathbf{x}_0)$ (that is, $v_i(\mathbf{x}_0) \geq w_i(\mathbf{x}_0)$), where $Req_k(\mathbf{x}_i) = \{r_{i1}, r_{i2}, \ldots, r_{ih}\}$, $h \in \mathbb{N}$, is defined as the set of b-terms listing all possible bootstrap variations where a subset of the datapoints in $Rm(\mathbf{x}_i)$ allows for $\mathbf{x}_i$ to be selectable as a k-PNN.

Proof. See Appendix B (proof of Theorem 4.2).

For Theorem 4.1, we first notice that Equation (8.9) does not need to change to account for the b-terms weights in the $k > 1$ case, since for a b-term $b_i \in E_{[l_p, l_m]}$ the corresponding weight in a random k-PNN selection would be $L_{RkS}(\mathbf{x}_i, b_i) = \left(\frac{1}{k}\right)\left(\frac{k}{l_p}\right) = \frac{1}{l_p}$. We can then define

$$P_{RkS} = P_{R1S}$$

and write the following theorem:

Theorem 8.4.2. Let us consider a datapoint $\mathbf{x}_0$ as our prediction target. Weights W_{RkS} for the $\hat{f}^*_{RkS}(x_0)$ regression estimate with arbitrary k have the form

$$w_i(\mathbf{x}_0) = P_{RkS}(\mathbf{x}_i, z_i(\mathbf{x}_0))$$

where

$$z_i(\mathbf{x}_0) = (\mathbf{x}_i)\left(\sum_{c\in Req_k(\mathbf{x}_i)} r_{ic}\right)\prod_{\mathbf{x}_j\in Ind(\mathbf{x}_i)}[Req_k(\mathbf{x}_j)](\mathbf{x}_j + \neg\mathbf{x}_j)$$

Proof. See Appendix B.

We finally have:

Theorem 8.4.3. Let us consider a datapoint $\mathbf{x}_0$ as our prediction target. The $\hat{f}^*_{RkS}(\mathbf{x}_0)$ regression estimate has the form

$$\hat{f}^*_{RkS}(\mathbf{x}_0) = \sum_{i=1}^{n} w_i(\mathbf{x}_0)y_i$$

where $w_i(\mathbf{x}_0)$'s are regarded as in the form of Theorem 4.2.

Proof. See Appendix B.

By proving Theorem 4.3 we have succeeded in our original objective of finding a more direct and accessible approach to compute the weights that in Equation (8.7). Also, with this theorem we have solved an open problem in Biau and Devroye [2010], since for the random k-PNN selection, the case $k = 1$ corresponds to the final weights W of the bagged layered NN regression estimate detailed in that paper, that is,

$$\hat{f}^*_{R1S}(\mathbf{x}_0) = \hat{f}^*_{1-PNN}(\mathbf{x}_0).$$

For the general case (since $L_{RkS} = L_{R1S}$), we also have

$$\hat{f}^*_{RkS}(\mathbf{x}_0) = \hat{f}^*_{k-PNN}(\mathbf{x}_0).$$

Finally and as a completing remark, we can now express the generalized bootstrap weights for f^*_{k-SA} as:

Theorem 8.4.4. Let m be the minimum value of k for which all datapoints are m-PNN. The bagged version of f_{k-SA} is of the form

$$f^*_{k-SA}(\mathbf{x}_0) = \sum_{i=1}^{m}\left(\sum_{\mathbf{x}_j\in F_i(\mathbf{x}_0)} v_j(\mathbf{x}_0)y_j\right),$$

where $v_j(\mathbf{x}_0)$'s are regarded as in the form of Lemma 4.3.

Proof. See Appendix B (Proof of Theorem 4.2).

8.4.3 Bagged estimators framework

From Equation (8.9), it is not difficult to imagine that other regression estimates may adjust to this model with a different $P_{\cdot}(\mathbf{x}_i, b_i)$ function. The $P_{\cdot}(\mathbf{x}_i, b_i)$ function general form is, for an estimator $\hat{f}_{\cdot}(\mathbf{x}_0)$ in

the calculation of the weight of $\mathbf{x}_i$:

$$P_{\cdot}(\mathbf{x}_i, b_i) = L_{\cdot}(\mathbf{x}_i, b_i)S(b_i). \tag{8.10}$$

We have seen how the weight calculation obtained in Theorem 4.2 accounts for all the bootstrap cases of interest within the k-PNNs, thus, the different ways in which we can specify the $L_{\cdot}(\mathbf{x}_i, b_i)$ function correspond to the different regression estimates. As an example, let us define

$$L_{1-NN}(\mathbf{x}_i, b_i) = \begin{cases} 1 & \text{if } b_i = \neg\mathbf{x}_1\neg\mathbf{x}_2 \ldots \neg\mathbf{x}_{i-1}\mathbf{x}_i b_j \\ \\ 0 & \text{otherwise} \end{cases}$$

where here, datapoints are sorted by increasing Euclidean distance, ($\mathbf{x}_1$ being the closest to $\mathbf{x}_0$ and $\mathbf{x}_n$ the furthest away) as the $L_{\cdot}(.,.)$ function that produces a bagged NN estimate for $k = 1$.

We will show now how we can use, in this case, the $P_{1-NN}(\mathbf{x}_i, b_i)$ function, to deduce the weights of Equation (8.3). We use the simple dataset of Figure 8.4.

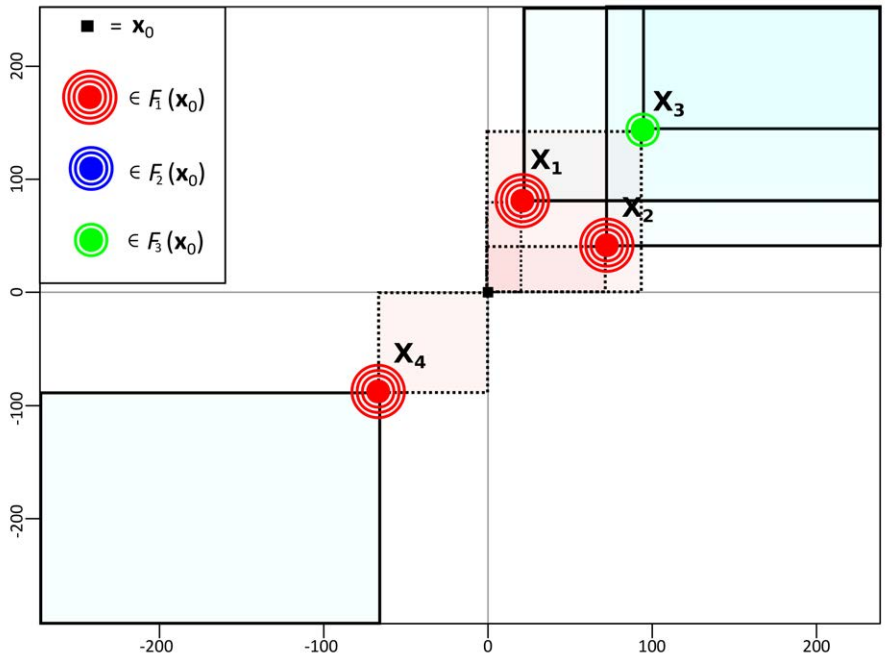


Figure 8.4: Feature space plot showing four datapoints and $F_k(\mathbf{x}_0)$ of $\mathbf{x}_0 = (0, 0)$ for values of k from 1 to 3.

For the dataset of Figure 8.4 and $k = 1$ we can write the b-terms sum expansion per datapoint as

follows:

$$\begin{aligned}
z_1(\mathbf{x}_0) =& \mathbf{x}_1\mathbf{x}_2\mathbf{x}_4 + \mathbf{x}_1\neg\mathbf{x}_2\mathbf{x}_4 + \mathbf{x}_1\mathbf{x}_2\neg\mathbf{x}_4 + \mathbf{x}_1\neg\mathbf{x}_2\neg\mathbf{x}_4 \\
z_2(\mathbf{x}_0) =& \mathbf{x}_1\mathbf{x}_2\mathbf{x}_4 + \neg\mathbf{x}_1\mathbf{x}_2\mathbf{x}_4 + \mathbf{x}_1\mathbf{x}_2\neg\mathbf{x}_4 + \neg\mathbf{x}_1\mathbf{x}_2\neg\mathbf{x}_4 \\
z_3(\mathbf{x}_0) =& \neg\mathbf{x}_1\neg\mathbf{x}_2\mathbf{x}_3\mathbf{x}_4 + \neg\mathbf{x}_1\neg\mathbf{x}_2\mathbf{x}_3\neg\mathbf{x}_4 \\
z_4(\mathbf{x}_0) =& \mathbf{x}_1\mathbf{x}_2\mathbf{x}_4 + \mathbf{x}_1\neg\mathbf{x}_2\mathbf{x}_4 + \neg\mathbf{x}_1\mathbf{x}_2\mathbf{x}_4 \\
& + \neg\mathbf{x}_1\neg\mathbf{x}_2\mathbf{x}_3\mathbf{x}_4 + \neg\mathbf{x}_1\neg\mathbf{x}_2\neg\mathbf{x}_3\mathbf{x}_4
\end{aligned}$$

where here, the b-term sums were obtained using $z_i(\mathbf{x}_0)$ of Theorem 4.1. For this example, we have the Euclidean distance sorting of the datapoints as $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_3$ (thus, the $L_{1-NN}(\mathbf{x}_0)$ function will compute them as $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$, respectively). If we now apply $P_{1-NN}(\mathbf{x}_i, b_i)$ we obtain:

$$\begin{aligned}
w_1(\mathbf{x}_0) &= S(\mathbf{x}_1\mathbf{x}_2\mathbf{x}_4 + \mathbf{x}_1\neg\mathbf{x}_2\mathbf{x}_4 + \mathbf{x}_1\mathbf{x}_2\neg\mathbf{x}_4 \\
&\quad + \mathbf{x}_1\neg\mathbf{x}_2\neg\mathbf{x}_4) \\
w_2(\mathbf{x}_0) &= S(\neg\mathbf{x}_1\mathbf{x}_2\mathbf{x}_4 + \neg\mathbf{x}_1\mathbf{x}_2\neg\mathbf{x}_4) \\
w_3(\mathbf{x}_0) &= S(\neg\mathbf{x}_1\neg\mathbf{x}_2\mathbf{x}_3\neg\mathbf{x}_4) \\
w_4(\mathbf{x}_0) &= S(\neg\mathbf{x}_1\neg\mathbf{x}_2\mathbf{x}_3\mathbf{x}_4 + \neg\mathbf{x}_1\neg\mathbf{x}_2\neg\mathbf{x}_3\mathbf{x}_4)
\end{aligned}$$

Now using the properties of the b-terms we can do:

$$\begin{aligned}
w_1(\mathbf{x}_0) &= S(\mathbf{x}_1\mathbf{x}_2\mathbf{x}_4 + \mathbf{x}_1\neg\mathbf{x}_2\mathbf{x}_4 + \mathbf{x}_1\mathbf{x}_2\neg\mathbf{x}_4 \\
&\quad + \mathbf{x}_1\neg\mathbf{x}_2\neg\mathbf{x}_4) \\
&= S(\mathbf{x}_1\mathbf{x}_4 + \mathbf{x}_1\neg\mathbf{x}_4) \\
&= S(\mathbf{x}_1) \\
w_2(\mathbf{x}_0) &= S(\neg\mathbf{x}_1\mathbf{x}_2\mathbf{x}_4 + \neg\mathbf{x}_1\mathbf{x}_2\neg\mathbf{x}_4) \\
&= S(\neg\mathbf{x}_1\mathbf{x}_2) \\
w_4(\mathbf{x}_0) &= S(\neg\mathbf{x}_1\neg\mathbf{x}_2\mathbf{x}_3\mathbf{x}_4 + \neg\mathbf{x}_1\neg\mathbf{x}_2\neg\mathbf{x}_3\mathbf{x}_4) \\
&= S(\neg\mathbf{x}_1\neg\mathbf{x}_2\mathbf{x}_4)
\end{aligned}$$

which effectively yields the weights of the bagged 1-NN as were known in Biau et al. [2010]. Using our framework and operations on the b-terms, we were able to deduce the form of the weights of the bagged estimator and reduce it to its known-form, requiring only one b-term per datapoint.

We argue here that any point selection strategy within the set of the k-PNNs can be adapted to this format, and by applying the b-terms properties to the result of the $P_{\cdot}(.,.)$ function, we can observe how the interplay between the b-terms and the $L_{\cdot}(.,.)$ function gives rise to the bagged version of many known regression estimates (for example, for all regression estimates using p-norm $\|.\|_p$ distances as point selectors, their bagged versions can be easily derived in a similar way than with the Euclidean distance in the 1-NN example). This includes the well-known predictive square error splitting criterion, considering that for an adaptive splitting criterion, decisions are made using an additional set of values Y.

8.4.4 Random forest with random split regression estimate

Now, we are looking to attain one of the main goals of this work: inducing the L_{RF} function that would allow us to build a regression estimate that outputs similar predictions to those obtained by the RF algorithm with random split. Using this framework, we reformulated the problem of inducing the RF estimator by traditional means to that of finding an expression for $p(\mathbf{x}_i|b_i)$ for any b_i in $\mathcal{D}_n$.

We found a recursive procedure that allows us to calculate $L_{RF}(\mathbf{x}_i, b_i)$, with $b_i \in E_{[l_p,l_m]}$ and for arbitrary k as:

$$L_{RF}(\mathbf{x}_i, b_i) = \begin{cases} G_{RF}(\mathbf{x}_i, b_i) \text{ if } b_i \text{ includes } \mathbf{x}_i, b_i \in E_{[l_p,l_m]} \text{ and } l_p > k \\ \frac{1}{l_p} \text{ if } b_i \text{ includes } \mathbf{x}_i, b_i \in E_{[l_p,l_m]} \text{ and } l_p \leq k \\ 0 \text{ otherwise,} \end{cases}$$

where

$$G_{RF}(\mathbf{x}_i, b_i) = \frac{1}{d}\sum_{l=1}^{d}\left(\sum_{k=1}^{l_p}\frac{I(k+1,l,b_i,\mathcal{D}_n) - I(k,l,b_i,\mathcal{D}_n)}{I_s(l,b_i,\mathcal{D}_n)} L_{RF}(\mathbf{x}_i, C(\mathbf{x}_i,k,l,b_i,\mathcal{D}_n))\right).$$

$I(k,l,b_i,\mathcal{D}_n)$ outputs the value in the l-th feature/column of the k-th datapoint in a sorted sample (from lowest to highest values) of the datapoints listed in $\mathbf{x}_0 b_i$ that appear in $\mathcal{D}_n \cup \mathbf{x}_0$. $I_s(l,b_i,\mathcal{D}_n)$ outputs the range of values of the l-th feature/column for the datapoints listed in $\mathbf{x}_0 b_i$ that appear in $\mathcal{D}_n \cup \mathbf{x}_0$. $C(\mathbf{x}_i,k,l,b_i,\mathcal{D}_n)$ outputs a b-term b_s that contains a subset of the listed datapoints of b_i. b-term b_s is defined as follows: Let us consider the sorted sample of $\mathcal{D}_n$ datapoints listed in b_i by their values in the l-th feature/column. We can then divide b_i into two b-terms b_{s1} and b_{s2} by splitting the sorted sample at the k-th position. Then we can define b_{s1} to be the b-term that lists the datapoints that in the sorted sample appeared before the k-th position, and b_{s2} containing the rest so that $b_{s1}b_{s2} = b_i$. Finally, we define b_s as $b_s = b_{s1}$ if the interval covered by the l-th feature values of the datapoints listed in b_{s1} includes the l-th feature value of $\mathbf{x}_0$. If it doesn't, we define it as $b_s = b_{s2}$.

Intuitively, $L_{RF}(\mathbf{x}_i, b_i)$, accounts for all possible cases that the classical RF algorithm with random split may produce. For $G_{RF}(\mathbf{x}_i, b_i)$, modeling the random subspace method implies d possible choices of coordinate. Each choice weighted by $\frac{1}{d}$ (Notice that in this type of RF, this is always the case regardless of how we tune this parameter). Then, for $b_i \in E_{[l_p,l_m]}$, l_p splits are possible ($l_p - 1$ provided by the l_p datapoints listed in b_i and 1 provided by $\mathbf{x}_0$) per coordinate. Each split is weighted by its probability of occurrence on the selected coordinate. After choosing the split, two mutually exclusive subsets of datapoints are created. Then, the one not containing $\mathbf{x}_0$ is discarded, while the other is selected. Repeating this process recursively for the selected subset of datapoints as a new b_i produces $G_{RF}(\mathbf{x}_i, b_i)$. The second case of $L_{RF}(\mathbf{x}_i, b_i)$ accounts for the stopping criteria, which can be plugged directly, and the third case accounts for the b-terms that do not contribute to the final weight of $\mathbf{x}_i$.

With this, we are ready to present the following theorem:

Theorem 8.4.5. Let us consider a datapoint $\mathbf{x}_0$ as our prediction target. The $\hat{f}_{RF}(\mathbf{x}_0)$ regression estimate

has the form

$$\hat{f}_{RF}(\mathbf{x}_0) = \sum_{i=1}^{n} w_i(\mathbf{x}_0) y_i$$

where $w_i(\mathbf{x}_0)$'s have the form

$$w_i(\mathbf{x}_0) = P_{RF}(\mathbf{x}_i, z_i(\mathbf{x}_0))$$

with

$$P_{RF}(\mathbf{x}_i, z_i(\mathbf{x}_0)) = L_{RF}(\mathbf{x}_i, z_i(\mathbf{x}_0)) S(z_i(\mathbf{x}_0))$$

and $z_i(\mathbf{x}_0)$ is regarded as in the form of Theorem 4.2.

Proof of this result is considered trivial after the proofs of Theorem 4.2 and Theorem 4.3.

This regression estimate, as we will show in Section 8.5, offers similar predictions to those of a RF in all cases and problems.

8.5 Towards practical implementation and random forest equivalence

After Section 8.4, we are provided with the means to rewrite bagged estimators that select points in the k-PNN set as sums of weighted b-terms. Additionally, we have succeeded in finding the explicit expression for these weights in the cases of a RF with random splitting criterion and our proposed $\hat{f}_{RkS}(\mathbf{x}_0)$ regression estimate. In this section, we seek to validate our findings at a practical level.

We started by implementing $\hat{f}_{RkS}(\mathbf{x}_0)$ with an algorithm that closely follows Theorem 4.3. In order to do so, we first noticed that function $S(.)$ as shown in Equation (8.8) displays a clear exponential growth with respect to l_p (in the binomial coefficient) and n (in the denominator inside the summation, as it is n^n) in computational complexity. In order to alleviate the complexity of both variables we use an easy workaround as any expression of the type $e(i) = \left(1 + \frac{i}{n}\right)^n$ satisfies $\lim_{n\to\infty} e(i) = e^i$. We can take advantage of this simply by approximating Equation (8.8) with

$$S(b_i) \approx \sum_{i=0}^{l_p} \binom{l_p}{i} (-1)^i e^{-i-l_m}, \tag{8.11}$$

offering an approximate result ($\sum_{i=1}^{n} w_i(\mathbf{x}_0) \leq 1$) that improves its accuracy the higher n becomes. We now do

$$\begin{aligned} S(b_i) &\approx \frac{1}{e^{l_m}} \sum_{i=0}^{l_p} \binom{l_p}{i} (-1)^i e^{-i} \\ &= \frac{1}{e^{l_m}} \sum_{i=0}^{l_p} \binom{l_p}{i} \left(\frac{-1}{e}\right)^i \\ &= \frac{1}{e^{l_m}} \left(\frac{e-1}{e}\right)^{l_p} \end{aligned}$$

which shows much clearly the relationship between bootstrapping and the b-terms. Also, this implies

that final weights W are, when $n \to \infty$, a sum of weighted exponential functions.

As for the number of b-terms to be computed, we can see that the b-terms expansion grows exponentially in the worst case scenario: Examining $z_i(\mathbf{x}_0)$ in Theorem 4.2, the number of b-terms doubles at each iteration of the product in the subset of b-terms allowed by the $[Req_k(.)]$ set of the datapoint to be computed. This is still an improvement with respect to classical analysis and Equation (8.7) for the complete computation of weights (from $\mathcal{O}(n^n)$ to $\mathcal{O}(2^n)$) and in some cases, as we have seen in section 8.4.1, the choice of the $L_.(.,.)$ can reduce final complexity to a sub-exponential form. For our testing, however, we limited ourselves to small sample sizes and k = 1.

In setting up our experimental environment, we use two datasets from UCI data repository (`Bike` and `Concrete`, with 5 and 6 variables, respectively) and six datasets from the R package mlbench (`Ozone`, `Boston Housing`, `Friedman1` with sd = 1, `Friedman2` with sd =125 and `Friedman3` with sd = 0.1, with 12,14,11,5 and 5 variables, respectively). For each dataset, we normalized the response values Y subtracting the mean and dividing by the standard deviation in order to control the scale of MSE values. Additionally, we implemented a full random RF that for each tree at each node to split, selects a random feature and performs a random split between its maximum and minimum values until there is a single datapoint in the leaves (k = 1).

We computed the MSE statistics between the true values and predictions given by the models to assess their performance. We additionally computed other statistics for analysis purposes. The results of the experiments are shown in Table 8.1.

Table 8.1: Comparisons of the results of $\hat{f}^{\star}_{R1S}$ and $\hat{f}_{RF}$ estimates for the selected datasets. The first two rows contain mean square error comparisons between real values and predicted values of both estimators, the range (±) is simply the standard deviation of the predictions, the third is the average of the average PNN distance that each testing point obtained w.r.t. the rest of the points in the dataset and the fourth is the ratio between the dimension of the dataset d and the sample size n.

	MSE - $\hat{f}^{\star}_{R1S}$	MSE - $\hat{f}_{RF}$	$\overline{\overline{PNN}}$	$\frac{d}{n}$
Bike	3.89±0.02	4.23±0.08	1.46	0.25
Concrete	1.18±0.00	2.56±0.60	0.00	0.30
Ozone	1.88±0.00	1.66±0.08	0.00	0.60
Boston	1.27±0.00	1.00±0.41	0.00	0.70
Friedman1	1.00±0.00	0.84±0.21	0.00	0.55
Friedman2	0.91±0.11	0.38±0.53	0.22	0.25
Friedman3	0.86± 0.14	0.41±0.54	0.16	0.25

In Table 8.1, most remarkable results come from analyzing the included statistics. In $\hat{f}^{\star}_{R1S}$ evaluations, four of the seven datasets present no variability between predictions, independently of the test point. In those sets, the average PNN distance is 0, result that can only occur if all datapoints in all cases were 1-PNNs of each datapoint used in testing. This is expected, as $\hat{f}^{\star}_{R1S}$ only distinguishes between predictions by differences in the k-PNN distances. Additionally, this case seems to occur at the highest levels of the $\frac{d}{n}$ ratio.

It is not difficult to notice that increasing dimensionality (d) while maintaining sample size could reduce the average PNN distance in the general case. To see this clearly, lets imagine a set of datapoints distributed in the perimeter of a circle around the testing datapoint $\mathbf{x}_0$ in a 2-dimensional setting (Figure 8.5, case B). It can be verified that all datapoints in this setting exhibit a PNN distance of 0 w.r.t. $\mathbf{x}_0$. If

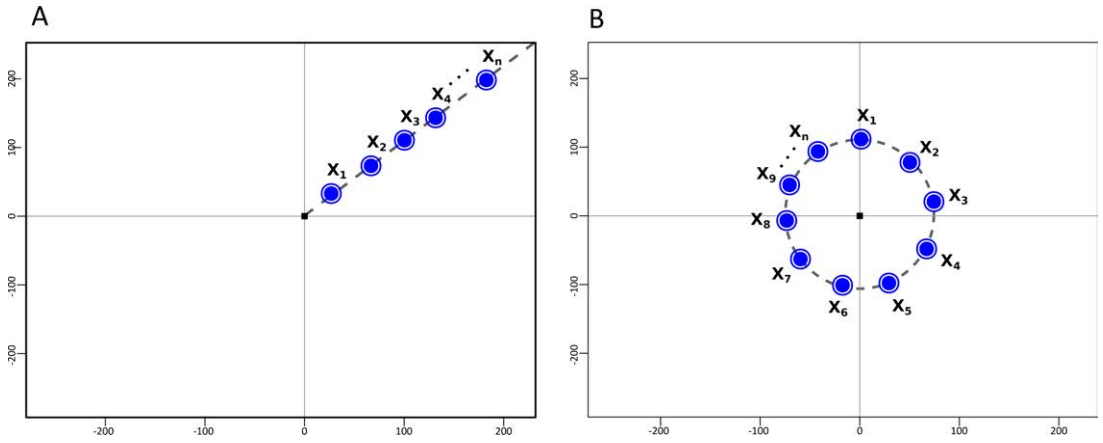


Figure 8.5: Two different cases showing maximum k-pnn distance arrangement and minimum k-pnn distance arrangement between a set of datapoints and the point to predict

we now project all datapoints onto a line by eliminating one of the dimensions, the result is that at most only the two immediate neighbors of $\mathbf{x}_0$ have PNN distance of 0.

To illustrate this practically, we removed all but two features on the previous datasets and repeated the computations, see Table 8.2.

Table 8.2: Comparisons of the results of $\hat{f}^{\star}_{R1S}$ and $\hat{f}_{RF}$ estimates for the selected datasets and the reduced number of variables.

	MSE - $\hat{f}^{\star}_{R1S}$	MSE - $\hat{f}_{RF}$	$\overline{PNN}$	$\frac{d}{n}$
Bike	4.40 ±0.09	4.58±0.14	8.14	0.15
Concrete	1.17±0.10	3.11±0.88	0.52	0.15
Ozone	1.82±0.05	1.86±0.13	2.12	0.15
Boston	1.22±0.39	1.33±0.51	3.77	0.15
Friedman1	0.72±0.28	0.71±0.49	2.06	0.15
Friedman2	0.73±0.35	0.82±0.59	2.11	0.15
Friedman3	0.70±0.42	0.73±0.70	1.67	0.15

For Table 8.2, MSE results lie in favor of $\hat{f}^{\star}_{R1S}$ for most cases, decreasing with respect to Table 8.1, in opposition with the tendency shown by MSE results of $\hat{f}_{RF}$. Variability in $\hat{f}^{\star}_{R1S}$, as expected, has increased yet remains substantially inferior to that of $\hat{f}_{RF}$. This seems to indicate that the reason for $\hat{f}^{\star}_{R1S}$ to perform better is that the effects of bootstrapping have a higher influence on the outcome of the estimator when the average PNN distance increases.

Further understanding allows for the following characterization: consider point arrangement cases of Figure 8.5. Computing the b-terms of any weight with any splitting criteria that requires to select $k = 1$ datapoints will output the bootstrap weights (case limit of equality in Lemma 4.1) for case A, as for any given setting, the splitting criterion always selects 1 out of 1 datapoints. Thus, the influence of the employed $L_{\cdot}(\mathbf{x}_i, b_i)S(b_i)$ function degenerates to its form of minimum variance, $L_{\cdot}(\mathbf{x}_i, b_i) = 1$. For case B, the b-terms sum expansion is the same for every datapoint, and differences in weights can be attributed exclusively to the $L_{\cdot}(\mathbf{x}_i, b_i)$ function variability.

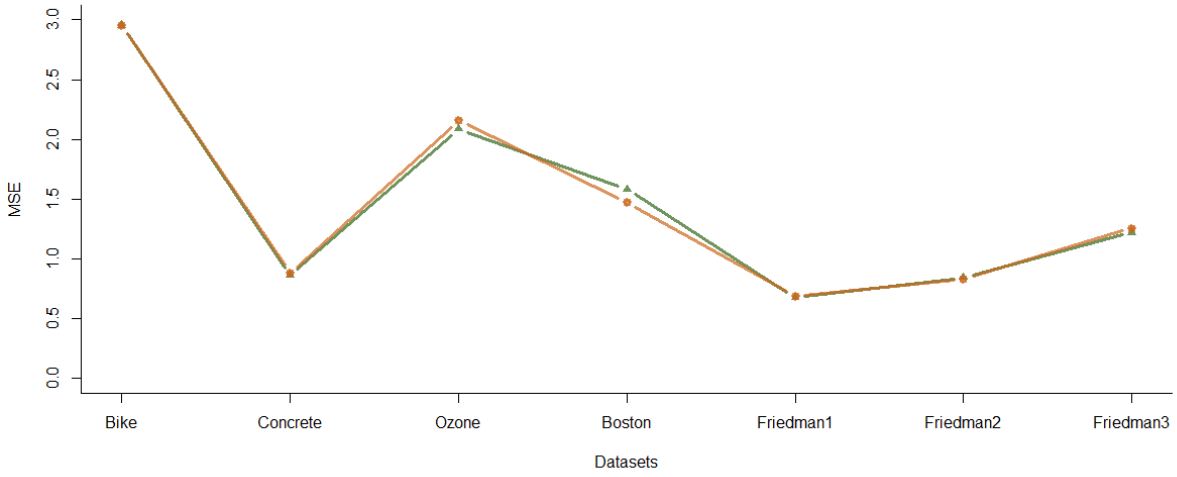


Figure 8.6: MSE comparisons between our regression estimate equipped with the L_{RF} (green triangles) and the RF method $\hat{f}_{RF}$ (orange circles) across the seven selected datasets.

Thus, we can safely argue that for any given problem, each prediction will fall between cases A and B, and thus the average PNN distance could be a good indicator of the overall contribution of each part of the estimator to the final outcome (and partially explain the results of Tables 8.1 and 8.2). Also this implies that the differences between bagged $\|.\|_p$ norm estimators (always case A) are governed exclusively by the differences in order in the ranking of datapoints. Our analysis seems to concur with literature (Karoui and Purdom [2016]) in the use of bootstrapping in high $\frac{d}{n}$ ratio (case B) problems as having a mild to poor effect on the overall quality of the predictions.

Finally, we repeated the experiments in Table 8.2 substituting L_{R1S} for L_{RF} and $k = 1$, to show that we can achieve a functional practical implementation of a RF using sums of weighted b-terms (Figure 8.6).

In Figure 8.6, it is clearly noticeable that virtually identical results were achieved for all datasets, where the minimal divergences in MSE can be safely attributed to a finite number of trees (less than all its possible variations) used to train $\hat{f}_{RF}$, together with the approximation of the bootstrap weights values shown in Equation (8.11).

With this we have shown that our results can produce models that are equivalent to traditional versions of RFs through an alternative path, without computing a single tree and effectively opening a new way of analyzing regression estimates that conform to the proposed framework. While we believe that the ideal use of b-terms and Equation (8.10) is analytical, on a practical sense we believe to have uncovered a way for new classes of algorithms to arise, perhaps taking advantage of heuristics to overcome the exponential expansion of b-terms while making affordable compromises in MSE values.

8.6 Summary and conclusions

In this work we have shown advances in our understanding of the statistical forces behind RFs, by means of their analogy with the k-PNNs. We first discovered that the developments to obtain the bagged 1-

NN regression estimate in Biau et al. [2010] could be extended to show the calculation of the bootstrap weights for k-PNN based regression estimates when substituting the corresponding monotone distance for its alternative PNN distance.

Then, we analyzed the influence of adding a point selection strategy to the previous results. A point selection strategy, such as any splitting criterion in a RF, turned out to act as an additional selector of k datapoints within the k-PNNs, causing some of k-PNNs to be finally selected, and some others to be not. Thus, the weights assigned to each datapoint had to be updated from bootstrap weights to the final weights. We first proved that the bootstrap weights act as upper bounds of the final weights for a RF equipped with any splitting criterion. Followingly, we obtained an explicit expression for the final weights considering a specific point selection strategy in the k-PNN, the random k-PNN selection, which induces the bagged regression estimate $\hat{f}^{\star}_{RkS}(\mathbf{x}_0)$, and showed that $\hat{f}^{\star}_{RkS}(\mathbf{x}_0) = \hat{f}^{\star}_{k-PNN}(\mathbf{x}_0)$. In doing so, we created the concept of b-terms as a list of inclusion/non-inclusion restrictions on the datapoints present in all bootstrap variations, defined the value of a b-term to be the proportion of bootstrap variations that comply with the list (and derived a mathematical expression to calculate that value). We then showed that datapoint weights can be expressed as sums of b-terms coupled with a local weight on each b-term.

Further understanding uncovered a framework for bagging estimators that included all classes of RF with a k datapoints stopping criterion. With this, we derived the regression estimate that corresponds with a RF equipped with random splitting criterion and showed the case of $k = 1$ at a practical level, where MSE values of our regression estimate and a full RF implemented in the classical way w.r.t. the real values were virtually identical. We were also able to conduct additional practical experiments that revealed how b-terms and k-PNN distance can be used to analyze the effect of bootstrapping in bagged regression estimators in contrast with the effect of the point selection strategy (splitting criterion in RF). With this, we validated the $\hat{f}^{\star}_{RkS}(\mathbf{x}_0)$ for $k = 1$ as a competent regression estimate. Our results suggest that it is recommended for problems with high scale disparity between features (since PNN are distance invariant) and high $\overline{PNN}$. Additionally, it is fast to implement (k-PNN calculation for a given datapoint with arbitrary k was $\mathcal{O}(n^2)$ in our methods) and intuitive to work with. It may also be a considerable choice over 1-NN (or bagged 1-NN) when the nearest neighbors assumptions (namely, that datapoints close in distance have also close y-associated values) do not hold.

We believe that the ideal use of our work would be as an analysis tool for other regression estimates and as a design platform for variants of random forests. In this setting, a researcher may follow a similar path to the one shown in this chapter to write a regression estimate as a weighted sum of datapoints were the weights are expressed as weighted sums of b-terms. Then, analysis on the particular form of that expression may allow for simplifications previously inaccessible, detection of grouping patterns/regularities in the addends of the sum and in general, to enjoy a higher degree of algebraic manipulation than in the initial proposal for that regression estimate.

This work opens numerous possibilities for regression estimates to have an alternative written form (as weighted sums of b-terms) that has desirable properties. Perhaps reductions in computational complexity for the calculation of well known methods, that have remained hidden so far, can now be unlocked, and Monte Carlo simulation as the preferred method for computation of those regression estimators can be eschewed.

As a future work, we believe that we have obtained the necessary tools to tackle the specification of other splitting criteria in RFs in terms of weighted PNNs, as well as provided access to a new form of

analysis of RF models and other regression estimates.

The present work has been published as Fernandez-Gonzalez, P., C. Bielza, and P. Larrañaga, "Random forests for regression as a weighted sum of k-potential nearest neighbors", *IEEE Access*, vol. 7, issue 1, pp. 25660-25672, 2019.

Part IV

CONCLUSIONS

Chapter 9

Conclusions and future work

9.1 Summary of contributions

1. In Chapter 5 we develop the theory for the univariate and bivariate cases for the truncated von Mises distribution for the modeling and simulation of angular phenomena in a restricted interval. The properties, maximum likelihood estimators and moments of the distribution are calculated for the univariate case, whereas maximum likelihood estimators, the conditional and the marginal distributions are studied for the bivariate case. The mentioned theoretical developments aim to establish the distribution as ready for practitioners to use with real data. Subsequently, we perform simulated studies in order to test the distribution and real data studies with leaf inclination angles and dihedral angles in protein chains. It is concluded that the distribution performs correctly in generalizing the von Mises distribution and models properly the data, which allows us to say it can be considered as a valuable option when directional statistics are needed for a scientific problem.

2. In Chapter 6 we put to use our truncated von Mises distribution developments to model branching angles of basal dendrites of pyramidal neurons in the human temporal cortex (layers III and V). We complement the study with population similarity comparison studies, where we form different subgroups of the total population of neurons and observe the statistical differences that emerge from those subdivisions. The performed studies are: study of branching angles by branch order, study of branching angles by branch order and maximum branch order, pairs of angles of contiguous orders comparison, layer IIIPost and layer VPost neurons comparisons, layer IIIPost and layer IIIAnt neurons comparisons, human and rat layer III neuron comparisons and different human neuron comparisons. For the first two studies, truncated von Mises distribution models are estimated from the data and their parameters used to draw statistical conclusions. Additionally, goodness-of-fit tests are employed to assess the difference with respect to the original von Mises distribution. Conclusions of the study reveal that the truncated von Mises distribution performs excellently in modeling the branching angle data, clearly outerperforming the von Mises distribution without truncation. The conducted study is able to produce meaningful insights into the principles that govern the branching angles patterns in the human brain.

3. In Chapter 7 we introduce a structural learning algorithm for multidimensional Gaussian Bayesian network classifiers. The algorithm makes use of the CB-decomposable property to break the MPE

classification problem into sub-problems with an overall gain in computational power. With this, we devise a strategy for incremental addition of complexity in the construction of a topologically unrestricted class subgraph that exploits the previous property to minimize the computational burden. The resultant network presents a class subgraph that can be examined for knowledge discovery and hypothesis generation, and it is not bounded by common topological restrictions in multidimensional Bayesian classifier literature.

4. In Chapter 8 we present a novel way to analyze the problem of RF for regression expressed as weighted sums of datapoints. We use the concept of k-PNNs in random forests and analyze their behavior under bootstrapping. We derive from this an upper bound on all splitting criterion-induced weights on the datapoints. Moreover, we use the previous bound together with the concept of b-terms (i.e., bootstrap terms) introduced in this work to create a framework from where we can derive a certain class of bagged regression estimators, including RFs, as weighted sums of datapoints. Finally, we make use of our obtained framework to produce a model that is equivalent to the RF regression estimate with random splitting criterion, obtaining also an explicit expression for writing the prediction as a weighted sum of datapoints. We show this equivalence both theoretically and practically for $k = 1$.

We believe that the compilation of these works is able to answer all hypotheses and fulfill all objectives detailed in Chapter 1. Hypotheses 1. and 2. and Objectives 1. and 2. are addressed in the works that employ the truncated von Mises distribution, that is, Chapters 5 and 6. We also show the benefits of using the CB-decomposable property and Gaussian feature nodes, detailed in Hypothesis 3., to build a competently performing multidimensional classifier, as specified in Objective 3. Finally, we have developed the theory and methodology to write a random forest regression estimate, and other regression estimates as well, as weighted sums of datapoints, fulfilling Objective 4. and answering affirmatively Hypothesis 4.

9.2 List of publications

The contents of this dissertation have been gathered in the following publications:

Q1 JCR journals

1. Fernandez-Gonzalez, P., C. Bielza, and P. Larrañaga, "Random forests for regression as a weighted sum of k-potential nearest neighbors", *IEEE Access*, vol. 7, issue 1, pp. 25660-25672, 2019.

2. Fernandez-Gonzalez, P., R. Benavides-Piccione, I. Leguey, C. Bielza, P. Larrañaga, and J. DeFelipe, "Dendritic branching angles of pyramidal neurons of the human cerebral cortex", *Brain Structure and Function*, vol. 222, issue 4, pp. 1847-1859, 2017.

Conference papers

1. Fernandez-Gonzalez, P., C. Bielza, and P. Larrañaga, "Multidimensional classifiers for neuroanatomical data", *ICML Workshop on Statistics, Machine Learning and Neuroscience (Stamlins 2015)*, pp. 0-6, 2015.

2. Fernandez-Gonzalez, P., P. Larrañaga, and C. Bielza, "Bayesian Gaussian networks for multidimensional classification of morphologically characterized neurons in the NeuroMorpho repository", In *Actas de la 17a Conferencia de la Asociación Española para la Inteligencia Artificial*, pp. 39-48, 2016

Non-JCR journals

1. Fernandez-Gonzalez, P., C. Bielza, and P. Larrañaga, "Univariate and bivariate truncated von Mises distributions", *Progress in Artificial Intelligence*, pp. 1-10, 2017.

9.3 Future work

In Chapter 5, we covered the univariate and bivariate cases of the truncated von Mises distribution. However, generalization to an arbitrary number of dimensions was not attempted. Additionally, further mathematical manipulation might improve our results on the marginal truncated von Mises distribution.

In Chapter 6, we could consider the attempt of other subgroupings of the data if more data and hypothesis are available. For example, we could include data from more species than rat and mice and produce more comparative studies. In a more technical vein, we could introduce other measures of independence (non-linear independence, unlike those derived from Gaussianity) that perhaps allow us to see the suspected dependency of lower branch orders on the branching angle of their parent order branching angles.

In Chapter 7, we observed in posterior efforts, when applying our model to different problems, that for difficult enough problems there may be an initialization problem with the wrapper approach using solely global accuracy as our metric: If a global accuracy higher than zero can only be achieved for models with at least a certain set of arcs, the procedure may never progress towards more complicated structures since it cannot distinguish benefiting candidates. For this reason, we have experimentally tried, but not yet published, a switch between Hamming score and global accuracy where the first would guide the network building process at early stages and the second would take over once the network satisfies the minimum required complexity. Additionally, while the computational cost of building our model is still lower than without the use of the CB-decomposable property, not limiting the maximum treewidth of our components would pose a problem when scaling to bigger problems. Thus, we consider attempting to find a good tradeoff between the loss of topological complexity and the gain in training complexity per component a valuable research direction. Finally, a mixed network able to handle both discrete and continuous feature nodes would further increase the flexibility of this model and broaden its range of application.

The work in Chapter 8 has many possible extensions and future work. Analysis of RFs in its original algorithm is of great interest and now can be adapted to a weighted sum of k-PNNs. We are currently enjoying a partially successful research direction, with written forms of this estimator already available but still immature for publication. As an analysis tool, the task of writing other regression estimates as weighted sums of datapoints and then observe emerging properties or regularities also hold the potential for improvements in those regression estimates. Another direction of research concerns the calculation of b-terms themselves, depending on what input is required for the local weighting of a b-term, further simplifications are also possible. For example, in the $f_{RkS}(.)$ regression estimate, we are only required to

know the l_p and l_m values of a b-term, making it possible to compute together groups of b-terms that belong to the same equivalence class, with the corresponding savings in computational time. Our research has also covered this topic and currently holds some results also immature for publication. In general, extending the practical appeal of the b-terms analysis may prove to be very rewarding; an algorithm that is able to alleviate the worst case scenario complexity of b-terms calculation may potentially redefine the preferred calculation method for some regression estimates, even when reduction of the b-terms sum expansion to a polynomial form was not attained.

Part V

APPENDICES

Appendix A

Univariate and bivariate truncated von Mises distributions

Proof of Lemma 5.2.1. We have, by means of the power series expansion of the $e^{(\cdot)}$ function,

$$I(\theta;\mu,\kappa) = \int f_{uvM}(\theta;\mu,\kappa)d\theta = \int e^{\kappa\cos(\theta-\mu)}d\theta = \int \sum_{n=0}^{\infty} \frac{(\kappa\cos(\theta-\mu))^n}{n!}d\theta,$$

where $f_{uvM}(\theta;\mu,\kappa)$ is the unnormalized von Mises distribution, and $I(\theta;\mu,\kappa)$ is its distribution function. Therefore, $\int_a^b f_{uvM}(\theta;\mu,\kappa) = I(b;\mu,\kappa) - I(a;\mu,\kappa)$.

Considering that $\sum_{n=0}^{\infty} \frac{|(\kappa\cos(\theta-\mu))^n|}{n!}$ is a solely positive continuous bounded function in $[1, e^{\kappa}]$, and, therefore, for any finite integral coefficients $i_1, i_2 \in \mathbb{R}$, it satisfies $\int_{i_1}^{i_2} \sum_{n=0}^{\infty} \frac{(\kappa\cos(\theta-\mu))^n}{n!}d\theta < \infty$, we can conclude that it satisfies the Fubini-Tonelli theorem conditions for integral summation exchange.

We then follow with the procedure for the indefinite integral:

$$\begin{aligned} I(\theta;\mu,\kappa) &= \int \sum_{n=0}^{\infty} \frac{(\kappa\cos(\theta-\mu))^n}{n!}d\theta \\ &= \sum_{n=0}^{\infty} \int \frac{(\kappa\cos(\theta-\mu))^n}{n!}d\theta \\ &= \sum_{n=0}^{\infty} \frac{\kappa^n}{n!} \int \cos^n(\theta-\mu)d\theta. \end{aligned} \tag{A.1}$$

The above integral is defined in a recursive way as

$$\int \cos^n(\theta-\mu)d\theta = \frac{\sin(\theta-\mu)\cos^{n-1}(\theta-\mu)}{n} + \frac{n-1}{n}\int \cos^{n-2}(\theta-\mu)d\theta.$$

And it can be calculated by the procedure of integration by parts. In this appendix, however, we give a non-recursive expression:

$$\int \cos^n(\theta-\mu)d\theta = \sin(\theta-\mu)\left(\sum_{i=0}^{\lfloor\frac{n}{2}\rfloor+\bmod\frac{n}{2}-1}\left(\cos^{n-2i-1}(\theta-\mu)\frac{\prod_{j=0}^{2i}(n-j)}{\prod_{j=0}^{i}(n-2j)^2}\right)\right) \quad \forall n \text{ such that } n = 2m+1$$

with $m \in \mathbb{N}$. This materializes out of the observation of the numerical regularities that appear when "unfolding" the recursive expression:

$$\begin{aligned}
\int \cos^n(\theta-\mu)d\theta &= \frac{\sin(\theta-\mu)\cos^{n-1}(\theta-\mu)}{n} + \frac{n-1}{n}\int \cos^{n-2}(\theta-\mu)d\theta \\
&= \frac{\sin(\theta-\mu)\cos^{n-1}(\theta-\mu)}{n} + \\
&\quad \frac{n-1}{n}\left(\frac{\sin(\theta-\mu)\cos^{n-3}(\theta-\mu)}{n-2} + \frac{n-3}{n-2}\int \cos^{n-4}(\theta-\mu)d\theta\right) \\
&= \frac{1}{n}\sin(\theta-\mu)\cos^{n-1}(\theta-\mu) + \frac{n-1}{n(n-2)}\sin(\theta-\mu)\cos^{n-3}(\theta-\mu) + \\
&\quad \frac{(n-1)(n-3)}{n(n-2)(n-4)}\sin(\theta-\mu)\cos^{n-5}(\theta-\mu) + \\
&\quad \frac{(n-1)(n-3)(n-5)}{n(n-2)(n-4)}\int \cos^{n-6}(\theta-\mu)d\theta
\end{aligned}$$

They can be primary generalized using the expression

$$\sin(\theta-\mu)\left(\sum_{i=0}^{\lfloor \frac{n}{2} \rfloor + \bmod \frac{n}{2}-1}\left(\cos^{n-2i-1}(\theta-\mu)\frac{\prod_{j=0}^{2i}(n-j)}{\prod_{j=0}^{i}(n-2j)^2}\right)\right)$$

However, while this first expression does suffice for odd n, an extra term appears if n is even as we reach the point at which the term $\int \cos^0(\theta-\mu)d\theta$ is computed. This can be reflected properly by adding an addend that takes into account the parity of the formula. In our case, it has the form:

$$g(n,x) = \frac{(-1)^n h(x) + h(x)}{2} = \frac{((-1)^n+1)h(x)}{2},$$

where $\forall n \in \mathbb{N}$ such that $n = 2m$ and $m \in \mathbb{N}, g(n,x) = h(x)$ and 0 otherwise.

In a shorter notation and adding the parity term, the expression becomes

$$\begin{aligned}
\int \cos^n(\theta-\mu)d\theta = \sin(\theta-\mu)\left(\sum_{i=0}^{\lfloor \frac{n}{2} \rfloor + \bmod \frac{n}{2}-1}\left(\cos^{n-2i-1}(\theta-\mu)\prod_{j=0}^{2i}(n-j)^{-(-1)^j}\right) + \right. \\
\left. \frac{((-1)^n+1)\prod_{j=0}^{\lfloor \frac{n}{2} \rfloor + \bmod \frac{n}{2}-1}(n-j)^{-(-1)^j}(\theta-\mu)}{2}\right).
\end{aligned}$$

Thus, substituting in Equation (A.1) we obtain the final expression for $\int e^{\kappa\cos(\theta-\mu)}d\theta$.

□

Proof of Theorem 5.3.1. The theorem is entirely derived by means of the trigonometrical equality:

$$\begin{aligned}&\kappa_2\cos(x)+c_2\sin(x)\\&=\left[\kappa_2\cos\left(\arctan\left(\frac{c_2}{\kappa_2}\right)\right)+c_2\sin\left(\arctan\left(\frac{c_2}{\kappa_2}\right)\right)\right]\cos\left(x-\arctan\left(\frac{c_2}{\kappa_2}\right)\right).\end{aligned} \tag{A.2}$$

From the equality we can express the exponent of the conditional distribution in (Equation 5.11) using a formula of the type $\kappa'\cos(x-\mu')$. Now if we consider that

$$\kappa_2\cos\left(\arctan\left(\frac{c_2}{\kappa_2}\right)\right)+c_2\sin\left(\arctan\left(\frac{c_2}{\kappa_2}\right)\right)=\frac{\kappa_2+\frac{c_2^2}{\kappa_2}}{\sqrt{1+\left(\frac{c_2}{\kappa_2}\right)^2}}=\sqrt{\kappa_2^2+c_2^2},$$

then Equation (A.2) becomes

$$\kappa_2\cos(x)+c_2\sin(x)=\sqrt{\kappa_2^2+c_2^2}\cos\left(x-\arctan\left(\frac{c_2}{\kappa_2}\right)\right). \tag{A.3}$$

Thus, we can adapt the truncated conditional distribution to the univariate truncated von Mises exponent by properly selecting:

$$\begin{aligned}\kappa' &= \sqrt{\kappa_2^2+c_2^2}\\ \mu' &= \mu_2+\arctan\left(\frac{c_2}{\kappa_2}\right),\end{aligned}$$

where $c_2=\lambda\sin(\theta_1-\mu_1)$.

□

Proof of Theorem 5.3.2. We consider

$$f_{umtvM}(\theta_{1'})=e^{\kappa_1\cos(\theta_{1'})}\int_{a_2}^{b_2}e^{\kappa_2\cos(\theta_2-\mu_2)+\lambda\sin(\theta_{1'})\sin(\theta_2-\mu_2)}d\theta_2 \tag{A.4}$$

to be the unnormalized marginal truncated von Mises distribution. For simplicity's sake, the proof is developed in a linear context (using classical intervals [x,y], with their associated constraints, instead of circular intervals $\mathbb{O}_{x,y}$), whose extension to the circle is deemed as known and trivial at this point. Also, unless otherwise specified, $\lambda>0$ is assumed and a_2, b_2 truncation parameters are referred to simply as the truncation parameters. The proof is as follows:

(a) Determination of the derivative expression and the $T(\cdot,\cdot,\cdot,\cdot,\cdot,\cdot)$ function

(b) Analysis of the marginal expression with focus on the case of symmetrical truncation parameters in order to prove cases 1 and 2

(c) Further analysis for the case of non-symmetrical truncation parameters, determining all distinctive behaviors of the integral subterm of the marginal expression

(d) Monotony study divided by cases of the circular distance of the truncation parameters w.r.t. μ_2 and subintervals of the $\theta_{1'} \in [-\pi, \pi]$ interval in order to prove case 3. Case 4 is proven by ruling out every other possible outcome.

In (a), $T(\cdot,\cdot,\cdot,\cdot,\cdot,\cdot)$ is derived from a particularization of the second derivative of the marginal function. The meaning of the value of the $T(\cdot,\cdot,\cdot,\cdot,\cdot,\cdot)$ function is clarified for the symmetrical truncation parameters. In (b) and (c), the analysis aims to characterize the behavior of the integral term of the marginal distribution. In (b), the analysis will first observe the particularities of the integral term, especially, how $\theta_{1'}$ modifies the location and concentration parameters of the von Mises distribution inside the integral, and then derive from it some properties and insights will also be used for the proof of case 3. We then prove how thsese variations affect the area under the curve and their relationships to the truncation parameters. Finally, partial and total analyses of the derivate of the integral term are performed, concluding the proof of the first two cases of the theorem. In (c), an analysis of the derivate of the integral term for non-symmetrical truncation parameters w.r.t. μ_2 is performed. Using the previous insights, the analysis first determines the cases where, according to the truncation parameter values, the marginal integral term follows a unimodal distribution. The analysis then focuses on the remaining cases in order to prove that the global maximum of the integral term necessarily appears at the associated point of the truncation parameter ($-\frac{\pi}{2}$ for a_2 and $\frac{\pi}{2}$ for b_2), which has the largest circular distance w.r.t. μ_2. Also, in the bi-modal case for non-symmetrical truncation parameters, we analyze how the minimum comprehended between the modes appears in the $\frac{\pi}{2}$–length interval with 0 as an extrema associated with the truncation parameter that has the smallest circular distance w.r.t. μ_2 ($[-\frac{\pi}{2}, 0]$ for a_2 and $[0, \frac{\pi}{2}]$ for b_2), and its relationship with the minimum that appears in $[-\pi, -\frac{\pi}{2}]$ for the associated interval $[-\frac{\pi}{2}, 0]$ or in $[\frac{\pi}{2}, \pi]$ for the associated interval $[0, \frac{\pi}{2}]$. In (d), the monotony study identifies all different behaviors and the subinterval in which more than one critical point can occur, thus enabling us to detect bi-modality with different valued maxima with the proposed criteria.

(a) By differentiating $f_{umtvM}(\theta_{1'})$ w.r.t. θ_1 we obtain:

$$\begin{aligned} f'_{umtvM}(\theta_{1'}) &= -\kappa_1 \sin(\theta_{1'}) e^{\kappa_1 \cos(\theta_{1'})} \int_{a_2}^{b_2} e^{\kappa_2 \cos(\theta_2-\mu_2)+\lambda \sin(\theta_{1'}) \sin(\theta_2-\mu_2)} d\theta_2 \\ &\quad +\lambda \cos(\theta_{1'}) e^{\kappa_1 \cos(\theta_{1'})} \int_{a_2}^{b_2} \sin(\theta_2-\mu_2) e^{\kappa_2 \cos(\theta_2-\mu_2)+\lambda \sin(\theta_{1'}) \sin(\theta_2-\mu_2)} d\theta_2 \\ &= e^{\kappa_1 \cos(\theta_{1'})} \left(-\kappa_1 \sin(\theta_{1'}) \int_{a_2}^{b_2} e^{\kappa_2 \cos(\theta_2-\mu_2)+\lambda \sin(\theta_{1'}) \sin(\theta_2-\mu_2)} d\theta_2 \right. \\ &\quad \left. +\lambda \cos(\theta_{1'}) \int_{a_2}^{b_2} \sin(\theta_2-\mu_2) e^{\kappa_2 \cos(\theta_2-\mu_2)+\lambda \sin(\theta_{1'}) \sin(\theta_2-\mu_2)} d\theta_2 \right). \end{aligned} \tag{A.5}$$

We observe that

$$\begin{aligned} f'_{umtvM}(0) &= \lambda e^{\kappa_1} \left(\int_{a_2}^{b_2} \sin(\theta_2-\mu_2) e^{\kappa_2 \cos(\theta_2-\mu_2)} d\theta_2 \right) \\ &= \frac{\lambda}{\kappa_2} e^{\kappa_1} \left(e^{\kappa_2 \cos(a_2-\mu_2)} - e^{\kappa_2 \cos(b_2-\mu_2)} \right). \end{aligned} \tag{A.6}$$

If and only if $\cos(b_2-\mu_2) = \cos(a_2-\mu_2)$, it follows that $f_{umtvM}(\theta_{1'})$ has a critical point at μ_1.

Solving and assessing the equation $f''_{umtvM}(\theta_{1'}) = 0$ in order to obtain information about the curvature for $\theta_{1'} = 0$ results in

$$-\frac{\kappa_1}{\lambda^2}+\frac{\int_{a_2}^{b_2}\sin^2(\theta_2-\mu_2)e^{\kappa_2\cos(\theta_2-\mu_2)}d\theta_2}{\int_{a_2}^{b_2}e^{\kappa_2\cos(\theta_2-\mu_2)}d\theta_2}=0,$$

from which we can define the $T(\cdot,\cdot,\cdot,\cdot,\cdot,\cdot)$ function as

$$T(\lambda,\mu_2,\kappa_1,\kappa_2,a_2,b_2)=-\frac{\kappa_1}{\lambda^2}+\frac{\int_{a_2}^{b_2}\sin^2(\theta_2-\mu_2)e^{\kappa_2\cos(\theta_2-\mu_2)}d\theta_2}{\int_{a_2}^{b_2}e^{\kappa_2\cos(\theta_2-\mu_2)}d\theta_2}. \tag{A.7}$$

However, we still need to understand whether Equation (A.7) is sufficient to distinguish between cases 1 and 2 established in the theorem.

(b) In order to understand the truncated marginal behavior, if we rewrite the integral term in $f_{umtvM}(\theta_{1'})$ by means of Equation (A.3) we have

$$f_{umtvM}(\theta_{1'})=e^{\kappa_1\cos(\theta_{1'})}\int_{a_2}^{b_2}e^{\sqrt{\kappa_2^2+(\lambda\sin(\theta_{1'}))^2}\cos\left(x_2-\mu_2-\arctan\left(\frac{\lambda\sin(\theta_{1'})}{\kappa_2}\right)\right)}d\theta_2.$$

It is apparent that the integral term computes the area of location-concentration varying von Mises distributions as $\int_{a_2}^{b_2}f_{tvM}\left(\theta_2;\mu_2+\arctan\left(\frac{\lambda\sin(\theta_{1'})}{\kappa_2}\right),\sqrt{\kappa_2^2+(\lambda\sin(\theta_{1'}))^2}\right)d\theta_2$. If we consider the location variations over $[-\pi,\pi]$ by means of the $\sin(\theta_{1'})$ function, the distribution in the integrand is displaced over the interval $[-\arctan\left(\frac{\lambda}{\kappa_2}\right),0]$ when $\sin(\theta_{1'})<0$ (from displacement 0 to displacement $-\arctan\left(\frac{\lambda}{\kappa_2}\right)$ when $\theta_{1'}\in[-\pi,-\frac{\pi}{2}]$ and from displacement $-\arctan\left(\frac{\lambda}{\kappa_2}\right)$ to displacement 0 when $\theta_{1'}\in[-\frac{\pi}{2},0]$), and over the interval $[0,\arctan\left(\frac{\lambda}{\kappa_2}\right)]$ when $\sin(\theta_{1'})>0$ (similary for $\theta_{1'}\in[0,\frac{\pi}{2}]$ and $\theta_{1'}\in[\frac{\pi}{2},\pi]$). If we consider concentration variations, we can regard the source of bi-modality of the integral term as the $\sqrt{\kappa_2^2+(\lambda\sin(\theta_{1'}))^2}$ subterm, given that $\sin^2(\theta_{1'})$ is a $\pi-$periodic solely positive function. Additionally, from $\theta_{1'}=0$ to $\theta_{1'}=\frac{\pi}{2}$ and from $\theta_{1'}=-\pi$ to $\theta_{1'}=-\frac{\pi}{2}$, the concentration parameter grows from its minimum value κ_2 to its maximum value $\sqrt{\kappa_2^2+\lambda^2}$, while it decreases from its maximum to its minimum value in the cases of $\theta_{1'}$ from $-\frac{\pi}{2}$ to 0 and from $\frac{\pi}{2}$ to π.

The proof then follows trivially by noting that, truncation parameters aside, the function's behavior in $[\mu_2-\pi,\mu_2]$ can be considered symmetrical w.r.t. μ_2 to the function's behavior in $[\mu_2,\mu_2+\pi]$. The symmetry w.r.t. μ_2 in the truncation parameters selects two subintervals of symmetrical behavior w.r.t. μ_1, thus producing a function that is symmetrical w.r.t. μ_1.

Further analyzing the integral term we look to determine the critical points and understand how the selection of truncation parameters affects the integral term behaviour. We take

$$\begin{aligned}v_1(\theta_{1'}) &= \int_{a_2}^{b_2}e^{\kappa_2\cos(\theta_2-\mu_2)+\lambda\sin(\theta_{1'})\sin(\theta_2-\mu_2)}d\theta_2\\ v_2(\theta_{1'}) &= \int_{a_2}^{b_2}\sin(\theta_2-\mu_2)e^{\kappa_2\cos(\theta_2-\mu_2)+\lambda\sin(\theta_{1'})\sin(\theta_2-\mu_2)}d\theta_2,\end{aligned}$$

where

$$\lambda\cos(\theta_{1'})v_2(\theta_{1'})=v_1'(\theta_{1'}).$$

We now want to analyze $v_2(\theta_{1'})$ as it is part of the derivate expression of $v_1(\theta_{1'})$. Taking the integrand of $v_2(\theta_{1'})$ to be

$$f_{v_2}(\theta_2;\theta_{1'})=\sin(\theta_2-\mu_2)e^{\kappa_2\cos(\theta_2-\mu_2)+\lambda\sin(\theta_{1'})\sin(\theta_2-\mu_2)}$$

Note that, in $f_{v_2}(\theta_2;\theta_{1'})$, the argument is θ_2 since it creates the area that is to be computed in $v_2(\theta_{1'})$. $\theta_{1'}$ can be considered here as a modifying parameter. The $f_{v_2}(\theta_2;\theta_{1'})$ function comprises the product of a strictly positive function $e^{(\cdot)}$ and a $\sin(\cdot)$ function. Therefore, the sign of $f_{v_2}(\theta_2;\theta_{1'})$ is solely determined by the sign of the $\sin(\cdot)$ function. To be precise, if $\theta_2 \in [\mu_2 - \pi, \mu_2]$ then $f_{v_2}(\theta_2;\theta_{1'}) \leq 0$ and if $\theta_2 \in [\mu_2, \mu_2 + \pi]$ then $f_{v_2}(\theta_2;\theta_{1'}) \geq 0$. Therefore, we can subdivide $v_2(\theta_{1'})$ as

$$v_2(\theta_{1'}) = \int_{a_2}^{\mu_2} f_{v_2}(\theta_2;\theta_{1'})d\theta_2 + \int_{\mu_2}^{b_2} f_{v_2}(\theta_2;\theta_{1'})d\theta_2,$$

where the first addend is a solely negative term and the second addend is a solely positive term provided that $\mu_2 \in (a_2, b_2)$. In the symmetry case, if $\theta_{1'} = 0$ we have

$$-\int_{a_2}^{\mu_2} f_{v_2}(\theta_2;0)d\theta_2 = \int_{\mu_2}^{b_2} f_{v_2}(\theta_2;0)d\theta_2; \tag{A.8}$$

for $\theta_{1'} \in (0, \pi)$ we have

$$-\int_{a_2}^{\mu_2} f_{v_2}(\theta_2;\theta_{1'})d\theta_2 < \int_{\mu_2}^{b_2} f_{v_2}(\theta_2;\theta_{1'})d\theta_2; \tag{A.9}$$

and for $\theta_{1'} \in (-\pi, 0)$ we have

$$-\int_{a_2}^{\mu_2} f_{v_2}(\theta_2;\theta_{1'})d\theta_2 > \int_{\mu_2}^{b_2} f_{v_2}(\theta_2;\theta_{1'})d\theta_2 \tag{A.10}$$

Intuitively, the displaced exponential w.r.t. the μ_2 term increases all the values of either the negative or the positive curve of the $\sin(\theta_2 - \mu_2)$ function and reduces the curve of the opposite sign in less amount, therefore defining the sign and the value of $v_2(\theta_{1'})$. Formally, this to hold, we need to prove that $\forall\theta_{1'} \in (-\pi, 0)$ $f_{v_2}(\theta_2;0) - f_{v_2}(\theta_2;\theta_{1'}) > 0$ if $\theta_2 \in (\mu_2 - \pi, \mu_2)$ and $\forall\theta_{1'} \in (-\pi, 0)$ $f_{v_2}(\theta_2;0) - f_{v_2}(\theta_2;\theta_{1'}) < 0$ if $\theta_2 \in (\mu_2, \mu_2 + \pi)$ for the negative displacement, and an analogous statement for $\theta_{1'} \in (0, \pi)$ positive displacement. For the negative displacement case, it follows that

$$\begin{aligned} \sin(\theta_2 - \mu_2)e^{\kappa_2\cos(\theta_2-\mu_2)} - \sin(\theta_2 - \mu_2)e^{\kappa_2\cos(\theta_2-\mu_2)+\lambda\sin(\theta_{1'})\sin(\theta_2-\mu_2)} &> 0 \\ \sin(\theta_2 - \mu_2)\left(e^{\kappa_2\cos(\theta_2-\mu_2)} - e^{\kappa_2\cos(\theta_2-\mu_2)+\lambda\sin(\theta_{1'})\sin(\theta_2-\mu_2)}\right) &> 0. \end{aligned}$$

As $\sin(\theta_2 - \mu_2) < 0$ in $\theta_2 \in [\mu_2 - \pi, \mu_2]$ it suffices if

$$e^{\kappa_2\cos(\theta_2-\mu_2)} - e^{\kappa_2\cos(\theta_2-\mu_2)+\lambda\sin(\theta_{1'})\sin(\theta_2-\mu_2)} < 0$$

in $\theta_2 \in [\mu_2 - \pi, \mu_2]$. We proceed as follows:

$$\begin{aligned} e^{\kappa_2\cos(\theta_2-\mu_2)} - e^{\kappa_2\cos(\theta_2-\mu_2)+\lambda\sin(\theta_{1'})\sin(\theta_2-\mu_2)} &< 0 \\ e^{-\lambda\sin(\theta_{1'})\sin(\theta_2-\mu_2)} &< 1 \\ -\lambda\sin(\theta_{1'})\sin(\theta_2 - \mu_2) &< 0 \end{aligned}$$

and, since we have specified $\theta_{1'} \in (-\pi, 0)$ and then $\sin(\theta_{1'}) < 0$, we have $-\lambda\sin(\theta_{1'}) > 0$. Therefore, the sign of $-\lambda\sin(\theta_{1'})\sin(\theta_2 - \mu_2)$ follows from that of $\sin(\theta_2 - \mu_2)$. This proves the statement for both θ_2 intervals in the case of negative displacement. The proof for positive displacement is analogous.

This result implies that the selection of truncation parameters that are symmetrical w.r.t. μ_2 does not change the monotony of $v_1(\theta_{1'})$. More generally, this result implies that no selection of truncation parameters changes the monotonicity of $v_2(\theta_{1'})$, that is, increasing in $[-\frac{\pi}{2}, \frac{\pi}{2}]$ and decreasing otherwise.

Since (A.8), (A.9) and (A.10) hold, we can now perform the sign and critical points analysis of $\lambda\cos(\theta_{1'})v_2(\theta_{1'})$ to obtain that $v_1(\theta_{1'})$ follows the monotony of $\sin^2(\theta_{1'})$ for any a_2, b_2 such that $\cos(b_2 - \mu_2) = \cos(a_2 - \mu_2)$, with critical points $\{-\frac{\pi}{2}, 0, \frac{\pi}{2}\}$. Therefore, in Equation (A.4), unimodal/bimodal observed distributions are "decided" for this case by the product of $v_1(\theta_{1'})$ with $e^{\kappa_1\cos(\theta_{1'})}$.

Therefore, if $T(\lambda, \mu_2, \kappa_1, \kappa_2, a_2, b_2) > 0$ then $f_{umtvM}(\theta_{1'})$ presents a minimum critical point at μ_1 and the distribution has two equal symmetrical maxima in $[-\frac{\pi}{2}, \frac{\pi}{2}]$ (the maxima location interval can be proven as a result of monotony and sign comparisons between $v_1(\theta_{1'})$ and $e^{\kappa_1\cos(\theta_{1'})}$). Respectively, if $T(\lambda, \mu_2, \kappa_1, \kappa_2, a_2, b_2) < 0$ then $f_{umtvM}(\theta_{1'})$ presents a maximum critical point and the distribution is unimodal. This result generalizes the outcome for the non-truncated case to symmetrical parameters other than a_2, b_2 such that $b_2 - a_2 = 2\pi$ (Singh [2002]). This suffices to prove cases 1 and 2 of the theorem.

(c) For case 3 we want to observe the behavior of the marginal distribution for different cases of circular distances of a_2, b_2 truncation parameters w.r.t. μ_2. Thus, we need knowledge about the subterm $v_2(\theta_{1'})$ when a_2, b_2 truncation parameters are not symmetrical w.r.t. μ_2 in order to reach useful results. We will address this point first.

If we now observe $\lambda\cos(\theta_{1'})v_2(\theta_{1'}) = 0$ for non-symmetrical parameters we can as before, isolate two critical points:

$$\begin{aligned}\theta_{1'} &= -\frac{\pi}{2},\\ \theta_{1'} &= \frac{\pi}{2}\end{aligned}$$

and a third critical point at some $\theta_{1'}$ such that $-\int_{a_2}^{\mu_2} f_{v_2}(\theta_2;\theta_{1'}) + \int_{\mu_2}^{b_2} f_{v_2}(\theta_2;\theta_{1'}) = 0$ if a_2, b_2 are not truncation parameters that satisfy any of the following conditions:

(i) $a_2, b_2 \in [\mu_2, \mu_2 + \pi]$ as then $v_2(\theta_{1'}) > 0\ \forall\theta_{1'} \in [-\pi, \pi]$

(ii) $a_2, b_2 \in [\mu_2 - \pi, \mu_2]$ as then $v_2(\theta_{1'}) < 0\ \forall\theta_{1'} \in [-\pi, \pi]$

(iii) $\mu_2 \in (a_2, b_2)$ such as $-\int_{a_2}^{\mu_2} f_{v_{2'}}(\theta_2; -\frac{\pi}{2})d\theta_2 \leq \int_{\mu_2}^{b_2} f_{v_{2'}}(\theta_2; -\frac{\pi}{2})d\theta_2$ as then $v_2(\theta_{1'}) > 0\ \forall\theta_{1'} \in [-\pi, \pi]$

(iv) $\mu_2 \in (a_2, b_2)$ such as $\int_{\mu_2}^{b_2} f_{v_{2'}}(\theta_2; \frac{\pi}{2})d\theta_2 \leq -\int_{a_2}^{\mu_2} f_{v_{2'}}(\theta_2; \frac{\pi}{2})d\theta_2$ as then $v_2(\theta_{1'}) < 0\ \forall\theta_{1'} \in [-\pi, \pi]$.

Notice that from the viewpoint of truncation parameters, cases (iii) and (iv) can be considered opposite. Also, as highlighted by the previous analysis, it is clear that case (iii) implies $\cos(b_2 - \mu_2) < \cos(a_2 - \mu_2)$ (more intuitively, $\cos(b_2 - \mu_2) \ll \cos(a_2 - \mu_2)$) and case (iv) $\cos(b_2 - \mu_2) > \cos(a_2 - \mu_2)$ (more intuitively, $\cos(b_2 - \mu_2) \gg \cos(a_2 - \mu_2)$). We will refer to cases (iii) and (iv) as the strong lower parameter cases.

Therefore, by manipulating a_2, b_2 truncation parameters, it is possible to reshape $v_1(\theta_{1'})$ to exhibit a minimum in $-\frac{\pi}{2}$ and a maximum in $\frac{\pi}{2}$ if case (i) or (iii) applies or to exhibit a maximum in $-\frac{\pi}{2}$ and

a minimum in $\frac{\pi}{2}$ if case (ii) or (iv) applies. In these cases, $v_1(\theta_{1'})$ is an integral term with unimodal behavior.

It follows that any other case for non-symmetrical truncation parameters implies $\mu_2 \in (a_2, b_2)$, and $v_1(\theta_{1'})$ exhibits two differentiated maxima in $-\frac{\pi}{2}$ and $\frac{\pi}{2}$. Also, $v_2(-\frac{\pi}{2}) < 0$ and $v_2(\frac{\pi}{2}) > 0$. If we examine the case of $\theta_{1'} = 0$ for truncation parameters a_2, b_2 such that $\cos(b_2 - \mu_2) > \cos(a_2 - \mu_2)$ then $-\int_{a_2}^{\mu_2} f_{v_2}(\theta_2; 0)d\theta_2 > \int_{\mu_2}^{b_2} f_{v_2}(\theta_2; 0)d\theta_2$ and therefore $v_2(\theta_{1'}) = 0$ for some $\theta^*_{1'} \in [0, \frac{\pi}{2}]$ such that $v_2(\theta_{1'}) < 0$ if $\theta_{1'} \in [0, \theta^*_{1'})$ and $v_2(\theta_{1'}) > 0$ if $\theta_{1'} \in (\theta^*_{1'}, \frac{\pi}{2}]$. It follows that this also impliesthe existence of another minimum in $[\frac{\pi}{2}, \pi]$ as $v_2(\theta_{1'}) > 0 \ \forall \theta_{1'} \in [\frac{\pi}{2}, \pi - \theta^*_{1'})$ and $v_2(\theta_{1'}) < 0 \ \forall \theta_{1'} \in (\pi - \theta^*_{1'}, \pi]$. Similarly, if $\cos(b_2 - \mu_2) < \cos(a_2 - \mu_2)$ then $-\int_{a_2}^{\mu_2} f_{v_2}(\theta_2; 0)d\theta_2 < \int_{\mu_2}^{b_2} f_{v_2}(\theta_2; 0)d\theta_2$ and therefore $v_2(\theta_{1'}) = 0$ for some $\theta^*_{1'} \in [-\frac{\pi}{2}, 0]$ and $-\pi - \theta^*_{1'} \in [-\pi, -\frac{\pi}{2}]$, that is, the minimum of $v_1(\theta_{1'})$ that appears in $[-\frac{\pi}{2}, \frac{\pi}{2}]$ is more precisely located in the $\frac{\pi}{2}$−length interval associated with the truncation parameter that presents the smallest circular distance w.r.t. μ_2 and imples an additional minimum located in the contiguous $\frac{\pi}{2}$−length interval more distant from $\theta_{1'} = 0$.

Additionally, the global maximum of the two differentiated maxima is that of the $\frac{\pi}{2}$−length interval associated with the truncation parameter that has the largest circular distance w.r.t. μ_2. We can prove this by comparing both maxima as follows:

$$v_1\left(-\frac{\pi}{2}\right) - v_1\left(\frac{\pi}{2}\right) > 0 \text{ if } \cos(b_2 - \mu_2) > \cos(a_2 - \mu_2).$$

Thus if we take $\kappa' = \sqrt{\kappa_2^2 + (\lambda)^2}$ we have

$$\int_{a_2}^{b_2} e^{\kappa' \cos\left(\theta_2 - \mu_2 - \arctan\left(-\frac{\lambda}{\kappa_2}\right)\right)} d\theta_2 - \int_{a_2}^{b_2} e^{\kappa' \cos\left(\theta_2 - \mu_2 - \arctan\left(\frac{\lambda}{\kappa_2}\right)\right)} d\theta_2 > 0.$$

Expressing this by means of the distribution function we obtain

$$\left[I(\theta, -\mu_2 - \arctan\left(-\frac{\lambda}{\kappa_2}\right), \kappa')\right]_{a_2}^{b_2} - \left[I(\theta, -\mu_2 - \arctan\left(\frac{\lambda}{\kappa_2}\right), \kappa')\right]_{a_2}^{b_2} > 0. \tag{A.11}$$

Clearly, $I(\theta, \mu, \kappa)$ is strictly increasing and $e^{\kappa' \cos\left(\theta_2 - \mu_2 - \arctan\left(-\frac{\lambda}{\kappa_2}\right)\right)}$ is symmetrical to $e^{\kappa' \cos\left(\theta_2 - \mu_2 - \arctan\left(\frac{\lambda}{\kappa_2}\right)\right)}$ w.r.t. μ_2. Therefore

1.
$$\left[I(\theta, -\mu_2 - \arctan\left(\frac{-\lambda}{\kappa_2}\right), \kappa')\right]_{2\mu_2 - b_2}^{\mu_2} = \left[I(\theta, -\mu_2 - \arctan\left(\frac{\lambda}{\kappa_2}\right), \kappa')\right]_{\mu_2}^{b_2}$$

2.
$$\left[I(\theta, -\mu_2 - \arctan\left(\frac{-\lambda}{\kappa_2}\right), \kappa')\right]_{\mu_2}^{2\mu_2 - a_2} = \left[I(\theta, -\mu_2 - \arctan\left(\frac{\lambda}{\kappa_2}\right), \kappa')\right]_{a_2}^{\mu_2}$$

taking

$$\left[I(\theta, -\mu_2 - \arctan\left(\frac{-\lambda}{\kappa_2}\right), \kappa')\right] = Ie_1(\theta)$$

$$\left[I(\theta, -\mu_2 - \arctan\left(\frac{\lambda}{\kappa_2}\right), \kappa')\right] = Ie_2(\theta),$$

we can rewrite inequation (A.11) as

$$[Ie_1(\theta)]_{a_2}^{\mu_2} + [Ie_1(\theta)]_{\mu_2}^{b_2} - [Ie_2(\theta)]_{a_2}^{\mu_2} - [Ie_2(\theta)]_{\mu_2}^{b_2} > 0,$$

substituting,

$$\begin{aligned} [Ie_1(\theta)]_{\mu_2}^{a_2} + [Ie_1(\theta)]_{b_2}^{\mu_2} - [Ie_1(\theta)]_{\mu_2}^{2\mu_2-a_2} - [Ie_1(\theta)]_{2\mu_2-b_2}^{\mu_2} &> 0 \\ -Ie_1(a_2) + Ie_1(b_2) - Ie_1(2\mu_2 - a_2) + Ie_1(2\mu_2 - b_2) &> 0 \\ [Ie_1(\theta)]_{a_2}^{2\mu_2-b_2} - [Ie_1(\theta)]_{b_2}^{2\mu_2-a_2} &> 0, \end{aligned}$$

that is, the inequation reduces to the comparison between the area in two subintervals of equal length that are symmetrical w.r.t. μ_2. By this symmetry and by the fact that the mode is in $(-\frac{\pi}{2}, 0)$ and the anti-mode in $(\frac{\pi}{2}, \pi)$ in $e^{\kappa' \cos\left(\theta_2-\mu_2-\arctan\left(-\frac{\lambda}{\kappa_2}\right)\right)}$, we can safely conclude that the inequation holds thus proving the statement. Therefore, for any marginal truncated distribution, the global maximum in the integral term is located in $\theta_{1'} = \frac{\pi}{2}$ if $\cos(a_2 - \mu_2) > \cos(b_2 - \mu_2)$ and in $\theta_{1'} = -\frac{\pi}{2}$ if $\cos(a_2 - \mu_2) < \cos(b_2 - \mu_2)$.

At this point all behaviors for critical points and monotony of $v_1(\theta_{1'})$ have been characterized. Analogously to the non-truncated case, the effect of the $e^{\kappa_1 \cos(\theta_{1'})}$ subterm has to be taken into consideration in order to determine the shape of the distribution. To do this, we perform a monotony study that incorporates all previous developments.

(d) After conducting the study on $v_2(\theta_{1'})$ and $v_1(\theta_{1'})$, we proceed by equating function (A.5) to zero, resulting in

$$-\kappa_1 \sin(\theta_{1'}) v_1(\theta_{1'}) + \lambda \cos(\theta_{1'}) v_2(\theta_{1'}) = 0.$$

If we consider the cases where $a_2, b_2 \in [\mu_2, \mu_2 + \pi]$ or a_2 is a strong lower parameter w.r.t b_2 we have:

1. $v_2(\theta_{1'}) > 0 \ \forall \theta_{1'} \in [-\pi, \pi]$.

2. If $\theta_{1'} \in [-\pi, -\frac{\pi}{2}]$, then $\sin(\theta_{1'}) \leq 0$ and $\cos(\theta_{1'}) \leq 0$. In this case, at least a minimum and a critical point of $f_{umtvM}(\theta_{1'})$ can be found in the examined interval as shown by:

$$\begin{aligned} f'_{umtvM}(-\pi) &= e^{-\kappa_1}\left(-\lambda \int_{a_2}^{b_2} \sin(\theta_2 - \mu_2) e^{\kappa_2 \cos(\theta_2-\mu_2)} d\theta_2\right) \\ f'_{umtvM}\left(-\frac{\pi}{2}\right) &= \kappa_1 \int_{a_2}^{b_2} e^{\kappa_2 \cos(\theta_2-\mu_2) - \lambda \sin(\theta_2-\mu_2)} d\theta_2 > 0, \end{aligned}$$

   where $f'_{umtvM}(-\pi) < 0$. Notice that if $a_2, b_2 \in [\mu_2, \mu_2 + \pi]$ the critical point neccesarily exists regardless of the effect of the other parameters.

3. If $\theta_{1'} \in [-\frac{\pi}{2}, 0]$, then $\sin(\theta_{1'}) \leq 0$ and $\cos(\theta_{1'}) \geq 0$. $f_{umtvM}(\theta_{1'})$ exhibits a monotonic increasing behavior, as all terms involved in the expression are positive.

4. If $\theta_{1'} \in [0, \frac{\pi}{2}]$, then $\sin(\theta_{1'}) \geq 0$ and $\cos(\theta_{1'}) \geq 0$. Here, at least a maximum and a critical point can be found in the interval by considering Equation (A.6), where $f'_{umtvM}(0) > 0$, and

$$f'_{umtvM}\left(\frac{\pi}{2}\right) = -\kappa_1 \int_{a_2}^{b_2} e^{\kappa_2 \cos(\theta_2-\mu_2) - \lambda \sin(\theta_2-\mu_2)} d\theta_2 < 0.$$

5. If $\theta_{1'} \in [\frac{\pi}{2}, \pi]$, then $\sin(\theta_{1'}) \geq 0$ and $\cos(\theta_{1'}) \leq 0$. $f_{umtvM}(\theta_{1'})$ exhibits a monotonic decreasing behavior, as all terms involved in the expression are negative.

Therefore, for this case, the distribution exhibits critical points in two non-contiguous intervals. By the previous developments, such a distribution of critical points would only correspond to the unimodal case and also, as the contribution of $e^{\kappa_1 \cos(\theta_{1'})}$ is symmetrical w.r.t. μ_1 or $\theta_{1'} = 0$, the marginal function could only have one global maximum in $\theta_{1'} \in [0, \frac{\pi}{2}]$ interval and one global minimum in $\theta_{1'} \in [-\pi, -\frac{\pi}{2}]$.

The case where $a_2, b_2 \in [\mu_2 - \pi, \mu_2]$ or b_2 is a strong lower parameter w.r.t a_2 can be understood as "symmetric behavior w.r.t μ_1", since the results for $\theta_{1'} \in [-\pi, -\frac{\pi}{2}]$ now hold for $\theta_{1'} \in [\frac{\pi}{2}, \pi]$ and the results for $\theta_{1'} \in [-\frac{\pi}{2}, 0]$ now hold for $\theta_{1'} \in [0, \frac{\pi}{2}]$. This property, general to the $[-\pi, \pi]$ interval, guarantees that in our case, it suffices to determine the behavior for one of the two remaining cases to completely determine the behavior of the marginal function.

We now consider the remaining parameter configurations that satisfy $\cos(b_2 - \mu_2) > \cos(a_2 - \mu_2)$.

1. If $\theta_{1'} \in [-\pi, -\frac{\pi}{2}]$, then $v_2(\theta_{1'}) < 0$, thus resulting in $f_{umtvM}(\theta_{1'})$, which exhibits a strictly increasing behavior, as all terms involved in the expression are now positive.

2. If $\theta_{1'} \in [-\frac{\pi}{2}, 0]$, then $v_2(\theta_{1'}) < 0$. In this case, after performing sign comparisons on the extrema, there is at least one critical point and one maximum in the interval.

3. If $\theta_{1'} \in [0, \frac{\pi}{2}]$, $v_2(\theta_{1'}) < 0 \ \forall \theta_{1'} \in [0, \theta^*_{1'})$ and $v_2(\theta_{1'}) > 0 \ \forall \theta_{1'} \in [\theta^*_{1'}, \frac{\pi}{2})$. Therefore, no critical point exists in $[0, \theta^*_{1'})$, since $f_{umtvM}(\theta_{1'})$ exhibits a decreasing behavior and all terms involved in the expression are negative. In $[\theta^*_{1'}, \frac{\pi}{2})$, no, one or two critical points can occur as both sign and monotony comparisons were not conclusive.

4. If $\theta_{1'} \in [\frac{\pi}{2}, \pi]$, then $v_2(\theta_{1'}) > 0 \ \forall \theta_{1'} \in [\frac{\pi}{2}, \pi - \theta^*_{1'})$ and $v_2(\theta_{1'}) < 0 \ \forall \theta_{1'} \in (\pi - \theta^*_{1'}, \pi]$. Therefore, no critical point exists in $[\frac{\pi}{2}, \pi - \theta^*_{1'})$ since $f_{umtvM}(\theta_{1'})$ exhibits a decreasing behavior as all terms involved in the expression are negative. In $(\pi - \theta^*_{1'}, \pi]$, after performing sign comparisons on the extrema, at least one critical point can occur. Therefore, for this case, the distribution has three contiguous intervals containing critical points. Since clearly no more than two critical points are allowed in a $\frac{\pi}{2}$–length interval, the case with two possible critical points in $[\theta^*_{1'}, \frac{\pi}{2})$ is the case of bi-maximality (differentiated maxima) with a minimum and a maximum in $\theta_{1'} \in [\theta^*_{1'}, \frac{\pi}{2})$ and a maximum in $\theta_{1'} \in [-\frac{\pi}{2}, 0]$. Complementarily, this distribution of critical points "corresponds" to the bi-maximal (differentiated maxima) behavior of $v_1(\theta_{1'})$, and, therefore, the critical point in $\theta_{1'} \in [-\frac{\pi}{2}, 0]$ is necessarily a maximum, and the critical point in $[\frac{\pi}{2}, \pi]$ is necessarily a minimum. Thus, it can be concuded that in the case of bimodality, the interval associated with the truncation parameter that has the shortest circular distance w.r.t. μ_2 contains the two critical points, whereas the interval associated with the truncation parameter that has the largest circular distance w.r.t. μ_2 contains the global maximum.

If $\lambda < 0$, the proof follows trivially by noting that the displacement caused by the $\sin(\cdot)$ function in the exponent that appears in the $v_1(\theta_{1'})$ subterm is the opposite. This in turn causes the distribution to have an opposite symmetrical behaviour w.r.t. μ_1. This suffices to prove case 3 of the theorem. Case 4 can also be proven with the developed theory. However, it can additionally be proven by ruling out any other possible outcome, considering the three previously developed cases.

□

Appendix B

Random forests for regression as a weighted sum of k-potential nearest neighbors

Proof of Lemma 8.3.1. We define $Rm^*(\mathbf{x}_i|\mathcal{D}_j^*) = \mathcal{D}_j^* \cap Rm(\mathbf{x}_i)$ and $R^*(\mathbf{x}_0, \mathbf{x}_i|\mathcal{D}_j^*) = Rm^*(\mathbf{x}_i|\mathcal{D}_j^*) \cup \{\mathbf{x}_i\}$,. That is $R^*(\mathbf{x}_0, \mathbf{x}_i|\mathcal{D}_j^*)$ is in $\mathcal{D}_j^*$ the equivalent of $R(\mathbf{x}_0, \mathbf{x}_i)$ in $\mathcal{D}_n$.

By the definition of k-PNN, $\mathbf{x}_i$ is a k-PNN of $\mathbf{x}_0$ in $\mathcal{D}_j^*$ if $|R^*(\mathbf{x}_0, \mathbf{x}_i|\mathcal{D}_j^*)| \leq k$. Then

$$|\mathcal{D}_j^* \cap Rm(\mathbf{x}_i)| \leq k - 1$$

□

Proof of Theorem 8.3.1. By Lemma 3.1, it is enough to show that each $\mathbf{x}_i$ in $f_{1-SA}^*(\mathbf{x}_0)$ is paired with a weight that accounts for the value of $|Req_1(\mathbf{x}_i)|$. We can use the proofs of results in Biau et al. [2010] for 1-NN together with Lemma 1 to prove the expression of the weight as

$$v_j(\mathbf{x}_0) = \left(1 - \frac{|R(\mathbf{x}_0, \mathbf{x}_j)| - 1}{n}\right)^n - \left(1 - \frac{|R(\mathbf{x}_0, \mathbf{x}_j)|}{n}\right)^n$$

where

$$|R(\mathbf{x}_0, \mathbf{x}_j)| = i$$

Now what remains is to show that the right-hand side of Equation (8.5) correctly pairs the bootstrap weights and the datapoints. This is trivial due to the structure of the double summation and the use of $F_i(\mathbf{x}_0)$. □

Proof of Lemma 8.4.1. We can prove it by contradiction.

Let us assume a RF regression estimate $\hat{f}_{RF}(\mathbf{x}_0)$ such that for $k = 1$, and $\mathbf{x}_i$, $w_i(\mathbf{x}_0) > v_i(\mathbf{x}_0)$.

We can find the set $\mathcal{V}_{\mathbf{x}_i}^* \subset B(\mathcal{D}_n)$ of bootstrap variations where $\mathbf{x}_i$ is a 1-PNN of $\mathbf{x}_0$. Since $v_i(\mathbf{x}_0)$ value is the proportion of bootstrap variations that do not contain the datapoints in $R(\mathbf{x}_0, \mathbf{x}_i)$, then $v_i(\mathbf{x}_0)$

can be expressed, by Lemma 1, as:

$$v_i(\mathbf{x}_0) = \frac{|\mathcal{V}^*_{\mathbf{x}_i}|}{n^n}$$

If our assumption holds ($w_i(\mathbf{x}_0) > v_i(\mathbf{x}_0)$), then a bootstrap variation selection set, $\mathcal{W}^*_{\mathbf{x}_i} \subset B(\mathcal{D}_n)$ can be found such that $|\mathcal{W}^*_{\mathbf{x}_i}| > |\mathcal{V}^*_{\mathbf{x}_i}|$. However, in that case $\mathcal{W}^*_{\mathbf{x}_i}$ necessarily has to account for bootstrap variation selections where $\mathbf{x}_i$ is not a 1-PNN.

□

Proof of Lemma 8.4.2. We can prove this by induction on the two parameters l_p and l_m of a b-term. For simplicity, let us define $e(i) = \left(1 - \frac{i}{n}\right)^n$.

If we rewrite the results in Biau et al. [2010] (bagged 1-NN) using the b-terms notation we have

$$\begin{aligned} v_i(\mathbf{x}_0) &= S(\neg\mathbf{x}_1 \ldots \neg\mathbf{x}_{i-1}) - S(\neg\mathbf{x}_1 \ldots \neg\mathbf{x}_{i-1}\neg\mathbf{x}_i) \\ v_i(\mathbf{x}_0) &= S(\neg\mathbf{x}_1 \ldots \neg\mathbf{x}_{i-1} - \neg\mathbf{x}_1 \ldots \neg\mathbf{x}_{i-1}\neg\mathbf{x}_i) \end{aligned}$$

and by b-terms Property 9, we have

$$v_i(\mathbf{x}_0) = S(\neg\mathbf{x}_1 \ldots \neg\mathbf{x}_{i-1}\mathbf{x}_i)$$

For purposes of readability, let us define $S'(l_p, l_m) = S(b_i)$ where $b_i \in E_{[l_p,l_m]}$. Now for the case of l_m, by b-terms property 4, we can eliminate the particular indexes and write

$$S'(0, l_m) = e(l_m)$$

and we assume by induction hypothesis on l_m

$$S'(1, l_m - 1) = e(l_m - 1) - e(l_m).$$

Then, for $S'(1, l_m)$ we can simply write by Property 9, the corresponding b-term as:

$$\begin{aligned} b_i &= \neg\mathbf{x}_1 \ldots \neg\mathbf{x}_{l_m}\mathbf{x}_{l_m+1} \\ &= \neg\mathbf{x}_1 \ldots \neg\mathbf{x}_{l_m} \\ &- \neg\mathbf{x}_1 \ldots \neg\mathbf{x}_{l_m}\neg\mathbf{x}_{l_m+1} \end{aligned}$$

and obtain

$$S'(1, l_m) = e(l_m) - e(l_m + 1).$$

Notice that for purposes of calculation, only parameters l_p and l_m are needed of a b-term, therefore when writing a particular expression of b_i as a member of $E_{[l_p,l_m]}$, the most convenient indexes can be chosen freely without loss of generality.

We can write the previous result as

$$S'(1, l_m) = S'(0, l_m) - S'(0, l_m + 1).$$

If we assume now

$$S'(l_p - 1, l_m) = S'(l_p - 2, l_m) - S'(l_p - 2, l_m + 1)$$

as our induction hypothesis for parameter l_p, we can use again Property 9, and write the corresponding b-term as

$$\begin{aligned} b_j &= \mathbf{x}_1 \dots \mathbf{x}_{l_p} \neg \mathbf{x}_{l_p+1} \dots \neg \mathbf{x}_{l_p+l_m} \\ &= \mathbf{x}_1 \dots \mathbf{x}_{l_p-1} \neg \mathbf{x}_{l_p+1} \dots \neg \mathbf{x}_{l_p+l_m} \\ &- \mathbf{x}_1 \dots \mathbf{x}_{l_p-1} \neg \mathbf{x}_{l_p} \dots \neg \mathbf{x}_{l_p+l_m} \end{aligned}$$

in order to obtain

$$S'(l_p, l_m) = S'(l_p - 1, l_m) - S'(l_p - 1, l_m + 1).$$

Finalizing the proof is now reduced to observe the pattern of how $S'(.,.)$ unfolds when calculated. A well-known property of the Pascal triangle is that $P(i, j) + P(i + 1, j) = P(i + 1, j + 1)$. This shows that the term-wise sum of two rows of the Pascal triangle where one is shifted by one produces the next row. This occurs in unfolding our recursive function with alternating signs because we have a difference rather than a sum. Accounting for all factors yields

$$S'(l_p, l_m) = S(b_i) = \sum_{i=1}^{l_p+1} P(l_p, i)(-1)^{i+1} \left(1 - \frac{i + l_m - 1}{n}\right)^n$$

□

Proof of Theorem 8.4.2. For $z_i(\mathbf{x}_0)$, it suffices to show that all relevant bootstrap variation selections (with relevant in the sense of Lemma 1) are listed.

Let the elements in $Req_1(\mathbf{x}_i)$ be regarded as the list of points that increase the PNN distance from $\mathbf{x}_i$ to $\mathbf{x}_0$. Then we can define $Dep(\mathbf{x}_i, \mathbf{x}_0)$ conversely as the set of points whose PNN distance to $\mathbf{x}_0$ is increased by $\mathbf{x}_i$ (Figure 5, solid line areas).

It can be seen that, regardless of the distribution of the datapoints in $Req_1(\mathbf{x}_i)$, all of them contain $\mathbf{x}_i$ in their $Dep(\mathbf{x}_i, \mathbf{x}_0)$ sets. Thus, for $k > 1$, $Req_k(\mathbf{x}_i)$ contains all combinations of $d_1 = l - k, d_2 = l - k + 1, \dots$ and $d_k = l - 1$ non included datapoints where $l = |R(\mathbf{x}_i, \mathbf{x}_0)| - 1$. Cardinality-wise we have

$$|Req_k(\mathbf{x}_i)| = \sum_{j=l-k}^{l-1} \binom{l-1}{j}.$$

All these combinations, coupled with the addition of the $\mathbf{x}_i$ restriction yields

$$v_i(\mathbf{x}_0) = (\mathbf{x}_i) \left(\sum_{c \in Req_k(\mathbf{x}_i)} r_{ic} \right)$$

which corresponds to the "always selected" case (Lemma 4.3).

We can also see this by considering that

$$\prod_{\mathbf{x}_j \in Ind(\mathbf{x}_i)} [Req_k(\mathbf{x}_j)](\mathbf{x}_j + \neg\mathbf{x}_j) = 1 \tag{B.1}$$

simply because the expression $(\mathbf{x}_j + \neg\mathbf{x}_j)$ inherently cancels to 1. That is, the restricted concatenation operator [.] applies a neutral element to the selected b-terms. More formally if we expand $P_{RkS}(\mathbf{x}_i, z_i(\mathbf{x}_0))$ we have

$$\begin{aligned} P_{RkS}(\mathbf{x}_i, z_i(\mathbf{x}_0)) = \ldots (\mathbf{x}_i)(r_{ig})(\mathbf{x}_j)(b_v)l_1 \\ + (\mathbf{x}_i)(r_{ig})(\neg\mathbf{x}_j)(b_v)l_2 \ldots \end{aligned}$$

where $g \in Req_k(\mathbf{x}_i)$, b_v is an arbitrary b-term, $l_1 = L_{RkS}(\mathbf{x}_i, (\mathbf{x}_i)(r_{ig})(\mathbf{x}_j)(b_v))$ and $l_2 = L_{RkS}(\mathbf{x}_i, (\mathbf{x}_i)(r_{ig})(\neg\mathbf{x}_j)(b_v))$.
Then, only if $l_1 = l_2 = 1$ the sum can be reduced to $(\mathbf{x}_i)(r_{ig})(b_v)$. Thus, splitting criteria determine the reduction possibilities among appearing b-terms (weight shrinking). If no $L_{\cdot}(\mathbf{x}_i, z_i(\mathbf{x}_0))$ is defined or $L_{\cdot}(\mathbf{x}_i, z_i(\mathbf{x}_0)) = 1$ operating the b-terms of $z_i(\mathbf{x}_0)$ yields $v_i(\mathbf{x}_0)$ (this proves Theorem 4.4 and Lemma 4.3).

Now, with left-hand side of Equation (B.1) in $z_i(\mathbf{x}_0)$ we ensure that each datapoint outside $R(\mathbf{x}_0, \mathbf{x}_i)$ is considered when their k-PNN requirements (Req_k) are met in the bootstrap variation selections. This covers the requirements for all datapoints.

This also proves Theorem 4.1. □

Proof of Theorem 8.4.3. Given Theorem 4.2, to complete the proof it suffices to show that for any $\mathcal{D}_n$, $L_{RkS}(\mathbf{x}_i, b_i)$ chooses k or less points in $P_k(\mathbf{x}_0)$ to guarantee $\sum_{i=1}^{n} w_i(\mathbf{x}_0) = 1$.

We have $L_{RkS}(\mathbf{x}_i, b_i) = \frac{1}{k}\frac{k}{l_p}$ for a b-term $b_i \in E_{[l_p, l_m]}$ that includes $\mathbf{x}_i$.

□

Bibliography

M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*. Applied Mathematics Series. Dover Publications, 1964.

A. H. Abuzaid, I. B. Mohamed, and A. G. Hussin. Boxplot for circular variables. *Computational Statistics*, 27(3):381–392, 2012.

R. Aghdam, V. Rezaei Tabar, and H. Pezeshk. Some node ordering methods for the k2 algorithm. *Computational Intelligence*, 35(1):42–58, 2019.

H. Akaike. A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*, pages 215–222. Springer, 1974.

L. Alonso-Nanclares, J. Gonzalez-Soriano, J. Rodriguez, and J. DeFelipe. Gender differences in human cortical synaptic density. *Proceedings of the National Academy of Sciences*, 105(38):14615–14619, 2008.

D. Arion, M. Sabatini, T. Unger, J. Pastor, L. Alonso-Nanclares, I. Ballesteros-Yáñez, R. García Sola, A. Muñoz, K. Mirnics, J. DeFelipe. Correlation of transcriptome profile with electrical activity in temporal lobe epilepsy. *Neurobiology of Disease*, 22(2):374–387, 2006.

S. Arlot and R. Genuer. Analysis of purely random forests bias. *CoRR*, abs/1407.3939, 2014.

G. A. Ascoli, D. E. Donohue, and M. Halavi. Neuromorpho.org: A central resource for neuronal morphologies. *Journal of Neuroscience*, 27(35):9247–9251, 2007.

Z.-D. Bai, L. Devroye, H.-K. Hwang, and T.-H. Tsai. Maxima in hypercubes. *Random Structures & Algorithms*, 27(3):290–309, 2005.

I. Ballesteros-Yáñez, R. Benavides-Piccione, J.-P. Bourgeois, J.-P. Changeux, and J. DeFelipe. Alterations of cortical pyramidal neurons in mice lacking high-affinity nicotinic receptors. *Proceedings of the National Academy of Sciences*, 107(25):11567–11572, 2010.

O. Barndorff-Nielsen and M. Sobel. On the distribution of the number of admissible points in a vector random sample. *Theory of Probability & its Applications*, 11(2):249–269, 1966.

R. Benavides-Piccione, F. Hamzei-Sichani, I. Ballesteros Yáñez, J. DeFelipe, and R. Yuste. Dendritic size of pyramidal neurons differs among mouse cortical regions. *Cerebral Cortex*, 16:990–1001, 2006.

S. Bernard, L. Heutte, and S. Adam. Forest-RK: A new random forest induction method. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, pages 430–437. Springer, 2008.

S. Bernard, S. Adam, and L. Heutte. Dynamic random forests. *Pattern Recognition Letters*, 33(12): 1580–1586, 2012.

G. Biau. Analysis of a random forests model. *Journal of Machine Learning Research*, 13:1063–1095, 2012.

G. Biau and L. Devroye. On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. *Journal of Multivariate Analysis*, 101(10):2499–2518, 2010.

G. Biau and E. Scornet. A random forest guided tour. *Test*, 25(2):197–227, 2016.

G. Biau, L. Devroye, and G. Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9:2015–2033, 2008.

G. Biau, F. Cérou, and A. Guyader. On the rate of convergence of the bagged nearest neighbor estimate. *Journal of Machine Learning Research*, 11:687–712, 2010.

C. Bielza and P. Larrañaga. Discrete Bayesian network classifiers: A survey. *ACM Computing Surveys*, 47(1):5, 2014a.

C. Bielza and P. Larrañaga. Bayesian networks in neuroscience: A survey. *Frontiers in Computational Neuroscience*, 8:Article 131, 2014b.

C. Bielza, G. Li, and P. Larrañaga. Multi-dimensional classification with Bayesian networks. *International Journal of Approximate Reasoning*, 52(6):705–727, 2011.

C. Bielza, R. Benavides-Piccione, P. López-Cruz, P. Larrañaga, and J. DeFelipe. Branching angles of pyramidal cell dendrites follow common geometrical design principles in different cortical areas. *Scientific Reports*, 4: 5909, 2014.

D. A. Bistrian and M. Iakob. One-dimensional truncated von Mises distribution in data modeling. *Annals of Faculty of Engineering Hunedoara*, tome VI, fascicule 3, 2008.

H. Borchani, C. Bielza, and P. Larrañaga. Learning CB-decomposable multi-dimensional Bayesian network classifiers. In *Proceedings of the 5th European Workshop on Probabilistic Graphical Models*, pages 25–32, 2010.

A.-L. Boulesteix, S. Janitza, J. Kruppa, and I. R. König. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):493–507, 2012.

C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in Bayesian networks. In *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*, pages 115–123. Morgan Kaufmann Publishers Inc., 1996.

N. M. T. Bowyer, P. and F. M. Danson. Safari 2000 canopy structural measurements, kalahari transect, wet season 2001. Dataset. 2005.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

L. Breiman. Some infinity theory for predictor ensembles. Technical report, Statistics Dept. UCB, 2000.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

L. Breiman. Consistency for a simple model of random forests. Technical report, University of California, Berkeley, 2004.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees. *Wadsworth International Group*, 1984.

P. Bülhmann and B. Yu. Analyzing bagging. *Annals of Statistics*, pages 927–961, 2002.

A. Buja and W. Stuetzle. The effect of bagging on variance, bias, and mean squared error. *Preprint. AT&T Labs-Research*, 2000.

W. Buntine. Theory refinement on Bayesian networks. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann Publishers Inc., 1991.

B. Caprile, S. Merler, C. Furlanello, and G. Jurman. Exact bagging with k-nearest neighbour classifiers. In *Multiple Classifier Systems*, pages 72–81. Springer, 2004.

E. Castillo, J. Gutierrez, and A. Hadi. Expert systems and probabilistic network models. *Computational Statistics and Data Analysis*, 2(25):244–245, 1997.

S. Cheamanunkul, E. Ettinger, and Y. Freund. Non-convex boosting overcomes random label noise. *CoRR*, abs/1409.2905, 2014.

R. Chen and J. Yu. An improved bagging neural network ensemble algorithm and its application. In *Proceedings of the Third International Conference on Natural Computation*, volume 5, pages 730–734. IEEE, 2007.

D. Chickering, D. Geiger, and D. Heckerman. Learning Bayesian networks: Search methods and experimental results. In *Proceedings of the Fifth Conference on Artificial Intelligence and Statistics*, pages 112–128, 1995.

D. M. Chickering. Learning equivalence classes of Bayesian network structures. *Journal of Machine Learning Research*, 2:445–498, 2002.

C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

G. F. Cooper. Nestor: A computer-based medical diagnostic aid that integrates causal and probabilistic knowledge. Technical report, Stanford University of California Department of Computer Science, 1984.

G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2-3):393–405, 1990.

G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

G. M. Cordeiro and S. L.P. Ferrari. A modified score test statistic having chi-squared distribution to order n-1. *Biometrika*, 78(3):573–582, 1991.

T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

R. G. Cowell, P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer Science & Business Media, 2006.

A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2-3):81–227, 2012.

A. Cutler and G. Zhao. PERT-perfect random tree ensembles. *Computing Science and Statistics*, 33: 490–497, 2001.

D. R. Cutler, T. C. Edwards Jr., K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler. Random forests for classification in ecology. *Ecology*, 88(11):2783–2792, 2007.

P. Dagum and M. Luby. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, 60(1):141–153, 1993.

A. Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

F. De Dombal, D. Leaper, J. R. Staniland, A. McCann, and J. C. Horrocks. Computer-aided diagnosis of acute abdominal pain. *British Medical Journal*, 2(5804):9–13, 1972.

P. R. De Waal and L. C. Van Der Gaag. Inference and learning in multi-dimensional Bayesian network classifiers. In *Proceedings of the Ninth European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 501–511. Springer, 2007.

R. Dechter. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113(1-2): 41–85, 1999.

J. DeFelipe. The evolution of the brain, the human nature of cortical circuits, and intellectual creativity. *Frontiers in Neuroanatomy*, 5:29, 2011.

J. DeFelipe and I. Fariñas. The pyramidal neuron of the cerebral cortex: Morphological and chemical characteristics of the synaptic inputs. *Progress in Neurobiology*, 39(6):563–607, 1992.

M. R. del Río and J. DeFelipe. A study of SMI 32-stained pyramidal cells, parvalbumin-immunoreactive Chandelier cells, and presumptive thalamocortical axons in the human temporal neocortex. *Journal of Comparative Neurology*, 342(3):389–408, 1994.

M. Denil, D. Matheson, and N. Freitas. Consistency of online random forests. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 1256–1264, 2013.

L. Devroye and L. Gyorfi. *Nonparametric Density Estimation: The L1 View*. Wiley Interscience Series in Discrete Mathematics. Wiley, 1985.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

R. Díaz-Uriarte and S. Alvarez de Andrés. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1):3, 2006.

T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.

M. M. Drugan and M. A. Wiering. Feature selection for Bayesian network classifiers using the MDL-FS score. *International Journal of Approximate Reasoning*, 51(6):695–717, 2010.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2012.

B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. CRC Press, 1994.

G. Elston. Specialization of the neocortical pyramidal cell during primate evolution In *Evolution of Nervous Systems*, pages 191–242. Academic Press, Oxford, 2007.

G. Elston and M. Rosa. The occipitoparietal pathway of the macaque monkey: Comparison of pyramidal cell morphology in Layer iii of functionally related cortical visual areas. *Cerebral Cortex*, 7(5):432–452, 1997.

G. Elston, R. Benavides-Piccione, and J. DeFelipe. The pyramidal cell in cognition: A comparative study in human and monkey. *Journal of Neuroscience*, 2001.

G. N. Elston, R. Benavides-Piccione, A. Elston, P. R. Manger, and J. DeFelipe. Pyramidal cells in prefrontal cortex of primates: Marked differences in neuronal structure among species. *Frontiers in Neuroanatomy*, article 5, 2011.

G. Eyal, H. D. Mansvelder, C. P. de Kock, and I. Segev. Dendrites impact the encoding capabilities of the axon. *Journal of Neuroscience*, 34(24):8063–8071, 2014.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2): 179–188, 1936.

E. Fix and J. L. Hodges Jr. Discriminatory analysis: Small sample performance. Technical report, USAF School of Aviation Medicine, 1952.

E. Fix and J. L. Hodges Jr. Discriminatory analysis-nonparametric discrimination: Consistency properties. Technical report, California University, Berkeley, 1951.

Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2): 256–285, 1995.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

J. Friedman, T. Hastie, R. Tibshirani. Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2):337–407, 2000.

J. H. Friedman and P. Hall. On bagging and nonlinear estimation. *Journal of Statistical Planning and Inference*, 137(3):669–683, 2007.

N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3): 131–163, 1997.

S. Garcia, J. Luengo, J. A. Sáez, V. Lopez, and F. Herrera. A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):734–750, 2013.

L. Garey. *Brodmann's Localisation in the Cerebral Cortex*. London: Smith-Gordon, 1994.

D. Geiger and D. Heckerman. Learning Gaussian networks. In *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence*, pages 235–243. Morgan Kaufmann Publishers Inc., 1994.

R. Genuer, J.-M. Poggi, and C. Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225 – 2236, 2010.

P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

E. Gibaja and S. Ventura. Multi-label learning: A review of the state of the art and ongoing research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(6):411–444, 2014.

S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 22–30. Springer, 2004.

L. A. Goodman. The multivariate analysis of qualitative data: Interactions among multiple classifications. *Journal of the American Statistical Association*, 65(329):226–256, 1970.

G. A. Gorry and G. O. Barnett. Experience with a model of sequential diagnosis. *Computers and Biomedical Research*, 1(5):490–507, 1968.

L.-A. Gottlieb, A. Kontorovich, and R. Krauthgamer. Efficient classification for metric data. *IEEE Transactions on Information Theory*, 60(9):5750–5759, 2014.

S. J. Haberman. *The General Log-Linear Model*. PhD thesis, University of Chicago, Department of Statistics, 1970.

T. Hancock, T. Jiang, M. Li, and J. Tromp. Lower bounds on learning decision lists and trees. *Information and Computation*, 126(2):114–122, 1996.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2009.

D. E. Heckerman and B. N. Nathwani. An evaluation of the diagnostic accuracy of pathfinder. *Computers and Biomedical Research*, 25(1):56–74, 1992a.

E. Heckerman and N. Nathwani. Toward normative expert systems: Part ii. Probability-based representations for efficient knowledge acquisition and inference. *Methods of Information in Medicine*, 31(02): 106–116, 1992b.

M. Henrion. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In *Machine Intelligence and Pattern Recognition*, pages 149–163, 1988.

M. Henrion. Towards efficient inference in multiply connected belief networks. In *Influence Diagrams, Belief Nets and Decision Analysis*, pages 385–407, 1990.

M. Henrion. Search-based methods to bound diagnostic probabilities in very large belief nets. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 142–150, 1991.

J. Howard and M. Bowles. The two most important algorithms in predictive modeling today. *Strata Conference Presentation*, 28, 2012.

C. Huang and A. Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15(3):225–263, 1996.

H. Ishwaran, U. B. Kogalur, E. H. Blackstone, and M. S. Lauer. Random survival forests. *Annals of Applied Statistics*, 2(3):841–860, 2008.

H. Ishwaran, U. B. Kogalur, X. Chen, and A. J. Minn. Random survival forests for high-dimensional data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 4(1):115–132, 2011.

B. Jacobs, M. Schall, M. Prather, E. Kapler, L. Driscoll, S. Baca, J. Jacobs, K. Ford, M. Wainwright, and M. Treml. Regional dendritic and spine variation in human cerebral cortex: A quantitative Golgi study. *Cerebral Cortex*, 11(6):558–571, 2001.

B. Jacobs, N. Johnson, D. Wahl, M. Schall, B. C. Maseko, A. Lewandowski, M. A. Raghanti, B. Wicinski, C. Butti, W. Hopkins, M. F. Bertelsen, T. Walsh, J. R. Roberts, R. Reep, P. R. Hof, C. C. Sherwood, and P. Manger. Comparative neuronal morphology of the cerebellar cortex in afrotherians, carnivores, cetartiodactyls, and primates. *Frontiers in Neuroanatomy*, 8(24), 2014.

F. V. Jensen. *An Introduction to Bayesian Networks*, UCL Press, 1996.

M. I. Jordan. *Learning in Graphical Models*, Springer Science & Business Media, 1998.

S. Jung, Y. Nam, and D. Lee. Inference of combinatorial neuronal synchrony with Bayesian networks. *Journal of Neuroscience Methods*, 186(1):130–139, 2010.

P. E. Jupp and K. V. Mardia. A unified view of the theory of directional statistics, 1975-1988. *International Statistical Review*, 57(3):261–294, 1989.

N. E. Karoui and E. Purdom. Can we trust the bootstrap in high-dimension? Technical Report, California University, Berkeley, 2015.

A. Kastanauskaite, L. Alonso-Nanclares, L. Blazquez-Llorca, J. Pastor, RG. Sola and J. DeFelipe. Alterations of the microvascular network in sclerotic hippocampi from patients with epilepsy. *Journal of Neuropathology & Experimental Neurology*, 68(8):939–950, 2009.

M. Kearns. Thoughts on hypothesis boosting. *ML Class Project*, 1988.

M. J. Kearns and L. G. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. In *Machine Learning: From Theory to Applications*, pages 29–49. Springer, 1993.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

K. B. Korb and A. E. Nicholson. *Bayesian Artificial Intelligence*. CRC Press, 2010.

W. H. Kruskal and W. A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952.

B. Lakshminarayanan, D. M. Roy, and Y. W. Teh. Mondrian forests: Efficient online random forests. In *Advances in Neural Information Processing Systems 27*, pages 3140–3148. Curran Associates, Inc., 2014.

P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, and Y. Yurramendi. Learning Bayesian network structures by searching for the best ordering with genetic algorithms. *IEEE Transactions on System, Man and Cybernetics. Part A: Systems and Humans*, 26(4):487–493, 1996.

H. Laurent and R. L. Rivest. Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, 5(1):15–17, 1976.

S. L. Lauritzen. *Graphical Models*. Clarendon Press, 1996.

I. Leguey, C. Bielza, P. Larrañaga, A. Kastanauskaite, C. Rojo, R. Benavides-Piccione, and J. DeFelipe. Dendritic branching angles of pyramidal cells across layers of the juvenile rat somatosensory cortex. *Journal of Comparative Neurology*, 524(13):2567–2576, 2016.

P. Leray and O. Francois. BNT structure learning package: Documentation and experiments. Laboratoire PSI, Universitè et INSA de Rouen, Technical Report, 2004.

T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, 40(3):203–228, 2000.

Y. Lin and Y. Jeon. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101(474):578–590, 2006.

C. X. Ling and H. Zhang. The representational power of discrete Bayesian networks. *Journal of Machine Learning Research*, 3:709–721, 2002.

P. M. Long and R. A. Servedio. Random classification noise defeats all convex potential boosters. *Machine Learning*, 78(3):287–304, 2010.

P. L. Lopez-Cruz, C. Bielza, P. Larrañaga, R. Benavides-Piccione, and J. DeFelipe. Models and simulation of 3d neuronal dendritic trees using Bayesian networks. *Neuroinformatics*, 9(4):347–369, 2011.

P. L. López-Cruz, P. Larrañaga, J. DeFelipe, and C. Bielza. Bayesian network modeling of the consensus between experts: An application to neuron classification. *International Journal of Approximate Reasoning*, 55(1):3–22, 2014.

P. L. López-Cruz, C. Bielza, and P. Larrañaga. Directional naive Bayes classifiers. *Pattern Analysis and Applications*, 18(2):225–246, 2015.

K. Mardia and P. Jupp. *Directional Statistics*. Wiley Series in Probability and Statistics. 2000.

K. V. Mardia. Statistics of directional data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 37(3): pp. 349–393, 1975.

K. V. Mardia and J. Voss. Some fundamental properties of a multivariate von Mises distribution. *Communications in Statistics-Theory and Methods*, 43(6):1132–1144, 2014.

K. V. Mardia, G. Hughes, C. C. Taylor, and H. Singh. A multivariate von Mises distribution with applications to bioinformatics. *Canadian Journal of Statistics*, 36:99–109, 2008.

L. Mason, J. Baxter, P. L. Bartlett, and M. R. Frean. Boosting algorithms as gradient descent. In *Advances in Neural Information Processing Systems*, pages 512–518, 2000.

M. Minsky. Steps toward artificial intelligence. *Proceedings of the Institute of Radio Engineers*, 49(1): 8–30, 1961.

H. Mohan, M. B. Verhoog, K. K. Doreswamy, G. Eyal, R. Aardse, B. N. Lodder, N. A. Goriounova, B. Asamoah, A. C. B. Brakspear, C. Groot, S. van der Sluis, G. Testa-Silva, J. Obermayer, ZS. Boudewijns, RT. Narayanan, J.C. Baayen, I. Segev, HD. Mansvelder and C.P. de Kock. Dendritic and axonal architecture of individual pyramidal neurons across layers of adult human neocortex. *Cerebral Cortex*, 25(12):4839–4853, 2015.

K. Murphy. The Bayes net toolbox for Matlab. *Computing Science and Statistics*, 33(2):1024–1034, 2001.

G. Naumov. NP-completeness of problems of construction of optimal decision trees. In *Soviet Physics Doklady*, volume 36, page 270, 1991.

R. M. Neal. Probabilistic Inference using Markov Chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.

R. E. Neapolitan. *Learning Bayesian Networks.* Pearson Prentice Hall, 2004.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988.

Y. Peng and J. A. Reggia. A probabilistic causal model for diagnostic problem solving Part i: Integrating symbolic causal inference with numeric probabilistic inference. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(2):146–162, 1987.

A. Pérez, P. Larrañaga, and I. Inza. Supervised classification with conditional Gaussian networks: Increasing the structure complexity from naive Bayes. *International Journal of Approximate Reasoning*, 43(1):1 – 25, 2006.

A. M. Prasad, L. R. Iverson, and A. Liaw. Newer classification and regression tree techniques: Bagging and random forests for ecological prediction. *Ecosystems*, 9(2):181–199, 2006.

J. R. Quinlan. *C4. 5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.

L. E. Raileanu and K. Stoffel. Theoretical comparison between the Gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93, 2004.

J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. In *Proceedings of the 20th Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 254–269. Springer, 2009.

J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society: Series B (Methodological)*, 49(3):223–239, 1987.

R. W. Robinson. Counting labeled acyclic digraphs. *New directions in the Theory of Graphs.* pp. 239–273. Academic Press, New York, 1973.

J. D. Rodríguez and J. A. Lozano. Multi-objective learning of multi-dimensional Bayesian classifiers. In *Proceedings of the Eighth International Conference on Hybrid Intelligent Systems*, pages 501–506. IEEE, 2008.

E. D. Rothman. Tests of coordinate independence for a bivariate sample on a torus. *The Annals of Mathematical Statistics*, 42(6):1962–1969, 1971.

S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2016.

A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line random forests. In *IEEE 12th International Conference on Computer Vision Workshops*, pages 1393–1400, 2009.

M. Sahami. Learning limited dependence Bayesian classifiers. In *Knowledge Discovery and Data Mining*, volume 96, pages 335–338, 1996.

R. J. Samworth. Optimal weighted nearest neighbour classifiers. *The Annals of Statistics*, 40(5):2733–2763, 2012.

R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

R. E. Schapire. The boosting approach to machine learning: An overview. In *Nonlinear Estimation and Classification*, pages 149–171. Springer, 2003.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

R. Scorcioni, S. Polavaram, and G. A. Ascoli. L-measure: A web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies. *Nature Protocols*, 3(5): 866–876, 2008.

E. Scornet. On the asymptotics of random forests. *Journal of Multivariate Analysis*, 146:72 – 83, 2016.

R. Shachter and M. Peot. Evidential reasoning using likelihood weighting. *Fifth Workshop on Uncertainty in Artificial Intelligence*, pages 18–20, 1989.

G. R. Shafer and P. P. Shenoy. Probability propagation. *Annals of Mathematics and Artificial Intelligence*, 2(1-4):327–351, 1990.

J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from a single depth image. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1297-1304, 2011.

C.-Y. Sin and H. White. Information criteria for selecting possibly misspecified parametric models. *Journal of Econometrics*, 71(1-2):207–225, 1996.

H. Singh. Probabilistic model for two dependent circular variables. *Biometrika*, 89(3):719–723, 2002.

V. A. Smith, J. Yu, T. V. Smulders, A. J. Hartemink, and E. D. Jarvis. Computational inference of neural information flow networks. *PLoS Computational Biology*, 2(11):e161, 2006.

R.G. Sola, V. Hernando-Requejo, J. Pastor, E. García-Navarrete, J. DeFelipe, MT. Alijarde, A. Sánchez, L. Domínguez-Gadea, P. Martín-Plasencia, F. Maestú, J. DeFelipe-Oroquieta, S. Ramón-Cajal and P. Pulido-Rivas. Epilepsia farmacorresistente del lóbulo temporal. Exploración con electrodos del foramen oval y resultados quirúrgicos. *Revista de Neurología*, 41(1):4–16, 2005.

D. J. Spiegelhalter, N. G. Best, B. P. Carlin, and A. Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4): 583–639, 2002.

P. Spirtes, C. N. Glymour and R. Scheines *Causation, Prediction, and Search*. Springer-Verlag, 1993.

B. M. Steele. Exact bootstrap k-nearest neighbor learners. *Machine Learning*, 74(3):235–255, 2009.

C. J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5(4): 595–620, 1977.

V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston. Random forest: A classification and regression tool for compound classification and QSAR modeling. *Journal of Chemical Information and Computer Sciences*, 43(6):1947–1958, 2003.

I. Syarif, E. Zaluska, A. Prugel-Bennett, and G. Wills. Application of bagging, boosting and stacking to intrusion detection. In *Proceedings of the Eighth International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 593–602. Springer, 2012.

V. R. Tabar. A simple node ordering method for the k2 algorithm based on the factor analysis. In *Proceedings of the International Conference on Pattern Recognition Applications and Methods*, pages 273–280, 2017.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

G. Valentini, M. Muselli, and F. Ruffino. Bagged ensembles of support vector machines for gene expression data analysis. In *Proceedings of the 2003 International Joint Conference on Neural Networks*, volume 3, pages 1844–1849. IEEE, 2003.

L. C. Van Der Gaag and P. R. De Waal. Multi-dimensional Bayesian network classifiers. In *Third European Conference on Probabilistic Graphical Models*, pages 107–114, 2006.

G. Varando, C. Bielza, and P. Larranaga. Decision boundary for discrete Bayesian network classifiers. *Journal of Machine Learning Research*, 16(1):2725–2749, 2015.

N. N. Vorob'ev. Consistent families of measures and their extensions. *Theory of Probability & its Applications*, 7(2):147–163, 1962.

H. Wallraff. Goal-oriented and compass-oriented movements of displaced homing pigeons after confinement in differentially shielded aviaries. *Behavioral Ecology and Sociobiology*, 5(2):201–225, 1979.

H. R. Warner, A. F. Toronto, L. G. Veasey, and R. Stephenson. A mathematical approach to medical diagnosis: Application to congenital heart disease. *Journal of the American Medical Assotiation*, 177 (3):177–183, 1961.

G. Watson. *Statistics on Spheres*. Wiley-Interscience, 1983.

G. S. Watson. Goodness-of-fit tests on a circle. *Biometrika*, 49(1/2):57–63, 1962.

S. J. Winham, R. R. Freimuth, and J. M. Biernacka. A weighted random forests approach to improve predictive performance. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 6(6): 496–505, 2013.

F. Yang, J. Wang, and G. Fan. Kernel induced random survival forests. *arXiv preprint arXiv:1008.3952*, 2010.

Z. Yi, S. Soatto, M. Dewan, and Y. Zhan. Information forests. In *2012 Information Theory and Applications Workshop*, pages 143–146, 2012.

D. Zhang, X. Zhou, S. C. Leung, and J. Zheng. Vertical bagging decision trees model for credit scoring. *Expert Systems with Applications*, 37(12):7838–7843, 2010.

M.-L. Zhang and Z.-H. Zhou. A k-nearest neighbor based algorithm for multi-label classification. In *Proceedings of the 2005 IEEE International Conference on Granular Computing*, 2:718–721, 2005.

N. L. Zhang and D. Poole. A simple approach to Bayesian network computations. In *Proceedings of the Biennial Conference-Canadian Society for Computational Studies of Intelligence*, pages 171–178. Canadian Information Processing Society, 1994.

Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC, 2012.

A. Ziegler, D. F. Schwarz, and I. R. KJönig. On safari to random jungle: A fast implementation of random forests for high-dimensional data. *Bioinformatics*, 26(14):1752–1758, 2010.